User's Manual

32

RA6T2 Group

User's Manual: Hardware

32-bit MCU
Renesas Advanced (RA) Family
Renesas RA6 Series

Renesas Electronics
www.renesas.com

Rev.1.30 Aug 2022

Notice

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

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

(Rev.5.0-1 October 2020)

Corporate Headquarters

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

Contact Information

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

Trademarks

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

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

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

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

Preface

1. About this document

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

2. Audience

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

3. Renesas Publications

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

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

4. Numbering Notation

The following numbering notation is used throughout this manual:

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

5. Typographic Notation

The following typographic notation is used throughout this manual:

Example	Description
WDT.WDTRCR.RSTIRQS	Periods separated a function module symbol (WDT), register symbol (WDTRCR), and bit field symbol (RSTIRQS).
WDT.WDTRCR	A period separated a function module symbol (WDT) and register symbol (WDTRCR).
WDTRCR.RSTIRQS	A period separated a register symbol (WDTRCR) and bit field symbol (RSTIRQS).
CKS[3:0]	Numbers in brackets expresses a bit number. For example, CKS[3:0] occupies bits 3 to 0 of the WDT Control Register (WDTCR) register.

6. Unit and Unit Prefix

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

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

7. Special Terms

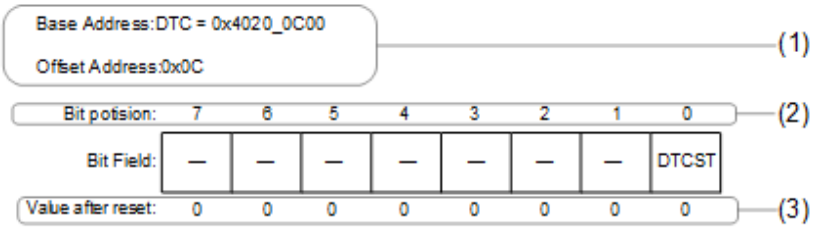
The following terms have special meanings.

Term	Description
NC	Not connected pin. NC means that pin is not connected to the MCU.
Hi-Z	High impedance.

8. Register Description

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

XXXX DTCST : DTC Module Start Register



(4) (5) (6)

Bit	Symbol	Function	R/W
0	DTCST	DTC Module Start 0:DTC module stop 1:DTC module start	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

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

Function module symbol, register symbol, and address assignment of this register are generally expressed. Base Address and Offset Address mean DTC Module Start Register (DTCST) of Data Transfer Controller (DTC) is assigned to address 0x4020_0C00.

(2) Bit number

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

(3) Value after reset

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

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

(4) Symbol

Symbol indicates the short name of bit field. Reserved bit is expressed with a —.

(5) Function

Function indicates the full name of the bit field and enumerated values.

(6) R/W

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

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

9. Abbreviations

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

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

10. Proprietary Notice

11. Feedback on the product

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

Contents

High-performance 240 MHz Arm Cortex-M33 core, up to 512 KB code flash memory with background operation, 16 KB Data flash memory, and 64 KB SRAM with ECC. Integrated A/D converter with channel-dedicated sample-and-hold circuit for simultaneous sampling and single-end/pseudo-differential input supportive amplifier. Integrated General PWM Timer with 200 MHz operation and high resolution. Integrated Secure Cryptographic Engine with cryptography accelerators and key management support in concert with Arm TrustZone for integrated secure element functionality.

Features

Arm® Cortex®-M33 Core

- Armv8-M architecture with the main extension
- Maximum operating frequency: 240 MHz
- Arm Memory Protection Unit (Arm MPU)
  - Protected Memory System Architecture (PMSAv8)
  - Secure MPU (MPU_S): 8 regions
  - Non-secure MPU (MPU_NS): 8 regions
- SysTick timer
  - Embeds two SysTick timers: Secure and Non-secure instance
  - Driven by LOCO or system clock
- CoreSight™ ETM-M33

Memory

- Up to 512-KB code flash memory
- 16-KB data flash memory (125,000 program/erase (P/E) cycles)
- 64-KB SRAM

Connectivity

- Serial Communications Interface (SCI) × 6
  - Asynchronous interfaces
  - 8-bit clock synchronous interface
  - Smart card interface
  - Simple IIC
  - Simple SPI
  - Simple LIN
  - Manchester coding
- I^2C bus interface (IIC) × 2
  - Transfer at up to 3.2 Mpbs (high speed mode)
- Serial Peripheral Interface (SPI) × 2
- CAN with Flexible Data-rate (CANFD)

Analog

- A/D Converter (ADC) × 2
  - Up to 16-bit resolution
  - Up to 6.25 Msps
  - Channel-dedicated sample-and-hold circuit × 6
  - Programmable Gain Amplifier (PGA) × 4
- High-Speed Analog Comparator (ACMPHS) × 4
- 12-bit D/A Converter (DAC12) × 4
- Temperature Sensor (TSN)

Timers

- General PWM Timer 32-bit (GPT32) with High Resolution × 4
  - 156 ps resolution in 200 MHz
- General PWM Timer 32-bit (GPT32) × 6
- Low Power Asynchronous General Purpose Timer (AGT) × 2

Security and Encryption

- Secure Cryptographic Engine (SCE5)
  - Symmetric algorithms: AES
  - Hash-value generation: GHASH
  - 128-bit unique ID
- Arm® TrustZone®
  - Up to three regions for the code flash
  - Up to two regions for the data flash
  - Up to three regions for the SRAM
  - Individual secure or non-secure security attribution for each peripheral
- Device lifecyle management

System and Power Management

- Low power modes
- Event Link Controller (ELC)
- Data Transfer Controller (DTC)
- DMA Controller (DMAC) × 8
- Power-on reset
- Low Voltage Detection (LVD) with voltage settings
- Watchdog Timer (WDT)
- Independent Watchdog Timer (IWDT)
- Key Interrupt Funciton (KINT)

Data Processing Accelerator

- Trigonometric Function Unit (TFU)
- IIR Filter Accelerator (IIRFA)

Multiple Clock Sources

- Main clock oscillator (MOSC) (8 to 24 MHz)
- High-speed on-chip oscillator (HOCO) (16/18/20 MHz)
- Middle-speed on-chip oscillator (MOCO) (8 MHz)
- Low-speed on-chip oscillator (LOCO) (32.768 kHz)
- IWDT-dedicated on-chip oscillator (15 kHz)
- Clock trim function for HOCO/MOCO/LOCO
- PLL/PLL2
- Clock out support

General-Purpose I/O Ports

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

Operating Voltage

- VCC: 2.7 to 3.6 V

Operating Temperature and Packages

- Ta = -40°C to +105°C
  - 100-pin LQFP (14 mm × 14 mm, 0.5 mm pitch)
  - 64-pin LQFP (10 mm × 10 mm, 0.5 mm pitch)
  - 48-pin LQFP (7 mm × 7 mm, 0.5 mm pitch)
  - 64-pin QFN (8 mm × 8 mm, 0.4 mm pitch)
  - 48-pin QFN (7 mm × 7 mm, 0.5 mm pitch)

1. Overview

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

The MCU in this series incorporates a high-performance Arm Cortex®-M33 core running up to 240 MHz with the following features:

- Up to 512 KB code flash memory
- 64 KB SRAM
- General PWM Timer (GPT) - Enhanced High Resolution
- Analog peripherals
- Security and safety features

1.1 Function Outline

Table 1.1 Arm core

Feature	Functional description
Arm Cortex-M33 core	• Maximum operating frequency: up to 240 MHz • Arm Cortex-M33 core: – Armv8-M architecture with security extension – Revision: r0p4-00rel0 • Arm Memory Protection Unit (Arm MPU) – Protected Memory System Architecture (PMSAv8) – Secure MPU (MPU_S): 8 regions – Non-secure MPU (MPU_NS): 8 regions • SysTick timer – Embeds two SysTick timers: Secure and Non-secure instance – Driven by SysTick timer clock (SYSTICCLK) or system clock (ICLK) • CoreSight™ ETM-M33

Table 1.2 Memory

Feature	Functional description
Code flash memory	Maximum 512 KB of code flash memory. See section 43, Flash Memory.
Data flash memory	16 KB of data flash memory. See section 43, Flash Memory.
Option-setting memory	The option-setting memory determines the state of the MCU after a reset. See section 6, Option-Setting Memory.
SRAM	On-chip high-speed SRAM with Error Correction Code (ECC). See section 41, SRAM.

Table 1.3 System (1 of 2)

Feature	Functional description
Operating modes	Two operating modes: • Single-chip mode • SCI boot mode See section 3, Operating Modes.
Resets	The MCU provides 14 resets. See section 5, Resets.
Low Voltage Detection (LVD)	The Low Voltage Detection (LVD) module monitors the voltage level input to the VCC pin. The detection level can be selected by register settings. The LVD module consists of three separate voltage level detectors (LVD0, LVD1, LVD2). LVD0, LVD1, and LVD2 measure the voltage level input to the VCC pin. LVD registers allow your application to configure detection of VCC changes at various voltage thresholds. See section 7, Low Voltage Detection (LVD).

Table 1.3 System (2 of 2)

Feature	Functional description
Clocks	• Main clock oscillator (MOSC) • High-speed on-chip oscillator (HOCO) • Middle-speed on-chip oscillator (MOCO) • Low-speed on-chip oscillator (LOCO) • IWDT-dedicated on-chip oscillator • PLL/PLL2 • Clock out support See section 8, Clock Generation Circuit.
Clock Frequency Accuracy Measurement Circuit (CAC)	The Clock Frequency Accuracy Measurement Circuit (CAC) counts pulses of the clock to be measured (measurement target clock) within the time generated by the clock selected as the measurement reference (measurement reference clock), and determines the accuracy depending on whether the number of pulses is within the allowable range. When measurement is complete or the number of pulses within the time generated by the measurement reference clock is not within the allowable range, an interrupt request is generated. See section 9, Clock Frequency Accuracy Measurement Circuit (CAC).
Interrupt Controller Unit (ICU)	The Interrupt Controller Unit (ICU) controls which event signals are linked to the Nested Vector Interrupt Controller (NVIC), the DMA Controller (DMAC), and the Data Transfer Controller (DTC) modules. The ICU also controls non-maskable interrupts. See section 12, Interrupt Controller Unit (ICU).
Key Interrupt Function (KINT)	The key interrupt function (KINT) generates the key interrupt by detecting rising or falling edge on the key interrupt input pins. See section 19, Key Interrupt Function (KINT).
Low power modes	Power consumption can be reduced in multiple ways, including setting clock dividers, stopping modules, selecting power control mode in normal operation, and transitioning to low power modes. See section 10, Low Power Modes.
Register write protection	The register write protection function protects important registers from being overwritten due to software errors. The registers to be protected are set with the Protect Register (PRCR). See section 11, Register Write Protection.
Memory Protection Unit (MPU)	The MCU has one Memory Protection Unit (MPU). See section 14, Memory Protection Unit (MPU).

Table 1.4 Event link

Feature	Functional description
Event Link Controller (ELC)	The Event Link Controller (ELC) uses the event requests generated by various peripheral modules as source signals to connect them to different modules, allowing direct link between the modules without CPU intervention. See section 17, Event Link Controller (ELC).

Table 1.5 Direct memory access

Feature	Functional description
Data Transfer Controller (DTC)	A Data Transfer Controller (DTC) module is provided for transferring data when activated by an interrupt request. See section 16, Data Transfer Controller (DTC).
DMA Controller (DMAC)	The MCU includes an 8-channel direct memory access controller (DMAC) that can transfer data without intervention from the CPU. When a DMA transfer request is generated, the DMAC transfers data stored at the transfer source address to the transfer destination address. See section 15, DMA Controller (DMAC).

Table 1.6 Timers (1 of 2)

Feature	Functional description
General PWM Timer (GPT)	The General PWM Timer (GPT) is a 32-bit timer with GPT32 × 10 channels. PWM waveforms can be generated by controlling the up-counter, down-counter, or the up- and down-counter. In addition, PWM waveforms can be generated for controlling brushless DC motors. The GPT can also be used as a general-purpose timer. See section 21, General PWM Timer (GPT).

Table 1.6 Timers (2 of 2)

Feature	Functional description
PWM Delay Generation Circuit (PDG)	The PWM Delay Generation circuit (PDG) has 4 channels delay circuits that can connect to the GPT. The PDG can control the rise and fall edge timing with which the PWM output for the GPT320 through the GPT323. See section 22, PWM Delay Generation Circuit (PDG).
Port Output Enable for GPT (POEG)	The POEG issues requests to stop output from output pins of the general PWM timer (GPT). Select the method of detection for stopping the output from the list below. See section 20, Port Output Enable for GPT (POEG).
Low power Asynchronous General Purpose Timer (AGT)	The low power Asynchronous General Purpose Timer (AGT) is a 32-bit timer that can be used for pulse output, external pulse width or period measurement, and counting external events. This timer consists of a reload register and a down counter. The reload register and the down counter are allocated to the same address, and can be accessed with the AGT register. See section 23, Low Power Asynchronous General Purpose Timer (AGTW).
Watchdog Timer (WDT)	The Watchdog Timer (WDT) is a 14-bit down counter that can be used to reset the MCU when the counter underflows because the system has run out of control and is unable to refresh the WDT. In addition, the WDT can be used to generate a non-maskable interrupt or an underflow interrupt. See section 24, Watchdog Timer (WDT).
Independent Watchdog Timer (IWDT)	The Independent Watchdog Timer (IWDT) consists of a 14-bit down counter that must be serviced periodically to prevent counter underflow. The IWDT provides functionality to reset the MCU or to generate a non-maskable interrupt or an underflow interrupt. Because the timer operates with an independent, dedicated clock source, it is particularly useful in returning the MCU to a known state as a fail-safe mechanism when the system runs out of control. The IWDT can be triggered automatically by a reset, underflow, refresh error, or a refresh of the count value in the registers. See section 25, Independent Watchdog Timer (IWDT).

Table 1.7 Communication interfaces

Feature	Functional description
Serial Communications Interface (SCI)	The Serial Communications Interface (SCI) × 6 channels have asynchronous and synchronous serial interfaces: ● Asynchronous interfaces (UART and Asynchronous Communications Interface Adapter (ACIA)) ● 8-bit clock synchronous interface ● Simple IIC (master-only) ● Simple SPI ● Simple LIN ● Smart card interface ● Manchester interface The smart card interface complies with the ISO/IEC 7816-3 standard for electronic signals and transmission protocol. SCIn (n = 0 to 4, 9) has FIFO buffers to enable continuous and full-duplex communication, and the data transfer speed can be configured independently using an on-chip baud rate generator. See section 26, Serial Communications Interface (SCI).
I^2C bus interface (IIC)	The I^2C bus interface (IIC) has 2 channels. The IIC module conform with and provide a subset of the NXP I^2C (Inter-Integrated Circuit) bus interface functions. See section 27, I^2C Bus Interface (IIC).
Serial Peripheral Interface (SPI)	The Serial Peripheral Interface (SPI) has 2 channels. The SPI provides high-speed full-duplex synchronous serial communications with multiple processors and peripheral devices. See section 30, Serial Peripheral Interface (SPI).
CAN with Flexible Data-rate (CANFD)	The CAN with Flexible Data-rate (CANFD) module can handle classical CAN frames and CAN-FD frames complied with ISO 11898-1 standard. The module supports 4 transmit buffers and 32 receive buffer. See section 28, CAN with Flexible Data-rate (CANFD).

Table 1.8 Analog

Feature	Functional description
A/D Converter (ADC)	The A/D Converter (ADC) has two units of noise-shaping SAR-type A/D converter. ● Hybrid architecture that combines the features of the successive approximation register-type and the delta-sigma modulation-type. ● Up to 16-bit resolution ● Up to 6.25 Msps ● Up to 29 analog input channels ● Support single-ended input or differential inputs ● Built-in channel-dedicated sample-and-hold circuit (SH) ● Built-in Programmable Gain Amplifier (PGA) ● Temperature sensor output and internal reference voltage, and D/A converters output are selectable for conversion. See section 36, A/D Converter.
12-bit D/A Converter (DAC12)	A 12-bit D/A converter (DAC12) is provided. See section 37, 12-Bit D/A Converter (DAC12).
High-Speed Analog Comparator (ACMPHS)	The High-Speed Analog Comparator (ACMPHS) compares a test voltage with a reference voltage and provides a digital output based on the conversion result. Both the test and reference voltages can be provided to the comparator from internal sources such as the DAC12 output and an internal PGA output, and an external source. Such flexibility is useful in applications that require go/no-go comparisons to be performed between analog signals without necessarily requiring A/D conversion. See section 39, High-Speed Analog Comparator (ACMPHS).
Temperature Sensor (TSN)	The on-chip Temperature Sensor (TSN) determines and monitors the die temperature for reliable operation of the device. The sensor outputs a voltage directly proportional to the die temperature, and the relationship between the die temperature and the output voltage is fairly linear. The output voltage is provided to the ADC for conversion and can be further used by the end application. See section 38, Temperature Sensor (TSN).

Table 1.9 Data processing

Feature	Functional description
Cyclic Redundancy Check (CRC)	The Cyclic Redundancy Check (CRC) generates CRC codes to detect errors in the data. The bit order of CRC calculation results can be switched for LSB-first or MSB-first communication. Additionally, various CRC-generation polynomials are available. The snoop function allows to monitor the access to specific addresses. This function is useful in applications that require CRC code to be generated automatically in certain events, such as monitoring writes to the serial transmit buffer and reads from the serial receive buffer. See section 31, Cyclic Redundancy Check (CRC).
Data Operation Circuit (DOC)	The data operation circuit (DOC) is used to compare, add, and subtract 16 or 32-bit data. An interrupt can be generated when the following conditions apply. ● When the 16 or 32-bit compared values match the detection condition ● When the result of 16 or 32-bit data addition overflows ● When the result of 16 or 32-bit data subtraction underflows See section 40, Data Operation Circuit (DOC).

Table 1.10 Data processing accelerator

Feature	Functional description
Trigonometric function unit (TFU)	Calculation of sine, cosine, arctangent, and hypot_k ($\sqrt{x^2+y^2}/k$) ● A sine and cosine can be simultaneously calculated. ● An arctangent and hypot_k can be simultaneously calculated.
IIR Filter Accelerator (IIRFA)	● 16 channels of biquad IIR filter ● cascaded biquad filter (max.32 stages) ● Operations using single-precision floating-point numbers

Table 1.11 Security (1 of 2)

Feature	Functional description
Securty function	● ARMv8-M TrustZone security ● Device lifecycle management ● Debug access level ● Key injection

Table 1.11 Security (2 of 2)

Feature	Functional description
Secure Cryptographic Engine (SCE5)	• Symmetric algorithms: AES • Hash-value generation: GHASH • 128-bit unique ID. See section 35, Secure Cryptographic Engine (SCE5).

1.2 Block Diagram

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

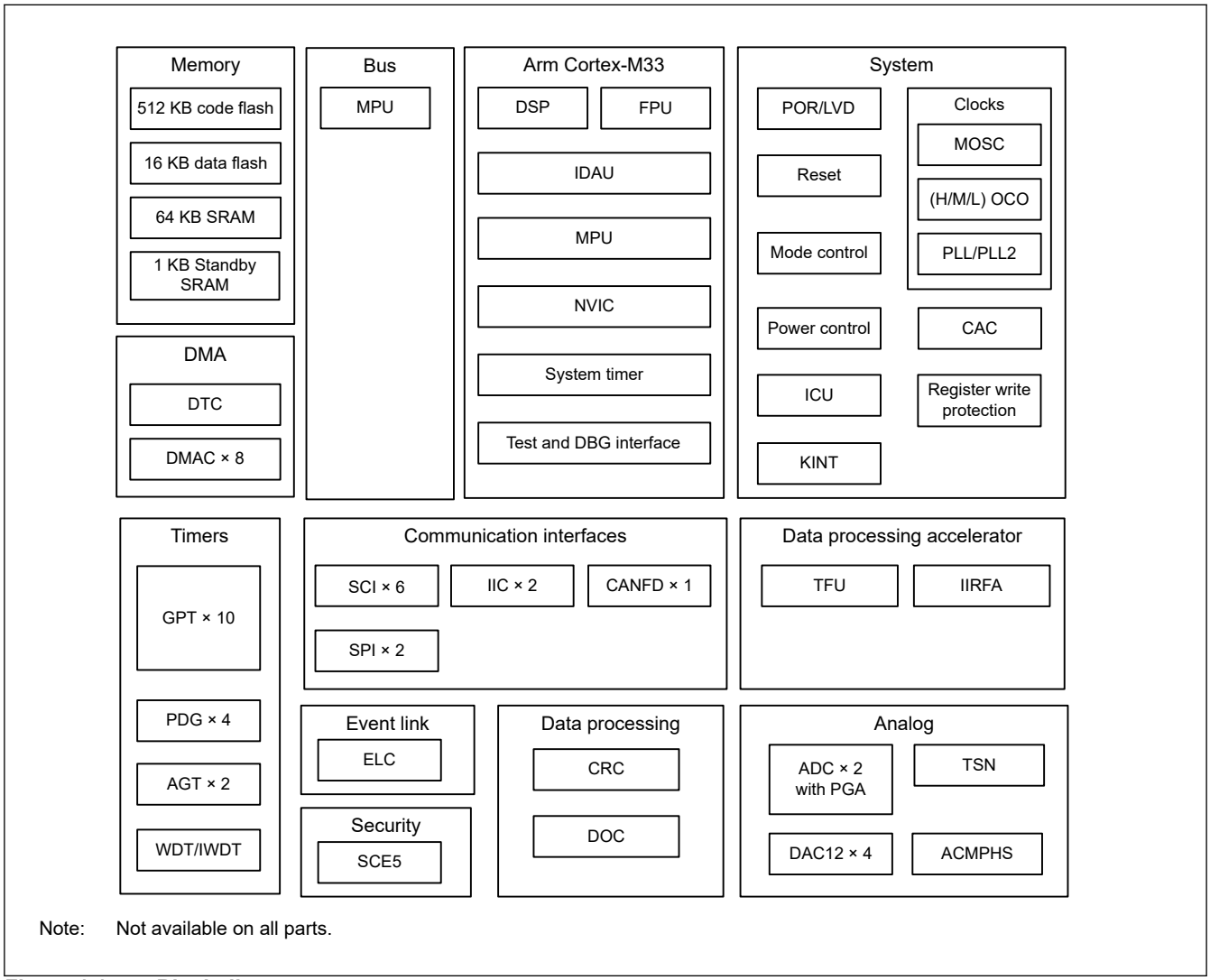
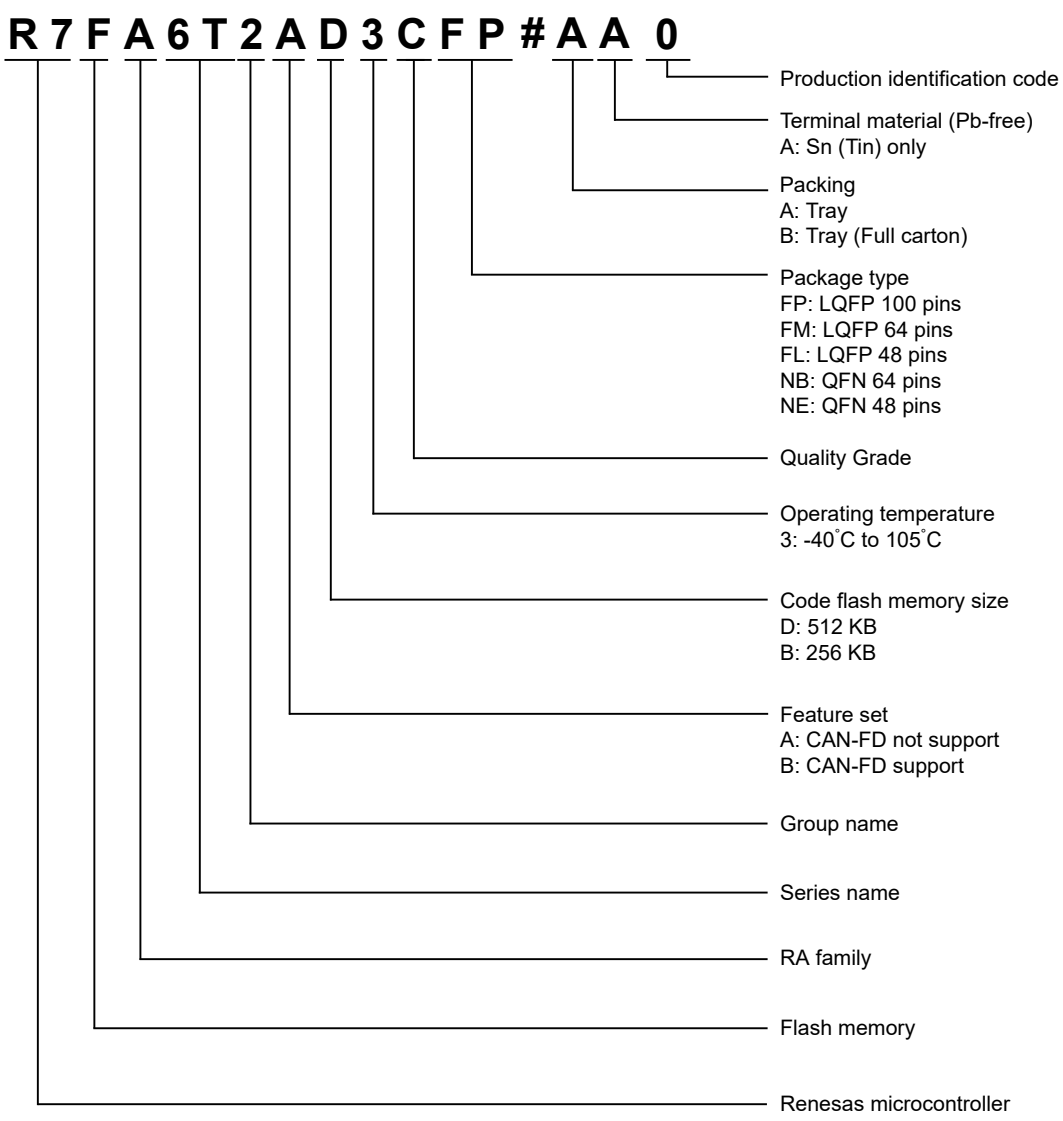


Figure 1.1 Block diagram

1.3 Part Numbering

Figure 1.2 shows the product part number information, including memory capacity and package type. Table 1.12 shows a list of products.



Note: Check the order screen for each product on the Renesas website for valid symbols after the #.

Figure 1.2 Part numbering scheme

Table 1.12 Product list (1 of 2)

Product part number	Package code	Code flash	Data flash	SRAM	CAN-FD	Operating temperature
R7FA6T2AD3CFP	PLQP0100KB-B	512 KB	16 KB	64 KB	Not support	-40 to +105°C
R7FA6T2AD3CFM	PLQP0064KB-C					
R7FA6T2AD3CFL	PLQP0048KB-B					
R7FA6T2AD3CNB	PWQN0064LB-A					
R7FA6T2AD3CNE	PWQN0048KC-A					
R7FA6T2AB3CFP	PLQP0100KB-B	256 KB				
R7FA6T2AB3CFM	PLQP0064KB-C					
R7FA6T2AB3CFL	PLQP0048KB-B					
R7FA6T2AB3CNB	PWQN0064LB-A					
R7FA6T2AB3CNE	PWQN0048KC-A					

Table 1.12 Product list (2 of 2)

Product part number	Package code	Code flash	Data flash	SRAM	CAN-FD	Operating temperature
R7FA6T2BD3CFP	PLQP0100KB-B	512 KB	16 KB	64 KB	Support	-40 to +105°C
R7FA6T2BD3CFM	PLQP0064KB-C					
R7FA6T2BD3CFL	PLQP0048KB-B					
R7FA6T2BD3CNB	PWQN0064LB-A					
R7FA6T2BD3CNE	PWQN0048KC-A					
R7FA6T2BB3CFP	PLQP0100KB-B	256 KB				
R7FA6T2BB3CFM	PLQP0064KB-C					
R7FA6T2BB3CFL	PLQP0048KB-B					
R7FA6T2BB3CNB	PWQN0064LB-A					
R7FA6T2BB3CNE	PWQN0048KC-A					

1.4 Function Comparison

Table 1.13 Function Comparison

Parts number		R7FA6T2XX3CFP	R7FA6T2XX3CFM	R7FA6T2XX3CFL	R7FA6T2XX3CNB	R7FA6T2XX3CNE
Pin count		100	64	48	64	48
Package		LQFP			QFN	
Code flash memory		512 KB, 256KB				
Data flash memory		16 KB				
SRAM	ECC	64 KB				
Standby SRAM	Parity	1 KB				
DMA	DTC	Yes				
	DMAC	8				
System	CPU clock	240 MHz (max.)				
	CPU clock sources	MOSC, HOCO, MOCO, LOCO, PLL				
	CAC	Yes				
	WDT/IWDT	Yes				
	KINT	Yes				
Communication	SCI	6				
	IIC	2*2				
	SPI	2				
	CANFD	1				
Timers	GPT*1	10				
	AGT*1	2				
Analog	ADC	Unit 0: 12 + 9*3, Unit 1: 8 + 9*3	Unit 0: 10, Unit 1: 8	Unit 0: 6, Unit 1: 4	Unit 0: 10, Unit 1: 8	Unit 0: 6, Unit 1: 4
	DAC12	4		2	4	2
	ACMPHS	4		3	4	3
	PGA	4		3	4	3
	TSN	Yes				
Data processing	CRC	Yes				
	DOC	Yes				
Event control	ELC	Yes				
Accelerator	TFU	Yes				
	IIRFA	Yes				
Security		SCE5, TrustZone and Lifecycle management				

Note: The product name differs depend on the memory size and CAN-FD support. see section 1.3. Part Numbering.
Note 1. Available pins depend on the Pin count, about details see section 1.7. Pin Lists.
Note 2. Fm+ and Hs-mode is only available for IIC channel IIC0.
Note 3. Shared terminal for UNIT0 and UNIT1.

1.5 Pin Functions

Table 1.14 Pin functions (1 of 3)

Function	Signal	I/O	Description
Power supply	VCC	Input	Power supply pin. Connect it to the system power supply. Connect this pin to VSS by a 0.1-μF capacitor. The capacitor should be placed close to the pin.
	VCL	I/O	Connect this pin to the VSS pin by the smoothing capacitor used to stabilize the internal power supply. Place the capacitor close to the pin.
	VSS	Input	Ground pin. Connect it to the system power supply (0 V).
Clock	EXTAL	Input	Pins for a crystal resonator. An external clock signal can be input through the EXTAL pin.
	XTAL	Output	
	CLKOUT	Output	Clock output pin
Operating mode control	MD	Input	Pin for setting the operating mode. The signal level on this pin must not be changed during operation mode transition on release from the reset state.
System control	RES	Input	Reset signal input pin. The MCU enters the reset state when this signal goes low.
CAC	CACREF	Input	Measurement reference clock input pin
On-chip emulator	TMS	Input	On-chip emulator or boundary scan pins
	TDI	Input	
	TCK	Input	
	TDO	Output	
	TCLK	Output	Output clock for synchronization with the trace data
	TDATA0 to TDATA3	Output	Trace data output
	SWO	Output	Serial wire trace output pin
	SWDIO	I/O	Serial wire debug data input/output pin
	SWCLK	Input	Serial wire clock pin
Interrupt	NMI	Input	Non-maskable interrupt request pin
	IRQn	Input	Maskable interrupt request pins
	IRQn-DS	Input	Maskable interrupt request pins that can also be used in Deep Software Standby mode
KINT	KR00 to KR07	Input	Key interrupt input pins

Table 1.14 Pin functions (2 of 3)

Function	Signal	I/O	Description
GPT	GTETRGA, GTETRGB, GTETRGC, GTETRGD	Input	External trigger input pins
	GTIOCnA, GTIOCnB	I/O	Input capture, output compare, or PWM output pins
	GTADSM0, GTADSM1	Output	A/D conversion start request monitoring output pins
	GTCPPO0 to GTCPPO4, GTCPPO7	Output	Toggle output synchronized with PWM period
	GTIU	Input	Hall sensor input pin U
	GTIV	Input	Hall sensor input pin V
	GTIW	Input	Hall sensor input pin W
	GTOUUP	Output	3-phase PWM output for BLDC motor control (positive U phase)
	GTOULO	Output	3-phase PWM output for BLDC motor control (negative U phase)
	GTOVUP	Output	3-phase PWM output for BLDC motor control (positive V phase)
	GTOVLO	Output	3-phase PWM output for BLDC motor control (negative V phase)
	GTOWUP	Output	3-phase PWM output for BLDC motor control (positive W phase)
	GTOWLO	Output	3-phase PWM output for BLDC motor control (negative W phase)
AGT	AGTEEn	Input	External event input enable signals
	AGTIOn	I/O	External event input and pulse output pins
	AGTOn	Output	Pulse output pins
	AGTOAn	Output	Output compare match A output pins
	AGTOBn	Output	Output compare match B output pins
SCI	SCKn	I/O	Input/output pins for the clock (clock synchronous mode)
	RXDn	Input	Input pins for received data (asynchronous mode/clock synchronous mode)
	TXDn	Output	Output pins for transmitted data (asynchronous mode/clock synchronous mode)
	CTSn_RTSn	I/O	Input/output pins for controlling the start of transmission and reception (asynchronous mode/clock synchronous mode), active-low.
	CTSn	Input	Input for the start of transmission.
	DEn	Output	Output pins for Driver Enable signal
	SCLn	I/O	Input/output pins for the IIC clock (simple IIC mode)
	SDAn	I/O	Input/output pins for the IIC data (simple IIC mode)
	SCKn	I/O	Input/output pins for the clock (simple SPI mode)
	MISOn	I/O	Input/output pins for slave transmission of data (simple SPI mode)
	MOSIn	I/O	Input/output pins for master transmission of data (simple SPI mode)
	SSn	Input	Chip-select input pins (simple SPI mode), active-low
IIC	SCLn	I/O	Input/output pins for the clock
	SDAn	I/O	Input/output pins for data
SPI	RSPCKA, RSPCKB	I/O	Clock input/output pin
	MOSIA, MOSIB	I/O	Input or output pins for data output from the master
	MISOA, MISOB	I/O	Input or output pins for data output from the slave
	SSLA0, SSLB0	I/O	Input or output pin for slave selection
	SSLA1 to SSLA3, SSLB1 to SSLB3	Output	Output pins for slave selection

Table 1.14 Pin functions (3 of 3)

Function	Signal	I/O	Description
CANFD	CRX0	Input	Receive data
	CTX0	Output	Transmit data
Analog power supply	AVCC0	Input	Analog voltage supply pin. This is used as the analog power supply for the respective modules. Supply this pin with the same voltage as the VCC pin.
	AVSS0	Input	Analog ground pin. This is used as the analog ground for the respective modules. Supply this pin with the same voltage as the VSS pin.
	VREFH0	Input	Analog reference voltage supply pin for the ADC. Connect this pin to AVCC0 when not using the ADC.
	VREFL0	Input	Analog reference ground pin for the ADC. Connect this pin to AVSS0 when not using the ADC.
ADC	AN000 to AN028	Input	Input pins for the analog signals to be processed by the A/D converter.
	PGAIN0 to PGAIN3	Input	Pseudo-differential input pins of programmable gain amplifier (Signal source side)
	PGAVSS0 to PGAVSS3	Input	Pseudo-differential input pins of programmable gain amplifier (reference ground side)
	PGAOUT0 to PGAOUT3	Output	Monitor output pins of programmable gain amplifier
	ADTRGm	Input	Input pins for the external trigger signals that start the A/D conversion, active-low.
DAC12	DAn	Output	Output pins for the analog signals processed by the D/A converter.
ACMPHS	VCOUT	Output	Comparator output pin (OR output of all units)
	CMPOUTm	Output	Comparator output pin (m:unit number)
	CMPOUT012	Output	Comparator output pin (OR output of units 0, 1 and 2)
	IVREF0, IVREF1	Input	Reference voltage input pins for comparator
	IVCMPm0, IVCMPm2, IVCMPm3	Input	Analog voltage input pins for comparator (m:unit number)
I/O ports	P201, P212, P213, PA08 to PA15, PB03 to PB10, PB12 to PB15, PC06 to PC12, PC14, PC15, PD00 to PD15, PE00 to PE06, PE08 to PE15	I/O	General-purpose input/output pins
	P000, P001, P002, PA00 to PA07, PB00 to PB02, PC00 to PC05, PC13	Input	General-purpose input pins

1.6 Pin Assignments

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

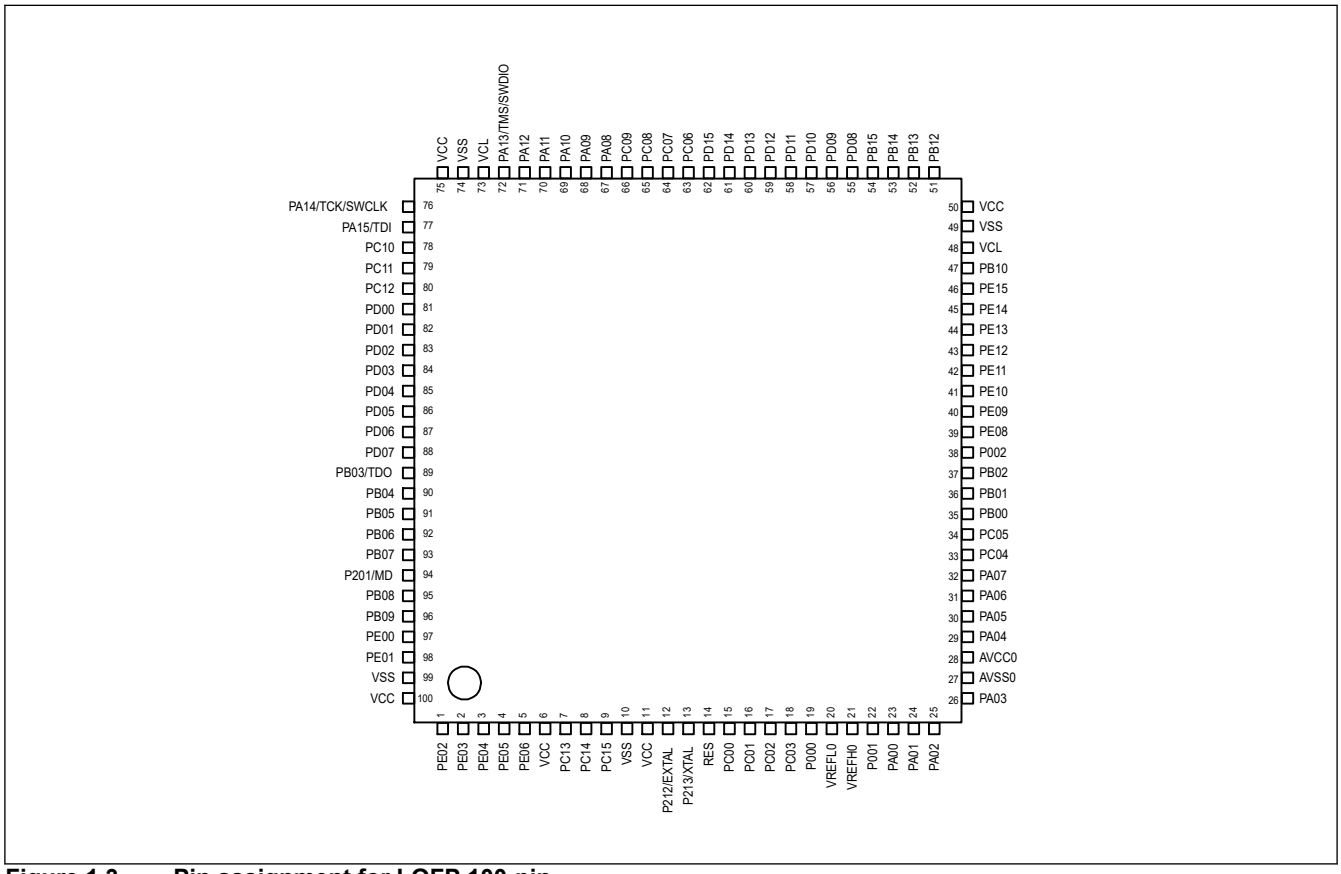


Figure 1.3 Pin assignment for LQFP 100-pin

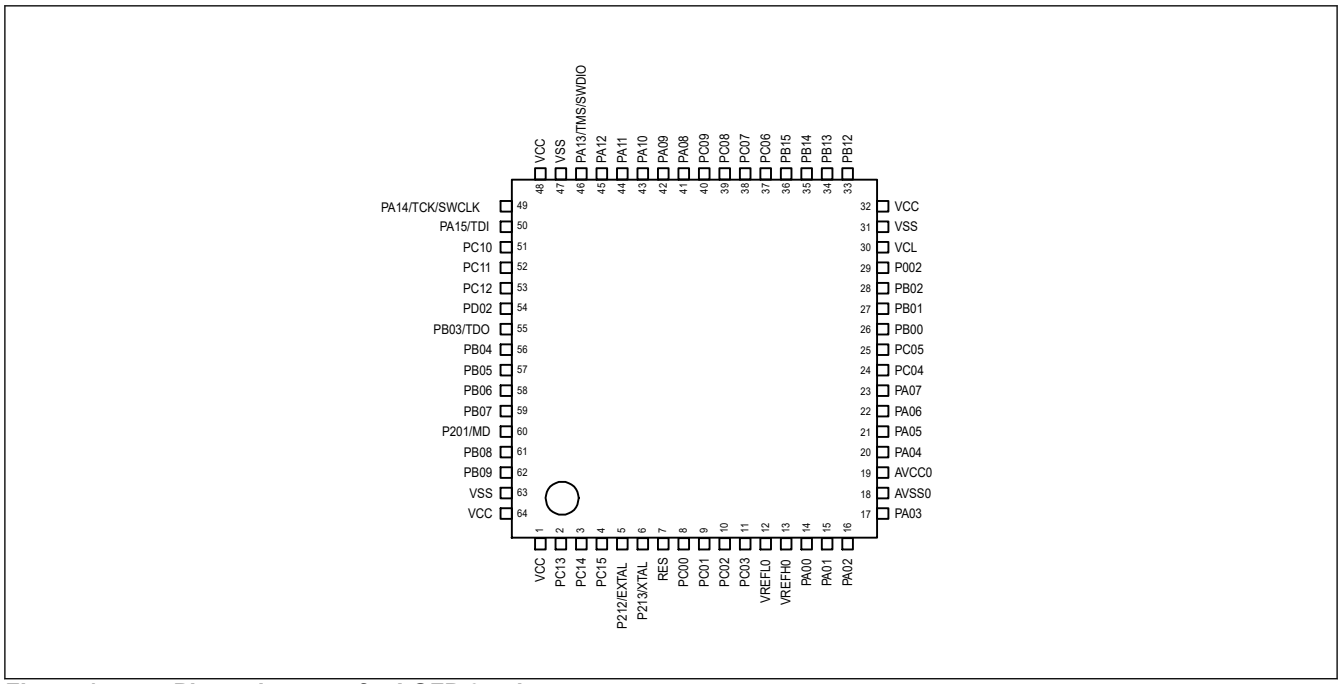


Figure 1.4 Pin assignment for LQFP 64-pin

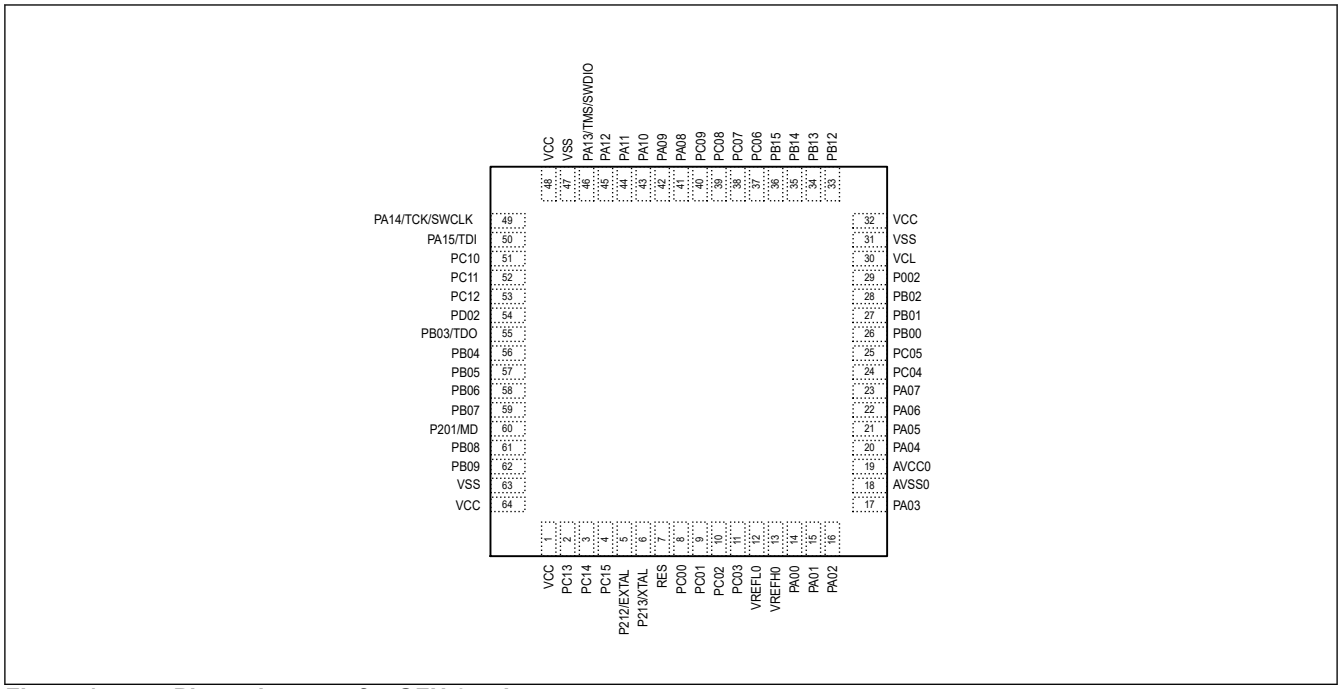


Figure 1.5 Pin assignment for QFN 64-pin

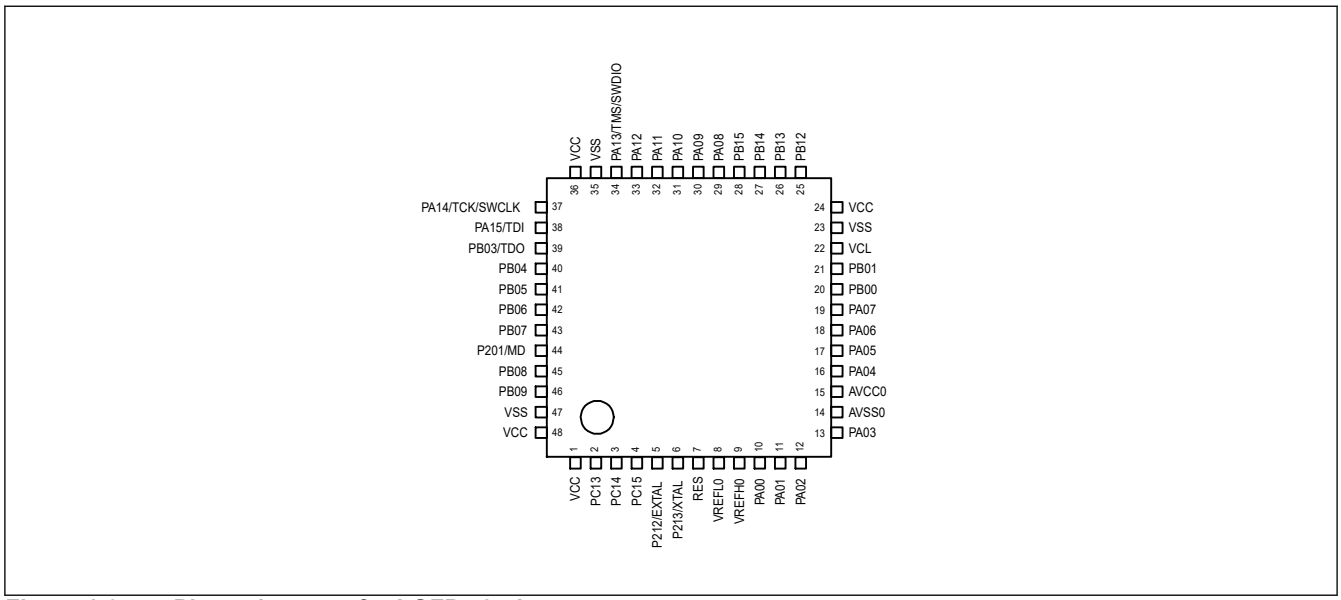


Figure 1.6 Pin assignment for LQFP 48-pin

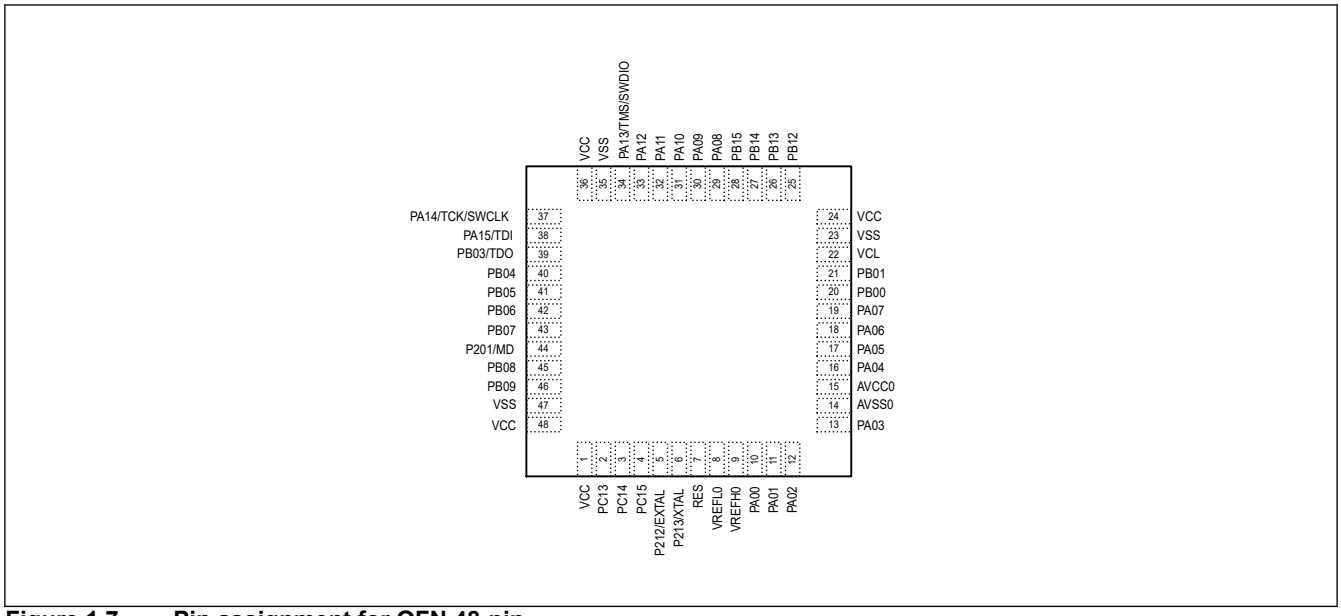


Figure 1.7 Pin assignment for QFN 48-pin

1.7 Pin Lists

Table 1.15 Pin list (1 of 3)

LQFP100	LQFP64, QFN64	LQFP48, QFN48	Power, System, Clock, Debug, CAC	I/O ports	Ex. Interrupt/KINT	SCI/IIC/SPI/CANFD	GPT/AGT	ADC/DAC12/ACMPHS
1	—	—	CLKOUT/TCLK	PE02	—	SCK0_B/DE0/SCK3_A/DE3/RSPCKB_C	GTOVLO/GTIOC7B/GTIOC8A	CMPOUT0
2	—	—	TDATA0	PE03	—	RXD0_B/MISO0_B/SCL0/CTS3_A/SSLB0_C	GTOWLO/GTIOC8A/GTIOC9A	CMPOUT1
3	—	—	TDATA1	PE04	—	TXD0_B/MOSI0_B/SDA0/CTS3_RTS3/SS3_A/DE3/ SSLB1_C	GTOUUP/GTIOC8B/GTIOC7B	CMPOUT2
4	—	—	TDATA2	PE05	—	CTS0_RTS0/SS0_B/DE0/RXD3_A/MISO3_A/SCL3/ MISOB_C	GTOVUP/GTIOC9A/GTIOC8B/ GTCPPO2	CMPOUT3
5	—	—	TDATA3	PE06	—	CTS0_B/TXD3_A/MOSI3_A/SDA3/MOSIB_C	GTOWUP/GTIOC9B/GTCPPO3	—
6	1	1	VCC	—	—	—	—	—
7	2	2	—	PC13	NMI	—	GTETRGD	—
8	3	3	—	PC14	IRQ14	—	GTETRGA/GTIOC3A/GTCPPO0/ GTADSM0/GTCPPO4/AGTIO0	ADTRG0/CMPOUT012
9	4	4	—	PC15	IRQ15	—	GTETRGB/GTIOC3B/GTCPPO1/ GTADSM1/GTCPPO7/AGTIO1	ADTRG1/CMPOUT3
10	—	—	VSS	—	—	—	—	—
11	—	—	VCC	—	—	—	—	—
12	5	5	EXTAL	P212	—	—	—	—
13	6	6	XTAL	P213	IRQ0	—	—	—
14	7	7	RES	—	—	—	—	—
15	8	—	—	PC00	IRQ11-DS	—	—	AN012/PGAOUT0/IVCMP00
16	9	—	—	PC01	IRQ12-DS	—	—	AN013/PGAOUT1/IVCMP10
17	10	—	—	PC02	IRQ13-DS	—	—	AN014/PGAOUT2/IVCMP20
18	11	—	—	PC03	IRQ14-DS	—	—	AN015/PGAOUT3/IVCMP30
19	—	—	—	P000	IRQ0	—	—	AN016/IVREF0
20	12	8	VREFL0	—	—	—	—	—
21	13	9	VREFH0	—	—	—	—	—
22	—	—	—	P001	IRQ2	—	—	AN017/IVREF1
23	14	10	—	PA00	IRQ0-DS	—	—	AN000/PGAIN0/IVCMP02/ IVCMP03
24	15	11	—	PA01	IRQ1	—	—	AN001/PGAVSS0
25	16	12	—	PA02	IRQ2	—	—	AN002/PGAIN1/IVCMP12/ IVCMP13
26	17	13	—	PA03	IRQ3	—	—	AN003/PGAVSS1
27	18	14	AVSS0	—	—	—	—	—
28	19	15	AVCC0	—	—	—	—	—
29	20	16	—	PA04	IRQ4	—	—	AN004/PGAIN2/IVCMP22/ IVCMP23
30	21	17	—	PA05	IRQ5	—	—	AN005/PGAVSS2
31	22	18	—	PA06	IRQ6	—	—	AN006/DA0
32	23	19	—	PA07	IRQ7	—	—	AN007/DA1
33	24	—	—	PC04	IRQ10	—	—	AN010/DA2
34	25	—	—	PC05	IRQ11	—	—	AN011/DA3
35	26	20	—	PB00	IRQ0	—	—	AN008/PGAOUT0/PGAOUT2
36	27	21	—	PB01	IRQ1	—	—	AN009/PGAOUT1/PGAOUT3
37	28	—	—	PB02	IRQ15-DS	—	—	AN018/PGAIN3/IVCMP32/ IVCMP33
38	29	—	—	P002	—	—	—	AN019/PGAVSS3
39	—	—	—	PE08	KR00	SSLA3_C	GTIV/GTIOC3A/GTETRGC/ GTADSM0	AN020/ADTRG0/ CMPOUT012
40	—	—	CACREF	PE09	KR01	SSLA2_C	GTIW/GTIOC3B/GTETRGD/ GTADSM1	AN021/ADTRG1/CMPOUT3
41	—	—	—	PE10	KR02	SSLA1_C	GTOULO/GTIOC2A/GTIOC4A/ GTIOC7A	AN022

Table 1.15 Pin list (2 of 3)

LQFP100	LQFP64, QFN64	LQFP48, QFN48	Power, System, Clock, Debug, CAC	I/O ports	Ex. Interrupt/KINT	SCI/IIC/SPI/CANFD	GPT/AGT	ADC/DAC12/ACMPHS
42	—	—	—	PE11	KR03	SSLA0_C	GTOUUP/GTIOC2B/GTIOC5A/ GTIOC8A	AN023
43	—	—	—	PE12	KR04	RSPCKA_C	GTOVLO/GTIOC1A/GTIOC6A/ GTIOC9A	AN024
44	—	—	—	PE13	KR05	MISOA_C	GTOVUP/GTIOC1B/GTIOC4B/ GTIOC7B	AN025
45	—	—	—	PE14	KR06	MOSIA_C	GTOWLO/GTIOC0A/GTIOC5B/ GTIOC8B	AN026
46	—	—	—	PE15	KR07	RXD4_A/MISO4_A/SCL4	GTOWUP/GTIOC0B/GTIOC6B/ GTIOC9B	AN027
47	—	—	CACREF/VCOUT	PB10	IRQ10-DS	TXD4_A/MOSI4_A/SDA4/CTS3_B	GTIU/GTETRGA/GTETRGB/ GTCPPO4/GTCPPO7	AN028
48	30	22	VCL	—	—	—	—	—
49	31	23	VSS	—	—	—	—	—
50	32	24	VCC	—	—	—	—	—
51	33	25	—	PB12	IRQ2	SCK4_A/DE4/RXD3_B/MISO3_B/SCL3/SSLB0_A/ CRX0	GTETRGA/GTIOC0A/GTIOC4A	ADTRG0
52	34	26	—	PB13	IRQ3	CTS4_A/TXD3_B/MOSI3_B/SDA3/RSPCKB_A/CTX0	GTOULO/GTIOC0B/GTIOC7A/ GTIOC5A	—
53	35	27	—	PB14	IRQ4	CTS4_RTS4/SS4_A/DE4/SCK3_B/DE3/SDA0_C/ MISOB_A	GTOVLO/GTIOC1A/GTIOC8A/ GTIOC6A	—
54	36	28	—	PB15	IRQ5	RXD4_A/MISO4_A/SCL4/CTS3_RTS3/SS3_B/DE3/ SCL0_C/MOSIB_A	GTOWLO/GTIOC1B/GTIOC9A/ GTIOC4B	—
55	—	—	—	PD08	KR00	CTS2_B/TXD1_A/MOSI1_A/SDA1/SSLB1_A	GTIOC2A	—
56	—	—	—	PD09	KR01	CTS2_RTS2/SS2_B/DE2/RXD1_A/MISO1_A/SCL1/ SSLB2_A	GTIOC2B	—
57	—	—	—	PD10	KR02	SCK2_C/DE2/SCK1_A/DE1/SSLB3_A	GTETRGC/GTIOC3A	—
58	—	—	—	PD11	KR03	RXD2_C/MISO2_C/SCL2/CTS1_A	GTIOC3B	—
59	—	—	—	PD12	IRQ12/KR04	TXD2_C/MOSI2_C/SDA2/CTS1_RTS1/SS1_A/DE1/ SCL1_D	GTIOC4A	—
60	—	—	—	PD13	IRQ13/KR05	SCK4_C/DE4/SCK9_C/DE9/SDA1_D	GTIOC4B	—
61	—	—	—	PD14	IRQ14/KR06	RXD4_C/MISO4_C/SCL4/RXD9_C/MISO9_C/SCL9/ SCL0_F	GTIOC5A	—
62	—	—	—	PD15	IRQ15/KR07	TXD4_C/MOSI4_C/SDA4/TXD9_C/MOSI9_C/ SDA9/DE9/SDA0_F	GTIOC5B	—
63	37	—	—	PC06	IRQ6	TXD2_B/MOSI2_B/SDA2/CTS9_RTS9/SS9_C/DE9/ SCL1_E	GTETRGD/GTIOC6A/GTIOC5B/ AGTO0	—
64	38	—	—	PC07	IRQ7	RXD2_B/MISO2_B/SCL2/CTS9_C/SDA1_E	GTETRGA/GTIOC6B/AGTEE0	—
65	39	—	CACREF	PC08	IRQ8	SCK2_B/DE2/CTS3_RTS3/SS3_C/DE3/SCL0_E/ SSLA3_B	GTIV/GTIOC7A/AGTOA0	—
66	40	—	CLKOUT	PC09	IRQ9	CTS2_RTS2/SS2_B/DE2/CTS3_C/SDA0_D/SDA0_E/ SSLA2_B	GTIW/GTIOC7B/GTIOC8A/AGTOB0	—
67	41	29	CLKOUT	PA08	IRQ8/KR00	SCK0_A/DE0/SCK1_C/DE1/SCL0_D/SSLA1_B	GTOUUP/GTIOC8A/GTIOC7B/ GTIOC2A/GTIOC9A/AGTIO0	CMPOUT2
68	42	30	—	PA09	IRQ9/KR01	TXD0_A/MOSI0_A/SDA0/SCL1_C/SSLA0_B	GTOVUP/GTIOC8B/GTIOC8B/ GTIOC2B/GTIOC7B	CMPOUT3
69	43	31	—	PA10	IRQ10/KR02	RXD0_A/MISO0_A/SCL0/SDA1_C/RSPCKA_B	GTOWUP/GTIOC9A/GTIOC9B/ GTIOC3A/GTIOC8B	CMPOUT0
70	44	32	—	PA11	IRQ11/KR03	CTS0_A/RXD1_C/MISO1_C/SCL1/MOSIA_B/CTX0	GTETRGD/GTIOC9B/GTETRGC/ GTIOC3B	CMPOUT1
71	45	33	CACREF	PA12	IRQ12/KR04	CTS0_RTS0/SS0_A/DE0/TXD1_C/MOSI1_C/SDA1/ MISOA_B/CRX0	GTETRGB/GTCPPO0/GTCPPO2/ GTADSM0/GTCPPO7	ADTRG1
72	46	34	TMS/SWDIO	PA13	—	SCK0_C/DE0/CTS1_RTS1/SS1_C/DE1	AGTO0	—
73	—	—	VCL	—	—	—	—	—
74	47	35	VSS	—	—	—	—	—
75	48	36	VCC	—	—	—	—	—
76	49	37	TCK/SWCLK	PA14	—	TXD0_C/MOSI0_C/SDA0/SCK9_B/DE9	AGTO1	—
77	50	38	TDI	PA15	IRQ1/KR02	RXD0_C/MISO0_C/SCL0/RXD9_B/MISO9_B/SCL9/ SSLA0_A	GTETRGB/GTADSM1/GTCPPO4	ADTRG0/CMPOUT012
78	51	—	—	PC10	IRQ6-DS/KR05	TXD1_B/MOSI1_B/SDA1/SCL0_B/RSPCKB_B	AGTIO1	CMPOUT0

Table 1.15 Pin list (3 of 3)

LQFP100	LQFP64, QFN64	LQFP48, QFN48	Power, System, Clock, Debug, CAC	I/O ports	Ex. Interrupt/KINT	SCI/IIC/SPI/CANFD	GPT/AGT	ADC/DAC12/ACMPHS
79	52	—	—	PC11	IRQ7-DS/KR06	RXD1_B/MISO1_B/SCL1/SDA0_B/MISOB_B	AGTOA1	CMPOUT1
80	53	—	—	PC12	IRQ8-DS/KR07	TXD4_B/MOSI4_B/SDA4/SCK1_B/DE1/MOSIB_B	AGTOB1	CMPOUT2
81	—	—	—	PD00	KR00	CTS2_A/RXD3_C/MISO3_C/SCL3/SSLB0_B/CRX0	GTADSM0/GTCPPO4	—
82	—	—	—	PD01	KR01	CTS2_RTS2/SS2_A/DE2/TXD3_C/MOSI3_C/SDA3/ SSLB1_B/CTX0	GTADSM1/GTCPPO7	—
83	54	—	CLKOUT	PD02	IRQ9-DS/KR02	RXD4_B/MISO4_B/SCL4/SCK3_C/DE3	GTCPPO0/GTCPPO2/AGTEE1	CMPOUT3
84	—	—	—	PD03	KR03	SCK4_B/DE4/CTS9_A/SSLB2_B	GTCPPO0	CMPOUT0
85	—	—	—	PD04	KR04	CTS4_RTS4/SS4_B/DE4/CTS9_RTS9/SS9_A/DE9/ SSLB3_B	GTCPPO1	CMPOUT1
86	—	—	—	PD05	KR05	TXD9_A/MOSI9_A/SDA9/SDA1_B/SSLA3_A	GTADSM0/GTCPPO3	—
87	—	—	—	PD06	KR06	RXD9_A/MISO9_A/SCL9/SCL1_B/SSLA2_A	GTCPPO4	—
88	—	—	—	PD07	KR07	SCK9_A/DE9/SSLA1_A	GTADSM1/GTCPPO7	—
89	55	39	TDO/SWO	PB03	IRQ0/KR03	TXD2_A/MOSI2_A/SDA2/TXD9_B/MOSI9_B/SDA9/ RSPCKA_A/CRX0	GTIOC4A/GTCPPO1/GTCPPO3/ AGTO1	ADTRG1/CMPOUT3
90	56	40	CACREF/VCOUT	PB04	IRQ13/KR04	RXD2_A/MISO2_A/SCL2/RXD3_D/MISO3_D/SCL3/ MISOA_A/CTX0	GTIOC4A/GTIOC5A/GTIOC0A/ AGTOA0	—
91	57	41	—	PB05	IRQ3-DS/KR05	SCK2_A/DE2/TXD3_D/MOSI3_D/SDA3/MOSIA_A/ CRX0	GTIU/GTIOC4B/GTIOC6A/GTIOC0B/ AGTOB0	—
92	58	42	—	PB06	IRQ4-DS/KR06	TXD0_D/MOSI0_D/SDA0/CTS3_RTS3/SS3_D/DE3/ SCL0_A/CTX0	GTIV/GTIOC5A/GTIOC4B/GTIOC1A/ AGTOA1	—
93	59	43	—	PB07	IRQ5-DS/KR07	RXD0_D/MISO0_D/SCL0/CTS1_RTS1/SS1_D/DE1/ SDA0_A	GTIW/GTIOC5B/GTETRGC/ GTIOC1B/AGTOB1	—
94	60	44	MD	P201	—	—	—	—
95	61	45	—	PB08	IRQ1-DS/KR00	RXD4/MISO4_C/SCL4/RXD1_D/MISO1_D/SCL1/ SCL1_A/CRX0	GTIOC6A/GTIOC5B/GTIOC2A/ AGTIO0	—
96	62	46	—	PB09	IRQ2-DS/KR01	TXD4/MOSI4_C/SDA4/TXD1_D/MOSI1_D/SDA1/ SDA1_A/CTX0	GTIOC6B/GTIOC2B/AGTIO1	—
97	—	—	CACREF	PE00	—	TXD0_E/MOSI0_E/SDA0/TXD9_D/MOSI9_D/SDA9/ SSLB3_C	GTETRGA/GTIOC4A/GTADSM0/ AGTEE0	ADTRG0
98	—	—	—	PE01	—	RXD0_E/MISO0_E/SCL0/RXD9_D/MISO9_D/SCL9/ SSLB2_C	GTOULO/GTIOC7A/GTIOC4B/ GTADSM1/AGTEE1	ADTRG1
99	63	47	VSS	—	—	—	—	—
100	64	48	VCC	—	—	—	—	—

Note: Several pin names have the added suffix of _A, _B, _C, _D, _E and _F. The suffix can be ignored when assigning functionality.

2. CPU

The MCU is based on the Arm® Cortex®-M33 core.

2.1 Overview

2.1.1 CPU

- Arm Cortex-M33
  - Revision: r0p4-00rel1
  - Armv8-M architecture profile
  - Single Precision Floating-Point Unit compliant with the ANSI/IEEE Std 754-2008
- SAU (Security Attribution Unit): 0 region
- IDAU (Implementation Defined Attribution Unit): 8 regions
  - Code flash (secure/non-secure callable/non-secure)
  - Data Flash (secure/non-secure)
  - SRAM0 (secure/non-secure callable/non-secure)
- Memory Protection Unit (MPU)
  - Armv8 Protected Memory System Architecture (PMSAv8)
  - Secure MPU (MPU_S): 8 regions
  - Non-secure MPU (MPU_NS): 8 regions
- SysTick timer
  - Two SysTick timers: Secure and Non-secure instance
  - Driven by SysTick timer clock (SYSTICCLK) or system clock (ICLK)

See reference 1. and reference 2. in section 2.14. References for details.

2.1.2 Debug

- Arm® CoreSight™ ETM-M33
  - Revision: r0p2-00rel0
  - ARM ETM Architecture version 4.2
- Instrumentation Trace Macrocell (ITM)
- Data Watchpoint and Trace Unit (DWT)
  - 4 comparators for watchpoints and triggers
- Breakpoint Unit (BPU)
  - Breakpoint function is available.
    - 8 instruction comparators
    - 0 literal comparators
- Time Stamp Generator (TSG)
  - Time stamp for ETM and ITM
  - Driven by CPU clock
- Debug Register Module (DBGREG)
  - Reset control
  - Halt control
- Debug Access Port (DAP)

  - JTAG Debug Port (JTAG-DP)
  - Serial Wire Debug Port (SW-DP)
- Cortex-M33 Trace Port Interface Unit (TPIU)
  - 4 bits TPIU formatter output
  - Serial Wire Output
- Cross Trigger Interface (CTI)
- Embedded Trace Buffer (ETB)
  - CoreSight Trace Memory Controller with ETB configuration
  - Buffer size: 2 KB

See reference 1. and reference 2. in section 2.14. References for details.

2.1.3 Operating Frequency

The operating frequencies for the MCU are as follows:

- CPU: maximum 240 MHz
- 4-bit TPIU trace interface: maximum 60 MHz
- Serial Write Output (SWO) trace interface: maximum 60 MHz
- Joint Test Action Group (JTAG) interface: maximum 25 MHz
- Serial Wire Debug (SWD) interface: maximum 25 MHz

2.1.4 Block Diagram

Figure 2.1 shows a block diagram of the Cortex-M33 core.

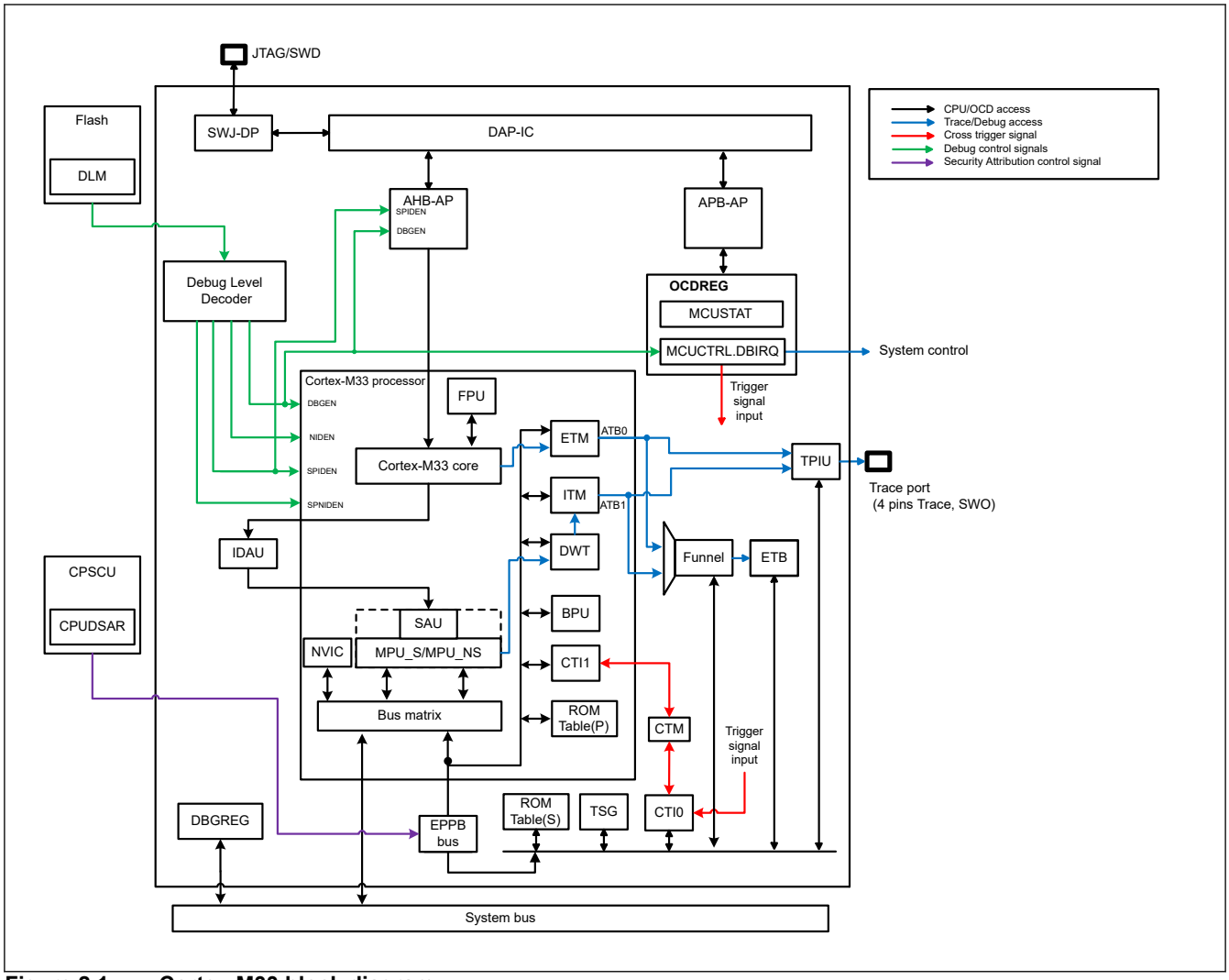


Figure 2.1 Cortex-M33 block diagram

2.2 Implementation Options

Table 2.1 shows the implementation options of the MCU.

Table 2.1 Implementation options (1 of 2)

Option	Implementation
SAU	Not included
IDAU	Included, 8 regions
MPU	Included, 8 regions for Secure and 8 regions for Non-secure
BPU	Included
Cross Trigger Interface (CTI)	Included
DWT	Included
Number of Wakeup Interrupt Controllers (WIC)	Not included ICU can wake up CPU instead of WIC. See section 12, Interrupt Controller Unit (ICU) for details.
TPIU	Included • 4 bits TPIU formatter output • Serial Wire Output
FPU	Included
DSP	Included

Table 2.1 Implementation options (2 of 2)

Option	Implementation
Embedded Trace Macrocell (ETM)	Included
Sleep mode power saving	Sleep mode and other low power modes are supported. For more details, see section 10, Low Power Modes. Note: SCB.SCR.SLEEPDEEP is ignored.
Interrupts	96
Priority bits	4 bits (16 levels)
Endianness	Little-endian
Memory features	Cacheable attribute is utilized in the MCU. See section 13, Buses for the detail.
SysTick	Included
SYST_CALIB register (0x4000_0147)	Bit [31] = 0 Reference clock provided Bit [30] = 1 TENMS value is inexact Bits [29:24] = 0x00 Reserved Bits [23:0] = 0x000147 TENMS: (32768 × 10 ms) - 1/32.768 kHz = 326.66 decimal = 327 with skew = 0x000147
Event input/output	Not implemented
Global exclusive monitor	Not implemented
System reset request output	The SYSRESETREQ bit in Application Interrupt and Reset Control Register causes a CPU reset

2.3 Trace Interface

A Trace Port Interface Unit (TPIU) and Serial Wire Output (SWO) provide trace output. Table 2.2 shows the MCU pins for the function. These pins are multiplexed with other functions.

Table 2.2 Trace function pins

Name	I/O	Function	When not in use
TCLK	Output	Trace clock	Open
TDATA0	Output	Trace data output 0	Open
TDATA1	Output	Trace data output 1	Open
TDATA2	Output	Trace data output 2	Open
TDATA3	Output	Trace data output 3	Open
TDO/SWO	Output	Serial wire output multiplexed with JTAG TDO pin	Open

2.4 JTAG/SWD Interface

Table 2.3 shows the JTAG/SWD pins.

Table 2.3 JTAG/SWD pins

Name	I/O	Function	When not in use
TDI	Input	JTAG TDI pin	Pull-up
TDO/SWO	Output	JTAG TDO pin multiplexed with serial wire output	Open
TCK/SWCLK	Input	JTAG clock pin Serial wire clock pin	Pull-up
TMS/SWDIO	I/O	JTAG TMS pin Serial wire data I/O pin	Pull-up

2.5 Security Attribution for Memory

In this MCU, SAU is not implemented and IDAU performs region definition for memory. IDAU divides the memory into 8 different areas as shown in Figure 2.2.

The code flash, the data flash, and the SRAM are divided into Secure (S), Non-secure (NS) and Non-secure callable (NSC) regions. These memory security attributions are set into the nonvolatile memory by the serial programming command when the device lifecycle is in SSD state. These memory security attributions are loaded into the IDAU and the memory controller before application execution. These memory security attributions cannot be updated by application but can read through the dedicated registers.

Note: When configuring, the memory regions should satisfy the setting condition of minimum address unit shown in Table 2.4.

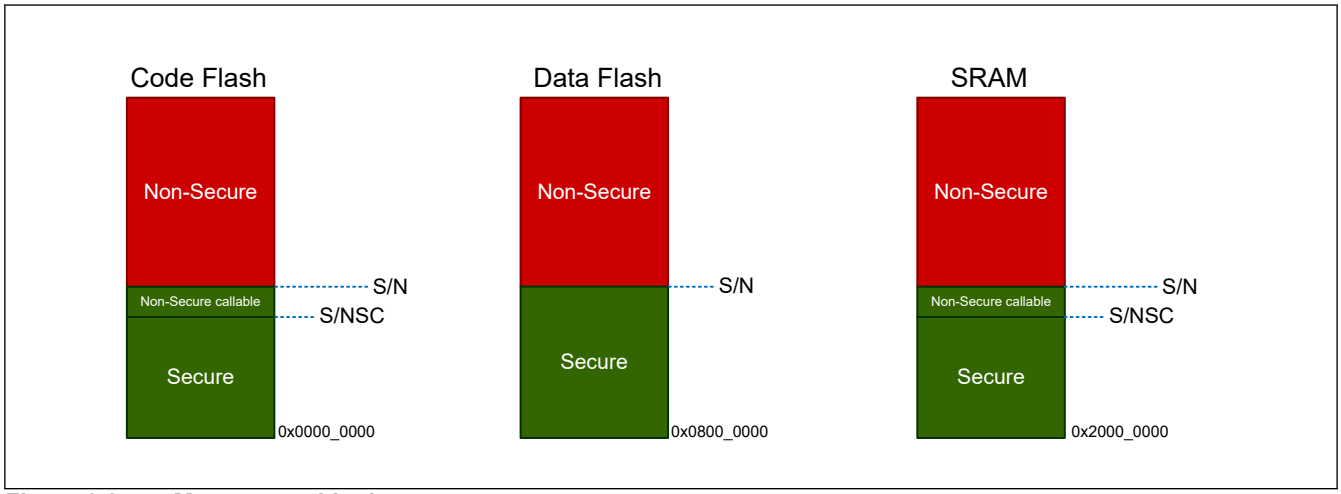


Figure 2.2 Memory partitioning

Table 2.4 S/NS and S/NSC boundary list

Boundary	Code flash	Data flash	SRAM
S/NS	32 KB	1 KB	8 KB
S/NSC	1 KB	—	1 KB

Each region has its dedicated ID as follows. For more details, see section 2.14. References.

IREGION (IDAU region number)	Description
0x0D	Non-secure SRAM
0x0E	Non-secure callable SRAM
0x0F	Secure SRAM
0x09	Non-secure data flash
0x0B	Secure data flash
0x05	Non-secure code flash
0x06	Non-secure callable code flash
0x07	Secure code flash

2.6 Debug Function

2.6.1 Debugger connectivity

In this MCU, debug function is considered in three levels, DBG0, DBG1, DBG2. At DBG0, no debug function is available. DBG1 level is defined as non-secure debug in ARMv-8 and the debugger can only access defined non-secure debug accessible regions. DBG2 level is defined as secure debug in ARMv-8 and at this level, nonsecure and secure debug function is enabled and can be accessible from the debugger.

Debug level is determined by the Device Lifecycle Management (DLM) state of the product.

See Figure 2.1 for debugger accessible regions.

Table 2.5 shows the CPU debug function and conditions.

Table 2.5 CPU debug function and conditions

Condition			Permitted debug function
OCD connect*1	DLM State	Debug level	Description
Connected	CM	DBG2	All debug functions are available
Connected	SSD	DBG2	All debug functions are available
Connected	NSECSD	DBG1	Only Non-secure debug function is available
Connected	DPL	DBG0	Debugger connection is not available
Connected	LCK_DBG	DBG0	Debugger connection is not available
Connected	LCK_BOOT	DBG0	Debugger connection is not available
Connected	RMA_REQ	DBG0	Debugger connection is not available
Connected	RMA_ACK	DBG2	All debug functions are available

Note 1. OCD connect is determined by the CDBGPWRUPREQ bit output in the SWJ-DP register. The bit can only be written by the OCD. However, the level of the bit can be confirmed by reading the DBGSTR.CDBGPWRUPREQ bit.

2.6.2 Emulator Connection

Renesas provides the emulator which supports both debugging using SWD or JTAG communication and serial programming using SCI communication. This emulator makes it easy to switch between debug-ging and serial programming.

Table 2.6 shows the pinout of 10 pin or 20 pin socket pinouts when using this emulator. The pinout of SWD and JTAG is ARM standard, and MD, TXD, RXD pins are added for the serial programming using SCI communication.

The serial programming interface must be used to program the TrustZone IDAU boundary register settings.

It is recommended to connect PA14/SWCLK/TCK and P201/MD pins using wired OR circuit on the board to use both debugging and serial programming.

Table 2.6 Pin assign for emulator

Pin No.	SWD	JTAG	Serial Programming using SCI
1	VCC	VCC	VCC
2	PA13/SWDIO	PA13/TMS	NC
4	PA14/SWCLK Wired OR with P201/MD	PA14/TCK Wired OR with P201/MD	P201/MD
6	PB03/SWO/TXD9	PB03/SWO/TXD9	PB03/TXD9
8	PA15/RXD9	PA15/TDI/RXD9	PA15/RXD9
9	GNDdetect	GNDdetect	GNDdetect
10	nRESET	nRESET	nRESET
12	PE02/TCLK	PE02/TCLK	NC
14	PE03/TDATA[0]	PE03/TDATA[0]	NC
16	PE04/TDATA[1]	PE04/TDATA[1]	NC
18	PE05/TDATA[2]	PE05/TDATA[2]	NC
20	PE06/TDATA[3]	PE06/TDATA[3]	NC
3, 5, 15, 17, 19	GND	GND	GND
7	NC	NC	NC
11, 13	NC	NC	NC

2.6.3 Self-Hosted Debug Function

As described in section 2.7.6. CPUDSAR : CPU Debug Security Attribution Register, at the initial setting access from the CPU in non-secure state to CoreSight debug components is protected, that is, the non-secure access to CoreSight debug components from the self-hosted debugger is not allowed when the debug level is DBG2 at the initial setting. Therfore, the CPUDSAR.CPUDSA0 must be set to 1 to enable the self-hosted debug for CPU in non-secure state.

Note: There is no restriction for the self-hosted debug function while the CPU is in the secure state.

2.6.4 Effect of Debug Function

The debug function effects inside and outside of CPU.

2.6.4.1 Low power mode

All CoreSight debug components can store the register settings even when the CPU enters Software Standby, Snooze or Deep Software Standby mode. However, AHB-AP cannot respond to On-Chip Debug (OCD) access in these low power modes. The OCD must wait for cancellation of the low power mode to access the CoreSight debug components. To request low power mode cancellation, the OCD can set the DBIRQ bit in the MCUCTRL register. For details, see section 2.7.5.2. MCUCTRL : MCU Control Register.

2.6.4.2 Reset

In OCD mode, some resets depend on the CPU status and the DBGSTOPCR register setting.

Table 2.7 Reset or interrupt and mode setting

Reset or interrupt name	Control in On-Chip Debug (OCD) mode	
	OCD break mode	OCD run mode
RES pin reset	Same as user mode	
Power-on reset	Same as user mode	
Independent watchdog timer reset/interrupt	Does not occur*1	Depends on DBGSTOPCR setting
Watchdog timer reset/interrupt	Does not occur*1	Depends on DBGSTOPCR setting
Voltage monitor 0 reset	Depends on DBGSTOPCR setting	
Voltage monitor 1 reset/interrupt	Depends on DBGSTOPCR setting	
Voltage monitor 2 reset/interrupt	Depends on DBGSTOPCR setting	
SRAM parity error reset/interrupt	Depends on DBGSTOPCR setting	
SRAM ECC error reset/interrupt	Depends on DBGSTOPCR setting	
Cache parity error reset/interrupt	Depends on DBGSTOPCR setting	
Bus master MPU error reset/interrupt	Same as user mode	
Deep software standby reset	Same as user mode	
Software reset	Same as user mode	

Note: In OCD break mode, the CPU is halted. In OCD run mode, the CPU is in OCD mode and the CPU is not halted.
Note 1. The IWDT and WDT always stop in this mode.

2.7 Programmers Model

2.7.1 Address Spaces

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

- AHB-AP, which is connected to the CPU bus matrix and has the same access to the system address space as the CPU
- APB-AP, which has a dedicated address space (OCD address space) and is connected to the OCDREG registers.

Figure 2.3 shows a block diagram of the AP connection and address spaces.

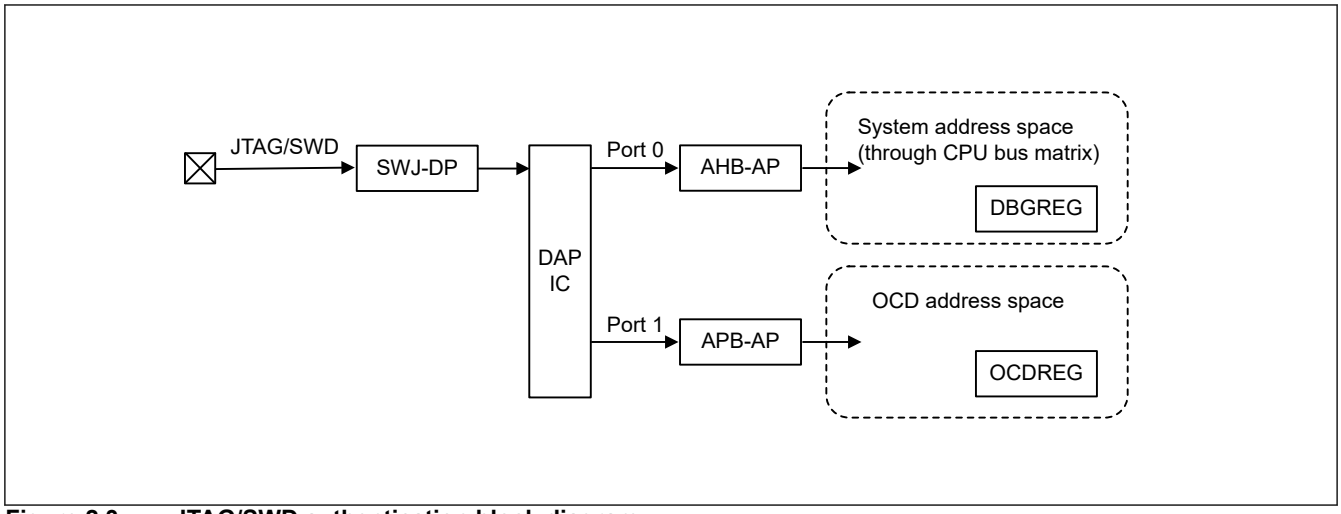


Figure 2.3 JTAG/SWD authentication block diagram

For debugging purposes, there are two register modules, DBGREG and OCDREG. DBGREG is located in the system address space and can be accessed from the OCD emulator, the CPU, and other bus masters in the MCU. OCDREG is located in the OCD address space and can only be accessed from the OCD tool. The CPU and other bus masters cannot access OCDREG.

2.7.2 Peripheral Address Map

In system address space, the Cortex-M33 core has a Private Peripheral Bus (PPB) which can be accessed only from CPU and OCD emulator. The PPB is expanded from the original implementation of the Cortex-M33 core for this MCU. Table 2.8 shows the address map of the MCU.

Table 2.8 Peripheral address map

Component name	Start address	End address	Note
ITM	0xE000_0000	0xE000_0FFF	See reference 2. in section 2.14. References
DWT	0xE000_1000	0xE000_1FFF	See reference 2. in section 2.14. References
BPU	0xE000_2000	0xE000_2FFF	See reference 2. in section 2.14. References
Secure SCS/SCS	0xE000_E000	0xE000_EFFF	See reference 1. in section 2.14. References
Non-Secure SCS	0xE002_E000	0xE002_EFFF	See reference 2. in section 2.14. References
TPIU	0xE004_0000	0xE004_0FFF	See reference 3. in section 2.14. References
ETM	0xE004_1000	0xE004_1FFF	See reference 1. in section 2.14. References
CTI1	0xE004_2000	0xE004_2FFF	See reference 2. in section 2.14. References
CTI0	0xE004_4000	0xE004_4FFF	See reference 4. in section 2.14. References
ATB Funnel	0xE004_7000	0xE004_7FFF	See section 2.9. CoreSight ATB Funnel and reference 4. in section 2.14. References
ETB	0xE004_8000	0xE004_8FFF	See reference 4. in section 2.14. References
Time Stamp Generator	0xE004_9000	0xE004_9FFF	See section 2.11. CoreSight Time Stamp Generator and reference 4. in section 2.14. References
System ROM Table	0xE00F_E000	0xE00F_EFFF	See reference 3. in section 2.14. References
Processor ROM Table	0xE00F_F000	0xE00F_FFFF	See reference 2. in section 2.14. References

2.7.3 CoreSight ROM Table

The MCU contains two CoreSight ROM Tables, the processor and system ROM Tables. The Processor ROM Table contains entries which hold a list of debug components inside the processor. The System ROM Table contains entries of Processor ROM Table and others debug components outside the processor.

2.7.3.1 ROM entries

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

Table 2.9 and Table 2.10 show the System ROM entries and Processor ROM entries. See reference 5. in section 2.14. References for details.

Table 2.9 System ROM entries

#	Address	Access size	R/W	Value	Target module pointer
0	0xE00F_E000	32 bits	R	0xFFF46003	CTI0
1	0xE00F_E004	32 bits	R	0xFFF49003	Funnel
2	0xE00F_E008	32 bits	R	0xFFF4A003	ETB
3	0xE00F_E00C	32 bits	R	0xFFF4B003	TSG
4	0xE00F_E010	32 bits	R	0xFFF42003	TPIU
5	0xE00F_E014	32 bits	R	0x00001003	Processor ROM table
6	0xE00F_E018	32 bits	R	0x00000000	End of entries

Table 2.10 Processor ROM Entries

#	Address	Access size	R/W	Value	Target module pointer
0	0xE00F_F000	32 bits	R	0xFFF0F003	SCS
1	0xE00F_F004	32 bits	R	0xFFF02003	DWT
2	0xE00F_F008	32 bits	R	0xFFF03003	BPU
3	0xE00F_F00C	32 bits	R	0xFFF01003	ITM
4	0xE00F_F014	32 bits	R	0xFFF42003	ETM
5	0xE00F_F018	32 bits	R	0xFFF43003	CTI1
6	0xE00F_F020	32 bits	R	0x00000000	End of entries

2.7.3.2 CoreSight component registers

The CoreSight ROM Table lists the CoreSight component registers defined in the Arm CoreSight architecture.

Table 2.11 shows the registers. See reference 5. in section 2.14. References for details of each register.

Table 2.11 CoreSight component registers in the CoreSight ROM Table

Name	Address	Access size	R/W	Initial value
PID4	0xE00F_EFD0	32 bits	R	0x00000004
PID5	0xE00F_EFD4	32 bits	R	0x00000000
PID6	0xE00F_EFD8	32 bits	R	0x00000000
PID7	0xE00F_EFDC	32 bits	R	0x00000000
PID0	0xE00F_EFE0	32 bits	R	0x0000003E
PID1	0xE00F_EFE4	32 bits	R	0x00000030
PID2	0xE00F_EFE8	32 bits	R	0x0000000A
PID3	0xE00F_EFEC	32 bits	R	0x00000000
CID0	0xE00F_EFF0	32 bits	R	0x0000000D
CID1	0xE00F_EFF4	32 bits	R	0x00000010
CID2	0xE00F_EFF8	32 bits	R	0x00000005
CID3	0xE00F_EFFC	32 bits	R	0x000000B1

2.7.4 DBGREG Module

The DBGREG module controls the debug functionalities and is implemented as a CoreSight-compliant component.

Table 2.12 shows the DBGREG registers other than the CoreSight component registers.

Table 2.12 Non-CoreSight DBGREG registers

Name		DAP port	Address	Access size	R/W
Debug Status Register	DBGSTR	Port 0	0x4001_B000	32 bits	R
Debug Stop Control Register	DBGSTOPCR	Port 0	0x4001_B010	32 bits	R/W

2.7.4.1 DBGSTR : Debug Status Register

Base address: DBG = 0x4001_B000

Offset address: 0x00

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

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

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

The DBGSTR register is a status register which indicates the state of the debug power-up request to the MCU from the emulator.

2.7.4.2 DBGSTOPCR : Debug Stop Control Register

Base address: DBG = 0x4001_B000

Offset address: 0x10

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	DBGSTOP_CPER	—	—	—	—	—	DBGSTOP_RECCR	DBGSTOP_RPER	—	—	—	—	—	DBGSTOP_LVD2	DBGSTOP_LVD1	DBGSTOP_LVD0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
0	DBGSTOP_IWDT	Mask bit for IWDT reset/interrupt in the OCD run mode In the OCD break mode, the reset/interrupt is masked and IWDT counter is stopped, regardless of this bit value. 0: Enable IWDT reset/interrupt 1: Mask IWDT reset/interrupt and stop IWDT counter	R/W
1	DBGSTOP_WDT	Mask bit for WDT reset/interrupt in the OCD run mode In the OCD break mode, the reset/interrupt is masked and WDT counter is stopped, regardless of this bit value. 0: Enable WDT reset/interrupt 1: Mask WDT reset/interrupt and stop WDT counter	R/W
15:2	—	These bits are read as 0. The write value should be 0.	R/W
16	DBGSTOP_LVD0	Mask bit for LVD0 reset 0: Enable LVD0 reset 1: Mask LVD0 reset	R/W
17	DBGSTOP_LVD1	Mask bit for LVD1 reset/interrupt 0: Enable LVD1 reset/interrupt 1: Mask LVD1 reset/interrupt	R/W
18	DBGSTOP_LVD2	Mask bit for LVD2 reset/interrupt 0: Enable LVD2 reset/interrupt 1: Mask LVD2 reset/interrupt	R/W
23:19	—	These bits are read as 0. The write value should be 0.	R/W
24	DBGSTOP_RPER	Mask bit for SRAM parity error reset/interrupt 0: Enable SRAM parity error reset/interrupt 1: Mask SRAM parity error reset/interrupt	R/W
25	DBGSTOP_RECCR	Mask bit for SRAM ECC error reset/interrupt 0: Enable SRAM ECC error reset/interrupt 1: Mask SRAM ECC error reset/interrupt	R/W
30:26	—	These bits are read as 0. The write value should be 0.	R/W
31	DBGSTOP_CPER	Mask bit for Cache SRAM parity error reset/interrupt 0: Enable Cache SRAM parity error reset/interrupt 1: Mask Cache SRAM parity error reset/interrupt	R/W

The Debug Stop Control Register (DBGSTOPCR) controls the functional stop in OCD mode. All bits in the register are regarded as 0 when the MCU is not in OCD mode.

2.7.4.3 DBGREG CoreSight component registers

The DBGREG module provides the CoreSight component registers defined in the Arm CoreSight architecture.

Table 2.13 shows the registers. See reference 4. in section 2.14. References for details of each register.

Table 2.13 DBGREG CoreSight component registers (1 of 2)

Name	Address	Access size	R/W	Initial value
PID4	0x4001_BFD0	32 bits	R	0x00000004
PID5	0x4001_BFD4	32 bits	R	0x00000000
PID6	0x4001_BFD8	32 bits	R	0x00000000
PID7	0x4001_BFDC	32 bits	R	0x00000000
PID0	0x4001_BFE0	32 bits	R	0x00000005
PID1	0x4001_BFE4	32 bits	R	0x00000030
PID2	0x4001_BFE8	32 bits	R	0x0000000A
PID3	0x4001_BFEC	32 bits	R	0x00000000
CID0	0x4001_BFF0	32 bits	R	0x0000000D
CID1	0x4001_BFF4	32 bits	R	0x000000F0

Table 2.13 DBGREG CoreSight component registers (2 of 2)

Name	Address	Access size	R/W	Initial value
CID2	0x4001_BFF8	32 bits	R	0x00000005
CID3	0x4001_BFFC	32 bits	R	0x000000B1

2.7.5 OCDREG Module

The OCDREG module are only accessible by the On-Chip Debug (OCD) emulator. OCDREG is implemented as a CoreSight-compliant component.

Table 2.14 lists the OCDREG registers.

Table 2.14 OCDREG registers

Name		DAP port	Address	Access size	R/W
MCU Status Register	MCUSTAT	Port 1	0x8000_0400	32 bits	R
MCU Control Register	MCUCTRL	Port 1	0x8000_0410	32 bits	R/W

Note: OCDREG is located in the dedicated OCD address space. This address map is independent from the system address map.

2.7.5.1 MCUSTAT : MCU Status Register

Base address: CPU_OCD = 0x8000_0000

Offset address: 0x400

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	SECDBG	DBGFUNCEN	BOOTMD	—	—	—	—	—	—	—	—	CPUSTOPCLK	CPUSLEEP	—
Value after reset:	0	0	1/0*1	1/0*1	1/0*1	0	0	1	0	0	0	0	0	1/0*1	1/0*1	0

Bit	Symbol	Function	R/W
0	—	These bits are read as 0.	R
1	CPUSLEEP	Sleep mode status This bit is unpredictable when the MCU is in Software Standby mode, Snooze mode, or Deep Software Standby mode. 0: CPU is not in Sleep mode 1: CPU in Sleep mode	R
2	CPUSTOPCLK	CPU clock status This bit is unpredictable when the MCU is in Deep Software Standby mode. 0: CPU clock is not stopped. 1: CPU clock is stopped.	R
7:3	—	These bits are read as 0.	R
8	—	These bits are read as 1.	R
10:9	—	These bits are read as 0.	R
11	BOOTMD	Boot mode status 0: Device is not in Boot mode 1: Device is in Boot mode	R
12	DBGFUNCEN	Debugger status 0: Debugger connection is not available 1: Debugger function is enabled	R

Bit	Symbol	Function	R/W
13	SECDBG	Secure Debug status 0: Secure Debug is not available 1: Secure Debug is available	R
31:14	—	These bits are read as 0.	R

Note 1. Depends on the MCU status.

2.7.5.2 MCUCTRL : MCU Control Register

Base address: CPU_OCD = 0x8000_0000

Offset address: 0x410

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

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

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0. The write value should be 0.	R/W
8	DBIRQ*2	Debug Interrupt Request Writing 1 to the bit wakes up the MCU from low power mode. The condition can be cleared by writing 0 to the DBIRQ bit. 0: Debug interrupt not requested 1: Debug interrupt requested	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W
16	CPUWAIT*2	CPU Wait Setting Write 1 to assert CPUWAIT, write 0 to deassert CPUWAIT*1. 0: Clear CPUWAIT to low 1: Set CPUWAIT to high	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. CPUWAIT is used to prevent the processor from executing code immediately after reset.
Note 2. Access (R/W) to bit is valid only when Debug Level is DBG1 or DBG2.

2.7.5.3 OCDREG CoreSight component registers

The OCDREG module provides the CoreSight component registers defined in the Arm CoreSight architecture.

Table 2.15 shows the registers. See reference 4. in section 2.14. References for details of each register.

Table 2.15 OCDREG CoreSight component registers (1 of 2)

Name	Address	Access size	R/W	Initial value
PID4	0x8000_0FD0	32 bits	R	0x00000004
PID5	0x8000_0FD4	32 bits	R	0x00000000
PID6	0x8000_0FD8	32 bits	R	0x00000000
PID7	0x8000_0FDC	32 bits	R	0x00000000
PID0	0x8000_0FE0	32 bits	R	0x00000004
PID1	0x8000_0FE4	32 bits	R	0x00000030
PID2	0x8000_0FE8	32 bits	R	0x0000000A
PID3	0x8000_0FEC	32 bits	R	0x00000000

Table 2.15 OCDREG CoreSight component registers (2 of 2)

Name	Address	Access size	R/W	Initial value
CID0	0x8000_0FF0	32 bits	R	0x0000000D
CID1	0x8000_0FF4	32 bits	R	0x000000F0
CID2	0x8000_0FF8	32 bits	R	0x00000005
CID3	0x8000_0FFC	32 bits	R	0x000000B1

2.7.6 CPUDSAR : CPU Debug Security Attribution Register

Base address: CPSCU = 0x4000_8000

Offset address: 0x1B0

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CPUDSA0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

Bit	Symbol	Function	R/W
0	CPUDSA0	CPU Debug Security Attribution 0 0: Secure 1: Non-secure	R/W
31:1	—	These bits are read as 1. The write value should be 1.	R/W

Note: Only secure access can write to this register. Both secure access and non-secure read access are allowed. Non-secure write access is denied, and no TrustZone access error is generated.

Note: This register is write-protected by PRCR register.

When Debug level of the MCU is DBG2, by guarding entire EPPB bus, the non-secure access from CPU to debug related components is completely controlled by the current value of the CPUDSA0 bit. Since this bit is modifiable only when CPU is in secure state, user must be aware of the CPUDSAR register before using CoreSight debug components.

CPUDSA0 bit (CPU Debug Security Attribution 0)

Security attributes of register for accessing the debug component of the CPU.

0: Debug component can only be accessed with secure access.

1: There is no restriction on accessing the debug component.

2.7.7 Processing on Error response generated by CPU access

In addition to the specific-error detection specification of the Arm Cortex-M33 processor, this MCU also provides additional error information which is described in section 13, Buses.

This section describes how to handle the additional error information with no conflict to that of the Arm Cortex-M33 processor.

Table 2.16 shows error detection modules, which are also described in section 13, Buses. These error detection modules not only provide error information on the bus module, but also notify the processor to trigger the exception handler.

Table 2.16 Error detection modules (1 of 2)

	NMI/RESET request	Interrupt	Bus error status register	Error address register Error RW register
Slave TZF	NMISR.TZFST	Bus Fault*1 (Hard Fault)	BUS.BUSnERRSTAT.STERRSTAT	BUS.BTZFnERRADD BUS.BTZFnERRRW

Table 2.16 Error detection modules (2 of 2)

	NMI/RESET request	Interrupt	Bus error status register	Error address register Error RW register
Slave bus error	—	Bus Fault*1 (Hard Fault)	BUS.BUSnERRSTAT.SLERRSTAT	BUS.BUSnERRADD BUS.BUSnERRRW
Illegal address access error	—	Bus Fault*1 (Hard Fault)	BUS.BUSnERRSTAT.ILERRSTAT	BUS.BUSnERRADD BUS.BUSnERRRW

Note 1. A Bus Fault can be treated as HardFault. For details, see *ARM® Cortex®-M33 Device Generic User Guide* in the section 2.14. References.

To prevent unexpected operation, when handling the exception, additional operation should be added into exception routing.

BusFault when occurred by error detected as shown in Table 2.16:

- See section 13, Buses for the error information in the corresponding register
- Clear the data in cache for the error address
- Clear the Error Status register in the bus module
- Service exception handling with Arm-guided operation

For a Bus Fault that is not detected in the Renesas-specific error detection module (occurred inside the Arm Cortex-M33 core), see the *ARM® Cortex®-M33 Device Generic User Guide* to handle this case.

In the system bus specification, there is a specific case for Slave TrustZone Filter, that is, if an error is selected to generate an NMI, then before the processor handles the Bus Fault exception, NMI with higher priority takes the exception first. Therefore, use the BusFault handler and not NMI handler to handle this error. In other words, the NMI status should be cleared but the error status bit should not be cleared to ensure that BusFault captures all the error information.

Figure 2.4 and Figure 2.5 show the recommended flows for NMI handler and BusFault handler for the errors described in Table 2.16.

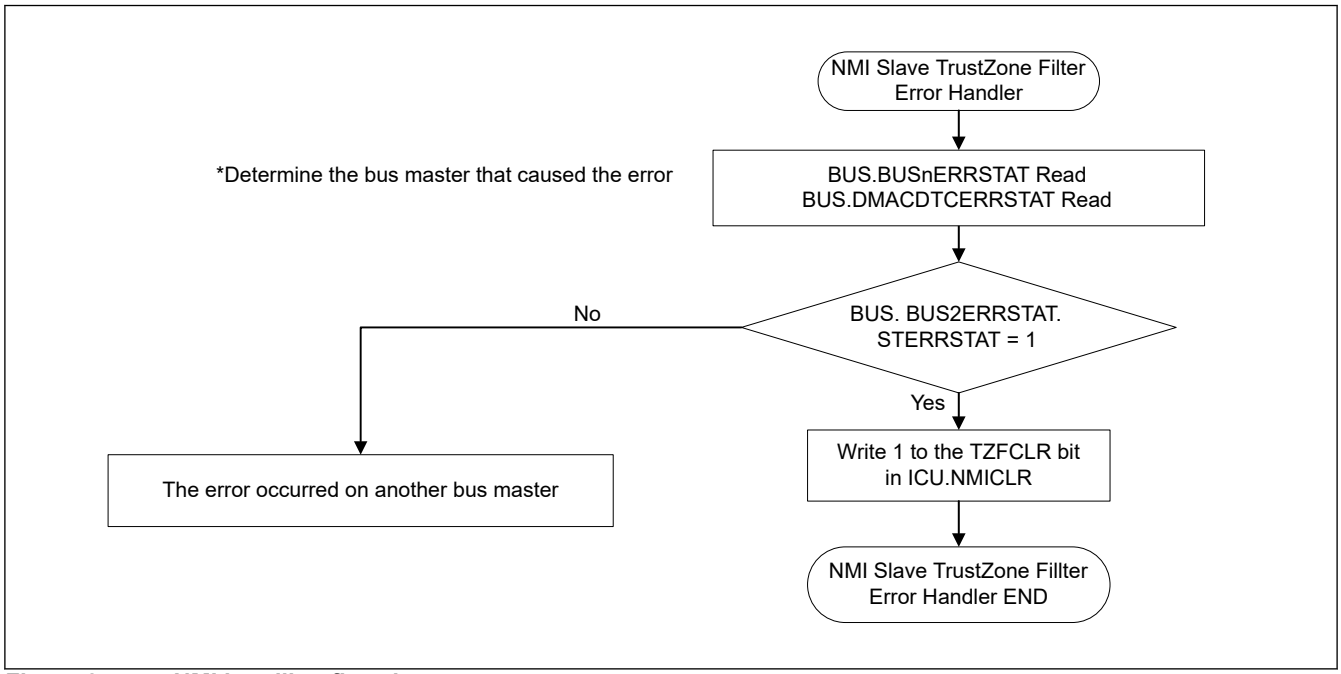


Figure 2.4 NMI handling flowchart

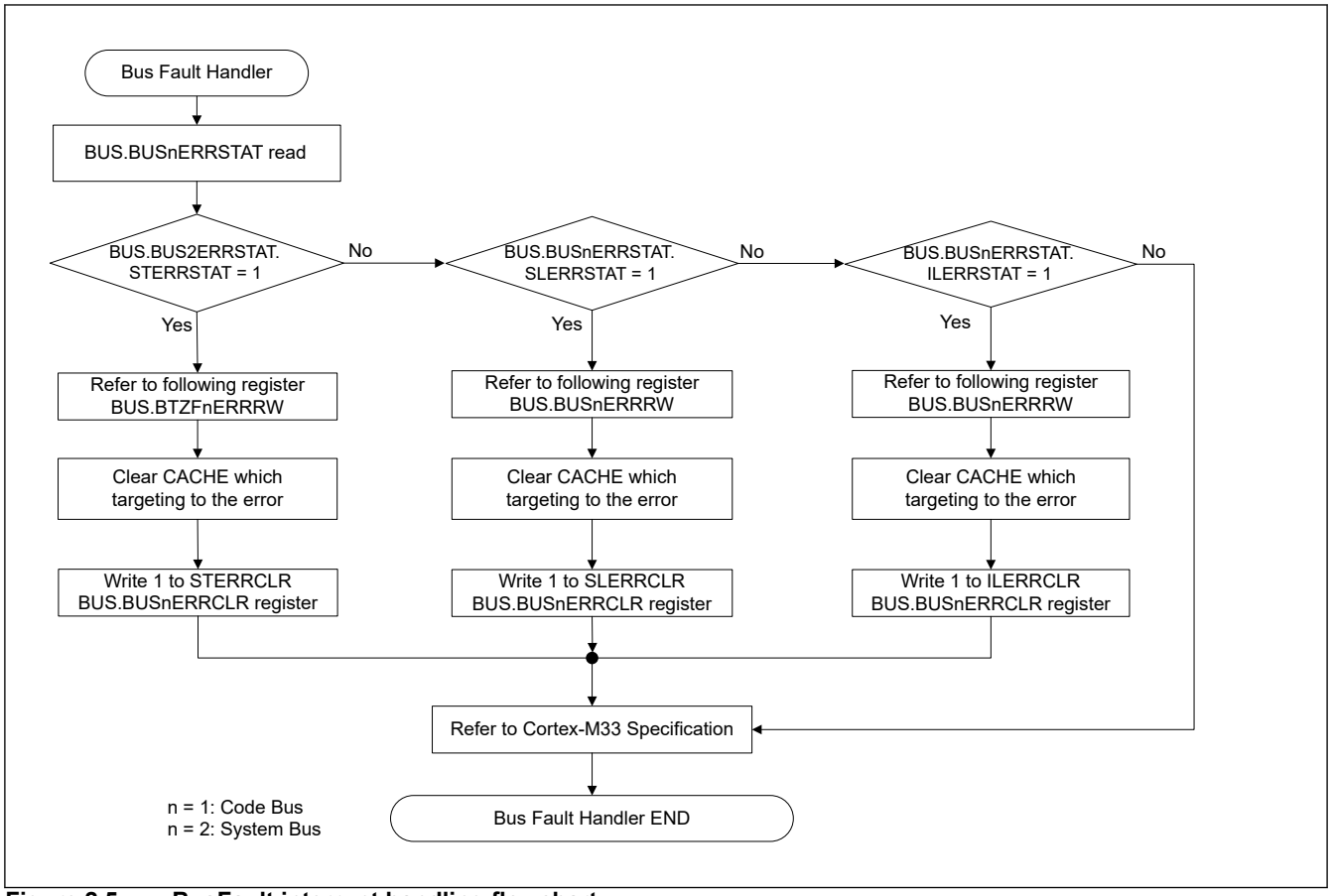


Figure 2.5 BusFault interrupt handling flowchart

2.8 CoreSight Cross Trigger Interface (CTI)

As shown in Figure 2.6, the input and output of a Cross Trigger Interface (CTI) interact with each other through four CTM channels. Input of a CTI can be used to trigger the output of another CTI using the four CTM channels.

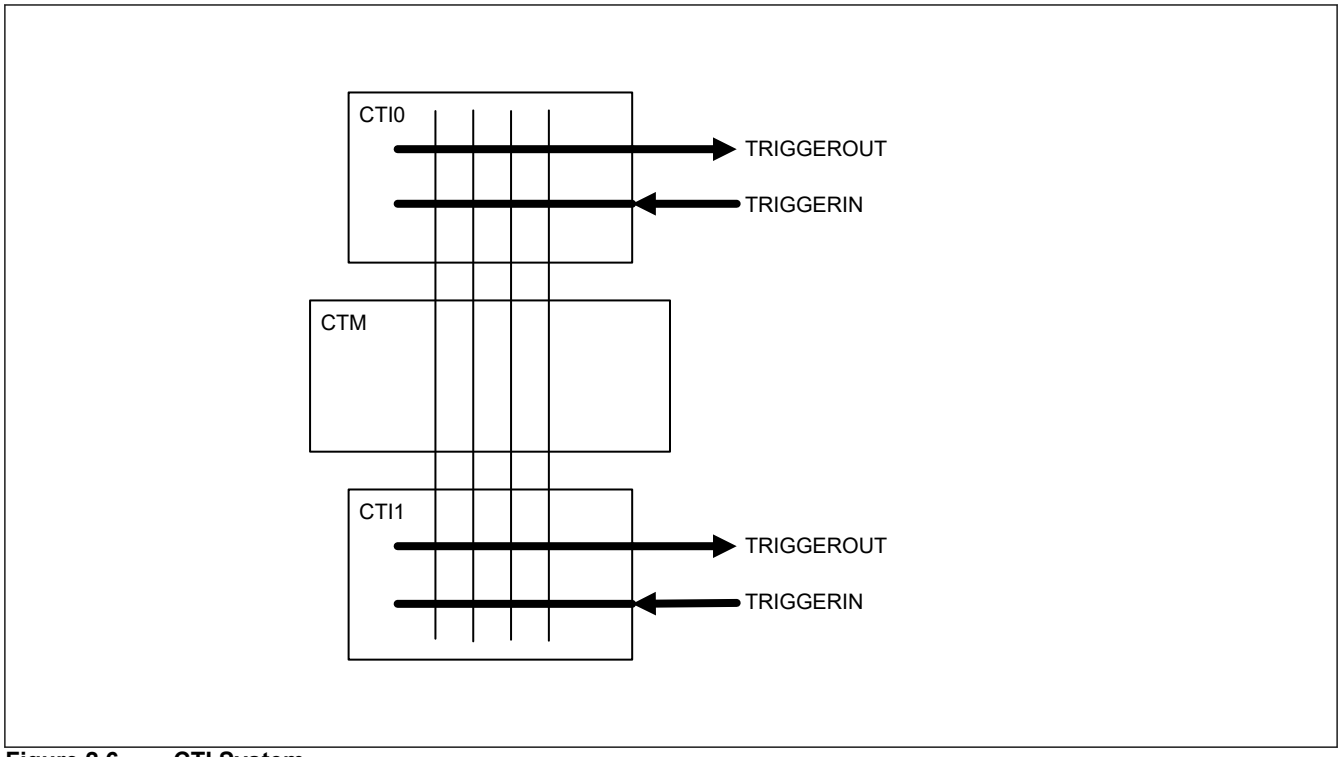


Figure 2.6 CTI System

Debug Interrupt Request (DBGIRQ) is controlled by MCUCTRL register in OCDREG module.

Table 2.17 CTI Trigger signals

Number of CTI channel	CTITRIGIN		CTITRIGOUT	
CTI0 (Debug common)	0	ACQCOMP	0	—
	1	FULL	1	—
	2	DBIRQ	2	ETB FLUSHIN
	3	—	3	ETB TRIGIN
	4	—	4	—
	5	—	5	—
	6	—	6	—
	7	—	7	—
CTI1 (CPU)	0	Processor Halted	0	Processor debug request
	1	DWT Comparator Output 0	1	Processor Restart
	2	DWT Comparator Output 1	2	CTIIRQ[0] (Connected to IRQ96)
	3	DWT Comparator Output 2	3	CTIIRQ[1] (Connected to IRQ97)
	4	ETM Event Output 0	4	ETM Event Input 0
	5	ETM Event Output 1	5	ETM Event Input 1
	6	—	6	ETM Event Input 2
	7	—	7	ETM Event Input 3

2.9 CoreSight ATB Funnel

There is one CoreSight ATB funnel in the MCU. The funnel has two ATB slaves and one ATB master, and it selects the debug trace source from ETM and ITM to ETB. Figure 2.7 shows the CoreSight ATB connection in the MCU.

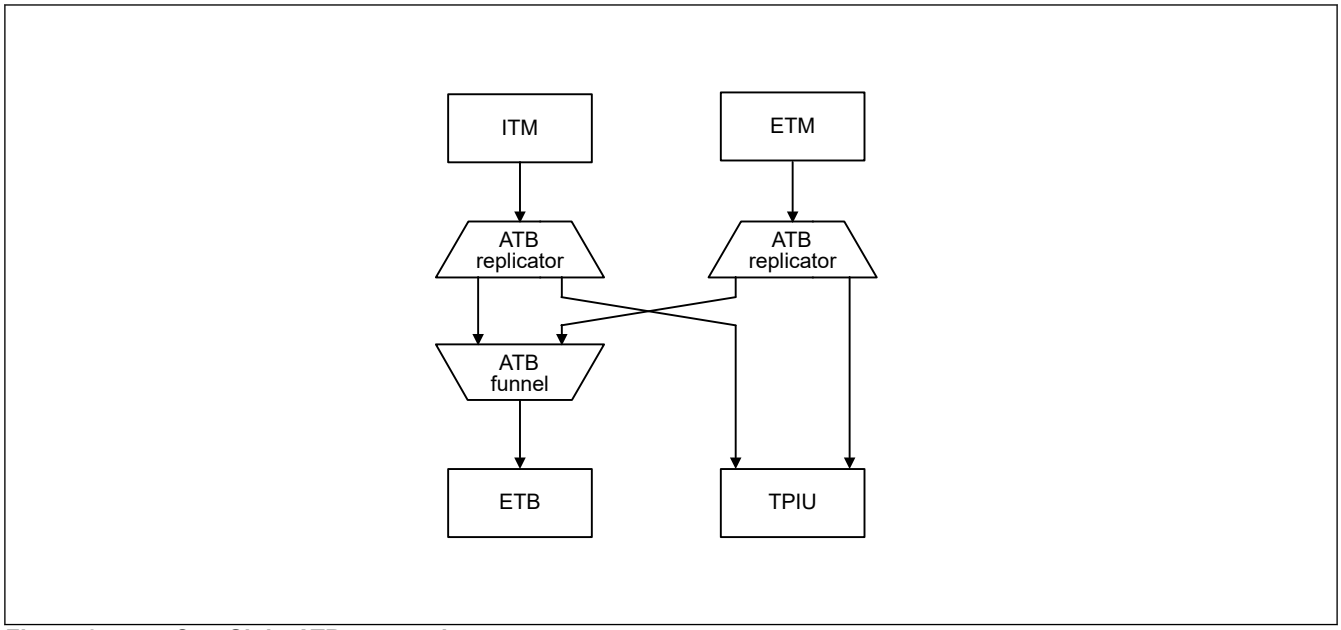


Figure 2.7 CoreSight ATB connection

Table 2.18 shows the ATB slave connection for the funnel.

Table 2.18 ATB slave connection

ATB slave number	Connected trace source
#0	ITM
#1	ETM

See reference 4. in section 2.14. References for details of the ATB and funnel.

2.10 Break Point Unit

The MCU has Break Point Unit. See BreakPoint unit chapter of reference 1. in section 2.14. References for details about register description of this module.

2.11 CoreSight Time Stamp Generator

A CoreSight Time Stamp Generator provides a CPU clock-based timestamp to ITM and ETM.The timestamp is generated by a 64-bit counter. See reference 4. in section 2.14. References for details.

2.12 SysTick Timer

The MCU has SysTick timer that provides two 24-bit down counters, non-secure and secure counters. The timer can select SysTick timer clock (SYSTICCLK) or System clock (ICLK).

See section 8, Clock Generation Circuit and reference 1. in section 2.14. References for details.

Note: SysTick timer counter operation is enabled by signal synchronized with CPU clock. Therefore, the counter might not operate correctly if the CPU clock is slower than the SysTick timer clock. In other words, clock setting must satisfy the following: CPU clock ≥ SysTick timer clock (LOCO: 32.768 kHz).

2.13 OCD Emulator Connection

In this product, the MCU confirms the access permission for Non-secure debug and Non-secure chip resources by checking Debug level is DBG1 or higher. For full access permission for debug and chip resources, Secure debug level DBG2 is required.

Figure 2.8 shows a block diagram of SWD authentication mechanism.

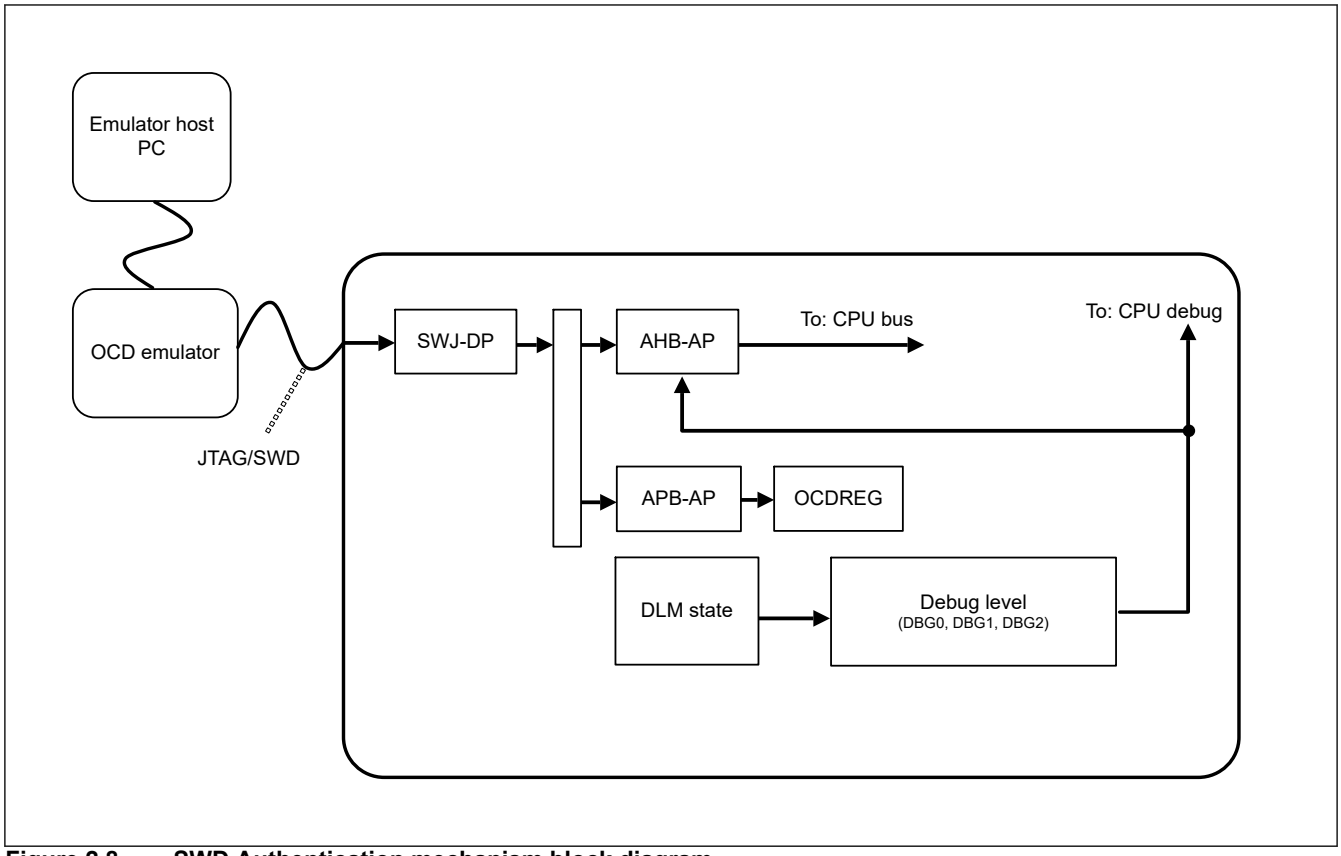


Figure 2.8 SWD Authentication mechanism block diagram

Three levels of debug capability are available, DBG0, DBG1, and DBG2, which correspond to the Device Level Management (DLM) states. When debug level is DBG0, access to debug components and system bus from OCD emulator is not permitted. When debug level is DBG1 or DBG2, the corresponding non-secure or secure debug components and system bus can be accessed from the OCD emulator. See Table 2.5 for more information about debug levels.

2.13.1 DBGEN

After the OCD emulator gets access permission, the OCD emulator must set the DBGEN bit in the System Control OCD Control Register (SYOCDCR). In addition, the OCD emulator must clear the DBGEN bit before disconnecting it. See section 10, Low Power Modes for details.

2.13.2 Restrictions on Connecting an OCD emulator

This section describes the restrictions on emulator access.

2.13.2.1 Starting connection while in low power mode

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

2.13.2.2 Changing low power mode while in OCD mode

When the MCU is in OCD mode, the low power mode can be changed. However, system bus access from AHB-AP is prohibited in Software Standby, Snooze or Deep Software Standby mode. Only SWJ-DP, APB-AP, and OCDREG can be accessed from the OCD emulator in these modes. Table 2.19 shows the restrictions.

Table 2.19 Restrictions by mode (1 of 2)

Active mode	Start OCD emulator connection	Change low power mode	Access AHB-AP and system bus	Access APB-AP and OCDREG
Normal	Yes	Yes	Yes	Yes

Table 2.19 Restrictions by mode (2 of 2)

Active mode	Start OCD emulator connection	Change low power mode	Access AHB-AP and system bus	Access APB-AP and OCDREG
Sleep	Yes	Yes	Yes	Yes
Software Standby	No	Yes	No	Yes
Snooze	No	Yes	No	Yes
Deep Software Standby	No	Yes	No	Yes

If system bus access is required in Software Standby, Snooze, or Deep Software Standby mode, set the MCUCTRL.DBIRQ bit in OCDREG to wake up the MCU from the low power modes. Simultaneously, by asserting the MCUCTRL.DBIRQ bit in OCDREG, the OCD emulator can wake up the MCU without starting CPU execution by using a CPU break.

2.13.2.3 Connecting sequence and JTAG/SWD authentication

1. Connect the OCD debugger to the MCU through the JTAG or SWD interface.
2. Set up SWJ-DP to access DAP bus.
   In the setup, the OCD emulator must assert CDBGPWRUPREQ in the SWJDP.
   Control Status Register, and then wait until CDBGPWRUPACK in the same register is asserted.
3. Set the APB-AP to access OCDREG. This APB-AP is connected to DAP bus port 1.
4. Set MCUCTRL.CPUWAIT = 1.
5. Confirm the debug capability of device by reading MCUSTAT:
   - If Debug function is prohibited, this device is not able to debug.
   - If Debug function is enabled and secure debug is not available, only non-secure debug is available.
   - If Debug function is enabled and secure debug is available, full debug functions are available.

   If Debug function is available, set debug-related register then clear MCUCTRL.CPUWAIT = 0.
6. Set up the AHB-AP to access the system address space. The AHB-AP is connected to DAP bus port 0.
7. Set SYOCDCR.DBGEN to 1.
8. Start accessing the CPU debug resources using the AHB-AP.

Note: Debug level is determined by the current DLM state of product.

2.14 References

1. *ARM®v8-M Architecture Reference Manual* (ARM DDI 0553B.a)
2. *ARM® Cortex®-M33 Processor Technical Reference Manual* (ARM 100230)
3. *ARM® Cortex®-M33 Device Generic User Guide* (ARM 100235)
4. *ARM® CoreSight™ SoC-400 Technical Reference Manual* (ARM DDI 0480G)
5. *ARM® CoreSight™ Architecture Specification* (ARM IHI 0029E)

3. Operating Modes

3.1 Overview

Table 3.1 shows the selection of operating modes by the mode-setting pin. For details, see section 3.2. Details of Operating Modes. Operation starts with the on-chip flash memory enabled, regardless of the mode in which operation started.

Table 3.1 Selection of operating modes by the mode-setting pin

Mode-setting pin (MD)	Operating mode	On-chip Flash
1	Single-chip mode	Enable
0	SCI boot mode	Enable

3.2 Details of Operating Modes

3.2.1 Single-Chip Mode

In single-chip mode, all I/O pins are available for use as input or output port, inputs or outputs for peripheral functions, or as interrupt inputs.

When a reset is released while the MD pin is high, the MCU starts in single-chip mode and the on-chip flash is enabled.

3.2.2 SCI Boot Mode

In this mode, the on-chip flash memory programming routine (SCI boot program), stored in the boot area within the MCU, is used. The on-chip flash, including code flash memory and data flash memory, can be modified from outside the MCU by using a universal asynchronous receiver/transmitter (UART) SCI. For details, see section 43, Flash Memory. The MCU starts in SCI boot mode if the MD pin is held low on release from the reset state.

3.3 Operating Modes Transitions

3.3.1 Operating Mode Transitions as Determined by the Mode-Setting Pin

Figure 3.1 shows operating mode transitions determined by the MD pin settings.

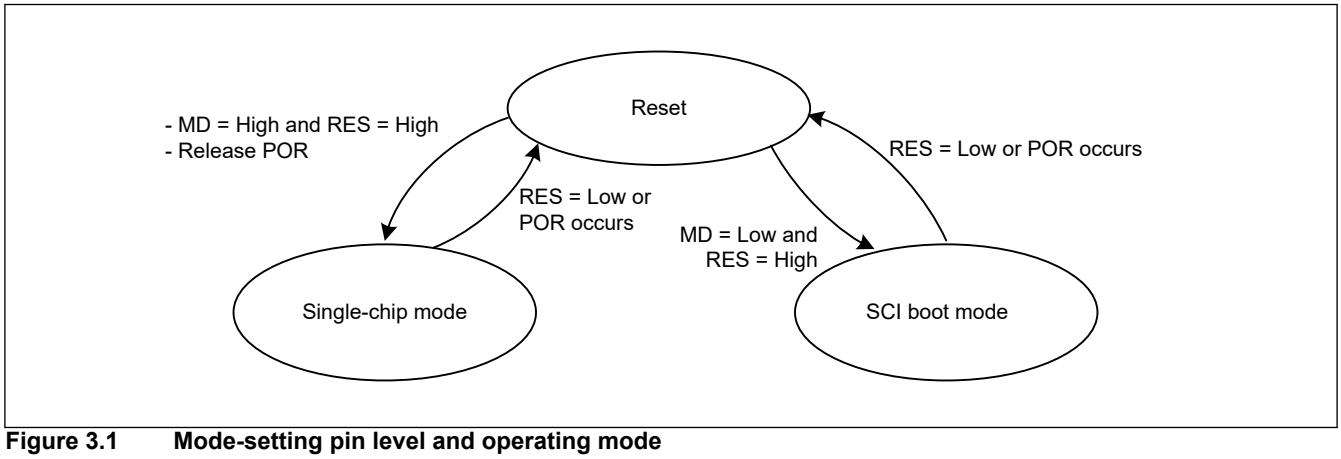
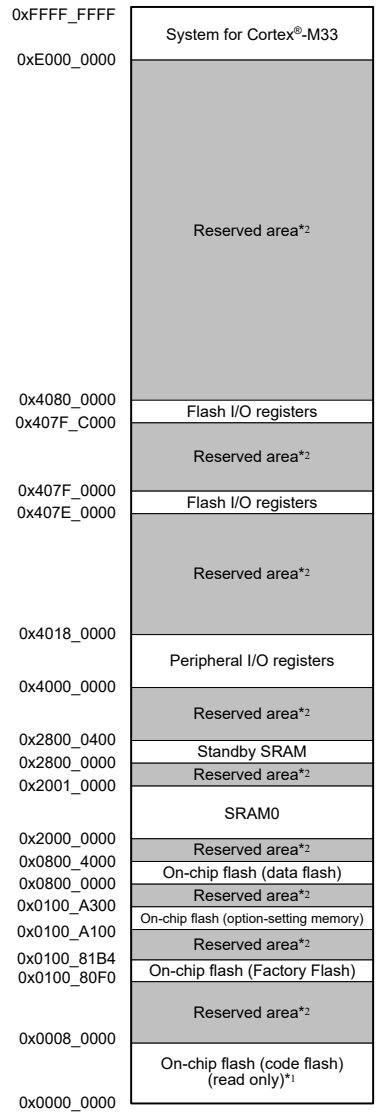


Figure 3.1 Mode-setting pin level and operating mode

4. Address Space

4.1 Address Space

The MCU supports a 4-GB linear address space ranging from 0x0000_0000 to 0xFFFF_FFFF that can contain both program and data. Figure 4.1 shows the memory map.



Note 1. See Table 4.1. The capacity of the flash differs depending on the product.
Note 2. Do not access reserved areas.

Figure 4.1 Memory map

Table 4.1 Capacity of the code flash memory, data flash memory, and SRAM0

Code flash memory		Data flash memory		SRAM0	
Capacity	**Address**	**Capacity**	**Address**	**Capacity**	**Address**
512 KB	0x0000_0000 - 0x0007_FFFF	16 KB	0x0800_0000 - 0x0800_3FFF	64 KB	0x2000_0000 - 0x2000_FFFF
256 KB	0x0000_0000 - 0x0003_FFFF				

5. Resets

5.1 Overview

The MCU provides 14 resets.

Table 5.1 lists the reset names and sources.

Table 5.1 Reset names and sources

Reset name	Source
RES pin reset	Voltage input to the RES pin is driven low
Power-on reset	VCC rise (voltage detection V_{POR})*1
Independent watchdog timer reset	IWDT underflow or refresh error
Watchdog timer reset	WDT underflow or refresh error
Voltage monitor 0 reset	VCC fall (voltage detection V_{det0})*1
Voltage monitor 1 reset	VCC fall (voltage detection V_{det1})*1
Voltage monitor 2 reset	VCC fall (voltage detection V_{det2})*1
SRAM parity error reset	SRAM parity error detection
SRAM ECC error reset	SRAM ECC error detection
Bus master MPU error reset	Bus master MPU error detection
TrustZone error reset	TrustZone error detection
Cache Parity error reset	Cache Parity error detection
Deep software standby reset	Deep software standby mode is canceled by an interrupt
Software reset	Register setting (use the software reset bit AIRCR.SYSRESETREQ)

Note 1. For details on the voltages to be monitored (V_{POR}, V_{det0}, V_{det1}, and V_{det2}), see section 7, Low Voltage Detection (LVD) and section 46, Electrical Characteristics.

The internal state and pins are initialized by a reset. Table 5.2 and Table 5.3 list the targets initialized by resets.

Table 5.2 Reset detect flags initialized by each reset source (1 of 4)

Flag to be initialized	Reset source							
	RES pin reset	Power-on reset	Voltage monitor 0 reset	Independent watchdog timer reset	Watchdog timer reset	Voltage monitor 1 reset	Voltage monitor 2 reset	Software reset
Power-On Reset Detect Flag (RSTSR0.PORF)	✓	—	—	—	—	—	—	—
Voltage Monitor 0 Reset Detect Flag (RSTSR0.LVD0RF)	✓	✓	—	—	—	—	—	—
Independent Watchdog Timer Reset Detect Flag (RSTSR1.IWDTRF)	✓	✓	✓	—	—	—	—	—
Watchdog Timer Reset Detect Flag (RSTSR1.WDTRF)	✓	✓	✓	—	—	—	—	—
Voltage Monitor 1 Reset Detect Flag (RSTSR0.LVD1RF)	✓	✓	✓	—	—	—	—	—
Voltage Monitor 2 Reset Detect Flag (RSTSR0.LVD2RF)	✓	✓	✓	—	—	—	—	—

Table 5.2 Reset detect flags initialized by each reset source (2 of 4)

Flag to be initialized	Reset source							
	RES pin reset	Power-on reset	Voltage monitor 0 reset	Independent watchdog timer reset	Watchdog timer reset	Voltage monitor 1 reset	Voltage monitor 2 reset	Software reset
Software Reset Detect Flag (RSTSR1.SWRF)	✓	✓	✓	—	—	—	—	—
SRAM Parity Error Reset Detect Flag (RSTSR1.RPERF)	✓	✓	✓	—	—	—	—	—
SRAM ECC Error Reset Detect Flag (RSTSR1.REERF)	✓	✓	✓	—	—	—	—	—
Bus Master MPU Error Reset Detect Flag (RSTSR1.BUSMRF)	✓	✓	✓	—	—	—	—	—
TrustZone Error Reset Detect Flag (RSTSR1.TZERF)	✓	✓	✓	—	—	—	—	—
Cache Parity Error Reset Detect Flag (RSTSR1.CPERF)	✓	✓	✓	—	—	—	—	—
Deep Software Standby Reset Detect Flag (RSTSR0.DPSRSTF)	✓	✓	✓	—	—	—	—	—
Cold Start/Warm Start Determination Flag (RSTSR2.CWSF)	—	✓	—	—	—	—	—	—

Table 5.2 Reset detect flags initialized by each reset source (3 of 4)

Flag to be initialized	Reset source						
	SRAM parity error reset	SRAM ECC error reset	Bus master MPU error reset	TrustZone error reset	Cache Parity error reset	Deep Software Standby reset	
						DEEPCUT[0] = 0	DEEPCUT[0] = 1
Power-On Reset Detect Flag (RSTSR0.PORF)	—	—	—	—	—	—	—
Voltage Monitor 0 Reset Detect Flag (RSTSR0.LVD0RF)	—	—	—	—	—	—	—
Independent Watchdog Timer Reset Detect Flag (RSTSR1.IWDTRF)	—	—	—	—	—	✓	✓
Watchdog Timer Reset Detect Flag (RSTSR1.WDTRF)	—	—	—	—	—	✓	✓
Voltage Monitor 1 Reset Detect Flag (RSTSR0.LVD1RF)	—	—	—	—	—	—	—
Voltage Monitor 2 Reset Detect Flag (RSTSR0.LVD2RF)	—	—	—	—	—	—	—
Software Reset Detect Flag (RSTSR1.SWRF)	—	—	—	—	—	✓	✓
SRAM Parity Error Reset Detect Flag (RSTSR1.RPERF)	—	—	—	—	—	✓	✓
SRAM ECC Error Reset Detect Flag (RSTSR1.REERF)	—	—	—	—	—	✓	✓
Bus Master MPU Error Reset Detect Flag (RSTSR1.BUSMRF)	—	—	—	—	—	✓	✓

Table 5.2 Reset detect flags initialized by each reset source (4 of 4)

Flag to be initialized	Reset source						
	SRAM parity error reset	SRAM ECC error reset	Bus master MPU error reset	TrustZone error reset	Cache Parity error reset	Deep Software Standby reset	
						DEEPCUT[0] = 0	DEEPCUT[0] = 1
TrustZone Error Reset Detect Flag (RSTSR1.TZERF)	—	—	—	—	—	✓	✓
Cache Parity Error Reset Detect Flag (RSTSR1.CPERF)	—	—	—	—	—	✓	✓
Deep Software Standby Reset Detect Flag (RSTSR0.DPSRSTF)	—	—	—	—	—	—	—
Cold Start/Warm Start Determination Flag (RSTSR2.CWSF)	—	—	—	—	—	—	—

Note: ✓ : Initialized to 0
— : Not initialized

Table 5.3 Module-related registers initialized by each reset source (1 of 4)

Registers to be initialized		Reset source							
		RES pin reset	Power-on reset	Voltage monitor 0 reset	Independent watchdog timer reset	Watchdog timer reset	Voltage monitor 1 reset	Voltage monitor 2 reset	Software reset
Independent watchdog timer registers	IWDTRR, IWDTSR	✓	✓	✓	✓	✓	✓	✓	✓
Watchdog timer registers	WDTRR, WDTCR, WDTSR, WDTRCR, WDTCSTPR	✓	✓	✓	✓	✓	✓	✓	✓
Voltage monitor function 1 registers	LVD1CR0,LVD1CMPCR	✓	✓	✓	✓	✓	—	—	—
	LVD1CR1/LVD1SR	✓	✓	✓	✓	✓	—	—	—
Voltage monitor function 2 registers	LVD2CR0, LVD2CMPCR	✓	✓	✓	✓	✓	—	—	—
	LVD2CR1/LVD2SR	✓	✓	✓	✓	✓	—	—	—
LOCO registers	LOCOCR	✓	✓	✓	✓	✓	✓	✓	✓
	LOCOUTCR	—	✓	✓	—	—	✓	✓	—
MOSC register	MOMCR	✓	✓	✓	✓	✓	✓	✓	✓
Bus, MPU and TrustZone error registers*2	BUS_ERROR_ADDRESS Register BUS_ERROR_STATUS Register	✓	✓	✓	✓	✓	✓	✓	✓
Pin states		✓	✓	✓	✓	✓	✓	✓	✓
Low-power function registers	DPSBYCR, DPSIER0 to DPSIER2, DPSIFR0 to DPSIFR2, DPSIEGR0 to DPSIEGR2	✓	✓	✓	✓	✓	✓	✓	✓
	SYOCDCR	—	✓	—	—	—	—	—	—

Table 5.3 Module-related registers initialized by each reset source (2 of 4)

Registers to be initialized		Reset source							
		RES pin reset	Power-on reset	Voltage monitor 0 reset	Independent watchdog timer reset	Watchdog timer reset	Voltage monitor 1 reset	Voltage monitor 2 reset	Software reset
Security Attribute Registers	CPUDSAR, RSTSAR, LVDSAR, CGFSAR, LPMSAR, DPFSAR, ICUSARx, BUSSARA, BUSSARB, CSAR, MMPUSARA, MMPUSARB, DMACSAR, DTCSAR, ELCSARA, ELCSARB, PmSAR, SRAMSAR, STBRAMSAR, FSAR, PSARB, PSARC, PSARD, PSARE, MSSAR, TZFSAR	✓*3	✓	✓*3	✓*3	✓*3	✓*3	✓*3	✓*3
Registers other than those shown, CPU, and internal state		✓	✓	✓	✓	✓	✓	✓	✓

Table 5.3 Module-related registers initialized by each reset source (3 of 4)

Registers to be initialized		Reset source						
		SRAM parity error reset	SRAM ECC error reset	Bus master MPU error reset	TrustZone error reset	Cache Parity error reset	Deep Software Standby reset	
							DEEPCUT[0] = 0	DEEPCUT[0] = 1
Independent watchdog timer registers	IWDTRR, IWDTSR	✓	✓	✓	✓	✓	✓	✓
Watchdog timer registers	WDTRR, WDTCR, WDTSR, WDTRCR, WDTCSTPR	✓	✓	✓	✓	✓	✓	✓
Voltage monitor function 1 registers	LVD1CR0,LVD1CMPCR	—	—	—	—	—	—	—
	LVD1CR1 / LVD1SR	—	—	—	—	—	✓	✓
Voltage monitor function 2 registers	LVD2CR0, LVD2CMPCR	—	—	—	—	—	—	—
	LVD2CR1/LVD2SR	—	—	—	—	—	✓	✓
LOCO registers	LOCOCR	✓	✓	✓	✓	✓	✓	✓
	LOCOUTCR	—	—	—	—	—	—	✓
MOSC register	MOMCR	✓	✓	✓	✓	✓	—	—
Bus, MPU and TrustZone error registers*2	BUS_ERROR_ADDRESS Register BUS_ERROR_STATUS Register	✓	✓	—	—	—	✓	✓
Pin states		✓	✓	✓	✓	✓	*1	*1
Low-power function registers	DPSBYCR, DPSIER0 to DPSIER2, DPSIFR0 to DPSIFR2, DPSIEGR0 to DPSIEGR2	✓	✓	✓	✓	✓	—	—
	SYOCDCR	—	—	—	—	—	—	—

Table 5.3 Module-related registers initialized by each reset source (4 of 4)

Registers to be initialized		Reset source						
		SRAM parity error reset	SRAM ECC error reset	Bus master MPU error reset	TrustZone error reset	Cache Parity error reset	Deep Software Standby reset	
							DEEPCUT[0] = 0	DEEPCUT[0] = 1
Security Attribute Registers	CPUDSAR, RSTSAR, LVDSAR, CGFSAR, LPMSAR, DPFSAR, ICUSARx, BUSSARA, BUSSARB, CSAR, MMPUSARA, MMPUSARB, DMACSAR, DTCSAR, ELCSARA, ELCSARB, PmSAR, SRAMSAR, STBRAMSAR, FSAR, PSARB, PSARC, PSARD, PSARE, MSSAR, TZFSAR	✓*3	✓*3	✓*3	✓*3	✓*3	✓*4	✓*4
Registers other than those shown, CPU, and internal state		✓	✓	✓	✓	✓	✓	✓

Note: ✓ : Initialized
— : Not initialized
Note 1. Depends on the setting of DPSBYCR.IOKEEP.
Note 2. Some control bits are not initialized by all types of resets. For details on the target bits, see section 13, Buses
Note 3. Reset does not occur while the debugger is connected (DBGSTR.CDBGPWRUPREQ = 1) even if On-chip debugger is disabled (SYOCDCR.DBGEN = 0).
Note 4. Reset does not occur while On-chip debugger is enabled (SYOCDCR.DBGEN = 1).

Table 5.4 shows the states of LOCO when a reset occurs.

Table 5.4 States of LOCO when a reset occurs

		Reset source	
		POR, LVD0, LVD1, LVD2, Deep Software Standby (DEEPCUT[0] = 1)	Other
LOCO	Enable or disable	Initialized to enable	
	Oscillation accuracy*1	Initialized to accuracy before trimming by power-on (accuracy: ± 10%)	Continue with the accuracy that was trimmed by LOCOUTCR

Note 1. The LOCO User Trimming Control Register (LOCOUTCR) is reset by POR, LVD0, LVD1, LVD2, and Deep Software Standby (DEEPCUT[0] = 1) resets, returning the LOCO to the default oscillation accuracy. To restore the pre-reset LOCO oscillation accuracy, reload the required trimming value into LOCOUTCR after any of these resets.

When a reset is released, reset exception handling starts.

Table 5.5 lists the pin related to the reset function.

Table 5.5 Pin related to reset

Pin name	I/O	Function
RES	Input	Reset pin

5.2 Register Descriptions

5.2.1 RSTSAR : Reset Security Attribution Register

Base address: SYSC = 0x4001_E000

Offset address: 0x3C4

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	NONSEC2	NONSEC1	NONSEC0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	NONSEC0	Non Secure Attribute bit 0 Target register: Reset Status Register 0 0: Secure 1: Non Secure	R/W
1	NONSEC1	Non Secure Attribute bit 1 Target register: Reset Status Register 1 0: Secure 1: Non Secure	R/W
2	NONSEC2	Non Secure Attribute bit 2 Target register: Reset Status Register 2 0: Secure 1: Non Secure	R/W
31:3	—	These bits are read as 1. The write value should be 1.	R/W

Note: Only secure access can write to this register. Both secure access and Non-secure read access are allowed but Non-secure write access is not allowed and TrustZone access error is not generated.

Note: This register is write-protected by PRCR register.

NONSEC0 bit (Non Secure Attribute bit 0)

This bit controls the security attribute of RSTSR0.

NONSEC1 bit (Non Secure Attribute bit 1)

This bit controls the security attribute of RSTSR1.

NONSEC2 bit (Non Secure Attribute bit 2)

This bit controls the security attribute of RSTSR2.

5.2.2 RSTSR0 : Reset Status Register 0

Base address: SYSC = 0x4001_E000

Offset address: 0x410

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DPSRSTF	—	—	—	LVD2RF	LVD1RF	LVD0RF	PORF
Value after reset:	x*1	0	0	0	x*1	x*1	x*1	x*1

Bit	Symbol	Function	R/W
0	PORF	Power-On Reset Detect Flag 0: Power-on reset not detected 1: Power-on reset detected	R/W*2
1	LVD0RF	Voltage Monitor 0 Reset Detect Flag 0: Voltage monitor 0 reset not detected 1: Voltage monitor 0 reset detected	R/W*2
2	LVD1RF	Voltage Monitor 1 Reset Detect Flag 0: Voltage monitor 1 reset not detected 1: Voltage monitor 1 reset detected	R/W*2
3	LVD2RF	Voltage Monitor 2 Reset Detect Flag 0: Voltage monitor 2 reset not detected 1: Voltage monitor 2 reset detected	R/W*2
6:4	—	These bits are read as 0. The write value should be 0.	R/W
7	DPSRSTF	Deep Software Standby Reset Detect Flag 0: Deep software standby mode cancellation not requested by an interrupt. 1: Deep software standby mode cancellation requested by an interrupt.	R/W*2

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note 1. The value after reset depends on the reset source.
Note 2. The register is cleared when a reset source listed in Table 5.2 occurs or when 0 is written to clear a flag. Bits other than the flag that is cleared should be set to 1.

PORF flag (Power-On Reset Detect Flag)

The PORF flag indicates that a power-on reset occurred.

[Setting condition]

- When a power-on reset occurs.

[Clearing conditions]

- When a reset listed in Table 5.2 occurs
- When PORF is read as 1 and then 0 is written to PORF

LVD0RF flag (Voltage Monitor 0 Reset Detect Flag)

The LVD0RF flag indicates that the VCC voltage fell below V_{det0}.

[Setting condition]

- When a voltage monitor 0 reset occurs.

[Clearing conditions]

- When a reset listed in Table 5.2 occurs
- When LVD0RF is read as 1 and then 0 is written to LVD0RF.

LVD1RF flag (Voltage Monitor 1 Reset Detect Flag)

The LVD1RF flag indicates that the VCC voltage fell below V_{det1}.

[Setting condition]

- When a voltage monitor 1 reset occurs.

[Clearing conditions]

- When a reset listed in Table 5.2 occurs
- When LVD1RF is read as 1 and then 0 is written to LVD1RF

LVD2RF flag (Voltage Monitor 2 Reset Detect Flag)

The LVD2RF flag indicates that the VCC voltage fell below V_{det2}.

[Setting condition]

- When a voltage monitor 2 reset occurs.

[Clearing conditions]

- When a reset listed in Table 5.2 occurs
- When LVD2RF is read as 1 and then 0 is written to LVD2RF

DPSRSTF flag (Deep Software Standby Reset Detect Flag)

The DPSRSTF flag indicates that Deep Software Standby mode has been canceled by an external or internal interrupt and that an internal reset (Deep Software Standby reset) occurred when the exception from Deep Software Standby Mode occur.

[Setting condition]

- When Deep Software Standby mode is cancelled by an external or an internal interrupt. For details, see section 10, Low Power Modes.

[Clearing conditions]

- When a reset listed in Table 5.2 occurs.
- When DPSRSTF is read as 1 and then 0 is written to DPSRSTF

5.2.3 RSTSR1 : Reset Status Register 1

Base address: SYSC = 0x4001_E000

Offset address: 0x0C0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CPERF	—	TZERF	—	BUSMRF	—	REERF	RPERF	—	—	—	—	—	SWRF	WDTRF	IWDTRF
Value after reset:	x*1	0	x*1	0	x*1	0	x*1	x*1	0	0	0	0	0	x*1	x*1	x*1

Bit	Symbol	Function	R/W
0	IWDTRF	Independent Watchdog Timer Reset Detect Flag 0: Independent watchdog timer reset not detected 1: Independent watchdog timer reset detected	R/W*2
1	WDTRF	Watchdog Timer Reset Detect Flag 0: Watchdog timer reset not detected 1: Watchdog timer reset detected	R/W*2
2	SWRF	Software Reset Detect Flag 0: Software reset not detected 1: Software reset detected	R/W*2
7:3	—	These bits are read as 0. The write value should be 0.	R/W
8	RPERF	SRAM Parity Error Reset Detect Flag 0: SRAM parity error reset not detected 1: SRAM parity error reset detected	R/W*2
9	REERF	SRAM ECC Error Reset Detect Flag 0: SRAM ECC error reset not detected 1: SRAM ECC error reset detected	R/W*2
10	—	This bit is read as 0. The write value should be 0.	R/W
11	BUSMRF	Bus Master MPU Error Reset Detect Flag 0: Bus master MPU error reset not detected 1: Bus master MPU error reset detected	R/W*2
12	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
13	TZERF	TrustZone Error Reset Detect Flag 0: TrustZone error reset not detected. 1: TrustZone error reset detected.	R/W*2
14	—	This bit is read as 0. The write value should be 0.	R/W
15	CPERF	Cache Parity Error Reset Detect Flag 0: Cache Parity error reset not detected. 1: Cache Parity error reset detected.	R/W*2

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note 1. The value after reset depends on the reset source.
Note 2. Only 0 can be written to clear the flag. The flag must be cleared by writing 0 after 1 is read.

IWDTRF flag (Independent Watchdog Timer Reset Detect Flag)

The IWDTRF flag indicates that an independent watchdog timer reset occurs.

[Setting condition]

- When an independent watchdog timer reset occurs.

[Clearing conditions]

- When a reset listed in Table 5.2 occurs
- When 1 is read and then 0 is written to IWDTRF.

WDTRF flag (Watchdog Timer Reset Detect Flag)

The WDTRF flag indicates that a watchdog timer reset occurs.

[Setting condition]

- When a watchdog timer reset occurs.

[Clearing conditions]

- When a reset listed in Table 5.2 occurs
- When 1 is read and then 0 is written WDTRF.

SWRF flag (Software Reset Detect Flag)

The SWRF flag indicates that a software reset occurs.

[Setting condition]

- When a software reset occurs.

[Clearing conditions]

- When a reset listed in Table 5.2 occurs
- When 1 is read and then 0 is written to SWRF.

RPERF flag (SRAM Parity Error Reset Detect Flag)

The RPERF flag indicates that an SRAM parity error reset occurs.

[Setting condition]

- When an SRAM parity error reset occurs.

[Clearing conditions]

- When a reset listed in Table 5.2 occurs
- When 1 is read as 1 and then 0 is written to RPERF.

REERF flag (SRAM ECC Error Reset Detect Flag)

The REERF flag indicates that an SRAM ECC error reset occurs.

[Setting condition]

- When an SRAM ECC error reset occurs.

[Clearing conditions]

- When a reset listed in Table 5.2 occurs
- When 1 is read as 1 and then 0 is written to REERF.

BUSMRF flag (Bus Master MPU Error Reset Detect Flag)

The BUSMRF flag indicates that a bus master MPU error reset occurs.

[Setting condition]

- When a bus master MPU error reset occurs.

[Clearing conditions]

- When a reset listed in Table 5.2 occurs
- When 1 is read and then 0 is written to BUSMRF.

TZERF flag (TrustZone Error Reset Detect Flag)

The TZERF flag indicates that a TrustZone error reset has occurred.

[Setting condition]

- When a TrustZone error reset occurs.

[Clearing conditions]

- When a reset listed in Table 5.2 occurs
- When 1 is read then and 0 is written to TZERF.

CPERF flag (Cache Parity Error Reset Detect Flag)

The CPERF flag indicates that a Cache Parity error reset has occurred.

[Setting condition]

- When a Cache Parity error reset occurs.

[Clearing conditions]

- When a reset listed in Table 5.2 occurs
- When 1 is read then and 0 is written to CPERF.

5.2.4 RSTSR2 : Reset Status Register 2

Base address: SYSC = 0x4001_E000

Offset address: 0x411

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

Bit	Symbol	Function	R/W
0	CWSF	Cold/Warm Start Determination Flag 0: Cold start 1: Warm start	R/W*2

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

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note 1. The value after reset depends on the reset source.
Note 2. Only 1 can be written to set the flag.

RSTSR2 determines whether a power-on reset caused the reset processing (cold start) or a reset signal input during operation caused the reset processing (warm start).

CWSF flag (Cold/Warm Start Determination Flag)

The CWSF flag indicates the type of reset processing, either cold start or warm start. The determines whether a power-on reset caused the reset processing (cold start) or a reset signal input during operation caused the reset processing (warm start). CWSF flag is initialized by a power-on reset. It is not initialized by a reset signal generated by the RES pin.

[Setting condition]

- When 1 is written by software.

[Clearing condition]

- When a reset listed in Table 5.2 occurs.

5.3 Operation

5.3.1 RES Pin Reset

The RES pin generates this reset. When the RES pin is driven low, all the processing in progress is aborted and the MCU enters a reset state. To successfully reset the MCU, the RES pin must be held low for the power supply stabilization time specified at power-on.

When the RES pin is driven high from low, the internal reset is canceled after the post-RES cancellation wait time (t_{RESWT}) elapses. The CPU then starts the reset exception handling.

For details, see section 46, Electrical Characteristics.

5.3.2 Power-On Reset

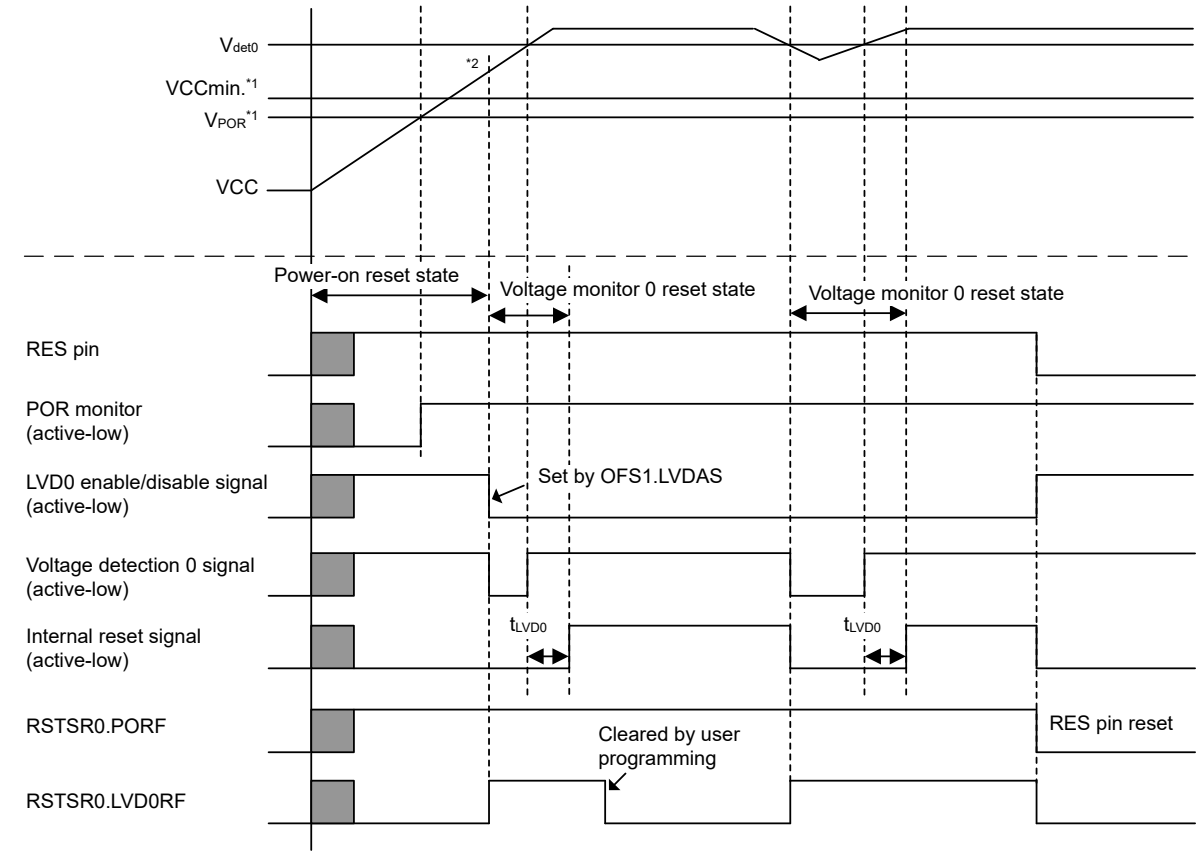
The power-on reset (POR) is an internal reset generated by the power-on reset circuit. A power-on reset is generated under the following conditions.

1. If the RES pin is in a high level state when power is supplied
2. If the RES pin is in a high level state when VCC is below V_{POR}

After VCC exceeds V_{POR} and the specified power-on reset time (t_{POR}) elapses, the CPU starts the reset exception handling. The power-on reset time is a stabilization period of the external power supply and the MCU circuit.

After a power-on reset is generated, the PORF flag in the RSTSR0 is set to 1. The PORF flag is initialized by the RES pin reset. When VCC falls below V_{POR}, a power-on reset state is occurred.

Figure 5.1 shows example of operations during a power-on reset.



Note: For details on the electrical characteristics, see section 46, Electrical Characteristics.
Note 1. V_{POR} shows a power-on reset detection level, and VCCmin shows the minimum guaranteed voltage of MCU.
Note 2. At power-on, VCC should rise to the minimum guaranteed voltage before the power-on reset is released.

Figure 5.1 Example of operations during a power-on reset

5.3.3 Voltage Monitor Reset

The voltage monitor i (i = 0, 1, 2) reset is an internal reset generated by the voltage monitor i circuit. If the Voltage Detection 0 Circuit Start (LVDAS) bit in the Option Function Select Register 1 (OFS1) is 0 (voltage monitor 0 reset is enabled after a reset) and VCC falls below V_{det0}, the RSTSR0.LVD0RF flag becomes 1 and the voltage detection circuit generates voltage monitor 0 reset. Clear the OFS1.LVDAS bit to 0 if the voltage monitor 0 reset is to be used. After VCC exceeds V_{det0} and the voltage monitor 0 reset time (t_{LVD0}) elapses, the internal reset is canceled and the CPU starts the reset exception handling.

When the Voltage Monitor 1 Interrupt/Reset Enable bit (RIE) is set to 1 (enabling generation of a reset or interrupt by the voltage detection circuit) and the Voltage Monitor 1 Circuit Mode Select bit (RI) is set to 1 (selecting generation of a reset in response to detection of a low voltage) in Voltage Monitor 1 Circuit Control Register 0 (LVD1CR0), the RSTSR0.LVD1RF flag is set to 1 and the voltage detection circuit generates a voltage monitor 1 reset if VCC falls to or below V_{det1}.

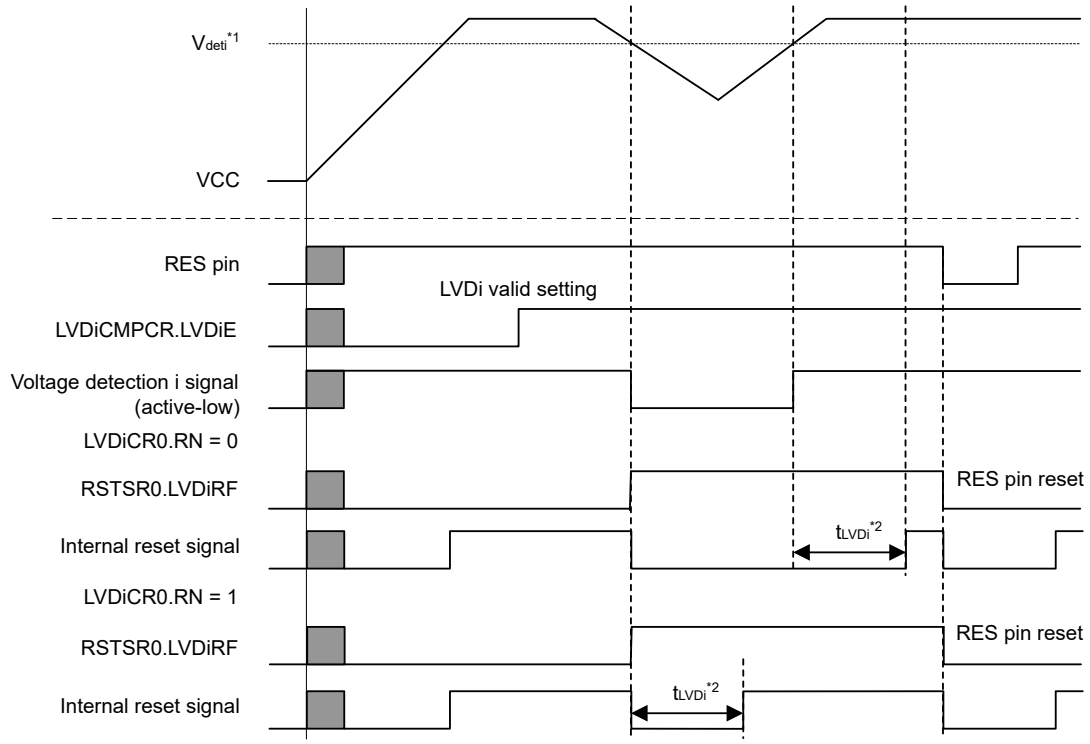
Likewise, when the Voltage Monitor 2 Interrupt/Reset Enable bit (RIE) is set to 1 (enabling generation of a reset or interrupt by the voltage detection circuit) and the Voltage Monitor 2 Circuit Mode Select bit (RI) is set to 1 (selecting generation of a reset in response to detection of a low voltage) in Voltage Monitor 2 Circuit Control Register 0 (LVD2CR0), the RSTSR0.LVD2RF flag is set to 1 and the voltage detection circuit generates a voltage monitor 2 reset if VCC falls to or below V_{det2}.

Similarly, timing for release from the voltage monitor 1 reset state is selectable with the Voltage Monitor 1 Reset Negate Select bit (RN) in the LVD1CR0. When the LVD1CR0.RN bit is 0 and VCC falls to or below V_{det1}, the CPU is released from the internal reset state and starts reset exception handling when the LVD1 reset time (t_{LVD1}) elapses after VCC rises above V_{det1}. When the LVD1CR0.RN bit is 1 and VCC falls to or below V_{det1}, the CPU is released from the internal reset state and starts reset exception handling when the LVD1 reset time (t_{LVD1}) elapses.

Likewise, timing for release from the voltage monitor 2 reset state is selectable by setting the Voltage Monitor 2 Reset Negate Select bit (RN) in the LDV2CR0 register.

Detection levels V_{det1} and V_{det2} can be changed in the Voltage Monitoring Comparator Control Register (LVD1CMPCR/ LVD2CMPCR).

Figure 5.2 shows example of operations during voltage monitor 1 and 2 resets. For details on the voltage monitor 1 reset and voltage monitor 2 reset, see section 7, Low Voltage Detection (LVD).



Note: For details on the electrical characteristics, see section 46, Electrical Characteristics.
Note 1. V_{deti} indicates the detection level of a voltage monitor 1 reset and a voltage monitor 2 reset (i = 1, 2).
Note 2. t_{LVDi} indicates the time for a voltage monitor 1 reset and a voltage monitor 2 reset (i = 1, 2).

Figure 5.2 Example of operations during voltage monitor 1 and voltage monitor 2 resets

5.3.4 Deep Software Standby Reset

This is an internal reset generated when Deep Software Standby mode is canceled by an interrupt.

When a Deep Software Standby mode cancelation source is generated, a Deep Software Standby reset is generated. The Deep Software Standby reset is canceled after t_{DSBY} (return time after Deep Software Standby mode cancelation) has elapsed. At the same time, Deep Software Standby mode is also canceled.

When t_{DSBYWT} (wait time after Deep Software Standby mode cancelation) has elapsed after Deep Software Standby mode has been canceled, the internal reset is canceled and the CPU starts the reset exception handling.

For details of the Deep Software Standby reset, see section 10, Low Power Modes.

5.3.5 Independent Watchdog Timer Reset

The independent watchdog timer reset is an internal reset generated from the Independent Watchdog Timer (IWDT). Output of the reset from the IWDT can be selected in the Option Function Select Register 0 (OFS0).

When output of the independent watchdog timer reset is selected, the reset is generated if the IWDT underflows, or if data is written when refresh operation is disabled. When the internal reset time (t_{RESW2}) elapses after the independent watchdog timer reset is generated, the internal reset is canceled and the CPU starts the reset exception handling.

For details on the independent watchdog timer reset, see section 25, Independent Watchdog Timer (IWDT).

5.3.6 Watchdog Timer Reset

The watchdog timer reset is an internal reset generated from the Watchdog Timer (WDT). Output of the reset from the WDT can be selected in the WDT Reset Control Register (WDTRCR) or Option Function Select register 0 (OFS0).

When output of the watchdog timer reset is selected, a watchdog timer reset is generated if the WDT underflows, or if data is written when refresh operation is disabled. When the internal reset time (t_{RESW2}) elapses after the watchdog timer reset is generated, the internal reset is canceled and the CPU starts the reset exception handling.

For details on the watchdog timer reset, see section 24, Watchdog Timer (WDT).

5.3.7 Software Reset

The software reset is an internal reset generated by a software setting of the SYSRESETREQ bit in the AIRCR register in the Arm core. When the SYSRESETREQ bit is set to 1, a software reset is generated. When the internal reset time (t_{RESW2}) elapses after the software reset is generated, the internal reset is canceled and the CPU starts the reset exception handling.

For details on the SYSRESETREQ bit, see the *ARM® Cortex®-M33 Technical Reference Manual*.

5.3.8 Determination of Cold/Warm Start

Read the CWSF flag in RSTSR2 to determine the cause of reset processing. This flag indicates whether a power-on reset caused the reset processing (cold start) or a reset signal input during operation caused the reset processing (warm start).

The CWSF flag is set to 0 when a power-on reset occurs (cold start), otherwise the flag is not set to 0. The flag is set to 1 when 1 is written to it through software. It is not set to 0 even on writing 0 to it.

Figure 5.3 shows an example of cold/warm start determination operation.

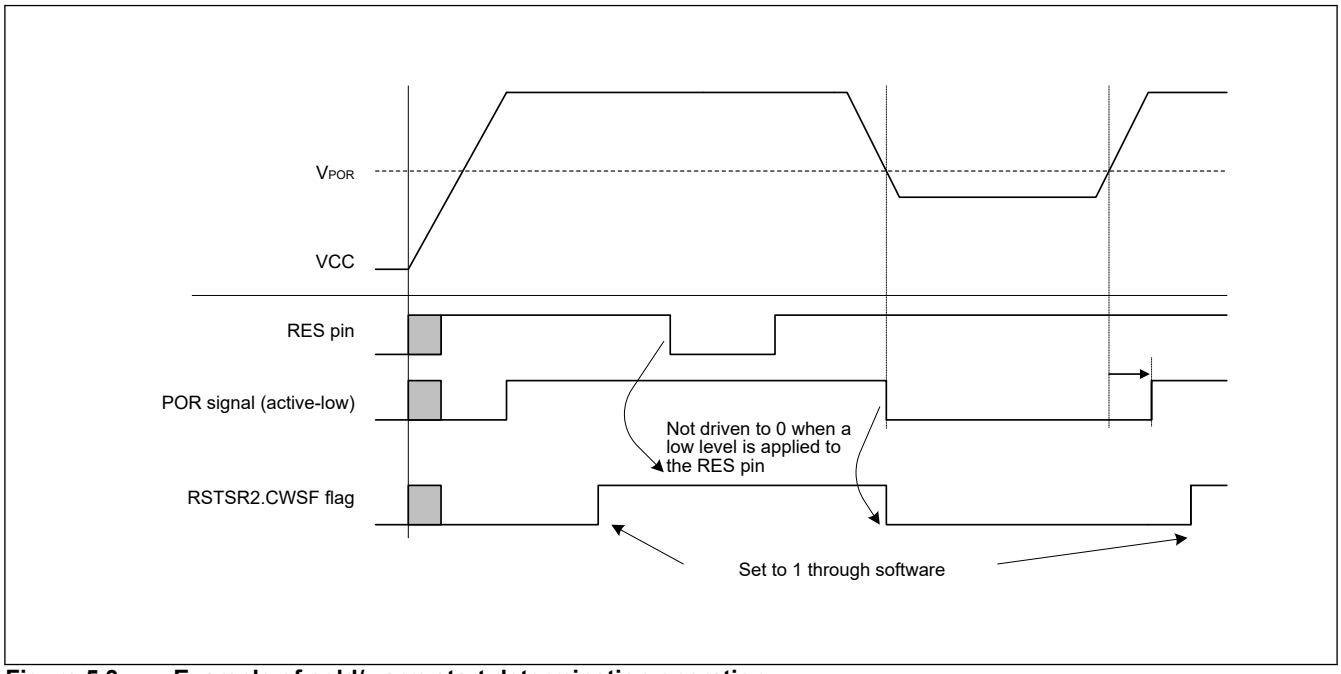
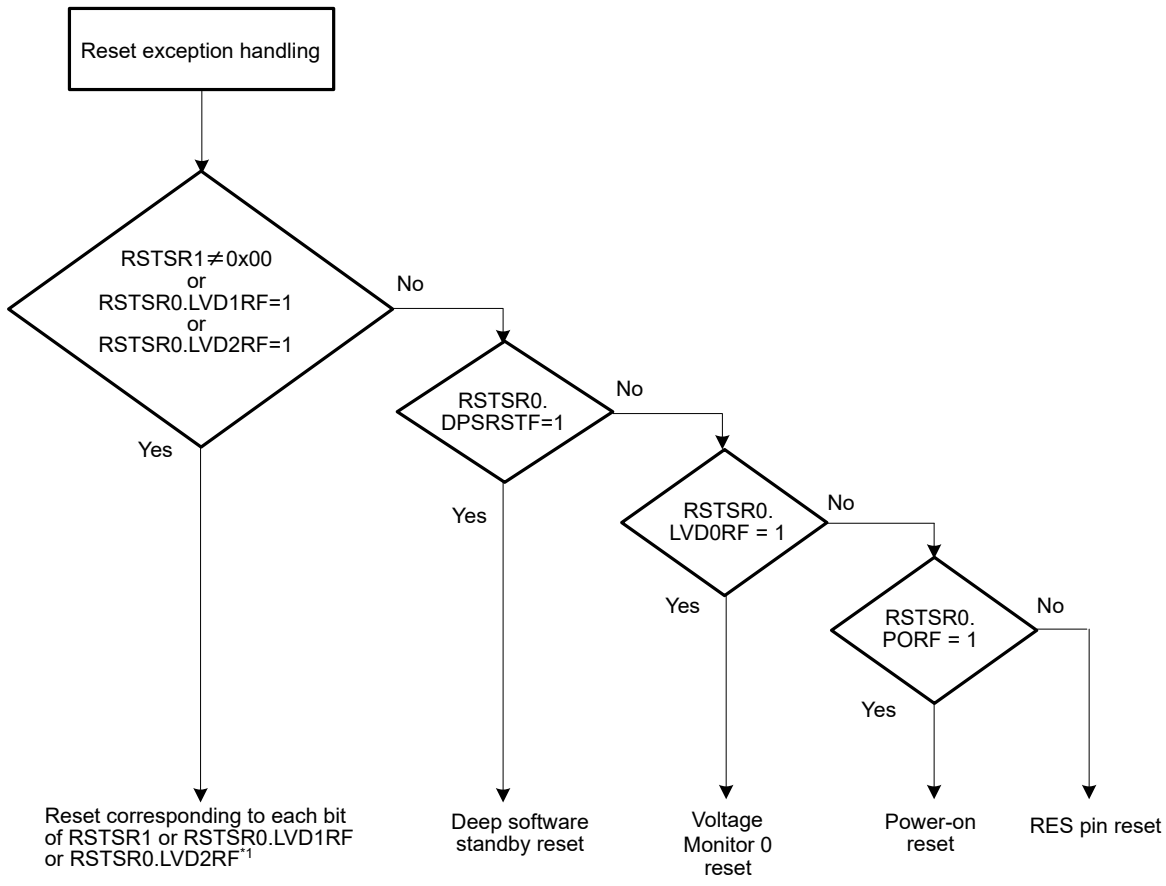


Figure 5.3 Example of cold/warm start determination operation

5.3.9 Determination of Reset Generation Source

Read RSTSR0 and RSTSR1 to determine which reset executes the reset exception handling.

Figure 5.4 shows an example of the flow to identify a reset generation source. The reset flag must be written with 0 after it is read as 1.



Note 1. When resets are generated simultaneously by two or more sources, the two or more corresponding reset flags will be set to 1.

Figure 5.4 Example of reset generation source determination flow

6. Option-Setting Memory

6.1 Overview

The option-setting memory determines the state of the MCU after a reset. The option-setting memory is allocated to the configuration setting area of the flash memory.

Figure 6.1 shows the option-setting memory area. The option-setting memory area has secure region. Table 6.1 shows the programming condition of the option-setting memory area.

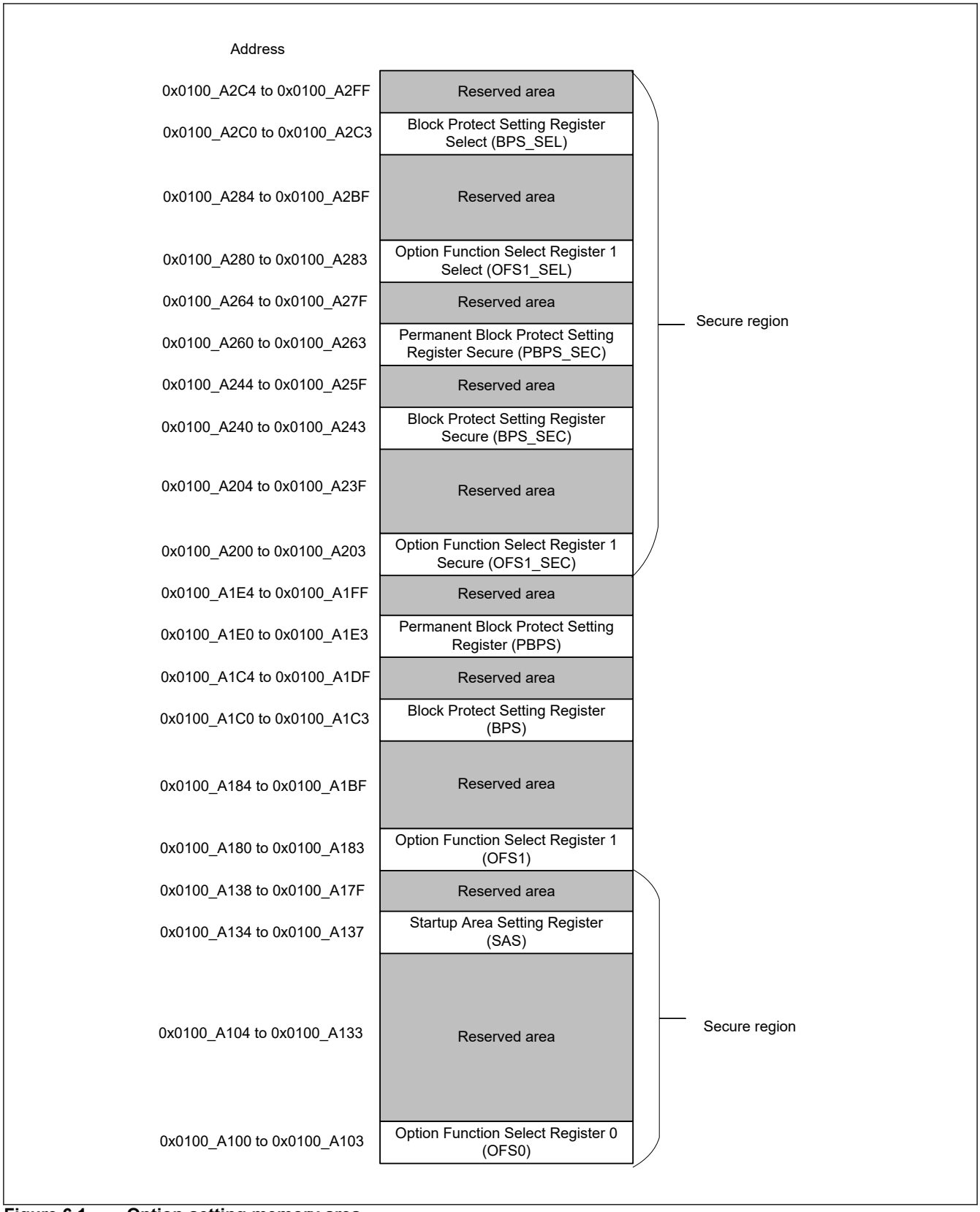


Figure 6.1 Option-setting memory area

Table 6.1 The programming condition of the option-setting memory area

	Self programming	Serial programming	Programming by the on-chip debugger
Secure region	Programming commands issued by secure access.	Programming commands issued when the device life cycle is SSD.	Programming commands issued when the debug level is DBG2.
Other region	Programming commands issued by secure or non-secure access.	Programming commands issued when the devise life cycle is SSD or NSECSD.	Programming commands issued when the debug level is DBG2 or DBG1.

6.2 Register Descriptions

6.2.1 OFS0 : Option Function Select Register 0

Address: 0x0100_A100

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	WDTSTPCTL	—	WDTRSTIRQS	WDTRPSS[1:0]		WDTRPES[1:0]		WDTCKS[3:0]				WDTTOPS[1:0]		WDTSTRT	—
Value after reset:	User setting*1															

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	IWDTSTPCTL	—	IWDTRSTIRQS	IWDTRPSS[1:0]		IWDTRPES[1:0]		IWDTCKS[3:0]				IWDTTOPS[1:0]		IWDTSTRT	—
Value after reset:	User setting*1															

Bit	Symbol	Function	R/W
0	—	When read, this bit returns the written value. The write value should be 1.	R
1	IWDTSTRT	IWDT Start Mode Select 0: Automatically activate IWDT after a reset (auto start mode) 1: Disable IWDT after a reset	R
3:2	IWDTTOPS[1:0]	IWDT Timeout Period Select 0 0: 128 cycles (0x007F) 0 1: 512 cycles (0x01FF) 1 0: 1024 cycles (0x03FF) 1 1: 2048 cycles (0x07FF)	R
7:4	IWDTCKS[3:0]	IWDT-Dedicated Clock Frequency Division Ratio Select 0x0: × 1 0x2: × 1/16 0x3: × 1/32 0x4: × 1/64 0xF: × 1/128 0x5: × 1/256 Others: Setting prohibited	R
9:8	IWDTRPES[1:0]	IWDT Window End Position Select 0 0: 75% 0 1: 50% 1 0: 25% 1 1: 0% (no window end position setting)	R
11:10	IWDTRPSS[1:0]	IWDT Window Start Position Select 0 0: 25% 0 1: 50% 1 0: 75% 1 1: 100% (no window start position setting)	R
12	IWDTRSTIRQS	IWDT Reset Interrupt Request Select 0: Interrupt 1: Reset	R

Bit	Symbol	Function	R/W
13	—	When read, this bit returns the written value. The write value should be 1.	R
14	IWDTSTPCTL	IWDT Stop Control 0: Continue counting 1: Stop counting when in Sleep, Snooze, or Software Standby mode	R
16:15	—	When read, these bits return the written value. The write value should be 1.	R
17	WDTSTRT	WDT Start Mode Select 0: Automatically activate WDT after a reset (auto start mode) 1: Stop WDT after a reset (register start mode)	R
19:18	WDTTOPS[1:0]	WDT Timeout Period Select 0 0: 1024 cycles (0x03FF) 0 1: 4096 cycles (0x0FFF) 1 0: 8192 cycles (0x1FFF) 1 1: 16384 cycles (0x3FFF)	R
23:20	WDTCKS[3:0]	WDT Clock Frequency Division Ratio Select 0x1: PCLKB divided by 4 0x4: PCLKB divided by 64 0xF: PCLKB divided by 128 0x6: PCLKB divided by 512 0x7: PCLKB divided by 2048 0x8: PCLKB divided by 8192 Others: Setting prohibited	R
25:24	WDTRPES[1:0]	WDT Window End Position Select 0 0: 75% 0 1: 50% 1 0: 25% 1 1: 0% (no window end position setting)	R
27:26	WDTRPSS[1:0]	WDT Window Start Position Select 0 0: 25% 0 1: 50% 1 0: 75% 1 1: 100% (no window start position setting)	R
28	WDTRSTIRQS	WDT Reset Interrupt Request Select 0: Interrupt 1: Reset	R
29	—	When read, these bits return the written value. The write value should be 1.	R
30	WDTSTPCTL	WDT Stop Control 0: Continue counting 1: Stop counting when entering Sleep mode	R
31	—	When read, these bits return the written value. The write value should be 1.	R

Note 1. The value in a blank product is 0xFFFFFFFF. It is set to the value written by your application.

IWDTSTRT bit (IWDT Start Mode Select)

The IWDTSTRT bit selects the mode in which the IWDT is activated after a reset (stopped state or activated state).

IWDTTOPS[1:0] bits (IWDT Timeout Period Select)

The IWDTTOPS[1:0] bits specify the timeout period, that is, the time it takes for the down counter to underflow, as 128, 512, 1024, or 2048 cycles of the frequency-divided clock set in the IWDTCKS[3:0] bits. The time it takes for the counter to underflow after a refresh operation is determined by the combination of the IWDTCKS[3:0] and IWDTTOPS[1:0] bits.

For details, see section 25, Independent Watchdog Timer (IWDT).

IWDTCKS[3:0] bits (IWDT-Dedicated Clock Frequency Division Ratio Select)

The IWDTCKS[3:0] bits specify the division ratio of the prescaler for dividing the frequency of the clock for the IWDT as 1/1, 1/16, 1/32, 1/64, 1/128, and 1/256. Using this setting combined with the IWDTTOPS[1:0] bits setting, the IWDT counting period can be set from 128 to 524288 IWDT clock cycles.

For details, see section 25, Independent Watchdog Timer (IWDT).

IWDTRPES[1:0] bits (IWDT Window End Position Select)

The IWDTRPES[1:0] bits specify the position where the window for the down counter ends as 0%, 25%, 50%, or 75% of the count value. The value of the window end position must be smaller than the value of the window start position, otherwise only the value for the window start position is valid.

The counter values associated with the settings for the start and end positions of the window in the IWDTRPSS[1:0] and IWDTRPES[1:0] bits vary with the setting in the IWDTTOPS[1:0] bits.

For details, see section 25, Independent Watchdog Timer (IWDT).

IWDTRPSS[1:0] bits (IWDT Window Start Position Select)

The IWDTRPSS[1:0] bits specify the position where the window for the down counter starts as 25%, 50%, 75%, or 100% of the counted value. The point at which counting starts is 100% and the point at which an underflow occurs is 0%. The interval between the window starts and ends positions becomes the period in which a refresh is possible. Refresh is not possible outside this period.

For details, see section 25, Independent Watchdog Timer (IWDT).

IWDTRSTIRQS bit (IWDT Reset Interrupt Request Select)

The IWDTRSTIRQS bit selects the operation on an underflow of the down counter or generation of a refresh error. The operation is selectable to an independent watchdog timer reset, a non-maskable interrupt request, or an interrupt request.

For details, see section 25, Independent Watchdog Timer (IWDT).

IWDTSTPCTL bit (IWDT Stop Control)

The IWDTSTPCTL bit specifies whether to stop counting when entering Sleep mode, Snooze mode, or Software Standby mode.

For details, see section 25, Independent Watchdog Timer (IWDT).

WDTSTRT bit (WDT Start Mode Select)

The WDTSTRT bit selects the mode in which the WDT is activated after a reset (stopped state or activated in auto start mode). When WDT is activated in auto start mode, the OFS0 register setting for the WDT is valid.

WDTTOPS[1:0] bits (WDT Timeout Period Select)

The WDTTOPS[1:0] bits specify the timeout period, that is, the time it takes for the down counter to underflow as 1024, 4096, 8192, or 16384 cycles of the frequency-divided clock set in the WDTCKS[3:0] bits. The number of PCLKB cycles that takes to underflow after a refresh operation is determined by a combination of the WDTCKS[3:0] and WDTTOPS[1:0] bits.

For details, see section 24, Watchdog Timer (WDT).

WDTCKS[3:0] bits (WDT Clock Frequency Division Ratio Select)

The WDTCKS[3:0] bits specify the division ratio of the prescaler for dividing the frequency of PCLKB as 1/4, 1/64, 1/128, 1/512, 1/2048, and 1/8192. Using this setting combined with the WDTTOPS[1:0] bits setting, the WDT counting period can be set from 4096 to 134217728 PCLKB cycles.

For details, see section 24, Watchdog Timer (WDT).

WDTRPES[1:0] bits (WDT Window End Position Select)

The WDTRPES[1:0] bits specify the position where the window on the down counter ends as 0%, 25%, 50%, or 75% of the counted value. The value of the window end position must be smaller than the value of the window start position, otherwise only the value for the window start position is valid.

The counter values associated with the settings for the start and end positions of the window in the WDTRPSS[1:0] and WDTRPES[1:0] bits vary with the setting of the WDTTOPS[1:0] bits.

For details, see section 24, Watchdog Timer (WDT).

WDTRPSS[1:0] bits (WDT Window Start Position Select)

The WDTRPSS[1:0] bits specify the position where the window for the down counter starts as 25%, 50%, 75%, or 100% of the counted value. The point at which counting starts is 100% and the point at which an underflow occurs is 0%. The interval between the positions where the window starts and ends becomes the period in which a refresh is possible.

Refresh is not possible outside this period.

For details, see section 24, Watchdog Timer (WDT).

WDTRSTIRQS bit (WDT Reset Interrupt Request Select)

The WDTRSTIRQS bit selects the operation on an underflow of the down-counter or generation of a refresh error. The operation is selectable to a watchdog timer reset, a non-maskable interrupt request, or an interrupt request.

For details, see section 24, Watchdog Timer (WDT).

WDTSTPCTL bit (WDT Stop Control)

The WDTSTPCTL bit specifies whether to stop counting when entering Sleep mode.

For details, see section 24, Watchdog Timer (WDT).

6.2.2 SAS : Startup Area Setting Register

Address: 0x0100_A134

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

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

Bit	Symbol	Function	R/W
14:0	—	When read, these bits return the written value. The write value should be 1.	R
15	FSPR	Protection of Startup Area Select Function This bit controls the programming of the write/erase protection for the Startup Area Select flag (SAS.BTFLG), and the temporary boot swap control. When this bit is set to 0, it cannot be changed to 1. 0: Executing the configuration setting command for programming the Startup Area Select flag (SAS.BTFLG) is invalid. 1: Executing the configuration setting command for programming the Startup Area Select flag (SAS.BTFLG) is valid.	R
30:16	—	When read, these bits return the written value. The write value should be 1.	R
31	BTFLG	Startup Area Select Flag This bit specifies whether the address of the startup area is exchanged for the boot swap function or not. 0: First 8-KB area (0x0000_0000 to 0x0000_1FFF) and second 8-KB area (0x0000_2000 to 0x0000_3FFF) are exchanged. 1: First 8-KB area (0x0000_0000 to 0x0000_1FFF) and second 8-KB area (0x0000_2000 to 0x0000_3FFF) are not exchanged.	R

6.2.3 OFS1, OFS1_SEC, OFS1_SEL : Option Function Select Register 1

Address: OFS1: 0x0100_A180
OFS1_SEC: 0x0100_A200
OFS1_SEL: 0x0100_A280

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	PGADEN[3:0]			
Value after reset:	The value set by the user*1															

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	HOCOFRQ0[1:0]		HOCOEN	—	—	—	—	—	LVDAS	VDSEL[1:0]	
Value after reset:	The value set by the user*1															

Bit	Symbol	Function	R/W
1:0	VDSEL[1:0]	Voltage Detection 0 Level Select 0 0: Setting prohibited 0 1: Select 2.94 V 1 0: Select 2.87 V 1 1: Select 2.80 V	R
2	LVDAS	Voltage Detection 0 Circuit Start 0: Enable voltage monitor 0 reset after a reset 1: Disable voltage monitor 0 reset after a reset	R
7:3	—	When read, these bits return the written value. The write value should be 1.	R
8	HOCOEN	HOCO Oscillation Enable 0: Enable HOCO oscillation after a reset 1: Disable HOCO oscillation after a reset	R
10:9	HOCOFRQ0[1:0]	HOCO Frequency Setting 0 0 0: 16 MHz 0 1: 18 MHz 1 0: 20 MHz 1 1: Setting prohibited	R
15:11	—	When read, these bits return the written value. The write value should be 1.	R
19:16	PGADEN[3:0]	PGAn pseudo-differential input Enable (n = 0 to 3) If not set, the pseudo-differential input of PGAn is enabled. Set this bit to 0 when pseudo-differential input to the PGAs is not to be used. 0: After reset, PGAn pseudo-differential input disabled (single-ended input) 1: After reset, PGAn pseudo-differential input enabled	R
31:20	—	When read, these bits return the written value. The write value shoud be 1.	R

Note 1. The value in a blank product is 0xFFFFFFFF. It is set to the value written by your application.

Only secure developer can program OFS1_SEC and OFS1_SEL registers. OFS1_SEC register is for secure developer, and OFS1 register is for non-secure developer. The applied setting value is determined by the setting value of the corresponding bit in OFS1_SEL register. For details, see section 6.3.3. Security attribution of option-setting memory.

VDSEL[1:0] bits (Voltage Detection 0 Level Select)

The VDSEL[1:0] bits select the voltage detection level of the voltage detection 0 circuit.

LVDAS bits (Voltage Detection 0 Circuit Start)

The LVDAS bit selects whether the voltage monitor 0 reset is enabled or disabled after a reset.

HOCOEN bit (HOCO Oscillation Enable)

The HOCOEN bit selects whether the HOCO oscillation is enabled or disabled after a reset. Setting this bit to 0 allows the HOCO oscillation to start before the CPU starts operation, which reduces the wait time for oscillation stabilization.

Note: When the HOCOEN bit is set to 0, the system clock source is not switched to HOCO. The system clock source is only switched to HOCO by setting the Clock Source Select bits (SCKSCR.CKSEL[2:0]). To use the HOCO clock, you must set the OFS1.HOCOFRQ0 bit to an optimum value.

HOCOFRQ0[1:0] bits (HOCO Frequency Setting 0)

The HOCOFRQ0[1:0] bits specify the HOCO frequency after a reset as 16, 18, or 20 MHz.

PGADEN[3:0] bits (PGAn pseudo-differential input Enable)

This bit selects whether the pseudo-differential input on the PGAn pin is enabled or disabled (single-ended input) after a reset.

6.2.4 BPS, BPS_SEC, BPS_SEL : Block Protect Setting Register

address:

BPS: 0x0100_A1C0
BPS_SEC: 0x0100_A240
BPS_SEL: 0x0100_A2C0

Bit position:	31	0
Bit field:		
Value after reset:	User setting*1	

Note 1. The value in a blank product is 0xFFFFFFFF. It is set to the value written by your application.

Only secure developer can program BPS_SEC and BPS_SEL registers. BPS_SEC register is for secure developer, and BPS register is for non-secure developer. The applied setting value is determined by the setting value of the corresponding bit in BPS_SEL register. For details, see section 6.3.3. Security attribution of option-setting memory.

The BPS and BPS_SEC registers invalidate the programming and erasure to the code flash memory. When the bit of this register is set to 0, the programming and erasure to the corresponding block are invalid. Figure 6.2 shows the code flash block structure of each product. Figure 6.3 shows the relationship between the bit of register and the block number. Unused bits are reserved and should be set to 1.

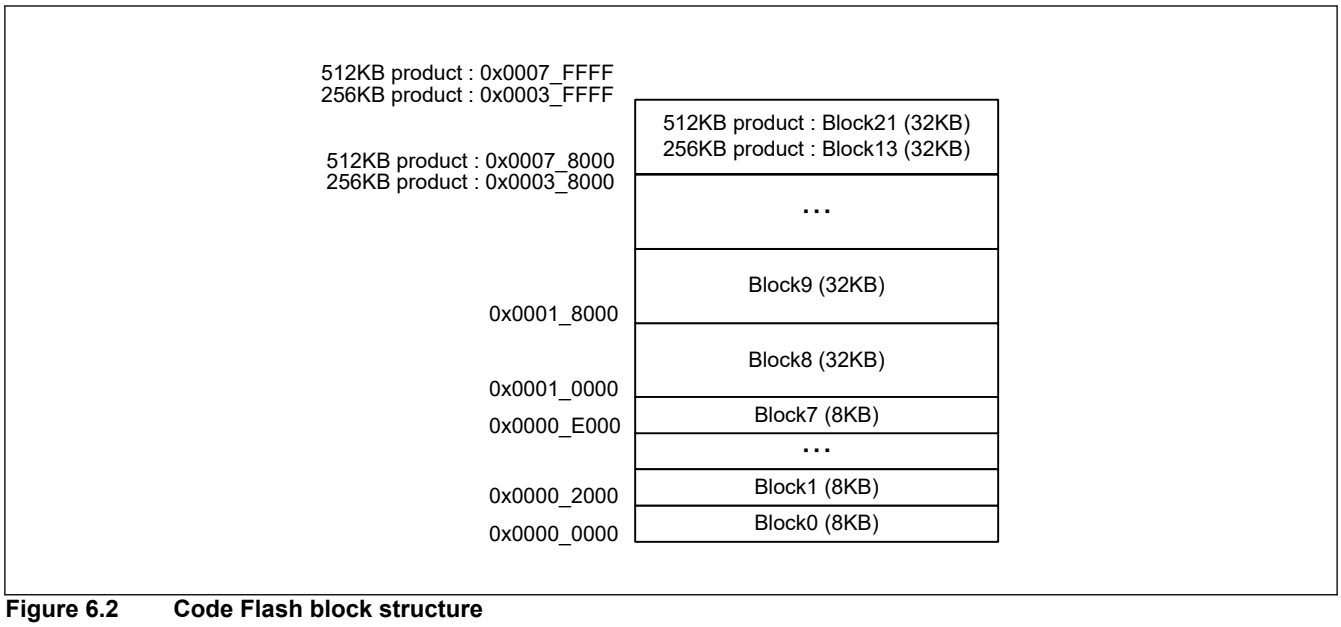


Figure 6.2 Code Flash block structure

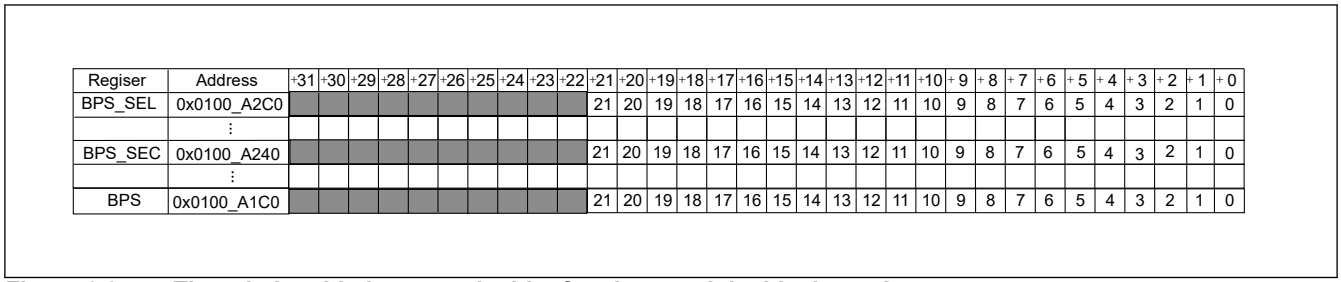
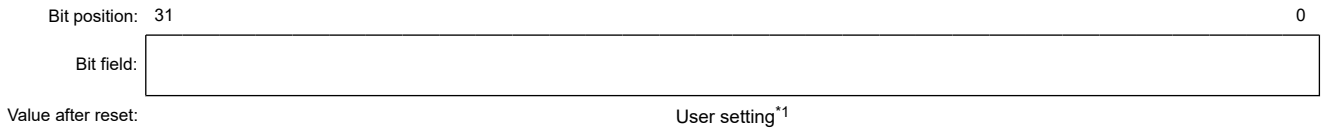


Figure 6.3 The relationship between the bit of register and the block number

6.2.5 PBPS, PBPS_SEC : Permanent Block Protect Setting Register

Address: PBPS: 0x0100_A1E0
PBPS_SEC: 0x0100_A260



Note 1. The value in a blank product is 0xFFFFFFFF. It is set to the value written by your application.

Only secure developer can program PBPS_SEC register. PBPS_SEC register is for secure developer, and PBPS register is for non-secure developer. The applied setting value is determined by the setting value of the corresponding bit in BPS_SEL register. For details, see section 6.3.3. Security attribution of option-setting memory. The security attribution register is same BPS_SEL register between the block protection and permanent block protection.

The PBPS and PBPS_SEC registers invalidate writes to bits of BPS and BPS_SEC. The bit of this register can be set to 0 when corresponding bit of BPS and BPS_SEC is set to 0. When the bit of this register is set to 0, writing the corresponding bit of BPS and BPS_SEC register is invalid. Once the bit of this register is set to 0, it is impossible to change the bit to 1. Table 6.2 shows the relationship between the bit of applied PBPS and bit of applied BPS.

The relationship between the bit of this register and the block number is same as BPS and BPS_SEC registers (section 6.2.4. BPS, BPS_SEC, BPS_SEL : Block Protect Setting Register). Unused bits are reserved and should be set to 1.

Table 6.2 The relationship between the bit of PBPS, PBPS_SEC and bit of BPS, BPS_SEC

The bit of applied PBPS	The bit of applied BPS	Content
1	1	Programming and erasure to the corresponding block is valid.
1	0	Programming and erasure to the corresponding block is invalid. This protection can be canceled by FBPROT0 or FBPROT1 registers.
0	1	Cannot set this condition
0	0	Programming and erasure to the corresponding block is invalid permanently

6.3 Setting Option-Setting Memory

6.3.1 Allocation of Data in Option-Setting Memory

Programming data is allocated to the addresses in the option-setting memory shown in Figure 6.1. The allocated data is used by tools such as a flash programming software or an on-chip debugger.

Note: Programming formats vary depending on the compiler. See the compiler manual for details.

6.3.2 Setting Data for Programming Option-Setting Memory

Allocating data according to the procedure described in section 6.3.1. Allocation of Data in Option-Setting Memory, alone does not actually write the data to the option-setting memory. You must also follow one of the actions described in this section.

(1) Changing the option-setting memory by self-programming

Use the configuration setting command to write data to the option-setting memory in the configuration setting area.

The option-setting memory does not support background operations (BGO). When write the option-setting memory, jump to SRAM after copying writing softwere to SRAM.

For details of the configuration setting command, see section 43, Flash Memory.

(2) Debugging through an OCD or programming by a flash writer

This procedure depends on the tool in use, see the tool manual for details.

The MCU provides two setting procedures:

- Read the data allocated as described in section 6.3.1. Allocation of Data in Option-Setting Memory, from an object file or Motorola S-format file generated by the compiler, and write the data to the MCU
- Use the GUI interface of the tool to program the same data as allocated in section 6.3.1. Allocation of Data in Option-Setting Memory.

6.3.3 Security attribution of option-setting memory

Some functionality has 3 registers for non-secure (FUNC NAME), and secure (FUNC NAME_SEC), and security attribution (FUNC NAME_SEL). Only secure developer can set the registers for secure and security attribution. As shown in Figure 6.4, when the bit of security attribution register is set to 0, the corresponding bit of secure register is applied. When the bit of security attribution register is set to 1, the corresponding bit of non-secure register is applied.

For example, if the secure developer wants to configure LVD of OFS1 as secure, HOCO and PGA of OFS1 as non-secure, the secure developer needs to set OFS1_SEL as follows.

OFS1_SEL = 0xFFFF_FFF8

By this setting, LVDAS and VDSEL[1:0] values of OFS1_SEC and HOCOFRQ0[1:0], HOCOEN and PGADEN[3:0] values of OFS1 are applied to MCU. The reserved bits of the security attribution register (FUNC NAME_SEL) should be set to 1.

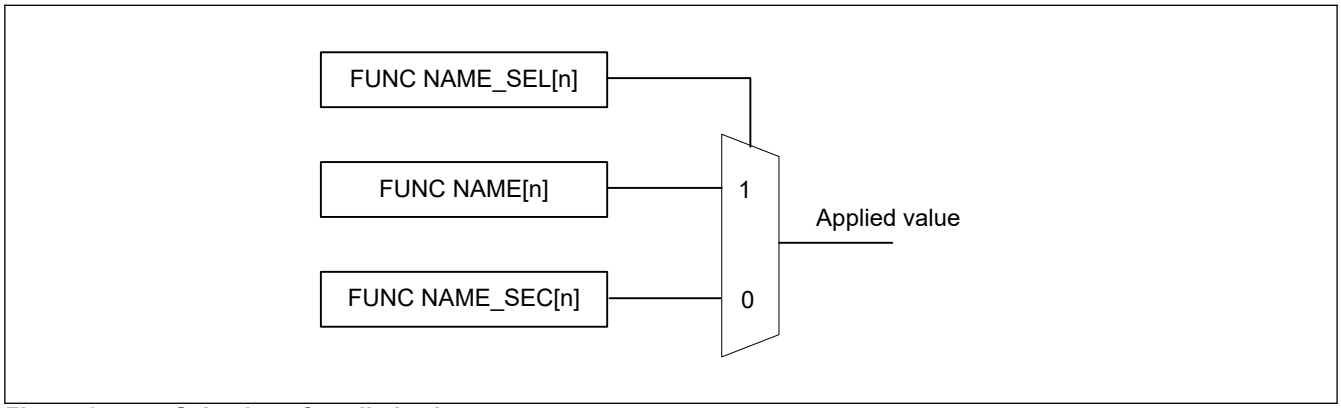


Figure 6.4 Selection of applied value

6.3.4 Timing of the Setting Value

For SAS, BPS, BPS_SEC, PBPS, and PBPS_SEC registers, the setting value of the related startup area and block protection is applied immediately after programming. For other registers, the setting value is applied after the MCU is reset.

In case the programming using the serial programming mode in customer’s factory, be careful that the block protection for secure user is applied after MCU is reset. Because initial value of the security attribution registers of block protection (BPS_SEL) is 1 (non-secure), the block protection setting for secure developer (BPS_SEC/PBPS_SEC) is not applied until MCU is reset even if the corresponding bit of BPS_SEL is programmed to 0 (secure).

6.4 Usage Notes

6.4.1 Data for Programming Reserved Areas and Reserved Bits in the Option-Setting Memory

When reserved areas and reserved bits in the option-setting memory are within the scope of programming, write 1 to all bits of reserved areas and all reserved bits. If 0 is written to these bits, normal operation cannot be guaranteed.

7. Low Voltage Detection (LVD)

7.1 Overview

The Low Voltage Detection (LVD) module monitors the voltage level input to the VCC pin. The detection level can be selected by register settings. The LVD module consists of three separate voltage level detectors (LVD0, LVD1, LVD2). LVD0, LVD1, and LVD2 measure the voltage level input to the VCC pin. LVD registers allow your application to configure detection of VCC changes at various voltage thresholds.

Voltage monitor registers are used to configure the LVD to trigger an interrupt, event link output, or reset when the thresholds are crossed.

Table 7.1 lists the LVD specifications. Figure 7.1 shows a block diagram of the voltage monitor 0 reset generation circuit. Figure 7.2 shows a block diagram of the voltage monitor 1 interrupt and reset circuit, and Figure 7.3 shows a block diagram of the voltage monitor 2 interrupt and reset circuit.

Table 7.1 LVD specifications

Parameter		Voltage monitor 0	Voltage monitor 1	Voltage monitor 2
Means for setting up operation		OFS1 register	Registers	Registers
Target for monitoring		VCC pin input voltage	VCC pin input voltage	VCC pin input voltage
Monitored voltage		V_{det0}	V_{det1}	V_{det2}
Detected event		Voltage falls past V_{det0}	Voltage rises or falls past V_{det1}	Voltage rises or falls past V_{det2}
Detection voltage		Selectable from 3 different levels in the OFS1.VDSEL[1:0] bits	Selectable from 3 different levels in the LVD1CMPCR.LVD1LVL[4:0] bits	Selectable from 3 different levels in the LVD2CMPCR.LVD2LVL[2:0] bits
Monitoring flag		None	LVD1SR.MON flag: Monitors whether voltage is higher or lower than V_{det1}	LVD2SR.MON flag: Monitors whether voltage is higher or lower than V_{det2}
			LVD1SR.DET flag: V_{det1} passage detection	LVD2SR.DET flag: V_{det2} passage detection
Process on voltage detection	Reset	Voltage monitor 0 reset	Voltage monitor 1 reset	Voltage monitor 2 reset
		Reset when V_{det0} > VCC CPU restart after specified time with VCC > V_{det0}	Reset when V_{det1} > VCC CPU restart timing selectable: after specified time with VCC > V_{det1} or V_{det1} > VCC	Reset when V_{det2} > VCC CPU restart timing selectable: after specified time with either VCC > V_{det2} or V_{det2} > VCC
	Interrupt	No interrupt	Voltage monitor 1 interrupt	Voltage monitor 2 interrupt
			Non-maskable or maskable interrupt selectable	Non-maskable or maskable interrupt selectable
			Interrupt request issued when V_{det1} > VCC and VCC > V_{det1} or either	Interrupt request issued when V_{det2} > VCC and VCC > V_{det2} or either
Digital filter	Switching between enable and disable	No digital filter function	Available	Available
	Sampling time	—	1/n LOCO frequency × 2 (n: 2, 4, 8, 16)	1/n LOCO frequency × 2 (n: 2, 4, 8, 16)
Event link function		None	Available Output of event signals on detection of V_{det1} crossings	Available Output of event signals on detection of V_{det2} crossings
TrustZone Filter		—	Security attribution can be set for each registers	

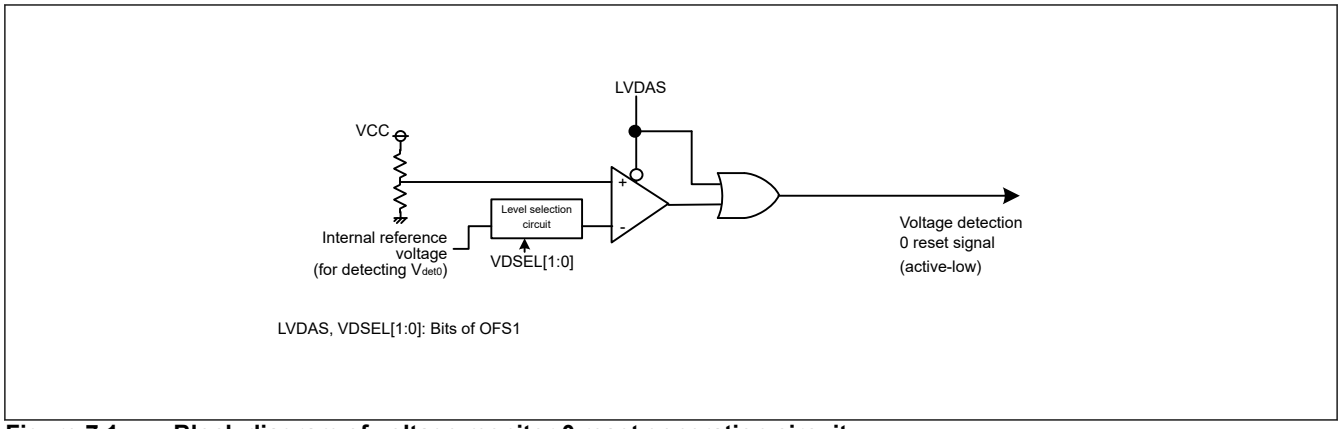


Figure 7.1 Block diagram of voltage monitor 0 reset generation circuit

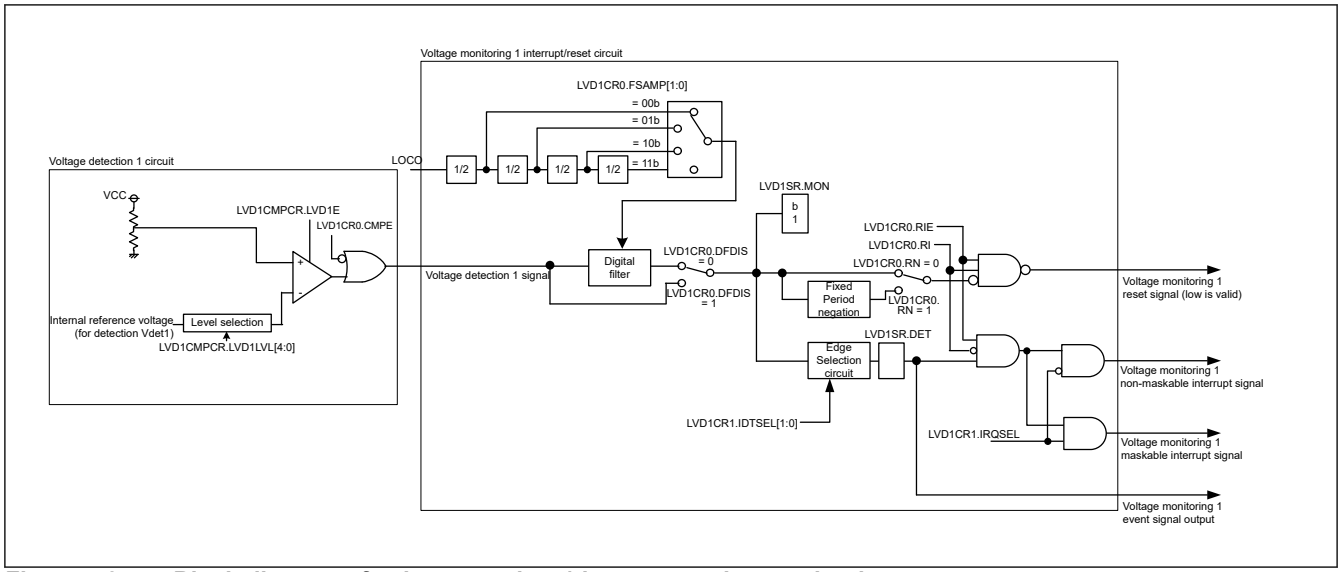


Figure 7.2 Block diagram of voltage monitor 1 interrupt and reset circuit

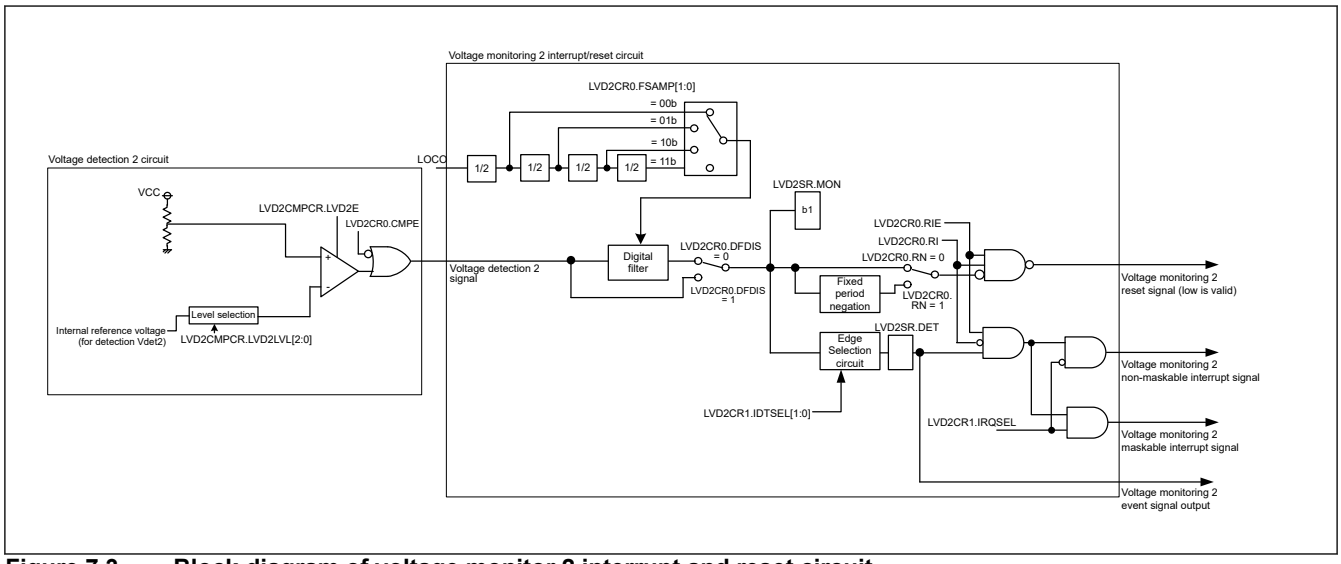


Figure 7.3 Block diagram of voltage monitor 2 interrupt and reset circuit

7.2 Register Descriptions

7.2.1 LVDSAR : Low Voltage Detection Security Attribution Register

Base address: SYSC = 0x4001_E000

Offset address: 0x3CC

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	NONSEC1	NONSEC0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	NONSEC0	Non Secure Attribute bit 0 Target register: registers for LVD1 0: Secure 1: Non Secure	R/W
1	NONSEC1	Non Secure Attribute bit 1 Target register: registers for LVD2 0: Secure 1: Non Secure	R/W
31:2	—	These bits are read as 1. The write value must be 1 when it is possible to write.	R/W

Note: Only secure access can write to this register. Both secure access and Non-secure read access are allowed but Non-secure write access is not allowed and TrustZone access error is not generated.

Note: This register is write-protected by PRCR register.

The LVDSAR register controls the secure attribute of LVD registers.

NONSEC0 bit (Non Secure Attribute bit 0)

This bit controls the security attribute of LVD1CMPCR, LVD1CR0, LVD1CR1, LVD1SR.

NONSEC1 bit (Non Secure Attribute bit 1)

This bit controls the security attribute of LVD2CMPCR, LVD2CR0, LVD2CR1, LVD2SR.

7.2.2 LVD1CMPCR : Voltage Monitoring 1 Comparator Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x417

Bit position:	7	6	5	4	3	2	1	0
Bit field:	LVD1E	—	—	LVD1LVL[4:0]				
Value after reset:	0	0	0	1	0	0	1	1

Bit	Symbol	Function	R/W
4:0	LVD1LVL[4:0]	Voltage Detection 1 Level Select (Standard voltage during drop in voltage) 0x11: 2.99 V (Vdet1_1) 0x12: 2.92 V (Vdet1_2) 0x13: 2.85 V (Vdet1_3) Others: Setting prohibited	R/W
6:5	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
7	LVD1E	Voltage Detection 1 Enable 0: Voltage detection 1 circuit disabled 1: Voltage detection 1 circuit enabled	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.

The LVD1CMPCR.LVD1LVL can be changed only if the LVD1CMPCR.LVD1E and LVD2CMPCR.LVD2E bits are both 0. The voltage detection circuits 1 and 2 should not be set at the same voltage detection level.

Do not change LVD1CMPCR.LVD1LVL and LVD1CMPCR.LVD1E at the same time.

LVD1E bit (Voltage Detection 1 Enable)

When using voltage detection 1 interrupt/reset or the LVD1SR.MON bit, set the LVD1E bit to 1. The voltage detection 1 circuit starts once $t_{d(E-A)}$ passes after the LVD1E bit value is changed from 0 to 1. When using the voltage detection 1 circuit in Deep Software Standby mode, do not set the DPSBYCR.DEEPCUT[1:0] bits to 11b.

7.2.3 LVD2CMPCR : Voltage Monitoring 2 Comparator Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x418

Bit position:	7	6	5	4	3	2	1	0
Bit field:	LVD2E	—	—	—	—	LVD2LVL[2:0]		
Value after reset:	0	0	0	0	0	1	1	1

Bit	Symbol	Function	R/W
2:0	LVD2LVL[2:0]	Voltage Detection 2 Level Select (Standard voltage during drop in voltage) 1 0 1: 2.99 V (Vdet2_1) 1 1 0: 2.92 V (Vdet2_2) 1 1 1: 2.85 V (Vdet2_3) Others: Setting prohibited	R/W
6:3	—	These bits are read as 0. The write value should be 0.	R/W
7	LVD2E	Voltage Detection 2 Enable 0: Voltage detection 2 circuit disabled 1: Voltage detection 2 circuit enabled	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.

The LVD2CMPCR.LVD2LVL can be changed only if the LVD1CMPCR.LVD1E and LVD2CMPCR.LVD2E bits are both 0. The voltage detection circuits 1 and 2 should not be set at the same voltage detection level.

Do not change LVD2CMPCR.LVD2LVL and LVD2CMPCR.LVD2E at the same time.

LVD2E bit (Voltage Detection 2 Enable)

When using voltage detection 2 interrupt/reset or the LVD2SR.MON bit, set the LVD2E bit to 1. The voltage detection 2 circuit starts once $t_{d(E-A)}$ passes after the LVD2E bit value is changed from 0 to 1. When using the voltage detection 2 circuit in Deep Software Standby mode, do not set the DPSBYCR.DEEPCUT[1:0] bits to 11b.

7.2.4 LVD1CR0 : Voltage Monitor 1 Circuit Control Register 0

Base address: SYSC = 0x4001_E000

Offset address: 0x41A

Bit position:	7	6	5	4	3	2	1	0
Bit field:	RN	RI	FSAMP[1:0]		—	CMPE	DFDIS	RIE
Value after reset:	1	0	0	0	x	0	1	0

Bit	Symbol	Function	R/W
0	RIE	Voltage Monitor 1 Interrupt/Reset Enable 0: Disable 1: Enable	R/W
1	DFDIS	Voltage monitor 1 Digital Filter Disabled Mode Select 0: Enable the digital filter 1: Disable the digital filter	R/W
2	CMPE	Voltage Monitor 1 Circuit Comparison Result Output Enable 0: Disable voltage monitor 1 circuit comparison result output 1: Enable voltage monitor 1 circuit comparison result output	R/W
3	—	The read value is undefined. The write value should be 1.	R/W
5:4	FSAMP[1:0]	Sampling Clock Select 0 0: 1/2 LOCO frequency 0 1: 1/4 LOCO frequency 1 0: 1/8 LOCO frequency 1 1: 1/16 LOCO frequency	R/W
6	RI	Voltage Monitor 1 Circuit Mode Select 0: Generate voltage monitor 1 interrupt on V_{det1} crossing 1: Enable voltage monitor 1 reset when the voltage falls to and below V_{det1}	R/W
7	RN	Voltage Monitor 1 Reset Negate Select 0: Negate after a stabilization time (t_{LVD1}) when VCC > V_{det1} is detected 1: Negate after a stabilization time (t_{LVD1}) on assertion of the LVD1 reset	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.

RIE bit (Voltage Monitor 1 Interrupt/Reset Enable)

The RIE bit enables or disables voltage monitor 1 interrupt/reset. Ensure that neither a voltage monitor 1 interrupt nor a voltage monitor 1 reset is generated during programming or erasure of the flash memory.

DFDIS bit (Voltage monitor 1 Digital Filter Disabled Mode Select)

The DFDIS bit disables the digital filter circuit. Set the LOCOCR.LCSTP bit to 0 (the LOCO operates) when this bit is 0(enabled). Set this bit to 1 (disabled) when using the voltage monitor 1 circuit in Software Standby mode or in Deep Software Standby mode.

CMPE bit (Voltage Monitor 1 Circuit Comparison Result Output Enable)

The CMPE bit enables or disables voltage monitor 1 circuit comparison result output. Set the CMPE bit to 1 after the voltage detection 1 circuit enables and stabilization time ($t_{d(E-A)}$) elapses. When stopping the voltage detection 1 circut, disable the voltage detection 1 circuit after setting the CMPE bit is 0.

FSAMP[1:0] bits (Sampling Clock Select)

The FSAMP[1:0] bits can be rewritten only when the LVD1CR0.DFDIS bit is 1 (digital filter circuit disabled). Do not rewrite these bits if the LVD1CR0.DFDIS bit is 0 (digital filter circuit enabled).

RI bit (Voltage Monitor 1 Circuit Mode Select)

When the RI bit is 1 (voltage monitor 1 reset selected), transition to Deep Software Standby mode cannot be made. In this case, transition to Software Standby mode is made. To enter Deep Software Standby mode, set the RI bit to 0 (voltage monitor 1 interrupt selected).

RN bit (Voltage Monitor 1 Reset Negate Select)

If the RN bit is set to 1 (negation follows a stabilization time on assertion of the LVD1 reset signal), set the LOCOCR.LCSTP bit to 0 (the LOCO operates). In addition, for a transition to Software Standby or Deep Software Standby mode, the only possible value for the RN bit is 0 (negation follows stabilization time when VCC > V_{det1} is detected). Do not set the RN bit to 1 when this is the case.

7.2.5 LVD2CR0 : Voltage Monitor 2 Circuit Control Register 0

Base address: SYSC = 0x4001_E000

Offset address: 0x41B

Bit position:	7	6	5	4	3	2	1	0
Bit field:	RN	RI	FSAMP[1:0]		—	CMPE	DFDIS	RIE
Value after reset:	1	0	0	0	x	0	1	0

Bit	Symbol	Function	R/W
0	RIE	Voltage Monitor 2 Interrupt/Reset Enable 0: Disable 1: Enable	R/W
1	DFDIS	Voltage monitor 2 Digital Filter Disabled Mode Select 0: Enable the digital filter 1: Disable the digital filter	R/W
2	CMPE	Voltage Monitor 2 Circuit Comparison Result Output Enable 0: Disable voltage monitor 2 circuit comparison result output 1: Enable voltage monitor 2 circuit comparison result output	R/W
3	—	The read value is undefined. The write value should be 1.	R/W
5:4	FSAMP[1:0]	Sampling Clock Select 0 0: 1/2 LOCO frequency 0 1: 1/4 LOCO frequency 1 0: 1/8 LOCO frequency 1 1: 1/16 LOCO frequency	R/W
6	RI	Voltage Monitor 2 Circuit Mode Select 0: Generate voltage monitor 2 interrupt on V_{det2} crossing 1: Enable voltage monitor 2 reset when the voltage falls to and below V_{det2}	R/W
7	RN	Voltage Monitor 2 Reset Negate Select 0: Negate after a stabilization time (t_{LVD2}) when VCC > V_{det2} is detected 1: Negate after a stabilization time (t_{LVD2}) on assertion of the LVD2 reset	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.

RIE bit (Voltage Monitor 2 Interrupt/Reset Enable)

The RIE bit enables or disables the voltage monitor 2 interrupt/reset. Ensure that neither a voltage monitor 2 interrupt nor a voltage monitor 2 reset is generated during programming or erasure of the flash memory.

DFDIS bit (Voltage monitor 2 Digital Filter Disabled Mode Select)

The DFDIS bit disables the digital filter circuit. Set the LOCOCR.LCSTP bit to 0 (the LOCO operates) when this bit is 0 (digital filter enabled). Set this bit to 1 (digital filter disabled) when using the voltage monitor 2 circuit in Software Standby mode or in Deep Software Standby mode.

CMPE bit (Voltage Monitor 2 Circuit Comparison Result Output Enable)

The CMPE bit enables or disables voltage monitor 2 circuit comparison result output. Set the CMPE bit to 1 after the voltage detection 2 circuit enables and stabilization time ($t_{d(E-A)}$) elapses. When stopping the voltage detection 2 circut, disable the voltage detection 2 circuit after setting the CMPE bit is 0.

FSAMP[1:0] bits (Sampling Clock Select)

The FSAMP[1:0] bits can be rewritten only when the LVD2CR0.DFDIS bit is 1 (digital filter circuit disabled). Do not rewrite these bits if the LVD2CR0.DFDIS bit is 0 (digital filter circuit enabled).

RI bit (Voltage Monitor 2 Circuit Mode Select)

When the RI bit is 1 (voltage monitor 2 reset selected), transition to Deep Software Standby mode cannot be made. In this case, transition to Software Standby mode is made. To enter Deep Software Standby mode, set the RI bit to 0 (voltage monitor 2 interrupt selected).

RN bit (Voltage Monitor 2 Reset Negate Select)

If the RN bit is set to 1 (negating LVD2 reset in a specified time after its assertion), set the LOCOCR.LCSTP bit to 0 (the LOCO operates). Additionally, for a transition to Software Standby or Deep Software Standby mode, the only possible value for the RN bit is 0 (negation follows a stabilization time when VCC > V_{det2} is detected). Do not set the RN bit to 1 (negation follows a stabilization time after assertion of the LVD2 reset signal) when this is the case.

7.2.6 LVD1CR1 : Voltage Monitor 1 Circuit Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x0E0

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	IRQSEL	IDTSEL[1:0]	
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
1:0	IDTSEL[1:0]	Voltage Monitor 1 Interrupt Generation Condition Select 0 0: When VCC ≥ V_{det1} (rise) is detected 0 1: When VCC < V_{det1} (fall) is detected 1 0: When fall and rise are detected 1 1: Settings prohibited	R/W
2	IRQSEL	Voltage Monitor 1 Interrupt Type Select 0: Non-maskable interrupt 1: Maskable interrupt*1	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note: Set the PRCR.PRC3 bit to 1 (writing enabled) before rewriting this register.

Note 1. When enabling maskable interrupts, do not change the NMIER.LVD1EN bit value in the ICU from the reset state.

7.2.7 LVD1SR : Voltage Monitor 1 Circuit Status Register

Base address: SYSC = 0x4001_E000

Offset address: 0x0E1

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

Bit	Symbol	Function	R/W
0	DET	Voltage Monitor 1 Voltage Variation Detection Flag 0: Not detected 1: V_{det1} crossing is detected	R/W*1
1	MON	Voltage Monitor 1 Signal Monitor Flag 0: VCC < V_{det1} 1: VCC ≥ V_{det1} or MON is disabled	R
7:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.

Note 1. Only 0 can be written to this bit. After writing 0 to this bit, 2 system clock cycles are required for the bit to be read as 0.

DET flag (Voltage Monitor 1 Voltage Variation Detection Flag)

The DET flag is enabled when the LVD1CMPCR.LVD1E bit is 1 (voltage detection 1 circuit enabled) and the LVD1CR0.CMPE bit is 1 (voltage monitor 1 circuit comparison result output enabled).

When detecting V_{det1}, set the DET flag to 0 after setting LVD1CR0.RIE is 0 (disabled). When setting LVD1CR0.RIE bit to 1 (enabled) after setting it to 0, wait for 2 or more PCLKB cycles which have elapsed.

MON flag (Voltage Monitor 1 Signal Monitor Flag)

The MON flag is enabled when the LVD1CMPCR.LVD1E bit is 1 (voltage detection 1 circuit enabled) and the LVD1CR0.CMPE bit is 1 (voltage monitor 1 circuit comparison result output enabled).

7.2.8 LVD2CR1 : Voltage Monitor 2 Circuit Control Register 1

Base address: SYSC = 0x4001_E000

Offset address: 0x0E2

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	IRQSEL	IDTSEL[1:0]	
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
1:0	IDTSEL[1:0]	Voltage Monitor 2 Interrupt Generation Condition Select 0 0: When VCC ≥ V_{det2} (rise) is detected 0 1: When VCC < V_{det2} (fall) is detected 1 0: When fall and rise are detected 1 1: Settings prohibited	R/W
2	IRQSEL	Voltage Monitor 2 Interrupt Type Select 0: Non-maskable interrupt 1: Maskable interrupt*1	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note: Set the PRCR.PRC3 bit to 1 (writing enabled) before rewriting this register.
Note 1. When enabling maskable interrupts, do not change the NMIER.LVD2EN bit value in the ICU from the reset state.

7.2.9 LVD2SR : Voltage Monitor 2 Circuit Status Register

Base address: SYSC = 0x4001_E000

Offset address: 0x0E3

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

Bit	Symbol	Function	R/W
0	DET	Voltage Monitor 2 Voltage Variation Detection Flag 0: Not detected 1: V_{det2} crossing is detected	R/W*1
1	MON	Voltage Monitor 2 Signal Monitor Flag 0: VCC < V_{det2} 1: VCC ≥ V_{det2} or MON is disabled	R
7:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.
Note 1. Only 0 can be written to this bit. After writing 0 to this bit, 2 system clock cycles are required for the bit to be read as 0.

DET flag (Voltage Monitor 2 Voltage Variation Detection Flag)

The DET flag is enabled when the LVD2CMPCR.LVD2E bit is 1 (voltage detection 2 circuit enabled) and the LVD2CR0.CMPE bit is 1 (voltage monitor 2 circuit comparison result output enabled).

Set the DET flag to 0 after setting LVD2CR0.RIE is 0 (disabled). When setting LVD2CR0.RIE bit to 1 (enabled) after setting it to 0, wait for 2 or more PCLKB cycles which have elapsed.

MON flag (Voltage Monitor 2 Signal Monitor Flag)

The MON flag is enabled when the LVD2CMPCR.LVD2E bit is 1 (voltage detection 2 circuit enabled) and the LVD2CR0.CMPE bit is 1 (voltage monitor 2 circuit comparison result output enabled).

7.3 VCC Input Voltage Monitor

7.3.1 Monitoring V_{det0}

The comparison results from voltage monitor 0 are not available for reading.

7.3.2 Monitoring V_{det1}

Table 7.2 shows the procedures to set up monitoring against V_{det1}. After the settings are complete, the comparison results from voltage monitor 1 can be monitored with the LVD1SR.MON flag.

Table 7.2 Procedures to set up monitoring against V_{det1}

Step		Monitoring the comparison results from voltage monitor 1
Setting up the voltage detection 1 circuit	1	Set LVD1CMPCR.LVD1E = 0 to disable voltage detection 1 before writing to the LVD1CMPCR.LVD1LVL[4:0] bits.
	2	Select the detection voltage in the LVD1CMPCR.LVD1LVL[4:0] bits.
	3	Set LVD1CMPCR.LVD1E = 1 to enable the voltage detection 1 circuit.
	4	Wait for at least $t_{d\ (E-A)}$ for the LVD1 operation stabilization time after LVD1 is enabled.*1
Setting the digital filter*2	5	Select the sampling clock for the digital filter in the LVD1CR0.FSAMP[1:0] bits.
	6	Set LVD1CR0.DFDIS = 0 to enable the digital filter.
	7	Wait for at least 2n + 3 cycles of the LOCO, where n = 2, 4, 8, or 16, and the sampling clock for the digital filter is the LOCO frequency-divided by n.
Enabling output	8	Set LVD1CR0.CMPE = 1 to enable output of the comparison results from voltage monitor 1.

Note 1. Steps 5 to 7 can be performed during the wait time of step 4. For details of $t_{d(E-A)}$, see section 46, Electrical Characteristics.
Note 2. Steps 5 to 7 are not required if the digital filter is not in use.

7.3.3 Monitoring V_{det2}

Table 7.3 shows the procedures to set up monitoring against V_{det2}. After the settings are complete, the comparison results from voltage monitor 2 can be monitored in the LVD2SR.MON flag.

Table 7.3 Procedures to set up monitoring against V_{det2}

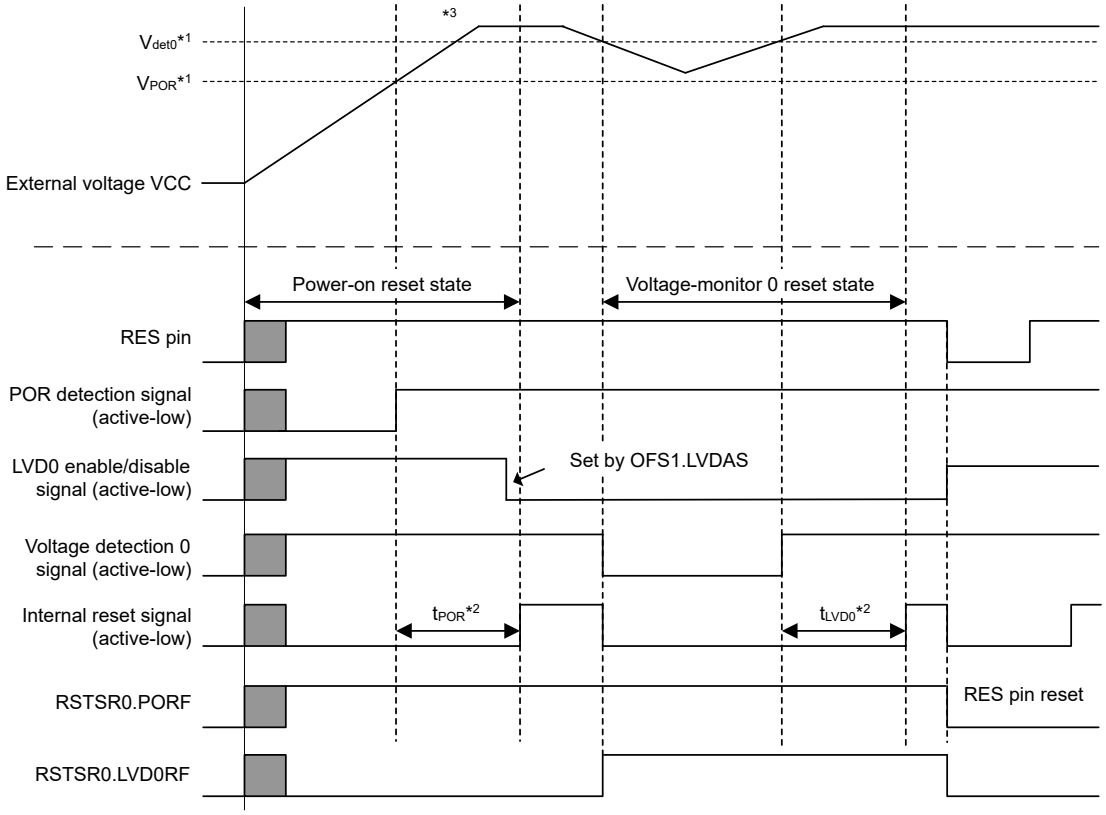
Step		Monitoring the results of comparison by voltage monitor 2
Setting up the voltage detection 2 circuit	1	Set LVD2CMPCR.LVD2E = 0 to disable voltage detection 2 before writing to the LVD2CMPCR.LVD2LVL[2:0] bits.
	2	Select the detection voltage in the LVD2CMPCR.LVD2LVL[2:0] bits.
	3	Set LVD2CMPCR.LVD2E = 1 to enable the voltage detection 2 circuit.
	4	Wait for at least $t_{d\ (E-A)}$ for the LVD2 operation stabilization time after LVD2 is enabled.*1
Setting the digital filter*2	5	Select the sampling clock for the digital filter in the LVD2CR0.FSAMP[1:0] bits.
	6	Set LVD2CR0.DFDIS = 0 to enable the digital filter.
	7	Wait for at least 2n + 3 cycles of the LOCO, where n = 2, 4, 8, or 16, and the sampling clock for the digital filter is the LOCO frequency-divided by n.
Enabling output	8	Set LVD2CR0.CMPE = 1 to enable output of the comparison results from voltage monitor 2.

Note 1. Steps 5 to 7 can be performed during the wait time of step 4. For details of $t_{d(E-A)}$, see section 46, Electrical Characteristics.
Note 2. Steps 5 to 7 are not required if the digital filter is not in use.

7.4 Reset from Voltage Monitor 0

When using the reset from voltage monitor 0, clear the OFS1.LVDAS bit to 0 to enable the voltage monitor 0 reset after a reset. However, at boot mode, the reset from voltage monitor 0 is disabled regardless of the value of the OFS1.LVDAS bit.

Figure 7.4 shows an example of operations for a voltage monitor 0 reset.



Note: For details of the electrical characteristics, see section 46, Electrical Characteristics.
Note 1. V_{POR} indicates the detection level for a power-on reset and V_{det0} indicates the detection level for a voltage monitor 0 reset.
Note 2. t_{POR}: the period until the power-on reset is released.
t_{LVD0}: the period until the LVD0 reset is released.
Note 3. At power-on, the POR reset is released after VCC has reached the minimum guaranteed voltage.

Figure 7.4 Example of voltage monitor 0 reset operation

7.5 Interrupt and Reset from Voltage Monitor 1

An interrupt or reset can be generated in response to the comparison results from the voltage monitor 1 circuit.

Table 7.4 shows the procedures for setting bits related to the voltage monitor 1 interrupt/reset so that voltage monitoring occurs. Table 7.5 shows the procedures for setting bits related to the voltage monitor 1 interrupt/reset so that voltage monitoring stops. Figure 7.5 shows an example of operations for a voltage monitor 1 interrupt. For the operation of the voltage monitor 1 reset, see Figure 5.2 in section 5, Resets.

When using the voltage monitor 1 circuit in Software Standby mode or Deep Software Standby mode, set up the circuit using the procedures in this section.

(1) Setting in Software Standby mode

- Disable the digital filter (LVD1CR0.DFDIS = 1).
- When VCC > V_{det1} is detected, negate the voltage monitor 1 reset signal (LVD1CR0.RN = 0) following a stabilization time.

(2) Settings in Deep Software Standby mode

- Disable the digital filter (LVD1CR0.DFDIS = 1).

- Enable the voltage monitor 1 interrupt (LVD1CR0.RI = 0). If the voltage monitor 1 reset is enabled (LVD1CR0.RI = 1), a transition to Deep Software Standby mode is not possible, and the operation transitions to Software Standby mode instead.
- When the DPSBYCR.DEEPCUT[1:0] bits are 11b, the voltage monitor 1 circuit stops. To use the voltage monitor 1 circuit in Deep Software Standby mode, set the DPSBYCR.DEEPCUT[1:0] bits to a value other than 11b.

Table 7.4 Procedures for setting bits related to voltage monitor 1 interrupt and voltage monitor 1 reset so that voltage monitoring occurs

Step		Voltage monitor 1 interrupt (voltage monitor 1 ELC event output)	Voltage monitor 1 reset
Setting up the voltage detection 1 circuit	1	Set LVD1CMPCR.LVD1E = 0 to disable voltage detection 1 before writing to the LVD1CMPCR register.	
	2	Select the detection voltage in the LVD1CMPCR.LVD1LVL[4:0] bits.	
	3	Set LVD1CMPCR.LVD1E = 1 to enable the voltage detection 1 circuit.	
	4	Wait for at least $t_{d\ (E-A)}$ for the LVD1 operation stabilization time after LVD1 is enabled.*1	
Setting the digital filter*3	5	Select the sampling clock for the digital filter in the LVD1CR0.FSAMP[1:0] bits.	
	6	Set LVD1CR0.DFDIS = 0 to enable the digital filter.	
	7	Wait for at least 2n + 3 LOCO cycles, where n = 2, 4, 8, or 16, and the sampling clock for the digital filter is the LOCO frequency-divided by n.*4	
Setting up the voltage monitor 1 interrupt or reset	8	Set LVD1CR0.RI = 0 to select the voltage monitor 1 interrupt.	• Set LVD1CR0.RI = 1 to select the voltage monitor 1 reset. • Select the type of reset negation in the LVD1CR0.RN bit.
	9	• Select the interrupt request condition in the LVD1CR1.IDTSEL[1:0] bits. • Select the interrupt type in the LVD1CR1.IRQSEL bit.	—
Enabling output	10	Set LVD1SR.DET = 0.	
	11	Set LVD1CR0.RIE = 1 to enable the voltage monitor 1 interrupt or reset.*2	
	12	Set LVD1CR0.CMPE = 1 to enable output of the comparison results from voltage monitor 1.	

Note 1. Steps 5 to 11 can be performed during the wait time in step 4. For details on $t_{d\ (E-A)}$, see section 46, Electrical Characteristics.
Note 2. Step 11 is not required if only the ELC event signal is to be output.
Note 3. Steps 5 to 7 are not required if the digital filter is not in use.
Note 4. Steps 8 to 11 can be performed during the wait time of step 7.

Table 7.5 Procedures for setting bits related to voltage monitor 1 interrupt and voltage monitor 1 reset so that voltage monitoring stops

Step		Voltage monitor 1 interrupt (voltage monitor 1 ELC event output), voltage monitor 1 reset
Stopping the enabling output	1	Set LVD1CR0.CMPE = 0 to disable output of the comparison results from voltage monitor 1.
	2	Wait for at least 2n + 3 cycles of the LOCO, where n = 2, 4, 8, or 16, and the sampling clock for the digital filter is the LOCO frequency-divided by n.*2
	3	Set LVD1CR0.RIE = 0 to disable the voltage monitor 1 interrupt or reset.*1
Stopping the digital filter	4	Set LVD1CR0.DFDIS = 1 to disable the digital filter.*2 *3
Stopping the voltage detection 1 circuit	5	Set LVD1CMPCR.LVD1E = 0 to disable the voltage detection 1 circuit.

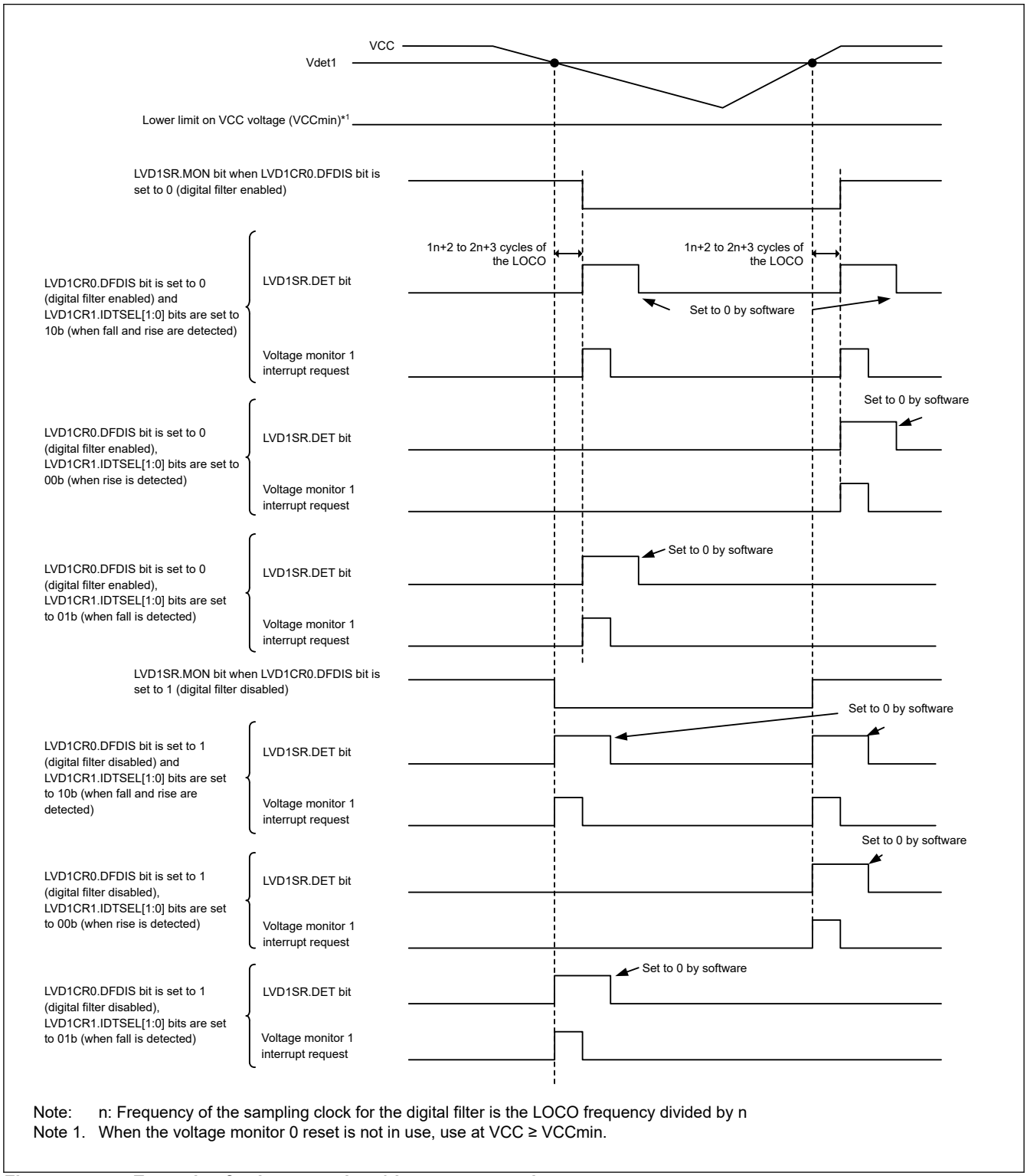
Note 1. Step 3 is not required if only the ELC event signal is to be output.
Note 2. Steps 2 and 4 are not required if the digital filter is not in use.
Note 3. To disable the digital filter from its enabled state and then re-enable it, disable it and wait for at least 2 LOCO clock cycles before re-enabling it.

If the voltage monitor 1 interrupt or reset setting is to be made again after it is used and stopped once, you can omit the following steps in the procedures for stopping and setting, depending on the conditions:

- Setting the voltage detection 1 circuit is not required if the settings for the circuit do not change.
- Setting the digital filter is not required if the settings for the circuit do not change.

- Setting the voltage monitor 1 interrupt or reset is not required if the settings for the voltage monitor 1 interrupt or voltage monitor 1 reset do not change.

Figure 7.5 shows an example of the voltage monitor 1 interrupt operation.



Note: n: Frequency of the sampling clock for the digital filter is the LOCO frequency divided by n
Note 1. When the voltage monitor 0 reset is not in use, use at VCC ≥ VCCmin.

Figure 7.5 Example of voltage monitor 1 interrupt operation

7.6 Interrupt and Reset from Voltage Monitor 2

An interrupt or reset can be generated in response to the comparison results from the voltage monitor 2 circuit.

Table 7.6 shows the procedures for setting bits related to the voltage monitor 2 interrupt/reset so that voltage monitoring occurs. Table 7.7 shows the procedures for setting bits related to the voltage monitor 2 interrupt/reset so that voltage monitoring stops. Figure 7.6 shows an example of operations for a voltage monitor 2 interrupt. For the operation of the voltage monitor 2 reset, see Figure 5.2 in section 5, Resets.

When using the voltage monitor 2 circuit in Software Standby mode or Deep Software Standby mode, set up the circuit with the following procedures.

(1) Setting in Software Standby mode

- Disable the digital filter (LVD2CR0.DFDIS = 1)
- When VCC > V_{det2} is detected, negate the voltage monitor 2 reset signal (LVD2CR0.RN = 0) following a LVD2 stabilization time.

(2) Settings in Deep Software Standby mode

- Disable the digital filter (LVD2CR0.DFDIS = 1).
- Enable the voltage monitor 2 interrupt (LVD2CR0.RI = 0). If the voltage monitor 2 reset is enabled (LVD2CR0.RI = 1), a transition to Deep Software Standby mode is not possible, and the operation transitions to Software Standby mode instead.
- When the DPSBYCR.DEEPCUT[1:0] bits are 11b, the voltage monitor 2 circuit stops. To use the voltage monitor 2 circuit in Deep Software Standby mode, set the DPSBYCR.DEEPCUT[1:0] bits to a value other than 11b.

Table 7.6 Procedures for setting bits related to voltage monitor 2 interrupt and voltage monitor 2 reset so that voltage monitoring occurs

Step		Voltage monitor 2 interrupt (voltage monitor 2 ELC event output)	Voltage monitor 2 reset
Setting up the voltage detection 2 circuit	1	Set LVD2CMPCR.LVD2E = 0 to disable voltage detection 2 before writing to the LVD2CMPCR register.	
	2	Select the detection voltage in the LVD2CMPCR.LVD2LVL[2:0] bits.	
	3	Set LVD2CMPCR.LVD2E = 1 to enable the voltage detection 2 circuit.	
	4	Wait for at least $t_{d\ (E-A)}$ for the LVD2 operation stabilization time after LVD2 is enabled.*1	
Setting the digital filter*3	5	Select the sampling clock for the digital filter in the LVD2CR0.FSAMP[1:0] bits.	
	6	Set LVD2CR0.DFDIS = 0 to enable the digital filter.	
	7	Wait for at least 2n + 3 LOCO cycles, where n = 2, 4, 8, or 16, and the sampling clock for the digital filter is the LOCO frequency-divided by n.*4	
Setting up the voltage monitor 2 interrupt or reset	8	Set LVD2CR0.RI = 0 to select the voltage monitor 2 interrupt.	• Set LVD2CR0.RI = 1 to select the voltage monitor 2 reset. • Select the type of reset negation in the LVD2CR0.RN bit.
	9	• Select the interrupt request condition in the LVD2CR1.IDTSEL[1:0] bits. • Select the interrupt type in the LVD2CR1.IRQSEL bit.	—
Enabling output	10	Set LVD2SR.DET = 0.	
	11	Set LVD2CR0.RIE = 1 to enable the voltage monitor 2 interrupt or reset.*2	
	12	Set LVD2CR0.CMPE = 1 to enable output of the comparison results from voltage monitor 2.	

Note 1. Steps 5 to 11 can be performed during the wait time in step 4. For details on $t_{d(E-A)}$, see section 46, Electrical Characteristics.
Note 2. Step 11 is not required if only the ELC event signal is to be output.
Note 3. Steps 5 to 7 are not required if the digital filter is not in use.
Note 4. Steps 8 to 11 can be performed during the wait time of step 7.

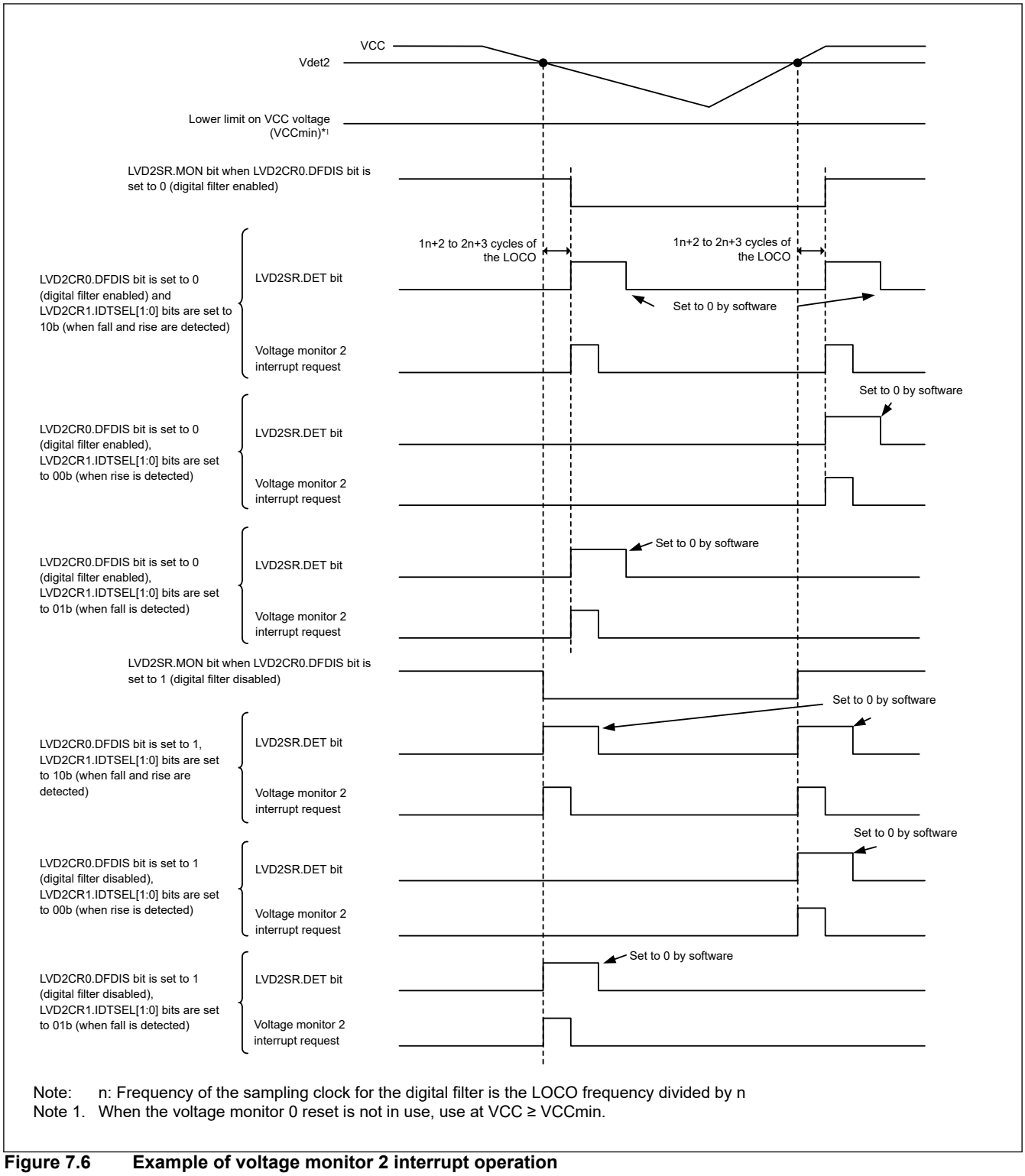
Table 7.7 Procedures for setting bits related to voltage monitor 2 interrupt and voltage monitor 2 reset so that voltage monitoring stops

Step		Voltage monitor 2 interrupt (voltage monitor 2 ELC event output), voltage monitor 2 reset
Settings to stop enabling output	1	Set LVD2CR0.CMPE = 0 to disable output of the comparison results from voltage monitor 2.
	2	Wait for at least 2n + 3 cycles of the LOCO, where n = 2, 4, 8, or 16, and the sampling clock for the digital filter is the LOCO frequency-divided by n.*2
	3	Set LVD2CR0.RIE = 0 to disable the voltage monitor 2 interrupt or reset.*1
Stopping the digital filter	4	Set LVD2CR0.DFDIS = 1 to disable the digital filter. *2 *3
Stopping the voltage detection 2 circuit	5	Set LVD2CMPCR.LVD2E = 0 to disable the voltage detection 2 circuit.

Note 1. Step 3 is not required if only the ELC event signal is to be output.
Note 2. Steps 2 and 4 are not required if the digital filter is not in use.
Note 3. To disable the digital filter from its enabled state and then re-enable it, disable it and wait for at least 2 LOCO clock cycles before re-enabling it.

If the voltage monitor 2 interrupt or reset setting is to be made again after it is used and stopped once, you can omit the following steps in the procedures for stopping and setting, depending on the conditions:

- Setting the voltage detection 2 is not required if the settings for the circuit do not change.
- Setting the digital filter is not required if the settings for the circuit do not change.
- Setting the voltage monitor 2 interrupt or reset is not required if the settings for the voltage monitor 2 interrupt or voltage monitor 2 reset do not change.



Note: n: Frequency of the sampling clock for the digital filter is the LOCO frequency divided by n
Note 1. When the voltage monitor 0 reset is not in use, use at VCC ≥ VCCmin.

Figure 7.6 Example of voltage monitor 2 interrupt operation

7.7 Event Link Controller (ELC) Output

The LVD can output the event signals to the Event Link Controller (ELC).

(1) V_{det1} Crossing Detection Event

The LVD outputs the event signal when it detects that the voltage has passed the V_{det1} voltage while both the voltage detection 1 circuit and the voltage monitor 1 circuit comparison result output are enabled.

(2) V_{det2} Crossing Detection Event

The LVD outputs the event signal when it detects that the voltage has passed the V_{det2} voltage while both the voltage detection 2 circuit and the voltage monitor 2 circuit comparison result output are enabled.

When enabling the event link output function of the LVD, you must enable the LVD before enabling the LVD event link function of the ELC. To stop the event link output function of the LVD, you must stop the LVD before disabling the LVD event link function of the ELC.

7.7.1 Interrupt Handling and Event Linking

The LVD provides bits to separately enable or disable the voltage monitor 1 and 2 interrupts. When an interrupt source is generated and the interrupt is enabled by the interrupt enable bit, the interrupt signal is output to the CPU.

In contrast, as soon as an interrupt source is generated, an event link signal is output as the event signal to the other module through the ELC, regardless of the state of the interrupt enable bit.

It is possible to output voltage monitor 1 and 2 interrupts in Software Standby and Deep Software Standby modes. The event signals for the ELC in Software Standby and Deep Software Standby modes are output as follows:

- When a V_{det1} or V_{det2} passage events is detected in Software Standby mode, event signals are not generated for the ELC because the clock is not supplied in Software Standby mode. Because the V_{det1} and V_{det2} passage detection flags are saved, when the clock supply resumes after returning from Software Standby mode, the event signals for the ELC are output based on the state of the V_{det1} and V_{det2} detection flags.
- When a Vdet1 or Vdet2 passage events are detected in Deep Software Standby mode, event signals are not generated for the ELC.

8. Clock Generation Circuit

8.1 Overview

The MCU provides a clock generation circuit. Table 8.1 and Table 8.2 list the clock generation circuit specifications. Figure 8.1 show a block diagram, and Table 8.3 lists the I/O pins.

Table 8.1 Clock generation circuit specifications for the clock sources

Clock source	Description	Specification
Main clock oscillator (MOSC)	Resonator frequency	8 MHz to 24 MHz
	External clock input frequency	Up to 24 MHz
	External resonator or additional circuit	ceramic resonator, crystal
	Connection pins	EXTAL, XTAL
	Drive capability switching	Available
	Oscillation stop detection function	Available
PLL circuit	Input clock source	MOSC, HOCO
	Input pulse frequency division ratio	Selectable from 1, 2, and 3
	Input frequency	8 MHz to 24 MHz
	Frequency multiplication ratio	Selectable from 10 to 30 (0.5 steps)
	Output pulse frequency division ratio	Unavailable
	PLL Output frequency	120 MHz to 240 MHz
PLL2 circuit	Input clock source	MOSC, HOCO
	Input pulse frequency division ratio	Selectable from 1, 2, and 3
	Input frequency	8 MHz to 24 MHz
	Frequency multiplication ratio	Selectable from 10 to 30 (0.5 steps)
	Output pulse frequency division ratio	Unavailable
	PLL Output frequency	120 MHz to 240 MHz
	PLL2-LDO stop function	Unavailable
High-speed on-chip oscillator (HOCO)	Oscillation frequency	16/18/20 MHz
	FLL function	Unavailable
	User trimming	Available
Middle-speed on-chip oscillator (MOCO)	Oscillation frequency	8 MHz
	User trimming	Available
Low-speed on-chip oscillator (LOCO)	Oscillation frequency	32.768 kHz
	User trimming	Available
IWDT-dedicated on-chip oscillator (IWDTLOCO)	Oscillation frequency	15 kHz
	User trimming	Unavailable
External clock input for JTAG (TCK)	Input clock frequency	Up to 25 MHz
External clock input for SWD (SWCLK)	Input clock frequency	Up to 25 MHz

Table 8.2 Clock generation circuit specifications for the internal clocks (1 of 2)

Item	Clock source	Clock supply	Specification
System clock (ICLK)	MOSC/HOCO/MOCO/ LOCO/PLL	CPU, DTC, DMAC, Flash, RAM, I/O ports, TFU, IIRFA	Up to 240 MHz Division ratios: 1/2/4/8/16/32/64

Table 8.2 Clock generation circuit specifications for the internal clocks (2 of 2)

Item	Clock source	Clock supply	Specification
Peripheral module clock A (PCLKA)	MOSC/HOCO/MOCO/ LOCO/PLL	Peripheral modules (SCI, CANFD-RAM, CNECC, SPI, CRC, DOC, ADC, DAC12, SCE5, GPT bus clock, PDG, IIC)	Up to 120 MHz Division ratio: 1/2/4/8/16/32/64
Peripheral module clock B (PCLKB)	MOSC/HOCO/MOCO/ LOCO/PLL	Peripheral modules (CAC, ELC, POEG, WDT, IWDT, AGT, CANFD, TSN, Standby SRAM, KINT ACMPHS)	Up to 60 MHz Division ratio: 1/2/4/8/16/32/64
Peripheral module clock C (PCLKC)	MOSC/HOCO/MOCO/ LOCO/PLL	Peripheral module (ADC)	Up to 60 MHz Division ratio: 1/2/4/8/16/32/64
Peripheral module clock D (PCLKD)	MOSC/HOCO/MOCO/ LOCO/PLL	Peripheral module (GPT)	Up to 120 MHz Division ratio: 1/2/4/8/16/32/64
FlashIF clock (FCLK)	MOSC/HOCO/MOCO/ LOCO/PLL	FlashIF	4 MHz to 60 MHz(P/E) Up to 60 MHz(read) Division ratio: 1/2/4/8/16/32/64
CANFD clock (CANFDCLK)	PLL/PLL2	CANFD	Up to 40 MHz Division ratio: 1/2/4/6/8
CAN clock (CANMCLK)	MOSC	CANFD	8 MHz to 24 MHz
Peripheral module asynchronous clock for GPT (GPTCLK)	MOSC/HOCO/MOCO/ LOCO/PLL/PLL2	GPT	Up to 200 MHz Division ratio: 1/2/4/6/8
Peripheral module asynchronous clock for IIC (IICCLK)	MOSC/HOCO/MOCO/ LOCO/PLL/PLL2	IIC	Up to 200 MHz Division ratio: 1/2/4/6/8
Peripheral module asynchronous clock for SCI/SPI (SCISPICLK)	MOSC/HOCO/MOCO/ LOCO/PLL/PLL2	SCI, SPI	Up to 120 MHz Division ratio: 1/2/4/6/8
AGT clock (AGTLCLK)	LOCO	AGT	32.768 kHz
CAC Main clock (CACMCLK)	MOSC	CAC	Up to 24 MHz
CAC LOCO clock (CACLCLK)	LOCO	CAC	32.768 kHz
CAC MOCO clock (CACMOCLK)	MOCO	CAC	8 MHz
CAC HOCO clock (CACHCLK)	HOCO	CAC	16/18/20 MHz
CAC IWDTLOCO clock (CACILCLK)	IWDTLOCO	CAC	15 kHz
IWDT clock (IWDTCLK)	IWDTLOCO	IWDT	15 kHz
SysTick timer clock (SYSTICCLK)	LOCO	SysTick timer	32.768 kHz
JTAG clock (JTAGTCK)	TCK	JTAG	Up to 25 MHz
Serial wire clock (SWCLK)	SWCLK	OCD	Up to 25 MHz
Trace clock (TRCLK)	MOSC/HOCO/MOCO/ LOCO/PLL	CPU-OCD	Up to 120 MHz Division ratio: 1/2/4
TCLK pin output (TCLK)	1/2 TRCLK	TCLK pin	Up to 60 MHz
Clock/buzzer output (CLKOUT)	MOSC/LOCO/MOCO/HOCO	CLKOUT pin	Up to 24 MHz Division ratios: 1/2/4/8/16/32/64/128

Note: Restrictions on setting clock frequency: ICLK ≥ PCLKA ≥ PCLKB, PCLKD ≥ PCLKA ≥ PCLKB, GPTCLK ≥ PCLKA
ICLK ≥ FCLK
Restrictions on clock frequency ratio: (N: integer, and up to 64)
ICLK:FCLK = N:1, ICLK:PCLKA = N:1, ICLK:PCLKB = N:1, ICLK:PCLKC = N:1 or 1:N , ICLK:PCLKD= N:1 or 1:N, ICLK:TRCLK= N:1 or 1:N
If the CANFD is used, clock frequency ratio is constrained to be PCLKA:PCLKB = 2:1.

Note: Restrictions on the minimum FCLK frequency 4MHz when P/E.

Note: The multiplication of PLL and PLL2 should be set to be within the output frequency range of PLL and PLL2, taking the frequency of HOCO into consideration. The division of PLL and PLL2 input also should be set to be within the input frequency range of PLL and PLL2, taking the frequency of HOCO into consideration.

Note: Clocks have a permissible frequency range (See Table 8.2).
Flash memory also have a permissible operating frequency range in each wait cycle setting. (See section 43, Flash Memory)
Those clock frequency ranges must be satisfied even if the HOCO has its maximum or minimum frequency. (See section 46, Electrical Characteristics).

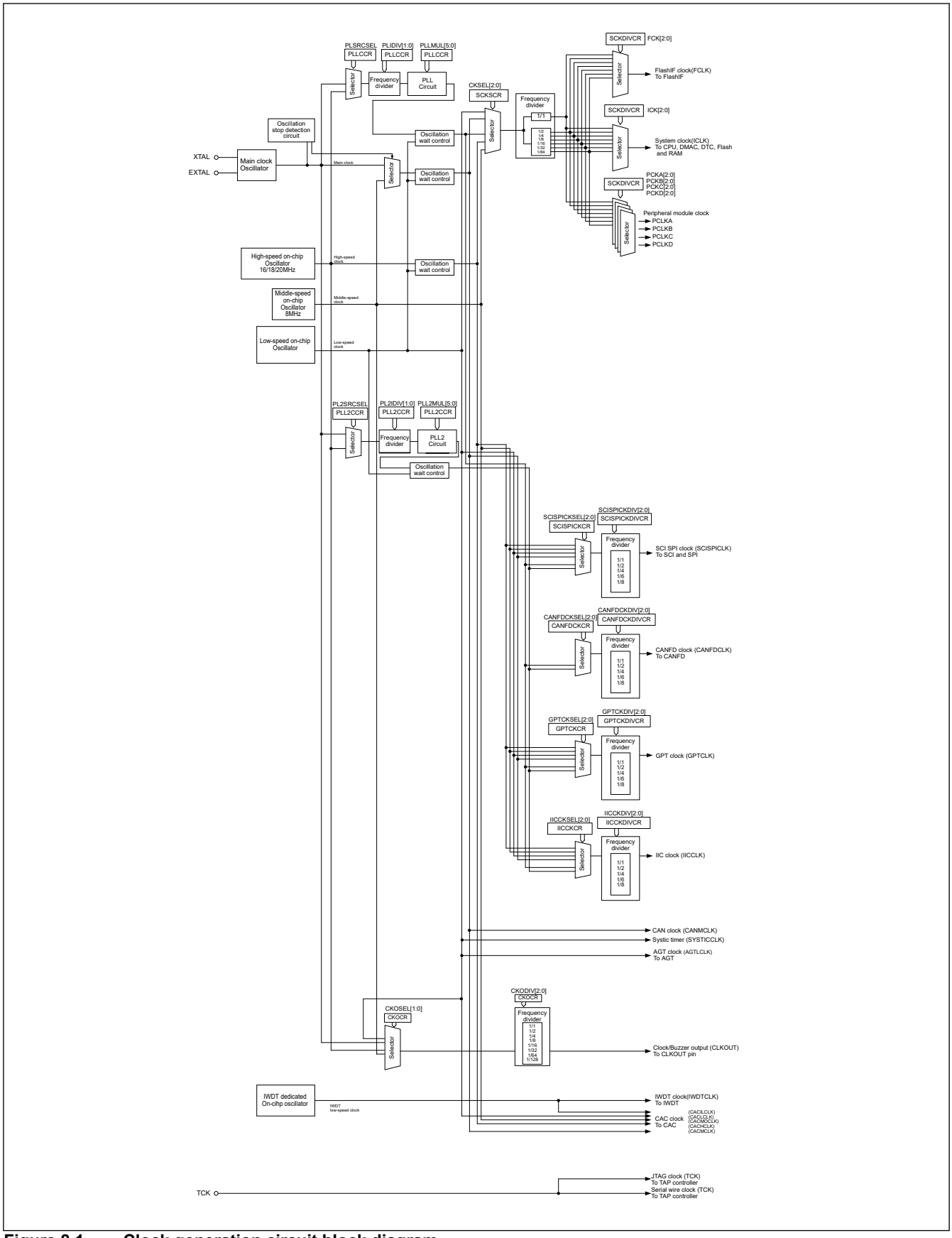


Figure 8.1 Clock generation circuit block diagram

Table 8.3 lists the input/output pins of the clock generation circuit.

Table 8.3 Input/Output Pins of Clock Generation Circuit

Pin name	I/O	Description
XTAL	Output	These pins are used to connect a ceramic resonator or crystal resonator. The EXTAL pin can also be used to input an external clock. For details, see section 8.3.2. External Clock Input.
EXTAL	Input	
TCK/SWCLK	Input	This pin is used to input the clock for the JTAG/SWD
CLKOUT	Output	This pin is used to output the CLKOUT/BUZZER clock

8.2 Register Descriptions

8.2.1 CGFSAR : Clock Generation Function Security Attribute Register

Base address: SYSC = 0x4001_E000

Offset address: 0x3C0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	NONSEC20	NONSEC19	NONSEC18	NONSEC17	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	NONSEC11	—	NONSEC09	NONSEC08	—	NONSEC06	NONSEC05	NONSEC04	NONSEC03	NONSEC02	—	NONSEC00
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	NONSEC00*1	Non Secure Attribute bit 00 Target register: SCKDIVCR, SCKSCR Target factor: system clock control 0: Secure 1: Non Secure	R/W
1	—	This bit is read as 1. The write value should be 1.	R/W
2	NONSEC02*1	Non Secure Attribute bit 02 Target register: HOCOCR, HOCOUTCR Target factor: HOCO 0: Secure 1: Non Secure	R/W
3	NONSEC03*1	Non Secure Attribute bit 03 Target register: MOCOCR, MOCOUTCR Target factor: MOCO 0: Secure 1: Non Secure	R/W
4	NONSEC04	Non Secure Attribute bit 04 Target register: LOCOCR, LOCOUTCR Target factor: LOCO 0: Secure 1: Non Secure	R/W
5	NONSEC05	Non Secure Attribute bit 05 Target register: MOSCCR, MOSCWTCR, MOMCR Target factor: MOSC 0: Secure 1: Non Secure	R/W
6	NONSEC06	Non Secure Attribute bit 06 Target register: OSTDCR, OSTDSR Target factor: oscillation stop detection control 0: Secure 1: Non Secure	R/W
7	—	This bit is read as 1. The write value should be 1.	R/W

Bit	Symbol	Function	R/W
8	NONSEC08*1	Non Secure Attribute bit 08 Target register: PLLCCR, PLLCR Target factor: PLL 0: Secure 1: Non Secure	R/W
9	NONSEC09	Non Secure Attribute bit 09 Target register: PLL2CCR, PLL2CR Target factor: PLL2 0: Secure 1: Non Secure	R/W
10	—	This bit is read as 1. The write value should be 1.	R/W
11	NONSEC11	Non Secure Attribute bit 11 Target register: CKOCR Target factor: CLKOUT control 0: Secure 1: Non Secure	R/W
16:12	—	These bits are read as 1. The write value should be 1.	R/W
17	NONSEC17	Non Secure Attribute bit 17 Target register: SCISPICKDIVCR, SCISPICKCR Target factor: SCISPICLK 0: Secure 1: Non Secure	R/W
18	NONSEC18	Non Secure Attribute bit 18 Target register: CANFDCKDIVCR, CANFDCKCR Target factor: CANFDCLK 0: Secure 1: Non Secure	R/W
19	NONSEC19	Non Secure Attribute bit 19 Target register: GPTCKDIVCR, GPTCKCR Target factor: GPTCLK 0: Secure 1: Non Secure	R/W
20	NONSEC20	Non Secure Attribute bit 20 Target register: IICCKDIVCR, IICCKCR Target factor: IICCLK 0: Secure 1: Non Secure	R/W
31:21	—	These bits are read as 1. The write value should be 1.	R/W

Note: Only secure access can write to this register. Both secure access and Non-secure read access are allowed but Non-secure write access is not allowed and TrustZone access error is not generated.
Note: This register is write-protected by PRCR register.
Note 1. It is recommended that these bits are configured as Non Secure when the device life cycle is NSECSD (DLMMON.DLMMON[3:0] = 0011b). See section 45.6.1. Restrictions on setting the security attribution for the details.

CGFSAR register controls the secure attribute of Clock Generation Function registers.

NONSEC00 bit (Non Secure Attribute bit 00)

This bit controls the security attribute of SCKDIVCR, SCKSCR.

NONSEC02 bit (Non Secure Attribute bit 02)

This bit controls the security attribute of HOCOCR, HOCOUTCR.

NONSEC03 bit (Non Secure Attribute bit 03)

This bit controls the security attribute of MOCOCR, MOCOUTCR.

NONSEC04 bit (Non Secure Attribute bit 04)

This bit controls the security attribute of LOCOCR, LOCOUTCR.

NONSEC05 bit (Non Secure Attribute bit 05)

This bit controls the security attribute of MOSCCR, MOSCWTCR, MOMCR.

NONSEC06 bit (Non Secure Attribute bit 06)

This bit controls the security attribute of OSTDCR, OSTDSR.

NONSEC08 bit (Non Secure Attribute bit 08)

This bit controls the security attribute of PLLCCR, PLLCR.

NONSEC09 bit (Non Secure Attribute bit 09)

This bit controls the security attribute of PLL2CCR, PLL2CR.

NONSEC11 bit (Non Secure Attribute bit 11)

This bit controls the security attribute of CKOCR.

NONSEC17 bit (Non Secure Attribute bit 17)

This bit controls the security attribute of SCISPICKDIVCR, SCISPICKCR.

NONSEC18 bit (Non Secure Attribute bit 18)

This bit controls the security attribute of CANFDCKDIVCR, CANFDCKCR.

NONSEC19 bit (Non Secure Attribute bit 19)

This bit controls the security attribute of GPTCKDIVCR, GPTCKCR.

NONSEC20 bit (Non Secure Attribute bit 20)

This bit controls the security attribute of IICCKDIVCR, IICCKCR.

8.2.2 SCKDIVCR : System Clock Division Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x020

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

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

Bit	Symbol	Function	R/W
2:0	PCKD[2:0]*3	Peripheral Module Clock D (PCLKD) Select 0 0 0: × 1/1 0 0 1: × 1/2 0 1 0: × 1/4 0 1 1: × 1/8 1 0 0: × 1/16 1 0 1: × 1/32 1 1 0: × 1/64 Others: Setting prohibited.	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
6:4	PCKC[2:0]*3	Peripheral Module Clock C (PCLKC) Select 0 0 0: × 1/1 0 0 1: × 1/2 0 1 0: × 1/4 0 1 1: × 1/8 1 0 0: × 1/16 1 0 1: × 1/32 1 1 0: × 1/64 Others: Setting prohibited.	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
10:8	PCKB[2:0]*2	Peripheral Module Clock B (PCLKB) Select 0 0 0: × 1/1 0 0 1: × 1/2 0 1 0: × 1/4 0 1 1: × 1/8 1 0 0: × 1/16 1 0 1: × 1/32 1 1 0: × 1/64 Others: Setting prohibited.	R/W
11	—	This bit is read as 0. The write value should be 0.	R/W
14:12	PCKA[2:0]*2	Peripheral Module Clock A (PCLKA) Select 0 0 0: × 1/1 0 0 1: × 1/2 0 1 0: × 1/4 0 1 1: × 1/8 1 0 0: × 1/16 1 0 1: × 1/32 1 1 0: × 1/64 Others: Setting prohibited.	R/W
15	—	This bit is read as 0. The write value should be 0.	R/W
18:16	RSV	Reserved. Set these bits to the same value as PCKB[2:0]. 0 0 0: × 1/1 0 0 1: × 1/2 0 1 0: × 1/4 0 1 1: × 1/8 1 0 0: × 1/16 1 0 1: × 1/32 1 1 0: × 1/64 Others: Settings prohibited	R/W
23:19	—	These bits are read as 0. The write value should be 0.	R/W
26:24	ICK[2:0]*1*2*3*4	System Clock (ICLK) Select 0 0 0: × 1/1 0 0 1: × 1/2 0 1 0: × 1/4 0 1 1: × 1/8 1 0 0: × 1/16 1 0 1: × 1/32 1 1 0: × 1/64 Others: Setting prohibited.	R/W
27	—	This bit is read as 0. The write value should be 0.	R/W
30:28	FCK[2:0]*1	FlashIF Clock (FCLK) Select 0 0 0: × 1/1 0 0 1: × 1/2 0 1 0: × 1/4 0 1 1: × 1/8 1 0 0: × 1/16 1 0 1: × 1/32 1 1 0: × 1/64 Others: Setting prohibited.	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note 1. The following relation is required between the frequencies of the system clock (ICLK) and the FlashIF clock (FCLK).
ICLK:FCLK=N:1 (N: integer)
Note 2. The following relation is required between the frequencies of the system clock (ICLK) and the peripheral module clocks (PCLKA, PCLKB)
ICLK:PCLKA = N:1, ICLK:PCLKB = N:1 (N: integer)
Note 3. The following relation is required between the frequencies of the system clock (ICLK) and the peripheral module clocks (PCLKC, PCLKD):
ICLK:PCLKC,PCLKD = N:1or1:N (N: integer)
Note 4. The frequency of the system clock (ICLK) is limited to the flash wait cycle register (FLWT). See section 43, Flash Memory.

SCKDIVCR selects the frequencies of the system clock (ICLK), peripheral module clock (PCLKA, PCLKB, PCLKC, PCLKD), FlashIF clock (FCLK).

When the PLL is selected as the clock source, you must enter the following modules in module-stop state before changing the value of this register: ADC, SCE5.

In addition, when changing any value in SCKDIVCR from a lower division ratio to a higher division ratio, wait at least 750 ns before the changing the value and change as follows according to the ICLK frequency before the change.

- When the ICLK frequency before change is more than 120 MHz:
  At first, wait 5 µs after setting ICLK frequency division ratio to 1/2, then wait 5 µs after setting SCKDIVCR.
- When the ICLK frequency before change is 120 MHz or less:
  Wait 5 µs after setting SCKDIVCR.

When changing any value from a higher division ratio to a lower division ratio, change as follows according to the changed ICLK frequency, before starting the subsequent processing.

- When the ICLK frequency after change is more than 120 MHz:
  At first, wait 1 µs after setting ICLK frequency division ratio to 1/2, then wait 1 µs after setting SCKDIVCR.
- When the ICLK frequency after change is 120 MHz or less:
  Wait 1 µs after setting SCKDIVCR.

The recommended method to measure the wait time is through software.

Be sure to consider the worst-case conditions to ensure that the required wait time elapses.

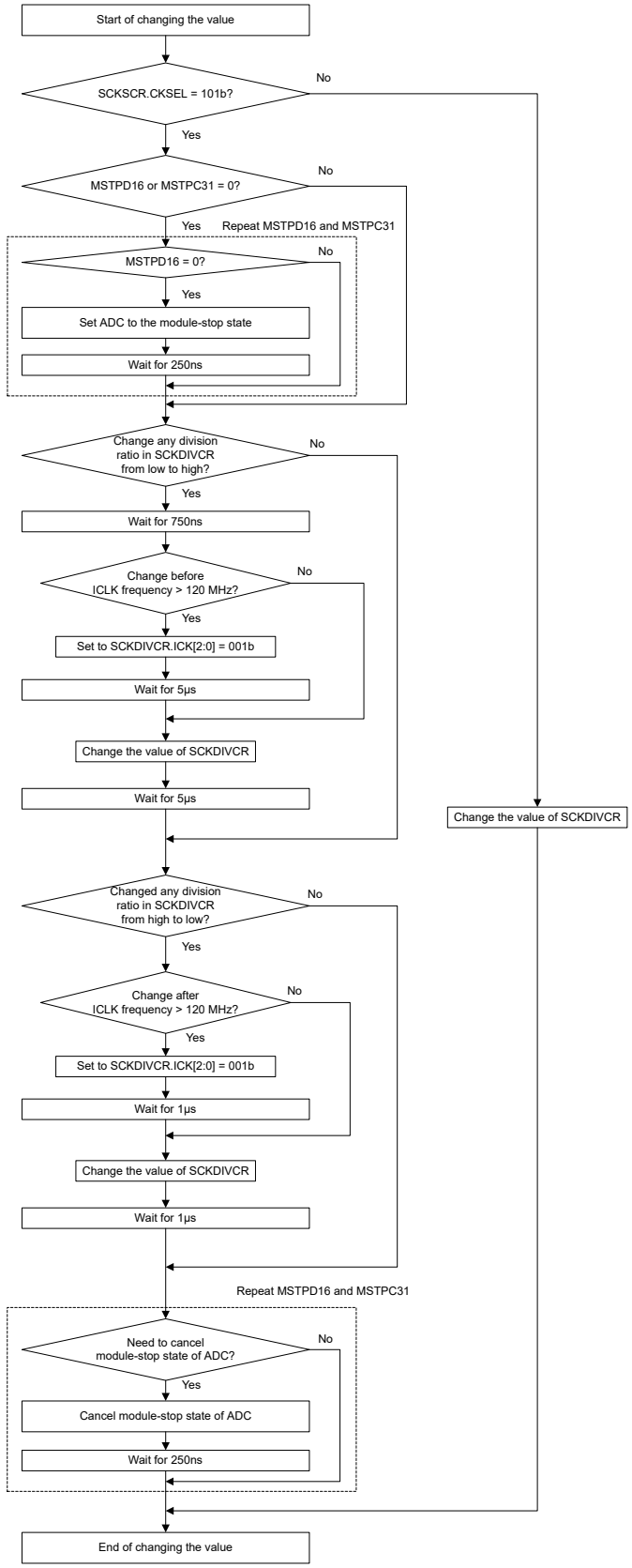


Figure 8.2 Example flow for changing the value of SCKDIVCR

8.2.3 SCKSCR : System Clock Source Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x026

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

Bit	Symbol	Function	R/W
2:0	CKSEL[2:0]	Clock Source Select 0 0 0: HOCO 0 0 1: MOCO 0 1 0: LOCO 0 1 1: Main clock oscillator (MOSC) 1 0 0: Setting prohibited 1 0 1: PLL 1 1 0: Setting prohibited 1 1 1: Setting prohibited	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

The SCKSCR register selects the clock source for the system clock.

When changing the value of SCKSCR to either select or diselect the PLL, set the following modules to the module-stop state before changing the SCKSCR value: ADC, SCE5.

In addition, when changing the value of SCKSCR from the PLL to a different clock source, wait at least 750 ns before the changing the value and change as follows according to the ICLK frequency before the change.

- When the ICLK frequency before change is more than 120 MHz:
  At first, wait 5 µs after setting ICLK frequency division ratio to 1/2, then wait 5 µs after setting SCKSCR.
- When the ICLK frequency before change is 120 MHz or less:
  Wait 5 µs after setting SCKSCR.

When changing the value from a non-PLL clock source to the PLL, change as follows according to the changed ICLK frequency, before starting the subsequent processing.

- When the ICLK frequency after change is more than 120 MHz:
  At first, wait 1 µs after setting ICLK frequency division ratio to 1/2, then wait 1 µs after setting SCKSCR.
- When the ICLK frequency after change is 120 MHz or less:
  Wait 1 µs after setting SCKSCR.

The recommended method to measure the wait time is through software. Be sure to consider the worst-case conditions to ensure that the required wait time elapses.

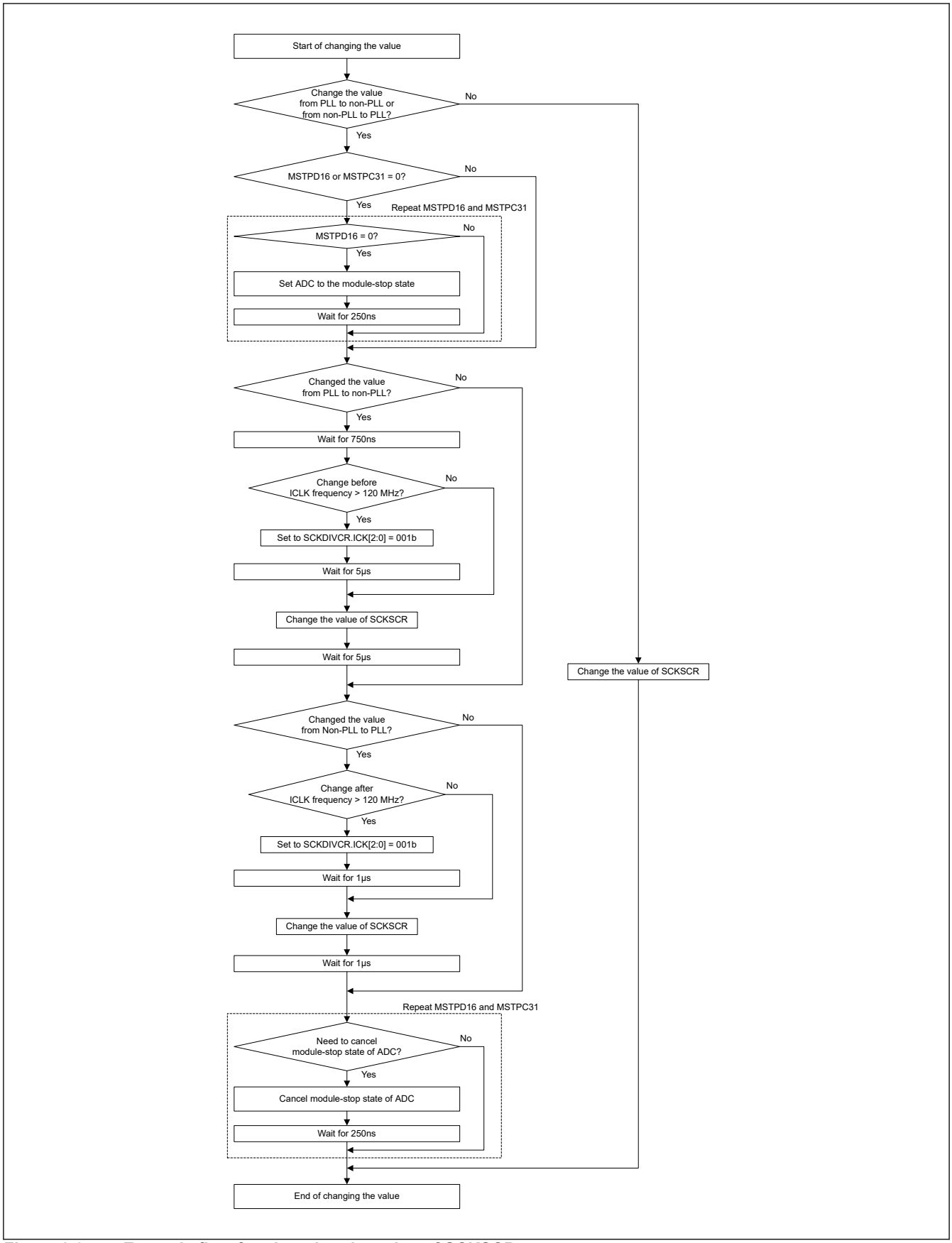


Figure 8.3 Example flow for changing the value of SCKSCR

CKSEL[2:0] bits (Clock Source Select)

The CKSEL[2:0] bits select the source for the following modules:

- System clock (ICLK)
- Peripheral module clocks (PCLKA, PCLKB, PCLKC, and PCLKD)
- FlashIF clock (FCLK)

The bits select from one of the following sources:

- Low-speed on-chip oscillator (LOCO)
- Middle-speed on-chip oscillator (MOCO)
- High-speed on-chip oscillator (HOCO)
- Main clock oscillator (MOSC)
- PLL

The operating state of each clock source is controlled not only by the clock oscillation enable settings but also by the operating modes of the product. Some clock sources might be forcibly stopped depending on the product operating mode being used.

Check the operation state of clock sources in each product operating mode, and do not select the clock source to be stopped in SCKSCR. The clock sources should be switched when there are no occurring internal asynchronous interrupt. For details, see section 10, Low Power Modes.

8.2.4 PLLCCR : PLL Clock Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x028

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

Bit	Symbol	Function	R/W
1:0	PLIDIV[1:0]*1	PLL Input Frequency Division Ratio Select 0 0: /1 0 1: /2 1 0: /3 Others: Setting prohibited.	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	PLSRCSEL	PLL Clock Source Select 0: Main clock oscillator 1: HOCO	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
13:8	PLLMUL[5:0]*2	PLL Frequency Multiplication Factor Select 0x13: × 10.0 (value after reset) 0x14: × 10.5 0x15: × 11.0 ⋮ 0x1C: × 14.5 0x1D: × 15.0 0x1E: × 15.5 ⋮ 0x3A: × 29.5 0x3B: × 30.0 Others: Setting prohibited.	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note 1. PLIDIV[1:0] should be set so that the frequency of PLL input signal is within the range of section 8.1. Overview.
Note 2. PLLMUL[5:0] should be set so that the frequency of PLL output signal is within the range of section 8.1. Overview.

The PLLCCR register sets the operation of the PLL circuit.

Writing to the PLLCCR is prohibited when the PLLCR.PLLSTP bit is 0 (the PLL operates).

PLIDIV[1:0] bits (PLL Input Frequency Division Ratio Select)

These bits select the frequency division ratio of the PLL clock source.

PLSRCSEL bit (PLL Clock Source Select)

This bit selects the clock source for the PLL.

PLLMUL[5:0] bits (PLL Frequency Multiplication Factor Select)

These bits select the frequency multiplication factor of the PLL circuit.

8.2.5 PLLCR : PLL Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x02A

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

Bit	Symbol	Function	R/W
0	PLLSTP	PLL Stop Control 0: PLL is operating 1: PLL is stopped.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

The PLLCR register controls the operation of the PLL circuit.

PLLSTP bit (PLL Stop Control)

This bit runs or stops the PLL circuit.

If the main clock oscillator is to be selected as the clock source for the PLL by the PLLCCR.PLSRCSEL bit, the Main Clock Oscillator Wait Control Register (MOSCWTCR) must be set.

After the PLLSTP bit setting is changed to run the PLL, only use the PLL clock after confirming that the OSCSF.PLLSF bit is set to 1. That is, a fixed time for stabilization is required after starting the PLL operation. A fixed time is also required for oscillation to stop after stopping the PLL operation. Additionally, apply the following limitations when starting and stopping the PLL operation by the PLLSTP bit:

- After stopping the PLL, confirm that the OSCSF.PLLSF bit is 0 before restarting the PLL.
- Confirm that the PLL is operating and that the OSCSF.PLLSF bit is 1 before stopping the PLL.
- Regardless of whether the PLL clock is selected as the system clock, confirm that the OSCSF.PLLSF is set to 1 before executing a WFI instruction to place the MCU in Software Standby or Deep Software Standby mode after operating the PLL.

- When transitioning to Software Standby or Deep Software Standby mode after stopping the PLL, confirm that the OSCSF.PLLSF bit is cleared to 0 before executing a WFI instruction.

Writing 1 to the PLLSTP bit is prohibited when SCKSCR.CKSEL[2:0] = 101 (system clock source = PLL).

Confirm the following conditions before writing 0 to PLLSTP:

- When PLL source clock = MOSC: MOSCCR.MOSTP = 0 (MOSC is enabled)
- When PLL source clock = HOCO: HOCOCR.HCSTP = 0 (HOCO is enabled).

8.2.6 PLL2CCR : PLL2 Clock Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x048

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	PLL2MUL[5:0]						—	—	—	PL2SRCSEL	—	—	PL2IDIV[1:0]	
Value after reset:	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	PL2IDIV[1:0]*1	PLL2 Input Frequency Division Ratio Select 0 0: /1 (value after reset) 0 1: /2 1 0: /3 Others: Setting prohibited.	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	PL2SRCSEL	PLL2 Clock Source Select 0: Main clock oscillator 1: HOCO	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
13:8	PLL2MUL[5:0]*2	PLL2 Frequency Multiplication Factor Select 0x13: × 10.0 (value after reset) 0x14: × 10.5 0x15: × 11.0 ⋮ 0x1C: × 14.5 0x1D: × 15.0 0x1E: × 15.5 ⋮ 0x3A: × 29.5 0x3B: × 30.0 Others: Setting prohibited.	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

Note 1. PL2IDIV[1:0] should be set so that the frequency of PLL2 input signal is within the range of section 8.1. Overview.

Note 2. PLL2MUL[5:0] should be set so that the frequency of PLL2 output signal is within the range of section 8.1. Overview.

The PLL2CCR register sets the operation of the PLL2 circuit.

Writing to the PLL2CCR register is prohibited when the PLL2CR.PLL2STP bit is 0 (the PLL2 operates).

PL2IDIV[1:0] bits (PLL2 Input Frequency Division Ratio Select)

These bits select the frequency division ratio of the PLL2 clock source.

PL2SRCSEL bit (PLL2 Clock Source Select)

This bit selects the clock source for the PLL2.

PLL2MUL[5:0] bits (PLL2 Frequency Multiplication Factor Select)

These bits select the frequency multiplication factor of the PLL2 circuit.

8.2.7 PLL2CR : PLL2 Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x04A

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	PLL2STP
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	PLL2STP	PLL2 Stop Control 0: PLL2 is operating 1: PLL2 is stopped.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

The PLL2CR register controls the operation of the PLL2 circuit.

PLL2STP bit (PLL2 Stop Control)

This bit runs or stops the PLL2 circuit.

If the main clock oscillator is to be selected as the clock source for the PLL2 by the PLL2CCR.PL2SRCSEL bit, the Main Clock Oscillator Wait Control Register (MOSCWTCR) must be set.

After the PLL2STP bit setting is changed to run the PLL2, only use the PLL2 clock after confirming that the OSCSF.PLL2SF bit is set to 1. That is, a fixed time for stabilization is required after starting the PLL2 operation. A fixed time is also required for oscillation to stop after stopping the PLL2 operation. Additionally, apply the following limitations when starting and stopping the PLL2 operation by the PLL2STP bit:

- After stopping the PLL2, confirm that the OSCSF.PLL2SF bit is 0 before restarting the PLL2.
- Confirm that the PLL2 is operating and that the OSCSF.PLL2SF bit is 1 before stopping the PLL2.
- Confirm that the OSCSF.PLL2SF bit is set to 1 before executing a WFI instruction to place the MCU in Software Standby or Deep Software Standby mode after operating the PLL2.
- When transitioning to Software Standby or Deep Software Standby mode after stopping the PLL2, confirm that the OSCSF.PLL2SF bit is cleared to 0 before executing a WFI instruction.

Confirm the following conditions before writing 0 to PLL2STP:

- When the PLL2 source clock = MOSC: MOSCCR.MOSTP = 0 (MOSC is enabled)
- When the PLL2 source clock = HOCO: HOCOCR.HCSTP = 0 (HOCO is enabled).

8.2.8 MOSCCR : Main Clock Oscillator Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x032

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

Bit	Symbol	Function	R/W
0	MOSTP	Main Clock Oscillator Stop 0: Operate the main clock oscillator*1 1: Stop the main clock oscillator	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note 1. MOMCR register must be set before setting MOSTP to 0.

The MOSCCR register controls the main clock oscillator.

MOSTP bit (Main Clock Oscillator Stop)

The MOSTP bit starts or stops the main clock oscillator.

When changing the value of the MOSTP bit, execute subsequent instructions only after reading the bit to check that the value is updated.

When using the main clock, the Main Clock Oscillator Mode Oscillation Control Register (MOMCR) and the Main Clock Oscillator Wait Control Register (MOSCWTCR) must be set before setting MOSTP to 0. After setting the MOSTP bit to 0, confirm that the OSCSF.MOSCSF bit is set to 1 before using the main clock oscillator.

A fixed stabilization wait time is required after setting the main clock oscillator to start operation. A fixed wait time is also required for oscillation to stop after stopping the main clock oscillator.

The following restrictions apply when starting and stopping operation:

- After stopping the main clock oscillator, confirm that the OSCSF.MOSCSF bit is 0 before restarting the main clock oscillator
- Confirm that the main clock oscillator operates and that the OSCSF.MOSCSF bit is 1 before stopping the main clock oscillator
- Regardless of whether the main clock oscillator is selected as the system clock, confirm that the OSCSF.MOSCSF bit is set to 1 before executing a WFI instruction to place the MCU in Software Standby after operating the main clock oscillator or Deep Software Standby mode.
- When a transition to Software Standby or Deep Software Standby mode is to follow the setting to stop the main clock oscillator, confirm that the OSCSF.MOSCSF bit is set to 0 before executing the WFI instruction.

Writing 1 to MOSTP is prohibited under the following condition:

- SCKSCR.CKSEL[2:0] = 011b (system clock source = MOSC).
- PLLCCR.PLSRCSEL = 0 (PLL source clock = MOSC) and SCKSCR.CKSEL[2:0] = 101b (system clock source = PLL)
- PLLCCR.PLSRCSEL = 0 (PLL source clock = MOSC) and PLLCR.PLLSTP = 0 (PLL is operating)
- PLL2CCR.PL2SRCSEL = 0 (PLL2 source clock = MOSC) and PLL2CR.PLL2STP = 0 (PLL2 is operating)

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

Base address: SYSC = 0x4001_E000

Offset address: 0x490

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

Bit	Symbol	Function	R/W
0	LCSTP	LOCO Stop 0: Operate the LOCO clock 1: Stop the LOCO clock	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

The LOCOCR register controls the LOCO clock.

LCSTP bit (LOCO Stop)

The LCSTP bit starts or stops the LOCO clock.

After setting the LCSTP bit to 0 to start the LOCO clock, only use the clock after the LOCO clock-oscillation stabilization wait time (t_{LOCOWT}) elapses. A fixed stabilization wait time is required after setting the LOCO clock to start operation. A fixed wait time is also required after setting the LOCO clock to stop.

The following restrictions apply when starting and stopping operation:

- After stopping the LOCO clock, allow a stop interval of at least 5 LOCO clock cycles before restarting it
- Confirm that LOCO oscillation is stable before stopping the LOCO clock
- Regardless of whether the LOCO is selected as the system clock, confirm that LOCO oscillation is stable before executing a WFI instruction to place the MCU in Software Standby or Deep Software Standby mode
- When a transition to Software Standby or Deep Software Standby mode is to follow the setting to stop the LOCO clock, wait for at least 3 LOCO cycles before executing the WFI instruction.

Writing 1 to LCSTP is prohibited under the following condition:

- SCKSCR.CKSEL[2:0] = 010b (system clock source = LOCO).

Because the LOCO clock measures the wait time for other oscillators, it continues to oscillate while measuring this time, regardless of the setting in LOCOCR.LCSTP. As a result, the LOCO clock might be unintentionally supplied even when the LCSTP is set to stop.

8.2.10 HOCOCR : High-Speed On-Chip Oscillator Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x036

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

Bit	Symbol	Function	R/W
0	HCSTP	HOCO Stop 0: Operate the HOCO clock *2 1: Stop the HOCO clock	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note 1. The HCSTP bit value after a reset is 0 when the OFS1.HOCOEN bit is 0. It is 1 when the OFS1.HOCOEN bit is 1.
Note 2. If you are using the HOCO (HCSTP = 0), set the OFS1.HOCOFRQ0[1:0] bit to an optimum value.

The HOCOCR register controls the HOCO clock.

HCSTP bit (HOCO Stop)

The HCSTP bit starts or stops the HOCO clock.

After setting the HCSTP bit to 0 to start the HOCO clock, confirm that the OSCSF.HOCOSF is set to 1 before using the clock. When OFS1.HOCOEN is set to 0, confirm that OSCSF.HOCOSF is also set to 1 before using the HOCO clock. A fixed stabilization wait time is required after setting the HOCO clock to start operation. A fixed wait time is also required after setting the HOCO clock to stop.

The following limitations apply when starting and stopping operation:

- After stopping the HOCO clock, confirm that the OSCSF.HOCOSF is 0 before restarting the HOCO clock.
- Confirm that the HOCO clock operates and that the OSCSF.HOCOSF is 1 before stopping the HOCO clock.
- Regardless of whether the HOCO clock is selected as the system clock, confirm that the OSCSF.HOCOSF is set to 1 before executing a WFI instruction to place the MCU in Software Standby or Deep Software Standby mode after setting HOCO operation with the HCSTP bit.
- When a transition to Software Standby or Deep Software Standby mode is to follow the setting of the HOCO clock to stop, confirm that the OSCSF.HOCOSF is set to 0 after setting the HOCO clock and before executing the WFI instruction.

Writing 1 to HCSTP is prohibited under the following conditions:

- SCKSCR.CKSEL[2:0] = 000b (system clock source = HOCO).
- PLLCCR.PLSRCSEL = 1 (PLL source clock = HOCO) and SCKSCR.CKSEL[2:0] = 101b (system clock source = PLL)
- PLLCCR.PLSRCSEL = 1 (PLL source clock = HOCO) and PLLCR.PLLSTP = 0 (PLL is operating)
- PLL2CCR.PL2SRCSEL = 1 (PLL2 source clock = HOCO) and PLL2CR.PLL2STP = 0 (PLL2 is operating)

8.2.11 MOCOCR : Middle-Speed On-Chip Oscillator Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x038

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

Bit	Symbol	Function	R/W
0	MCSTP	MOCO Stop 0: MOCO clock is operating 1: MOCO clock is stopped	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

The MOCOCR register controls the MOCO clock.

MCSTP bit (MOCO Stop)

The MCSTP bit starts or stops the MOCO clock.

After setting MCSTP to 0, use the MOCO clock only after the MOCO clock oscillation stabilization time (t_{MOCOWT}) elapses. A fixed stabilization wait time is required after setting the MOCO clock to start operation. A fixed wait time is also required for oscillation to stop after setting the MOCO clock to stop operation.

The following restrictions apply when starting and stopping the oscillator:

- After stopping the MOCO clock, allow a stop interval of at least 5 MOCO clock cycles before restarting it
- Confirm that MOCO clock oscillation is stable before stopping the MOCO clock
- Regardless of whether the MOCO clock is selected as the system clock, confirm that MOCO clock oscillation is stable before executing a WFI instruction to place the MCU in Software Standby or Deep Software Standby mode
- When a transition to Software Standby or Deep Software Standby mode is to follow the setting to stop the MOCO clock, wait for at least 3 MOCO clock cycles before executing the WFI instruction.

Writing 1 to MCSTP is prohibited under the following condition:

- SCKSCR.CKSEL[2:0] = 001b (system clock source = MOCO).

Writing 1 to the MCSTP bit (stopping the MOCO) is prohibited if oscillation stop detection is enabled in the Oscillation Stop Detection Control Register (OSTDCR.OSTDE).

8.2.12 OSCSF : Oscillation Stabilization Flag Register

Base address: SYSC = 0x4001_E000

Offset address: 0x03C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	PLL2SF	PLLSF	—	MOSCSF	—	—	HOCOSF
Value after reset:	0	0	0	0	0	0	0	0/1*1

Bit	Symbol	Function	R/W
0	HOCOSF	HOCO Clock Oscillation Stabilization Flag 0: The HOCO clock is stopped or is not yet stable 1: The HOCO clock is stable, so is available for use as the system clock	R
2:1	—	These bits are read as 0.	R
3	MOSCSF	Main Clock Oscillation Stabilization Flag 0: The main clock oscillator is stopped (MOSTP = 1) or is not yet stable*2 1: The main clock oscillator is stable, so is available for use as the system clock	R
4	—	This bit is read as 0.	R
5	PLLSF	PLL Clock Oscillation Stabilization Flag 0: The PLL clock is stopped, or oscillation of the PLL clock is not stable yet 1: The PLL clock is stable, so is available for use as the system clock	R
6	PLL2SF	PLL2 Clock Oscillation Stabilization Flag 0: The PLL2 clock is stopped, or oscillation of the PLL2 clock is not stable yet 1: The PLL2 clock is stable	R
7	—	These bits are read as 0.	R

Note 1. The value after reset depends on the OFS1.HOCOEN setting.

When OFS1.HOCOEN = 1 (disable HOCO), the value after reset of HOCOSF is 0.
When OFS1.HOCOEN = 0 (enable HOCO), the HOCOSF value is set to 0 immediately after reset is released, and the HOCOSF value is set to 1 after the HOCO oscillation stabilization wait time elapses.

Note 2. This is true when an appropriate value is set in the Wait Control register for the main clock oscillator. If the wait time value is not sufficient, the oscillation stabilization flag is set to 1 and supply of the clock signal to the internal circuits starts before oscillation is stable.

This register is not controlled by CGFSAR register.

The OSCSF register contains flags to indicate the operating status of the counters in the oscillation stabilization wait circuits for the individual oscillators. After oscillation starts, these counters measure the wait time until each oscillator output clock is supplied to the internal circuits. An overflow of a counter indicates that the clock supply is stable and available for the associated circuit.

HOCOSF flag (HOCO Clock Oscillation Stabilization Flag)

The HOCOSF flag indicates the operating status of the counter that measures the wait time for the high-speed clock oscillator (HOCO). When OFS1.HOCOEN is set to 0, confirm that OSCSF.HOCOSF is set to 1 before using the HOCO clock.

[Setting condition]

- When the HOCO clock is stopped and the HOCOCR.HCSTP bit is set to 0, and then the HOCO oscillation stabilization time is counted by the LOCO clock and supply of the HOCO clock within the MCU is started. For the HOCO oscillation stabilization time, see section 46, Electrical Characteristics.

[Clearing condition]

- When the HOCO clock is operating and then is deactivated because the HOCOCR.HCSTP bit is set to 1.

MOSCSF flag (Main Clock Oscillation Stabilization Flag)

The MOSCSF flag indicates the operating status of the counter that measures the wait time for the main clock oscillator.

[Setting condition]

- When the main clock oscillator is stopped and the MOSCCR.MOSTP bit is set to 0, and then the number of LOCO clock cycles corresponding to the setting of the MOSCWTCR register is counted and supply of the main clock within the MCU is started.

[Clearing condition]

- When the main clock oscillator is operating and then is deactivated because the MOSCCR.MOSTP bit is set to 1.

PLLSF flag (PLL Clock Oscillation Stabilization Flag)

The PLLSF flag indicates the operating state of the counter that measures the wait time of the PLL.

[Setting condition]

- When the PLL is stopped and the PLLCR.PLLSTP bit is set to 0, and then the PLL oscillation stabilization time is counted by the LOCO clock and supply of the PLL clock within the MCU is started. If oscillation by the PLL clock source is not stable when the PLLCR.PLLSTP bit is set to 0, counting of the LOCO cycles continues even after the PLL clock source oscillation is stabilized. For the PLL oscillation stabilization time, see section 46, Electrical Characteristics.

[Clearing condition]

- When the PLL is operating and then is deactivated because the PLLCR.PLLSTP bit is set to 1.

PLL2SF flag (PLL2 Clock Oscillation Stabilization Flag)

The PLL2SF flag indicates the operating state of the counter that measures the wait time of the PLL2.

[Setting condition]

- When the PLL2 is stopped and the PLL2CR.PLL2STP bit is set to 0, and then the PLL2 oscillation stabilization time is counted by the LOCO clock and supply of the PLL2 clock within the MCU is started. If oscillation by the PLL2 clock source is not stable when the PLL2CR.PLL2STP bit is set to 0, counting of the LOCO cycles continues even after the PLL2 clock source oscillation is stabilized. For the PLL2 oscillation stabilization time, see section 46, Electrical Characteristics.

[Clearing condition]

- When the PLL2 is operating and then is deactivated because the PLL2CR.PLL2STP bit is set to 1.

8.2.13 OSTDCR : Oscillation Stop Detection Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x040

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

Bit	Symbol	Function	R/W
0	OSTDIE	Oscillation Stop Detection Interrupt Enable 0: Disable oscillation stop detection interrupt (do not notify the POEG) 1: Enable oscillation stop detection interrupt (notify the POEG)	R/W
6:1	—	These bits are read as 0. The write value should be 0.	R/W
7	OSTDE	Oscillation Stop Detection Function Enable 0: Disable oscillation stop detection function 1: Enable oscillation stop detection function	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

The OSTDCR register controls the oscillation stop detection function.

OSTDIE bit (Oscillation Stop Detection Interrupt Enable)

The OSTDIE bit enables the oscillation stop detection function interrupt. It also controls whether oscillation stop detection is reported to the POEG.

If the Oscillation Stop Detection flag in the Oscillation Stop Detection Status Register (OSTDSR.OSTDF) requires clearing, set the OSTDIE bit to 0 before clearing OSTDF. Wait for at least 2 PCLKB cycles before setting the OSTDIE bit to 1. By reading the I/O register whose access cycle number is defined by PCLKB, it is possible to secure waiting time of 2 or more cycles of PCLKB.

OSTDE bit (Oscillation Stop Detection Function Enable)

The OSTDE bit enables the oscillation stop detection function.

When the OSTDE bit is 1 (enabled), the MOCO stop bit (MOCOCR.MCSTP) is set to 0 and the MOCO operation starts. The MOCO clock cannot be stopped while the oscillation stop detection function is enabled. Writing 1 to the MOCOCR.MCSTP bit (MOCO stopped) is invalid.

When the Oscillation Stop Detection flag in the Oscillation Stop Detection Status Register (OSTDSR.OSTDF) is 1 (main clock oscillation stop detected), writing 0 to the OSTDE bit is invalid.

The OSTDE bit must be set to 0 before transitioning to Software Standby or Deep Software Standby mode. To transition to Software Standby or Deep Software Standby mode, first set the OSTDE bit to 0, then execute the WFI instruction.

The following restrictions apply when using the oscillation stop detection function:
In low-speed mode, selecting division by 1, 2, 4, 8 for ICLK, FCLK, PCLKA, PCLKB, PCLKC, and PCLKD is prohibited.

8.2.14 OSTDSR : Oscillation Stop Detection Status Register

Base address: SYSC = 0x4001_E000

Offset address: 0x041

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

Bit	Symbol	Function	R/W
0	OSTDF	Oscillation Stop Detection Flag 0: Main clock oscillation stop not detected 1: Main clock oscillation stop detected	R/W*1
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note 1. This bit can only be set to 0. This bit is cleared to 0 by writing 0 after reading it as 1.

The OSTDSR register indicates the stop detection status of the main clock oscillator.

OSTDF flag (Oscillation Stop Detection Flag)

The OSTDF flag indicates the main clock oscillator status. When this flag is 1, it indicates that the main clock oscillation stop was detected. After this stop is detected, the OSTDF flag is not set to 0 even when the main clock oscillation is restarted. The OSTDF bit is cleared to 0 by writing 0 after reading it as 1.

At least 3 ICLK cycles of wait time are required between writing 0 to OSTDF and reading it as 0. If the OSTDF flag is set to 0 when the main clock oscillation is stopped, the OSTDF flag becomes 0 then returns to 1.

The OSTDF flag cannot be set to 0 under the following conditions:

- SCKSCR.CKSEL[2:0] = 011b (system clock source = MOSC).
- PLLCCR.PLSRCSEL = 0 (PLL source clock = MOSC) and SCKSCR.CKSEL[2:0] = 101b (System clock source = PLL)

The OSTDF flag must be set to 0 after switching the clock source to sources other than the main clock oscillator and PLL.

[Setting condition]

- The main clock oscillator is stopped when OSTDCR.OSTDE = 1 (oscillation stop detection function enabled).

[Clearing condition]

- 1 is read and then 0 is written when the SCKSCR.CKSEL[2:0] bits are neither 011b (system clock is MOSC) nor 101b (system clock is PLL) and PLLCCR.PLSRCSEL bit is not 0 (PLL source clock is MOSC).

8.2.15 MOSCWTCR : Main Clock Oscillator Wait Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x0A2

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

Bit	Symbol	Function	R/W
3:0	MSTS[3:0]	Main Clock Oscillator Wait Time Setting 0x0: Wait time = 3 cycles (11.4 µs) 0x1: Wait time = 35 cycles (133.5 µs) 0x2: Wait time = 67 cycles (255.6 µs) 0x3: Wait time = 131 cycles (499.7 µs) 0x4: Wait time = 259 cycles (988.0 µs) 0x5: Wait time = 547 cycles (2086.6 µs) 0x6: Wait time = 1059 cycles (4039.8 µs) 0x7: Wait time = 2147 cycles (8190.2 µs) 0x8: Wait time = 4291 cycles (16368.9 µs) 0x9: Wait time = 8163 cycles (31139.4 µs) Others: Setting prohibited	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

MSTS[3:0] bits (Main Clock Oscillator Wait Time Setting)

The MSTS[3:0] bits specify the oscillation stabilization wait time for the main clock oscillator.

Set the main clock oscillation stabilization time to a period longer than or equal to the stabilization time recommended by the oscillator manufacturer. When the main clock is input externally, set these bits to 0x0 because the oscillation stabilization time is not required.

The wait time set in these bits is counted using: 1 cycle (µs) = $1/(f_{LOCO}[MHz] \times 8) = 1/(0.032768 \times 8) = 3.81$ (µs) (min.)
The LOCO clock automatically oscillates when necessary, regardless of the value of the LOCO.LCSTP bit. After the specified wait time elapses, supply of the main clock starts internally in the MCU, and the OSCSF.MOSCSF flag is set to 1. If the specified wait time is short, supply of the main clock starts before oscillation of the clock becomes stable.

Only rewrite the MOSCWTCR register when the MOSCCR.MOSTP bit is 1 and the OSCSF.MOSCSF flag is 0. Do not rewrite this register under any other conditions.

8.2.16 MOMCR : Main Clock Oscillator Mode Oscillation Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x413

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	MOSEL	MODRV[1:0]		—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
5:4	MODRV[1:0]	Main Clock Oscillator Drive Capability 0 Switching 0 0: 20 MHz to 24 MHz 0 1: 16 MHz to 20 MHz 1 0: 8 MHz to 16 MHz 1 1: 8 MHz	R/W
6	MOSEL	Main Clock Oscillator Switching 0: Resonator 1: External clock input	R/W
7	—	These bits are read as 0. The write value should be 0.	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:

- Secure and Non-secure access are allowed.

Note: The EXTAL/XTAL pins are also used as ports. In the initial state, the pin is set as a port.
Note: The MOSCCR.MOSTP bit must be 1 (MOSC is stopped) before changing this register.
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

MODRV[1:0] bit (Main Clock Oscillator Drive Capability 0 Switching)

The MODRV[1:0] bit switches the drive capability of the main clock oscillator.

MOSEL bit (Main Clock Oscillator Switching)

The MOSEL bit switches the source for the main clock oscillator.

8.2.17 CKOCR : Clock Out Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x03E

Bit position:	7	6	5	4	3	2	1	0
Bit field:	CKOEN	CKODIV[2:0]			—	CKOSEL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	CKOSEL[2:0]	Clock Out Source Select 0 0 0: HOCO (value after reset) 0 0 1: MOCO 0 1 0: LOCO 0 1 1: MOSC 1 0 0: Setting prohibited 1 0 1: Setting prohibited Others: Setting prohibited	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
6:4	CKODIV[2:0]	Clock Output Frequency Division Ratio 0 0 0: × 1/1 0 0 1: × 1/2 0 1 0: × 1/4 0 1 1: × 1/8 1 0 0: × 1/16 1 0 1: × 1/32 1 1 0: × 1/64 1 1 1: × 1/128	R/W
7	CKOEN	Clock Out Enable 0: Disable clock out 1: Enable clock out	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note: If the security attribution is configured as secure:

- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:

- Secure and Non-secure access are allowed.

CKOSEL[2:0] bits (Clock Out Source Select)

The CKOSEL[2:0] bits select the source of the clock to be output from the CLKOUT pin. When changing the clock source, set the CKOEN bit to 0.

CKODIV[2:0] bits (Clock Output Frequency Division Ratio)

The CKODIV[2:0] bits specify the clock division ratio. Set the CKOEN bit to 0 when changing the division ratio.

CKOEN bit (Clock Out Enable)

The CKOEN bit enables output from the CLKOUT pin.

When this bit is set to 1, the selected clock is output. When this bit is set to 0, low is output. When changing this bit, confirm that the clock out source clock selected in the CKOSEL[2:0] bits is stable. Otherwise, a glitch might be generated in the output.

Clear this bit before entering Software Standby or Deep Software Standby mode if the selecting clock out source clock is stopped in that mode.

8.2.18 LOCOUTCR : LOCO User Trimming Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x492

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

Bit	Symbol	Function	R/W
7:0	LOCOUTRM[7:0]	LOCO User Trimming 0x80: -128 0x81: -127 ⋮ 0xFF: -1 0x00: Center Code 0x01: +1 ⋮ 0x7E: +126 0x7F: +127	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

The LOCOUTCR register is added to the original LOCO trimming data.

MCU operation is not guaranteed when LOCOUTCR is set to a value that causes the LOCO frequency to be outside of the specification range. When LOCOUTCR is modified, the frequency stabilization time corresponds to the frequency stabilization time at the start of MCU operation. When the ratio of the LOCO frequency and the other oscillation frequency is an integer value, changing the LOCOUTCR value is prohibited.

8.2.19 MOCOUTCR : MOCO User Trimming Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x061

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

Bit	Symbol	Function	R/W
7:0	MOCOUTRM[7:0]	MOCO User Trimming 0x80: -128 0x81: -127 ⋮ 0xFF: -1 0x00: Center Code 0x01: +1 ⋮ 0x7E: +126 0x7F: +127	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

The MOCOUTCR register is added to the original MOCO trimming data.

MCU operation is not guaranteed when MOCOUTCR is set to a value that causes the MOCO frequency to be outside of the specification range. When MOCOUTCR is modified, the frequency stabilization wait time corresponds to the frequency stabilization wait time at the start of the MCU operation. When the ratio of the MOCO frequency and the other oscillation frequency is an integer value, changing the MOCOUTCR value is prohibited.

8.2.20 HOCOUTCR : HOCO User Trimming Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x062

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

Bit	Symbol	Function	R/W
7:0	HOCOUTRM[7:0]	HOCO User Trimming 0x80: -128 0x81: -127 ⋮ 0xFF: -1 0x00: Center Code 0x01: +1 ⋮ 0x7E: +126 0x7F: +127	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

The HOCOUTCR register is added to the original HOCO trimming data.

MCU operation is not guaranteed when HOCOUTCR is set to a value that causes the HOCO frequency to be outside of the specification range. When HOCOUTCR is modified, the frequency stabilization wait time corresponds to the frequency stabilization wait time at the start of the MCU operation.

8.2.21 SCISPICKDIVCR : SCI SPI Clock Division Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x06D

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

Bit	Symbol	Function	R/W
2:0	SCISPICKDIV[2:0]	SCI SPI Clock (SCISPICLK) Division Select 0 0 0: ⁄1 (value after reset) 0 0 1: ⁄2 0 1 0: ⁄4 0 1 1: ⁄6 1 0 0: ⁄8 Others: Setting prohibited.	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

SCISPICKDIVCR controls the SCI SPI clock (SCISPICLK).

SCISPICKDIV[2:0] bits (SCI SPI Clock (SCISPICLK) Division Select)

These bits select the frequency of the SCI SPI clock (SCISPICLK) and must be modified when SCISPICKCR.SCISPICKSRDY = 1.

8.2.22 CANFDCKDIVCR : CANFD Clock Division Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x06E

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

Bit	Symbol	Function	R/W
2:0	CANFDCKDIV[2:0]	CANFD clock (CANFDCLK) Division Select 0 0 0: /1 (value after reset) 0 0 1: /2 0 1 0: /4 0 1 1: /6 1 0 0: /8 Others: Setting prohibited	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

CANFDCKDIVCR controls the CANFD clock (CANFDCLK).

CANFDCKDIV[2:0] bit (CANFD clock (CANFDCLK) Division Select)

These bits select the frequency of the CANFD clock (CANFDCLK).

These bits must change when CANFDCKCR.CANFDCKSRDY = 1.

8.2.23 GPTCKDIVCR : GPT Clock Division Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x06F

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

Bit	Symbol	Function	R/W
2:0	GPTCKDIV[2:0]	GPT clock (GPTCLK) Division Select 0 0 0: /1 (value after reset) 0 0 1: /2 0 1 0: /4 0 1 1: /6 1 0 0: /8 Others: Setting prohibited	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

GPTCKDIVCR controls the GPT clock (GPTCLK).

GPTCKDIV[2:0] bit (GPT clock (GPTCLK) Division Select)

These bits select the frequency of the GPT clock (GPTCLK).

These bits must change when GPTCKCR.GPTCKCRSRDY = 1.

8.2.24 IICCKDIVCR : IIC Clock Division Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x070

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

Bit	Symbol	Function	R/W
2:0	IICCKDIV[2:0]	IIC clock (IICCLK) Division Select 0 0 0: /1 (value after reset) 0 0 1: /2 0 1 0: /4 0 1 1: /6 1 0 0: /8 Others: Setting prohibited	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

IICCKDIVCR controls the IIC clock (IICCLK).

IICCKDIV[2:0] bit (IIC clock (IICCLK) Division Select)

These bits select the frequency of the IIC clock (IICCLK).

These bits must change when IICCKCR.IICCKSRDY = 1.

8.2.25 SCISPICKCR : SCI SPI Clock Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x075

Bit position:	7	6	5	4	3	2	1	0
Bit field:	SCISPICKSRDY	SCISPICKSREQ	—	—	—	SCISPICKSEL[2:0]		
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
2:0	SCISPICKSEL[2:0]	SCI SPI Clock (SCISPICLK) Source Select 0 0 0: HOCO 0 0 1: MOCO (value after reset) 0 1 0: LOCO 0 1 1: Main clock oscillator 1 0 1: PLL 1 1 0: PLL2 Others: Setting prohibited.	R/W
5:3	—	These bits are read as 0. The write value should be 0.	R/W
6	SCISPICKSREQ	SCI SPI Clock (SCISPICLK) Switching Request 0: No request 1: Request switching.	R/W
7	SCISPICKSRDY	SCI SPI Clock (SCISPICLK) Switching Ready state flag 0: Switching not possible 1: Switching possible.	R

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

The SCISPICKCR register controls the SCI SPI clock (SCISPICLK).

When switching the clock source, ensure that the clock before the switch and the clock after the switch generate stable output.

To change the set value of SCISPICKDIVCR.SCISPICKDIV[2:0] and SCISPICKSEL[2:0], use the following procedure:

1. Write 1 to SCISPICKSREQ.
2. Poll until SCISPICKSRDY is read as 1. While SCISPICKSRDY = 1, no clock is output to SCISPICLK.
3. Write to SCISPICKDIVCR.SCISPICKDIV[2:0] and SCISPICKSEL[2:0].
4. Write 0 to SCISPICKSREQ.
5. Poll until SCISPICKSRDY is read as 0.
6. When SCISPICKSRDY becomes 0, SCISPICLK starts to output. Clock switching is complete.

When transitioning to Software Standby or Deep Software Standby mode, do not execute the WFI instruction while performing clock switching. That is, do not execute the WFI instruction when SCISPICKSREQ = 1 and SCISPICKSRDY = 0, or when SCISPICKSREQ = 0 and SCISPICKSRDY = 1.

SCISPICKSEL[2:0] bits (SCI SPI Clock (SCISPICLK) Source Select)

These bits select the clock source of the SCI SPI clock (SCISPICLK) and must be modified when SCISPICKCR.SCISPICKSRDY = 1.

SCISPICKSREQ bit (SCI SPI Clock (SCISPICLK) Switching Request)

This bit selects the SCISPICLK switching request.

SCISPICKSRDY flag (SCI SPI Clock (SCISPICLK) Switching Ready state flag)

This flag indicates the state of switching ready for the SCISPICLK. When SCISPICKSRDY = 1, no clock is output to SCISPICLK.

8.2.26 CANFDCKCR : CANFD Clock Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x076

Bit position:	7	6	5	4	3	2	1	0
Bit field:	CANFDCKSRDY	CANFDCKSREQ	—	—	—	CANFDCKSEL[2:0]		
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
2:0	CANFDCKSEL[2:0]	CANFD clock (CANFDCLK) Source Select 1 0 1: PLL 1 1 0: PLL2 Others: Setting prohibited	R/W
5:3	—	These bits are read as 0. The write value should be 0.	R/W
6	CANFDCKSREQ	CANFD clock (CANFDCLK) Switching Request 0: No request 1: Request switching	R/W
7	CANFDCKSRDY	CANFD clock (CANFDCLK) Switching Ready state flag 0: Impossible to Switch 1: Possible to Switch	R

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

The CANFDCKCR register controls the CANFD clock (CANFDCLK).

When switching the clock source, ensure that the clock before the switch and the clock after the switch generate stable output. To change the set value of CANFDCKDIVCR.CANFDCKDIV[2:0] and CANFDCKSEL[2:0], use the following procedure:

1. Write 1 to CANFDCKSREQ.
2. Poll until CANFDCKSRDY is read as 1. While CANFDCKSRDY = 1, no clock is output to CANFDCLK.
3. Write to CANFDCKDIVCR.CANFDCKDIV[2:0] and CANFDCKSEL[2:0].
4. Write 0 to CANFDCKSREQ.
5. Poll until CANFDCKSRDY is read as 0.
6. When CANFDCKSRDY becomes 0, CANFDCLK starts to output. Clock switching is complete.

When transitioning to Software Standby or Deep Software Standby mode, do not execute the WFI instruction while performing clock switching. That is, do not execute the WFI instruction when CANFDCKSREQ = 1 and CANFDCKSRDY = 0, or when CANFDCKSREQ = 0 and CANFDCKSRDY = 1.

CANFDCKSEL[2:0] bits (CANFD clock (CANFDCLK) Source Select)

These bits select the clock source of the CANFD clock (CANFDCLK) and must be modified when CANFDCKCR.CANFDCKSRDY = 1.

CANFDCKSREQ bit (CANFD clock (CANFDCLK) Switching Request)

This bit selects the CANFDCLK switching request.

CANFDCKSRDY flag (CANFD clock (CANFDCLK) Switching Ready state flag)

This flag indicates the state of switching ready for the CANFDCLK. When CANFDCKSRDY = 1, no clock is output to CANFDCLK.

8.2.27 GPTCKCR : GPT Clock Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x077

Bit position:	7	6	5	4	3	2	1	0
Bit field:	GPTCKSRDY	GPTCKSREQ	—	—	—	GPTCKSEL[2:0]		
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
2:0	GPTCKSEL[2:0]	GPT clock (GPTCLK) Source Select 0 0 0: HOCO 0 0 1: MOCO (value after reset) 0 1 0: LOCO 0 1 1: Main clock oscillator 1 0 1: PLL 1 1 0: PLL2 Others: Setting prohibited	R/W
5:3	—	These bits are read as 0. The write value should be 0.	R/W
6	GPTCKSREQ	GPT clock (GPTCLK) Switching Request 0: No request 1: Request switching	R/W
7	GPTCKSRDY	GPT clock (GPTCLK) Switching Ready state flag 0: Impossible to Switch 1: Possible to Switch	R

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

The GPTCKCR register controls the GPT clock (GPTCLK).

When switching the clock source, ensure that the clock before the switch and the clock after the switch generate stable output. To change the set value of GPTCKDIVCR.GPTCKDIV[2:0] and GPTCKSEL[2:0], use the following procedure:

1. Write 1 to GPTCKSREQ.
2. Poll until GPTCKSRDY is read as 1. While GPTCKSRDY = 1, no clock is output to GPTCLK.
3. Write to GPTCKDIVCR.GPTCKDIV[2:0] and GPTCKSEL[2:0].
4. Write 0 to GPTCKSREQ.
5. Poll until GPTCKSRDY is read as 0.
6. When GPTCKSRDY becomes 0, GPTCLK starts to output. Clock switching is complete.

When transitioning to Software Standby or Deep Software Standby mode, do not execute the WFI instruction while performing clock switching. That is, do not execute the WFI instruction when GPTCKSREQ = 1 and GPTCKSRDY = 0, or when GPTCKSREQ = 0 and GPTCKSRDY = 1.

GPTCKSEL[2:0] bits (GPT clock (GPTCLK) Source Select)

These bits select the clock source of the GPT clock (GPTCLK) and must be modified when GPTCKCR.GPTCKSRDY = 1.

GPTCKSREQ bit (GPT clock (GPTCLK) Switching Request)

This bit selects the GPTCLK switching request.

GPTCKSRDY flag (GPT clock (GPTCLK) Switching Ready state flag)

This flag indicates the state of switching ready for the GPTCLK. When GPTCKSRDY = 1, no clock is output to GPTCLK.

8.2.28 IICCKCR : IIC Clock Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x078

Bit position:	7	6	5	4	3	2	1	0
Bit field:	IICCK SRDY	IICCK SREQ	—	—	—	IICCKSEL[2:0]		
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
2:0	IICCKSEL[2:0]	IIC clock (IICCLK) Source Select 0 0 0: HOCO 0 0 1: MOCO (value after reset) 0 1 0: LOCO 0 1 1: Main clock oscillator 1 0 1: PLL 1 1 0: PLL2 Others: Setting prohibited	R/W
5:3	—	These bits are read as 0. The write value should be 0.	R/W
6	IICCKSREQ	IIC clock (IICCLK) Switching Request 0: No request 1: Request switching	R/W

Bit	Symbol	Function	R/W
7	IICCKSRDY	IIC clock (IICCLK) Switching Ready state flag 0: Impossible to Switch 1: Possible to Switch	R

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

The IICCKCR register controls the IIC clock (IICCLK).

When switching the clock source, ensure that the clock before the switch and the clock after the switch generate stable output. To change the set value of IICCKDIVCR.IICCKDIV[2:0] and IICCKSEL[2:0], use the following procedure:

1. Write 1 to IICCKSREQ.
2. Poll until IICCKSRDY is read as 1. While IICCKSRDY = 1, no clock is output to IICCLK.
3. Write to IICCKDIVCR.IICCKDIV[2:0] and IICCKSEL[2:0].
4. Write 0 to IICCKSREQ.
5. Poll until IICCKSRDY is read as 0.
6. When IICCKSRDY becomes 0, IICCLK starts to output. Clock switching is complete.

When transitioning to Software Standby or Deep Software Standby mode, do not execute the WFI instruction while performing clock switching. That is, do not execute the WFI instruction when IICCKSREQ = 1 and IICCKSRDY = 0, or when IICCKSREQ = 0 and IICCKSRDY = 1.

IICCKSEL[2:0] bits (IIC clock (IICCLK) Source Select)

These bits select the clock source of the IIC clock (IICCLK) and must be modified when IICCKCR.IICCKSRDY = 1.

IICCKSREQ bit (IIC clock (IICCLK) Switching Request)

This bit selects the IICCLK switching request.

IICCKSRDY flag (IIC clock (IICCLK) Switching Ready state flag)

This flag indicates the state of switching ready for the IICCLK. When IICCKSRDY = 1, no clock is output to IICCLK.

8.2.29 TRCKCR : Trace Clock Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x03F

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TRCKEN	—	—	—	TRCK[3:0]			
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
3:0	TRCK[3:0]	Trace Clock operating frequency select 0x0: /1 0x1: /2 (value after reset) 0x2: /4 Others: Setting prohibited	R/W
6:4	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
7	TRCKEN	Trace Clock operating Enable 0: Stop 1: Operation enable	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

Trace Clock Control Register controls switching the trace clock.

TRCKCR can be written only when the debugger is connected (DBGSTR.CDBGPWRUPREQ = 1).

Change the TRCLK frequency in the state of TRCKEN = 0.

Factor of the initialization of TRCKCR register is all resets.

8.3 Main Clock Oscillator

To supply the clock signal to the main clock oscillator, use one of the following ways:

- Connect an oscillator
- Connect the input of an external clock signal.

8.3.1 Connecting a Crystal Resonator

Figure 8.4 shows an example of connecting a crystal resonator. A damping resistor (Rd) can be added, if required.

Because the resistor values vary according to the resonator and the oscillation drive capability, use values recommended by the resonator manufacturer. If the manufacturer recommends using an external feedback resistor (Rf), insert an Rf between EXTAL and XTAL by following the instructions.

When connecting a resonator to supply the clock, the frequency of the resonator must be in the frequency range of the resonator for the main clock oscillator as described in Table 8.1.

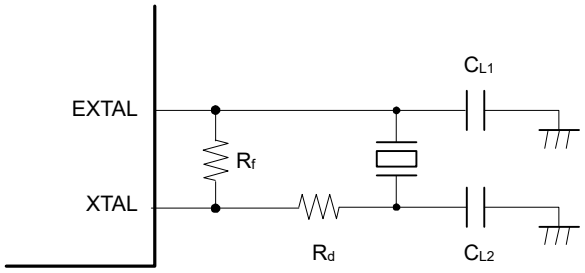


Figure 8.4 Example of crystal resonator connection

Figure 8.5 shows an equivalent circuit of the crystal resonator.

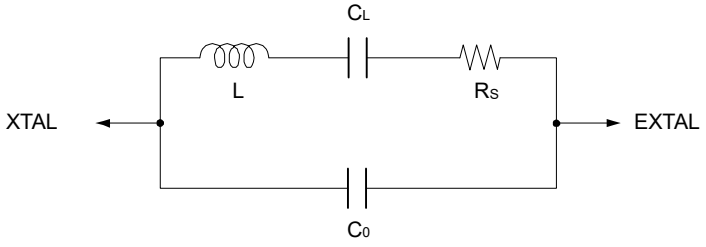


Figure 8.5 Equivalent circuit of the crystal resonator

8.3.2 External Clock Input

Figure 8.6 shows an example of connecting an external clock input. To operate the oscillator with an external clock signal, set the MOMCR.MOSEL bit to 1. The XTAL pin becomes high impedance.

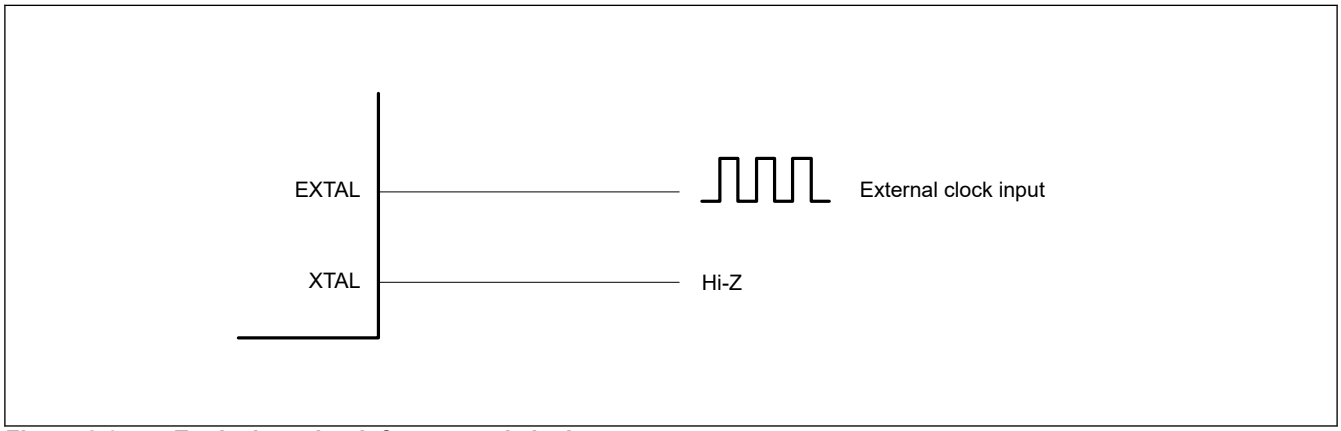


Figure 8.6 Equivalent circuit for external clock

8.3.3 Notes on External Clock Input

The frequency of the external clock input can only be changed when the main clock oscillator is stopped. Do not change the frequency of the external clock input when the setting of the Main Clock Oscillator Stop bit (MOSCCR.MOSTP) is 0.

8.4 Oscillation Stop Detection Function

8.4.1 Oscillation Stop Detection and Operation after Detection

The oscillation stop detection function detects the main clock oscillator stop. When oscillation stop is detected, the system clock switches as follows:

- If an oscillation stop is detected with SCKSCR.CKSEL[2:0] = 011b (system clock source = MOSC), the system clock source switches to the MOCO clock.
- If an oscillation stop is detected with PLLCCR.PLSRCSEL = 0 (PLL source clock = MOSC) and SCKSCR.CKSEL[2:0] = 101b (system clock source = PLL), PLL clock remains the system clock source. However, the frequency becomes a free-running oscillation frequency.

An oscillation stop detection interrupt request can be generated when an oscillation stop is detected. In addition, the General PWM Timer (GPT) output can be forced to a high-impedance state on detection.

The main clock oscillation stop is detected when the input clock remains at 0 or 1 for a certain period, for example, when a malfunction occurs in the main clock oscillator. See section 46, Electrical Characteristics.

Switching between the main clock oscillator and the MOCO clock or between the PLL clock and PLL free-running clock is controlled by the Oscillation Stop Detection Flag (OSTDSR.OSTDF).

OSTDF controls the switched clock as follows:

- When SCKSCR.CKSEL[2:0] = 011b (system clock source = MOSC):
  - When OSTDF changes from 0 to 1, the clock source switches to the MOCO clock.
  - When OSTDF changes from 1 to 0, the clock source switches back to MOSC.
- When PLLCCR.PLSRCSEL = 0 (PLL source clock = MOSC) and SCKSCR.CKSEL[2:0] = 101b (System clock source = PLL):
  - When OSTDF changes 0 to 1, the clock source switches to the PLL free-running oscillation clock.
  - When OSTDF changes 1 to 0, the clock source switches back to PLL.

To switch the clock source to the main clock or PLL clock again after the oscillation stop detection, set the CKSEL[2:0] bits to a clock source other than the main clock or PLL clock and clear the OSTDF flag to 0. Also, check that the OSTDF flag is not 1, then set the CKSEL[2:0] bits to the main clock or PLL clock after the specified oscillation stabilization time elapses.

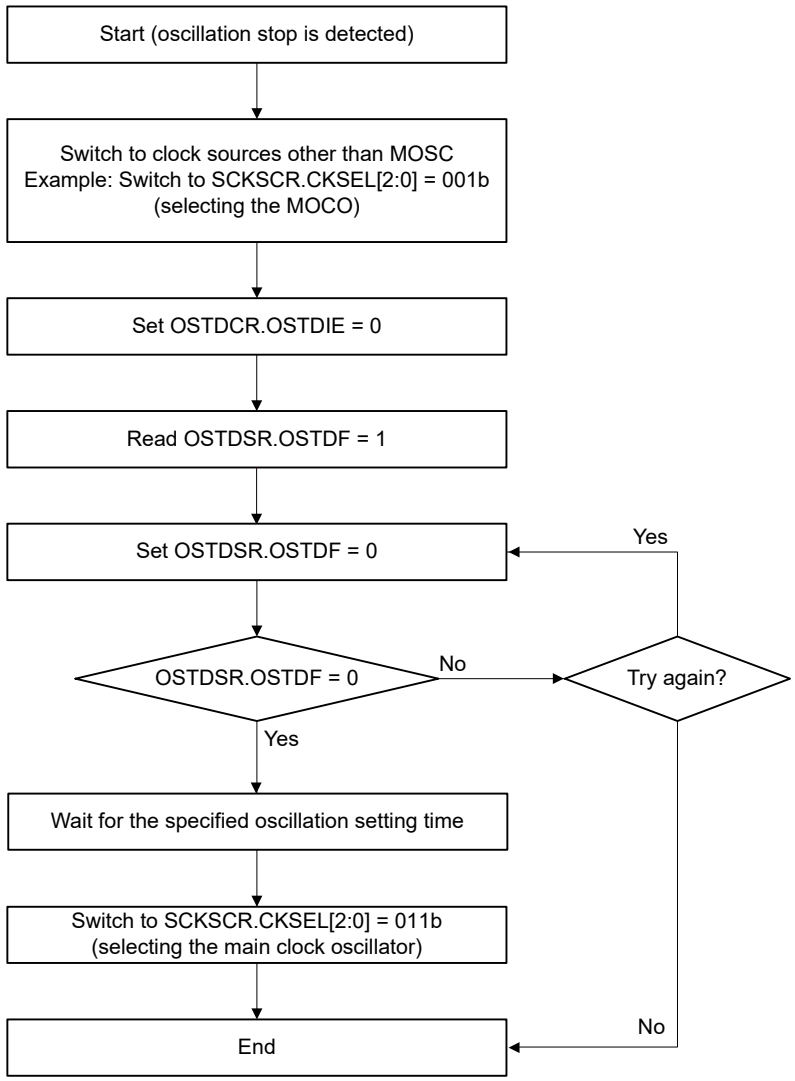
After a reset release, the main clock oscillator is stopped and the oscillation stop detection function is disabled. To enable the oscillation stop detection function, activate the main clock oscillator and write 1 to the oscillation stop detection function enable bit (OSTDCR.OSTDE) after a specified oscillation stabilization time elapses.

The oscillation stop detection function detects when the main clock is stopped by an external cause. Therefore, the oscillation stop detection function must be disabled before the main clock oscillator is stopped by software or a transition is made to Software Standby or Deep Software Standby mode.

The oscillation stop detection function switches all clocks that can be selected as the MOSC clock except CLKOUT to the MOCO (when system clock is MOSC) or PLL free-running (when system clock is PLL).

The system clock (ICLK) frequency during the MOCO(when system clock is MOSC) or PLL free-running (when system clock is PLL) operation is specified by the MOCO oscillation frequency and the division ratio set by the system clock select bits (SCKDIVCR.ICK[2:0])

Example of returning when CKSEL[2:0] = 011b (selecting the main clock oscillator) after an oscillation stop is detected



Note: On return from the oscillation-stopped state, the factor responsible for stopping the main clock oscillation circuit must be removed from the system to allow oscillation to resume.

Figure 8.7 Flow of recovery on detection of oscillator stop

8.4.2 Oscillation Stop Detection Interrupts

An oscillation stop detection interrupt (MOSC_STOP) is generated when the Oscillation Stop Detection Flag (OSTDSR.OSTDF) is 1 and the Oscillation Stop Detection Interrupt Enable bit in the Oscillation Stop Detection Control Register (OSTDCR.OSTDIE) is 1 (enabled). The Port Output Enable for GPT (POEG) is notified of the main clock oscillator stop. On receiving the notification, the POEG sets the Oscillation Stop Detection Flag in the POEG Group n Setting Register (POEGGn.OSTPF) to 1 (n = A, B, C, D).

After the oscillation stop is detected, wait at least 10 PCLKB clock cycles before writing to the POEGGn.OSTPF flag. When the OSTDSR.OSTDF flag requires clearing, do so after clearing the Oscillation Stop Detection Interrupt Enable bit in the Oscillation Stop Detection Control Register (OSTDCR.OSTDIE). Wait at least 2 PCLKB clock cycles before setting the OSTDCR.OSTDIE bit to 1 again. A longer PCLKB wait time might be required, depending on the number of cycles required to read a given I/O register.

The oscillation stop detection interrupt is a non-maskable interrupt. Because non-maskable interrupts are disabled in the initial state after a reset release, enable non-maskable interrupts through software before using oscillation stop detection interrupts. For details, see section 12, Interrupt Controller Unit (ICU).

8.5 PLL Circuit

The PLL circuit has a function to multiply the frequency from the oscillator.

8.6 Internal Clock

Clock sources for the internal clock signals include:

- Main clock
- HOCO clock
- MOCO clock
- LOCO clock
- PLL clock
- PLL2 clock
- IWDT-dedicated clock
- JTAG clock

The following internal clocks are produced from these sources.

- Operating clock of the CPU, DMAC, DTC, Flash, I/O ports, TFU, IIRFA and RAM: System clock (ICLK)
- Operating clocks of peripheral modules: Peripheral module clocks (PCLKA, PCLKB, PCLKC, PCLKD)
- Operating clock of the FlashIF: FlashIF clock (FCLK)
- Operating clock for the GPT : GPT clock (GPTCLK)
- Operating clock for the SCI and SPI: SCI SPI clock (SCISPICLK)
- Operating clock for the CANFD: CANFD clock (CANFDCLK)
- Operating clock for the CANFD: CAN clock (CANMCLK)
- Operating clocks for the IIC: IIC clock (IICCLK)
- Operating clocks for the CAC: CAC clock (CACCLK)
- Operating clock for the IWDT: IWDT-dedicated clock (IWDTCLK)
- Operating clock for the AGT: AGT-dedicated LOCO clock (AGTLCLK)
- Operating clock for the SysTick Timer: SysTick Timer-dedicated clock (SYSTICCLK)
- Clock for external pin output: Clock/Buzzer output clock(CLKOUT)
- Operating clock for the JTAG: JTAG clock (JTAGTCK)

For details on the registers used to set the frequencies of the internal clocks, see section 8.6.1. System Clock (ICLK) to section 8.6.14. JTAG Clock (JTAGTCK)

If the value of any of these bits is changed, subsequent operation is at the frequency determined by the new value.

8.6.1 System Clock (ICLK)

The system clock (ICLK) is the operating clock of the CPU, DMAC, DTC, Flash, I/O ports, TFU, IIRFA and SRAM.

The ICLK frequency is specified by the ICK[2:0] bits in SCKDIVCR, the CKSEL[2:0] bits in SCKSCR, the PLLMUL[5:0], and PLIDIV[1:0] bits in PLLCCR, and the HOCOFRQ0[1:0] bits in OFS1.

When the ICLK clock source is switched, the duration of the ICLK clock cycle becomes longer during the clock source transition period. See Figure 8.8 and Figure 8.9.

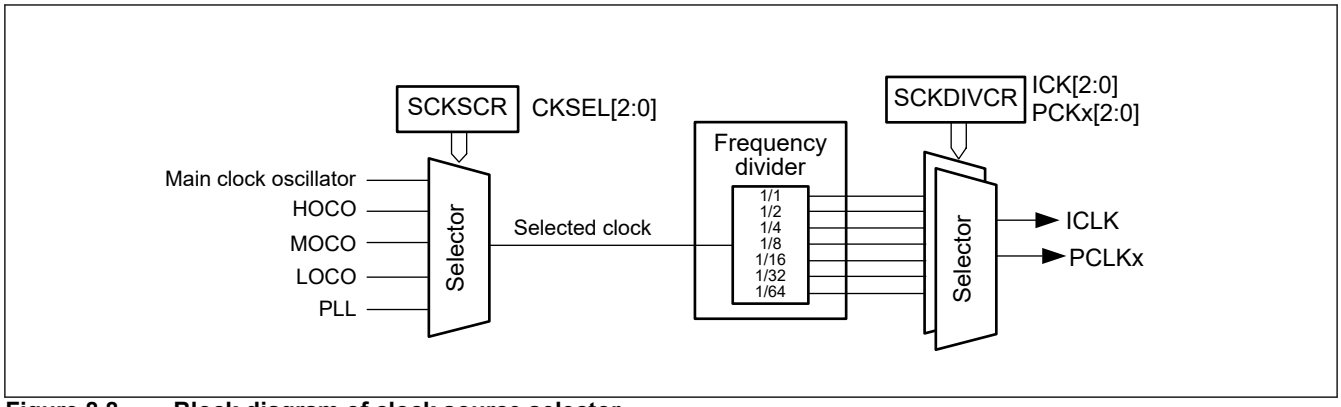


Figure 8.8 Block diagram of clock source selector

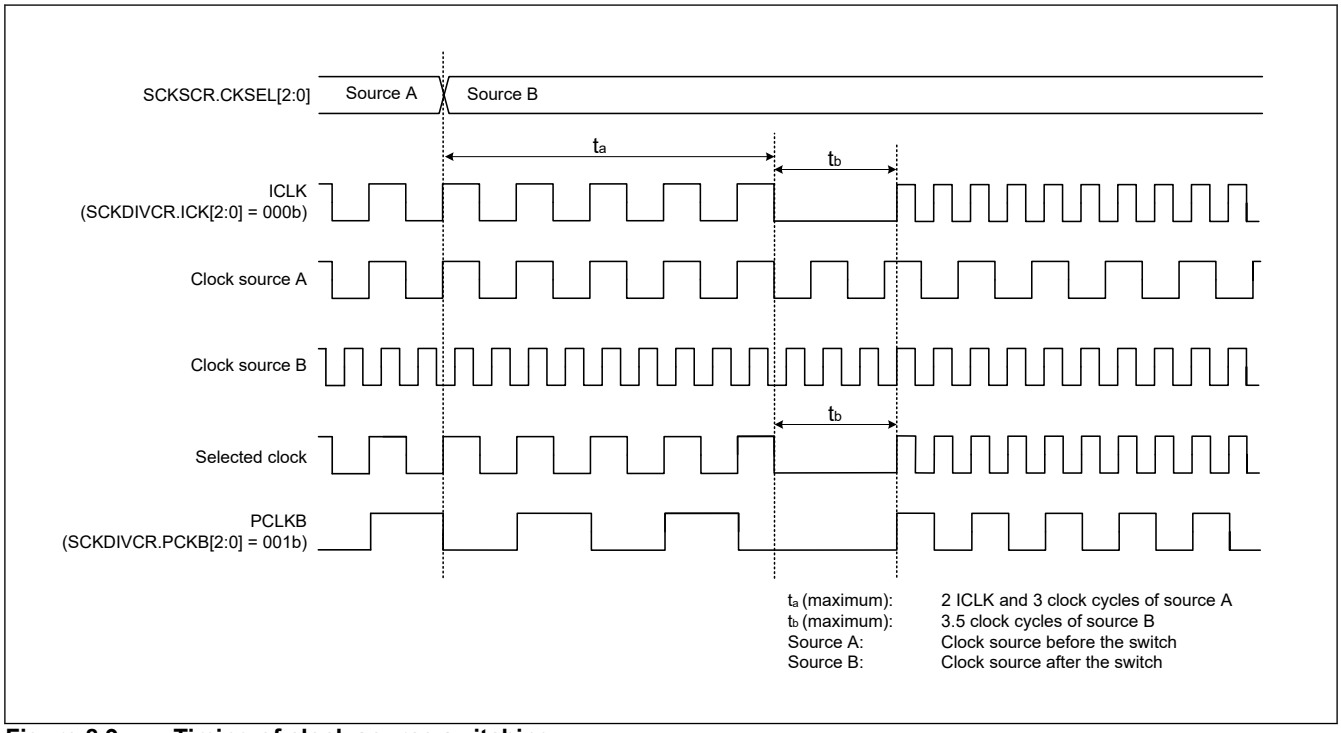


Figure 8.9 Timing of clock source switching

8.6.2 Peripheral Module Clock (PCLKA, PCLKB, PCLKC, PCLKD)

The peripheral module clocks (PCLKA, PCLKB, PCLKC and PCLKD) are the operating clocks for the peripheral modules.

The frequency of the given clock is specified in the following bits:

- PCKA[2:0], PCKB[2:0], PCKC[2:0] and PCKD[2:0] bits in SCKDIVCR
- CKSEL[2:0] bits in SCKSCR
- PLLMUL[5:0] and PLIDIV[1:0] bits in PLLCCR
- HOCOFRQ0[1:0] bits in OFS1.

When the clock source of the peripheral module clock is switched, the duration of the peripheral module clock cycle becomes longer during the clock source transition period. See Figure 8.8 and Figure 8.9.

8.6.3 FlashIF Clock (FCLK)

The flash interface clock (FCLK) is the operating clock for the flash memory interface. In addition to reading from the data flash, FCLK is used for the programming and erasure of the code flash and data flash.

The FCLK frequency is specified in the following bits:

- FCK[2:0] bits in SCKDIVCR
- CKSEL[2:0] bits in SCKSCR
- PLLMUL[5:0] and PLIDIV[1:0] bits in PLLCCR
- HOCOFRQ0[1:0] bits in OFS1.

8.6.4 GPT Clock (GPTCLK)

The GPT clock (GPTCLK) is the operating clock for the GPT module.

The GPTCLK frequency is specified in the following bits:

- GPTCKSEL[2:0] bits in GPTCKCR
- GPTCKDIV[2:0] bits in GPTCKDIVCR
- PLLMUL[5:0] and PLIDIV[1:0] bits in PLLCCR
- PLL2MUL[5:0] and PL2IDIV[1:0] bits in PLL2CCR
- HOCOFRQ0[1:0] bits in OFS1.

8.6.5 SCI SPI clock (SCISPICLK)

The SCI SPI clock (SCISPICLK) is the operating clock for the SCI and SPI module.

The SCISPICLK frequency is specified in the following bits:

- SCISPICKSEL[2:0] bits in SCISPICKCR
- SCISPICKDIV[2:0] bits in SCISPICKDIVCR
- PLLMUL[5:0] and PLIDIV[1:0] bits in PLLCCR
- PLL2MUL[5:0] and PL2IDIV[1:0] bits in PLL2CCR
- HOCOFRQ0[1:0] bits in OFS1.

8.6.6 CAN Clock (CANMCLK)

The CAN clock, CANMCLK, is the operating clock for the CANFD module. CANMCLK is generated by the main clock oscillator.

8.6.7 CANFD Clock (CANFDCLK)

The CANFD clock (CANFDCLK) is the operating clock for the CANFD module.

The CANFDCLK frequency is specified in the following bits:

- CANFDCKSEL[2:0] bits in CANFDCKCR
- CANFDCKDIV[2:0] bits in CANFDCKDIVCR
- PLLMUL[5:0] and PLIDIV[1:0] bits in PLLCCR
- PLL2MUL[5:0] bits and PL2IDIV[1:0] bits in PLL2CCR.

8.6.8 CAC Clock (CACCLK)

The CAC clock, CACCLK, is the operating clock for the CAC. CACCLK is generated by the following oscillators:

- Main clock oscillator
- High-speed clock oscillator (HOCO)
- Middle-speed clock oscillator (MOCO)
- Low-speed on-chip oscillator (LOCO)
- IWDT-dedicated on-chip oscillator. (IWDTLOCO)

8.6.9 IIC clock (IICCLK)

The IIC clock (IICCLK) is the operating clock for the IIC module.

The IICCLK frequency is specified in the following bits:

- IICCKSEL[2:0] bits in IICCKCR
- IICCKDIV[2:0] bits in IICCKDIVCR
- PLLMUL[5:0] and PLIDIV[1:0] bits in PLLCCR
- PLL2MUL[5:0] and PL2IDIV[1:0] bits in PLL2CCR
- HOCOFRQ0[1:0] bits in OFS1.

8.6.10 IWDT-Dedicated Clock (IWDTCLK)

The IWDT-dedicated clock (IWDTCLK) is the operating clock for the IWDT. IWDTCLK is internally generated by the IWDT-dedicated on-chip oscillator.

8.6.11 AGT-Dedicated LOCO Clock (AGTLCLK)

The AGT-dedicated LOCO clock (AGTLCLK) is the operating clock for the AGT. AGTLCLK is generated by the LOCO clock.

8.6.12 SysTick Timer-Dedicated Clock (SYSTICCLK)

The SysTick timer-dedicated clock, SYSTICCLK, is the operating clock for the SysTick timer. SYSTICCLK is generated by the LOCO clock.

8.6.13 External Pin Output Clock (CLKOUT)

The CLKOUT is output externally from the CLKOUT pin for the clock or buzzer output. The CLKOUT is output to the CLKOUT pin when the CKOCR.CKOEN bit is set to 1. Only change the value in the CKODIV[2:0] bits or CKOSEL[2:0] bits in CKOCR when the CKOCR.CKOEN bit is 0.

The CLKOUT clock frequency is specified in the following bits:

- CKODIV[2:0] bits or CKOSEL[2:0] bits in CKOCR
- HOCOFRQ0[1:0] bits in OFS1

8.6.14 JTAG Clock (JTAGTCK)

The JTAG clock (JTAGTCK) is the clock for the JTAG.

JTAGTCK is generated by the JTAG external clock (TCK).

8.7 Usage Notes

8.7.1 Notes on Clock Generation Circuit

The frequency of the following clocks supplied to each module changes according to the setting of the SCKDIVCR register:

- System clock (ICLK)
- Peripheral module clocks (PCLKA, PCLKB, PCLKC, and PCLKD)
- FlashIF clock (FCLK)

Each frequency must meet the following conditions:

- Each frequency must be selected within the operation-guaranteed range of the operating frequency (f) specified in the AC characteristics. See section 46, Electrical Characteristics.
- The system clock, peripheral module clock must be set according to Table 8.2.

To ensure correct processing after the clock frequency changes, first write to the relevant Clock Control register to change the frequency, then read the value from the register, and finally perform the subsequent processing.

8.7.2 Notes on Board Design

When using a crystal resonator, place the resonator and its load capacitors as close to the XTAL and EXTAL pins as possible. Other signal lines should be routed away from the oscillation circuit as shown in Figure 8.10 to prevent electromagnetic induction from interfering with correct oscillation. Figure 8.10 shows the case which the main clock oscillator is used.

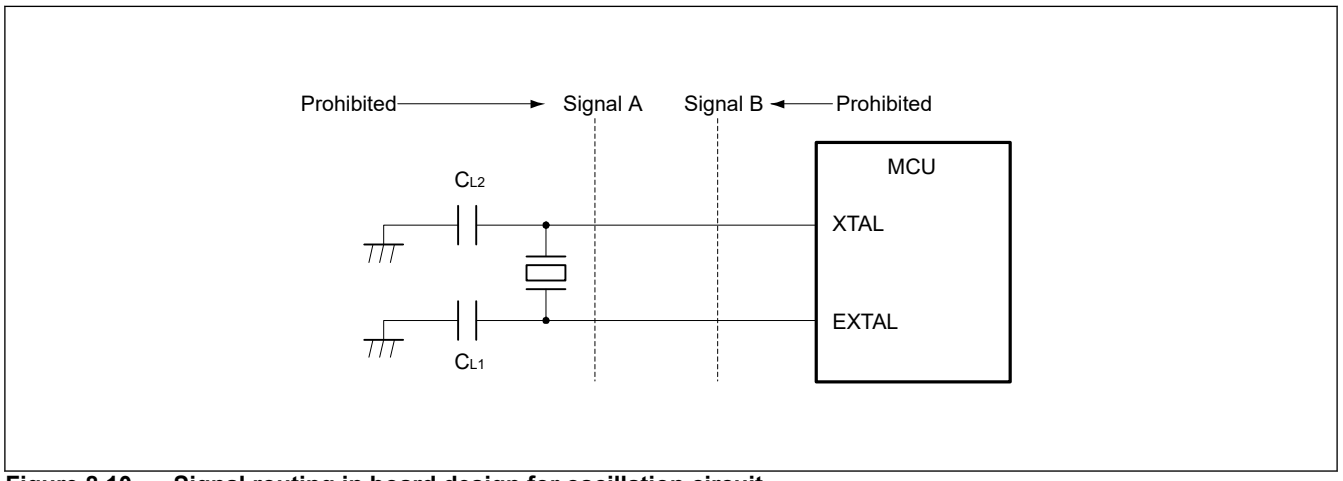


Figure 8.10 Signal routing in board design for oscillation circuit

8.7.3 Notes on Resonator Connect Pin

When the main clock is not used, the EXTAL and XTAL pins can be used as general ports. When these pins are used as general ports, the main clock must be stopped (MOSCCR.MOSTP bit should be set to 1).

9. Clock Frequency Accuracy Measurement Circuit (CAC)

9.1 Overview

The Clock Frequency Accuracy Measurement Circuit (CAC) counts pulses of the clock to be measured (measurement target clock) within the time generated by the clock selected as the measurement reference (measurement reference clock), and determines the accuracy depending on whether the number of pulses is within the allowable range. When measurement is complete or the number of pulses within the time generated by the measurement reference clock is not within the allowable range, an interrupt request is generated.

Table 9.1 lists the CAC specifications, Figure 9.1 shows the CAC block diagram, and Table 9.2 lists the CAC I/O pin.

Table 9.1 CAC specifications

Parameter	Specifications
Measurement target clocks	Frequency can be measured for: • Main clock oscillator • HOCO clock • MOCO clock • LOCO clock • Peripheral module clock B (PCLKB) • IWDT-dedicated clock
Measurement reference clocks	Frequency can be referenced to: • External clock input to the CACREF pin • Main clock oscillator • HOCO clock • MOCO clock • LOCO clock • Peripheral module clock B (PCLKB) • IWDT-dedicated clock
Selectable function	Digital filter
Interrupt sources	• Measurement end • Frequency error • Overflow
Module-stop function	Module-stop state can be set to reduce power consumption
TrustZone Filter	Security attribution can be set

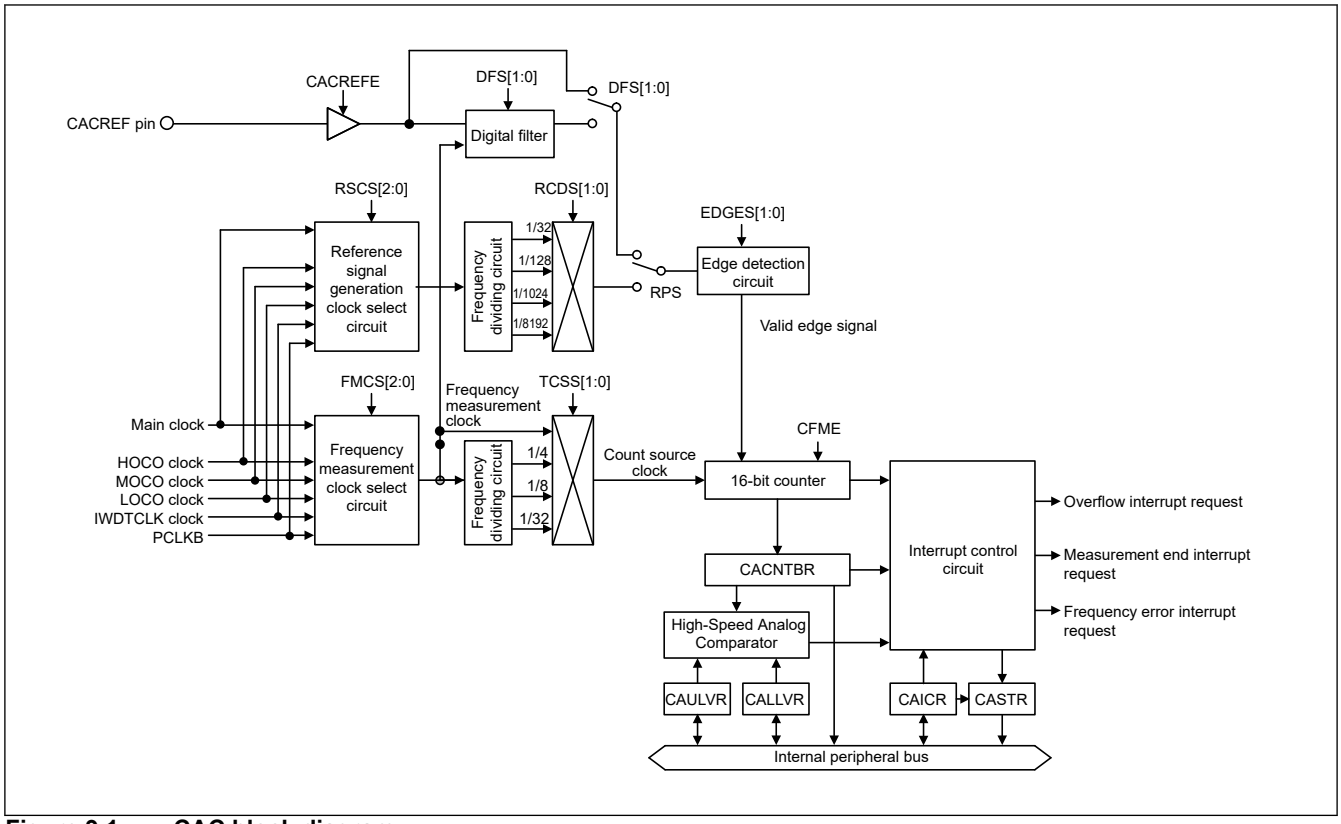


Figure 9.1 CAC block diagram

Table 9.2 CAC I/O pin

Function	Pin name	I/O	Description
CAC	CACREF	Input	Measurement reference clock input pin

9.2 Register Descriptions

9.2.1 CACR0 : CAC Control Register 0

Base address: CAC = 0x4008_3600

Offset address: 0x00

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

Bit	Symbol	Function	R/W
0	CFME	Clock Frequency Measurement Enable 0: Disable 1: Enable	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

CFME bit (Clock Frequency Measurement Enable)

The CFME bit enables clock frequency measurement. Changes made to this bit are not immediately reflected to the internal circuit. Read the bit to confirm that the change has been reflected.

9.2.2 CACR1 : CAC Control Register 1

Base address: CAC = 0x4008_3600

Offset address: 0x01

Bit position:	7	6	5	4	3	2	1	0
Bit field:	EDGES[1:0]		TCSS[1:0]		FMCS[2:0]			CACREFE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CACREFE	CACREF Pin Input Enable 0: Disable 1: Enable	R/W
3:1	FMCS[2:0]	Measurement Target Clock Select 0 0 0: Main clock oscillator 0 0 1: Setting prohibited 0 1 0: HOCO clock 0 1 1: MOCO clock 1 0 0: LOCO clock 1 0 1: Peripheral module clock B (PCLKB) 1 1 0: IWDT-dedicated clock 1 1 1: Setting prohibited	R/W
5:4	TCSS[1:0]	Timer Count Clock Source Select 0 0: No division 0 1: × 1/4 clock 1 0: × 1/8 clock 1 1: × 1/32 clock	R/W
7:6	EDGES[1:0]	Valid Edge Select 0 0: Rising edge 0 1: Falling edge 1 0: Both rising and falling edges 1 1: Setting prohibited	R/W

Note: Set the CACR1 register when the CACR0.CFME bit is 0.

CACREFE bit (CACREF Pin Input Enable)

The CACREFE bit enables the CACREF pin input.

FMCS[2:0] bits (Measurement Target Clock Select)

The FMCS[2:0] bits select the measurement target clock whose frequency is to be measured.

TCSS[1:0] bits (Timer Count Clock Source Select)

The TCSS[1:0] bits select the division ratio of the measurement target clock.

EDGES[1:0] bits (Valid Edge Select)

The EDGES[1:0] bits select the valid edge for the reference signal.

9.2.3 CACR2 : CAC Control Register 2

Base address: CAC = 0x4008_3600

Offset address: 0x02

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DFS[1:0]		RCDS[1:0]		RSCS[2:0]			RPS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RPS	Reference Signal Select 0: CACREF pin input 1: Internal clock (internally generated signal)	R/W
3:1	RSCS[2:0]	Measurement Reference Clock Select 0 0 0: Main clock oscillator 0 0 1: Setting prohibited 0 1 0: HOCO clock 0 1 1: MOCO clock 1 0 0: LOCO clock 1 0 1: Peripheral module clock B (PCLKB) 1 1 0: IWDT-dedicated clock 1 1 1: Setting prohibited	R/W
5:4	RCDS[1:0]	Measurement Reference Clock Frequency Division Ratio Select 0 0: × 1/32 clock 0 1: × 1/128 clock 1 0: × 1/1024 clock 1 1: × 1/8192 clock	R/W
7:6	DFS[1:0]	Digital Filter Select 0 0: Disable digital filtering 0 1: Use sampling clock for the digital filter as the frequency measuring clock 1 0: Use sampling clock for the digital filter as the frequency measuring clock divided by 4 1 1: Use sampling clock for the digital filter as the frequency measuring clock divided by 16.	R/W

Note: Set the CACR2 register when the CACR0.CFME bit is 0.

RPS bit (Reference Signal Select)

The RPS bit selects whether to use the CACREF pin input or an internal clock (internally generated signal) as the reference signal.

RSCS[2:0] bits (Measurement Reference Clock Select)

The RSCS[2:0] bits select the reference clock for measurement.

RCDS[1:0] bits (Measurement Reference Clock Frequency Division Ratio Select)

The RCDS[1:0] bits select the frequency-divisor of the reference clock for measurement when an internal reference clock is selected. When RPS = 0 (CACREF pin is used as the reference clock source), the reference clock is not divided.

DFS[1:0] bits (Digital Filter Select)

The DFS[1:0] bits enable or disable the digital filter and selects its sampling clock.

9.2.4 CAICR : CAC Interrupt Control Register

Base address: CAC = 0x4008_3600

Offset address: 0x03

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	OVFFCL	MENDFCL	FERRFCL	—	OVFIE	MENDIE	FERRIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	FERRIE	Frequency Error Interrupt Request Enable 0: Disable 1: Enable	R/W
1	MENDIE	Measurement End Interrupt Request Enable 0: Disable 1: Enable	R/W

Bit	Symbol	Function	R/W
2	OVFIE	Overflow Interrupt Request Enable 0: Disable 1: Enable	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	FERRFCL	FERRF Clear 0: No effect 1: The CASTR.FERRF flag is cleared	W
5	MENDFCL	MENDF Clear 0: No effect 1: The CASTR.MENDF flag is cleared	W
6	OVFFCL	OVFF Clear 0: No effect 1: The CASTR.OVFF flag is cleared.	W
7	—	This bit is read as 0. The write value should be 0.	R/W

FERRIE bit (Frequency Error Interrupt Request Enable)

The FERRIE bit enables or disables the frequency error interrupt request.

MENDIE bit (Measurement End Interrupt Request Enable)

The MENDIE bit enables or disables the measurement end interrupt request.

OVFIE bit (Overflow Interrupt Request Enable)

The OVFIE bit enables or disables the overflow interrupt request.

FERRFCL bit (FERRF Clear)

Setting the FERRFCL bit to 1 clears the CASTR.FERRF flag.

MENDFCL bit (MENDF Clear)

Setting the MENDFCL bit to 1 clears the CASTR.MENDF flag.

OVFFCL bit (OVFF Clear)

Setting the OVFFCL bit to 1 clears the CASTR.OVFF flag.

9.2.5 CASTR : CAC Status Register

Base address: CAC = 0x4008_3600

Offset address: 0x04

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

Bit	Symbol	Function	R/W
0	FERRF	Frequency Error Flag 0: Clock frequency is within the allowable range 1: Clock frequency has deviated beyond the allowable range (frequency error).	R
1	MENDF	Measurement End Flag 0: Measurement is in progress 1: Measurement ended	R
2	OVFF	Overflow Flag 0: Counter has not overflowed 1: Counter overflowed	R
7:3	—	These bits are read as 0.	R

FERRF flag (Frequency Error Flag)

The FERRF flag indicates a deviation of the clock frequency from the set value (frequency error).

[Setting condition]

- The clock frequency is outside the allowable range defined in the CAULVR and CALLVR registers.

[Clearing condition]

- 1 is written to the FERRFCL bit.

MENDF flag (Measurement End Flag)

The MENDF flag indicates the end of measurement.

[Setting condition]

- Measurement ends.

[Clearing condition]

- 1 is written to the MENDFCL bit.

OVFF flag (Overflow Flag)

The OVFF flag indicates that the counter overflowed.

[Setting condition]

- The counter overflows.

[Clearing condition]

- 1 is written to the CAICR.OVFFCL bit.

9.2.6 CAULVR : CAC Upper-Limit Value Setting Register

Base address: CAC = 0x4008_3600

Offset address: 0x06

Bit position:	15															0
Bit field:																
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	n/a	The Upper Value of the Allowable Range The CAULVR register is a 16-bit read/write register that specifies the upper value of the allowable range. When the counter value exceeds the value specified in this register, a frequency error is detected. Write to this register when the CACR0.CFME bit is 0. The counter value stored in CACNTBR can vary depending on the difference between the phases of the digital filter and edge-detection circuit, and the signal on the CACREF pin. Ensure that this setting allows an adequate margin.	R/W

9.2.7 CALLVR : CAC Lower-Limit Value Setting Register

Base address: CAC = 0x4008_3600

Offset address: 0x08

Bit position:	15															0
Bit field:																
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	n/a	The Lower Value of the Allowable Range The CALLVR register is a 16-bit read/write register that specifies the lower value of the allowable range. When the counter value falls below the value specified in this register, a frequency error is detected. Write to this register when the CACR0.CFME bit is 0. The counter value stored in CACNTBR can vary depending on the difference between the phases of the digital filter and edge-detection circuit, and the signal on the CACREF pin. Ensure that this setting allows an adequate margin.	R/W

9.2.8 CACNTBR : CAC Counter Buffer Register

Base address: CAC = 0x4008_3600

Offset address: 0x0A

Bit position:	15															0
Bit field:																
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	n/a	The Measurement Result The CACNTBR register is a 16-bit read-only register that stores the measurement result.	R

9.3 Operation

9.3.1 Measuring Clock Frequency

The CAC measures the clock frequency using the CACREF pin input or an internal clock as a reference. Figure 9.2 shows an operating example of the CAC.

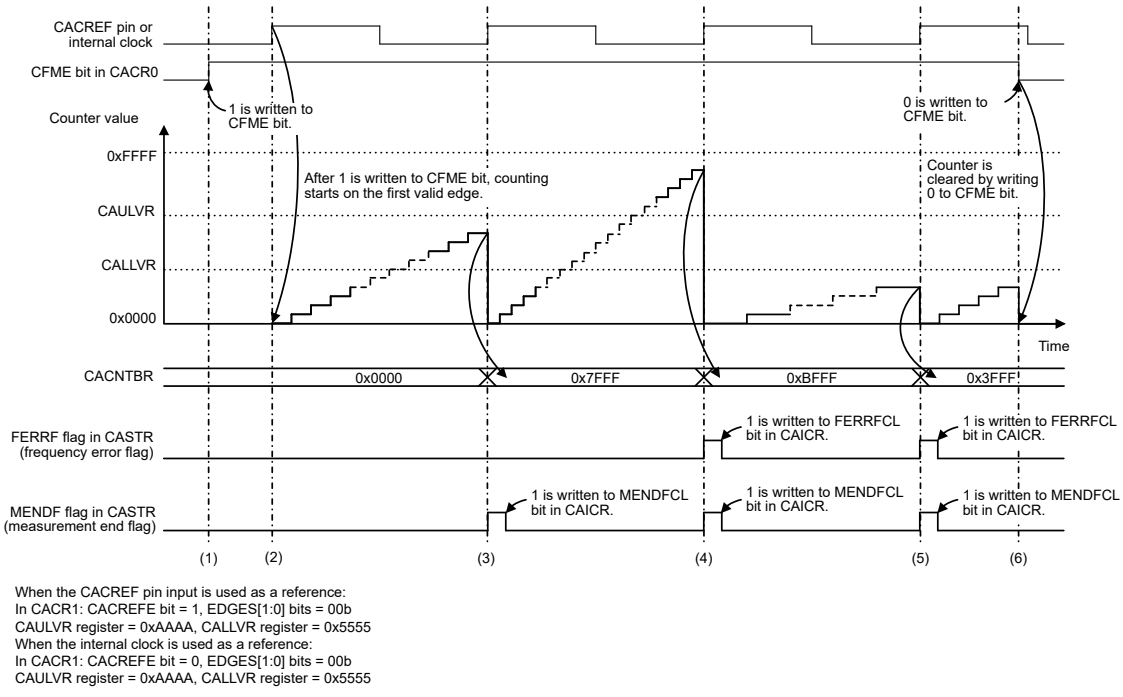


Figure 9.2 CAC operating example

The events in Figure 9.2 are:

1. When the CACREF pin input is used as reference (CACR1.CACREFE = 1), frequency measurement is enabled by writing 1 to the CACR0.CFME bit while the CACR2.RPS bit is set to 0 and the CACR1.CACREFE bit is set to 1. When the internal clock is used as reference (CACR1.CACREFE = 0), frequency measurement is enabled by writing 1 to the CACR0.CFME bit while the CACR2.RPS bit is set to 1.
2. When the CACREF pin input is used as reference, after 1 is written to the CFME bit, the timer starts up-counting if the valid edge selected by the CACR1.EDGES[1:0] bits (rising edge (CACR1.EDGES[1:0] = 00b) in Figure 9.2) is input from the CACREF pin. When the internal clock is used as reference, after 1 is written to the CFME bit, the timer starts up-counting if the valid edge selected by the CACR1.EDGES[1:0] bits (rising edge (CACR1.EDGES[1:0] = 00b) in Figure 9.2) is input based on the clock source selected by the CACR2.RSCS[2:0] bits.
3. When the next valid edge is input, the counter value is transferred to CACNTBR and compared with the values in CAULVR and CALLVR. If both CACNTBR ≤ CAULVR and CACNTBR ≥ CALLVR are true, only the MENDF flag in CASTR is set to 1, because the clock frequency is correct. If the MENDIE bit in CAICR is 1, a measurement end interrupt is generated.
4. When the next valid edge is input, the counter value is transferred to CACNTBR and compared with the values in CAULVR and CALLVR. If CACNTBR > CAULVR, the FERRF flag in CASTR is set to 1, because the clock frequency is erroneous. If the FERRIE bit in CAICR is 1, a frequency error interrupt is generated. The MENDF flag in CASTR is set to 1 at the end of measurement. If the MENDIE bit in CAICR is 1, a measurement end interrupt is generated.
5. When the next valid edge is input, the counter value is transferred to CACNTBR and compared with the values in CAULVR and CALLVR. If CACNTBR < CALLVR, the FERRF flag in CASTR is set to 1, because the clock frequency is erroneous. If the FERRIE bit in CAICR is 1, a frequency error interrupt is generated. The MENDF flag in CASTR is set to 1 at the end of measurement. If the MENDIE bit in CAICR is 1, a measurement end interrupt is generated.
6. When the CFME bit in CACR0 is 1, the counter value is transferred to CACNTBR and compared with the values in CAULVR and CALLVR every time a valid edge is input. Writing 0 to the CFME bit in CACR0 clears the counter and stops up-counting.

9.3.2 Digital Filtering of Signals on CACREF Pin

The CACREF pin has a digital filter, and levels on the CACREF pin are transmitted to the internal circuitry after three consecutive matches in the selected sampling interval. The same level continues to be transmitted internally until the level on the pin has three consecutive matches again. Enabling or disabling of the digital filter and its sampling clock are selectable.

The counter value transferred to CACNTBR might be in error by up to 1 cycle of the sampling clock because of the difference between the phases of the digital filter and the signal input to the CACREF pin. When a frequency dividing clock is selected as a count source clock, the counter value error is obtained using the following formula:

$$\text{Counter value error} = (1 \text{ cycle of the count source clock}) / (1 \text{ cycle of the sampling clock})$$

9.4 Interrupt Requests

The CAC generates three types of interrupt requests:

- Frequency error interrupt
- Measurement end interrupt
- Overflow interrupt

When an interrupt source is generated, the associated status flag is set to 1. Table 9.3 provides information on the CAC interrupt requests.

Table 9.3 CAC interrupt requests (1 of 2)

Interrupt request	Interrupt enable bit	Status flag	Interrupt sources
Frequency error interrupt	CAICR.FERRIE	CASTR.FERRF	The result of comparing CACNTBR with CAULVR and CALLVR is either CACNTBR > CAULVR or CACNTBR < CALLVR

Table 9.3 CAC interrupt requests (2 of 2)

Interrupt request	Interrupt enable bit	Status flag	Interrupt sources
Measurement end interrupt	CAICR.MENDIE	CASTR.MENDF	• Valid edge is input from the CACREF pin or internal clock • Measurement end interrupt does not occur at the first valid edge after writing 1 to the CACR0.CFME bit
Overflow interrupt	CAICR.OVFIE	CASTR.OVFF	Counter overflows

9.5 Usage Notes

9.5.1 Settings for the Module-Stop Function

The Module Stop Control Register C (MSTPCRC) can enable or disable CAC operation. The CAC module is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 10, Low Power Modes.

10. Low Power Modes

10.1 Overview

The MCU has several functions for reducing power consumption, such as setting clock dividers, stopping modules, selecting power control mode in Normal mode, and transitioning to low power modes.

Table 10.1 lists the specifications of the low power mode functions. Table 10.2 lists the conditions to transition to low power modes, the states of the CPU and peripheral modules, and the method for canceling each mode. After a reset, the MCU enters the program execution state, but only the DTC, DMAC and SRAM operate.

Table 10.1 Specifications of the low power mode functions

Item	Specification
Reducing power consumption by switching clock signals	The frequency division ratio can be selected independently for the system clock (ICLK), peripheral module clocks (PCLKA, PCLKB, PCLKC, PCLKD), and flash interface clock (FCLK). *1
Module stop	Functions can be stopped independently for each peripheral module
Low-power modes	• Sleep mode • Software Standby mode • Snooze mode • Deep Software Standby mode
Power control modes	• Power consumption can be reduced in Normal and Sleep modes by selecting an appropriate operating power control mode according to the operating frequency. • Two operating power control modes are available: High-speed mode Low-speed mode
TrustZone Filter	Security attribution can be set for each registers

Note 1. For details, see section 8, Clock Generation Circuit

Table 10.2 Operating conditions of each low power mode (1 of 2)

Item	Sleep mode	Software Standby mode	Snooze mode	Deep Software Standby mode
Transition condition	WFI instruction while SBYCR.SSBY = 0	WFI instruction while SBYCR.SSBY = 1 and DPSBYCR.DPSBY = 0	Snooze request trigger in Software Standby mode. SNZCR.SNZE=1.	WFI instruction while SBYCR.SSBY = 1 and DPSBYCR.DPSBY = 1
Canceling method	All interrupts. Any reset available in the mode.	Interrupts shown in Table 10.3. Any reset available in the mode.	Interrupts shown in Table 10.3. Any reset available in the mode.	Interrupts shown in Table 10.3. Any reset available in the mode.
State after cancellation by an interrupt	Program execution state (interrupt processing)	Program execution state (interrupt processing)	Program execution state (interrupt processing)	Reset state
State after cancellation by a reset	Reset state	Reset state	Reset state	Reset state
Main clock oscillator	Selectable	Stop	Selectable*4	Stop
High-speed on-chip oscillator	Selectable	Stop	Selectable	Stop
Middle-speed on-chip oscillator	Selectable	Stop	Selectable	Stop
Low-speed on-chip oscillator	Selectable	Selectable	Selectable	Selectable*7
IWDT-dedicated on-chip oscillator	Selectable*1	Selectable*1	Selectable*1	Stop
PLL	Selectable	Stop	Selectable*4	Stop
PLL2	Selectable	Stop	Selectable*4	Stop
Oscillation stop detection function	Selectable	Operation prohibited	Operation prohibited	Operation prohibited
Clock/buzzer output function	Selectable	Selectable*2	Selectable	Stop (Undefined)
CPU	Stop (Retained)	Stop (Retained)	Stop (Retained)	Stop (Undefined)
SRAMn (n = 0)	Selectable	Stop (Retained)	Selectable	Stop (Undefined)
Standby SRAM	Selectable	Stop (Retained)	Selectable	Stop (Retained/Undefined)*8
Flash memory	Operating	Stop (Retained)	Stop (Retained)	Stop (Retained)

Table 10.2 Operating conditions of each low power mode (2 of 2)

Item	Sleep mode	Software Standby mode	Snooze mode	Deep Software Standby mode
DMA Controller (DMAC)	Selectable	Stop (Retained)	Operation prohibited	Stop (Undefined)
Data Transfer Controller (DTC)	Selectable	Stop (Retained)	Selectable	Stop (Undefined)
Watchdog Timer (WDT)	Selectable*1	Stop (Retained)	Stop (Retained)	Stop (Undefined)
Independent Watchdog Timer (IWDT)	Selectable*1	Selectable*1	Selectable*1	Stop (Undefined)
Asynchronous General Purpose Timer (AGTn (n = 0, 1))	Selectable	Selectable*12	Selectable*12	Stop (Undefined)
A/D Converter (ADC)	Selectable	Stop (Retained)	Operation prohibited	Stop (Undefined)
Programmable Gain Amplifiers (PGAs)	Selectable*13	Stop (Retained)	Selectable*13	Stop (Undefined)
12-Bit D/A Converter (DAC12)	Selectable	Stop (Retained)	Selectable	Stop (Undefined)
Data Operation Circuit (DOC)	Selectable	Stop (Retained)	Selectable	Stop (Undefined)
Serial Communications Interface (SCI0)	Selectable	Stop (Retained)	Selectable (RXD0 falling edge is available, to enter snooze mode) (only in asynchronous mode).*5	Stop (Undefined)
Serial Communications Interface (SCIn (n = 1 to 4, 9))	Selectable	Stop (Retained)	Operation prohibited	Stop (Undefined)
I2C Bus Interface (IIC0)	Selectable	Selectable*3	Selectable*3 Only wakeup interrupt is available.	Stop (Undefined)
I2C Bus Interface (IIC1)	Selectable	Stop (Retained)	Operation prohibited	Stop (Undefined)
Event Link Controller (ELC)	Selectable	Stop (Retained)	Selectable*6	Stop (Undefined)
High-Speed Analog Comparator (ACMPHSn, n = 0 to 3)	Selectable	Stop (Retained)	Selectable VCOUT function only.*9	Stop (Undefined)
IRQn (n = 0 to 15) pin interrupt	Selectable	Selectable	Selectable	Stop (Undefined)
NMI, IRQn-DS (n = 0 to 15) pin interrupt	Selectable	Selectable	Selectable	Selectable
Key Interrupt Function (KINT)	Selectable	Selectable	Selectable	Stop (Undefined)
Low voltage detection (LVD)	Selectable	Selectable	Selectable	Selectable*10
Power-on reset circuit	Operating	Operating	Operating	Operating*11
Other peripheral modules	Selectable	Stop (Retained)	Operation prohibited	Stop (Undefined)
I/O Ports	Operating	Retained	Operating	Retained

Note: Selectable means that operating or not operating can be selected by the control registers.
Stop (Retained) means that the contents of the internal registers are retained but the operations are suspended.
Operation prohibited means that the function must be stopped before entering Software Standby mode.
Stop (Undefined) means that the contents of the internal registers are undefined and power to the internal circuit is cut off.
All modules whose module-stop bits are 0 start as soon as PCLKs are supplied after entering Snooze mode. In order to avoid increase in power consumption in Snooze mode, module-stop bit of modules which are unnecessary in Snooze mode must be set to 1 before entering Software Standby mode.

Note 1. In IWDT-dedicated on-chip oscillator and IWDT, operating or stopping is selected by setting the IWDT Stop Control bit (IWDTSTPCTL) in Option Function Select register 0 (OFS0) in IWDT auto start mode. In WDT, operating or stopping is selected by setting the WDT Stop Control bit (WDTSTPCTL) in Option Function Select Register 0 (OFS0) in WDT auto start mode.Power consumption can be reduced in Normal and Sleep modes by selecting an appropriate operating power control mode according to the operating frequency.

Note 2. Stopped when the clock output source select bits (CKOCR.CKOSEL[2:0]) are set to a value other than 010b (LOCO).

Note 3. IIC0 wakeup interrupt is available.

Note 4. When using SCI0 in Snooze mode, MOSCCR.MOSTP and PLLCR.PLLSTP and PLL2CR.PLL2STP bits must be 1.

Note 5. Serial communication modes of SCI0 is only in asynchronous mode.

Note 6. Event lists the restrictions described in section 10.10.12. ELC Events in Snooze Mode.

Note 7. If the DPSBYCR.DEEPCUT[1:0] bits are 00b, the oscillator status is the same as before entering Deep Software Standby mode. When the DPSBYCR.DEEPCUT[1:0] bits are not 00b, the oscillator stops when the MCU enters Deep Software Standby mode.

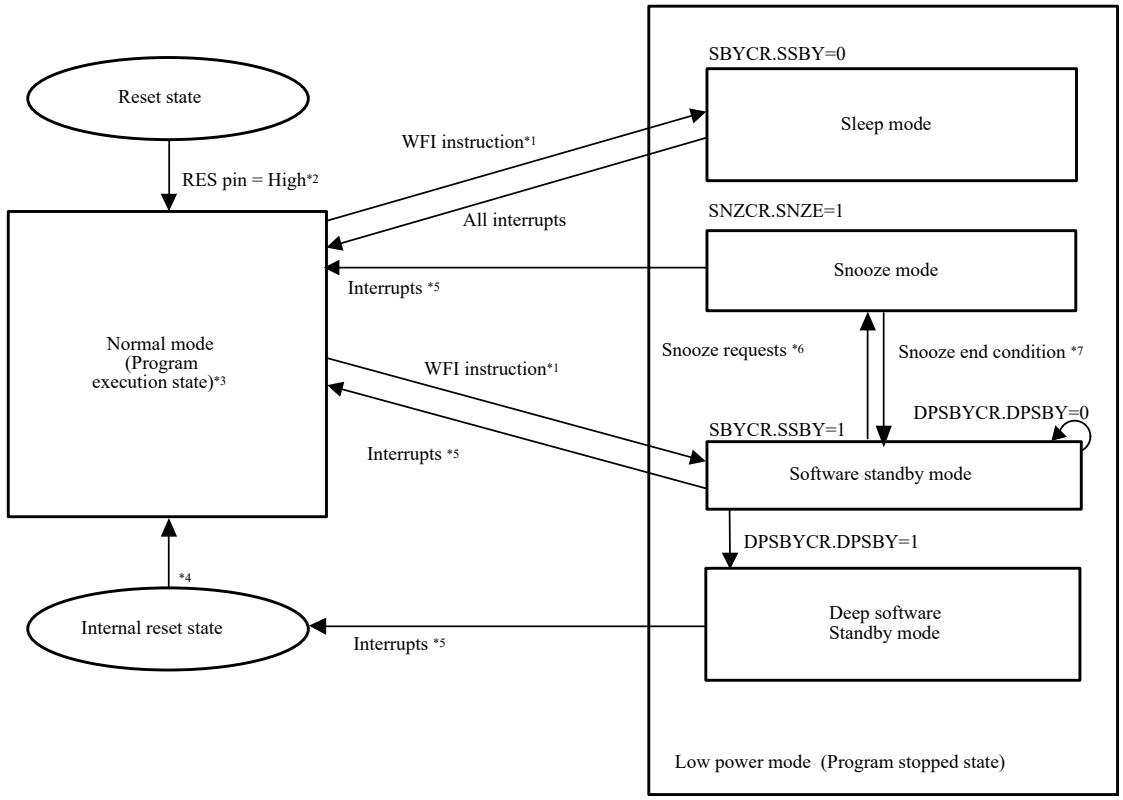
Note 8. If the DPSBYCR.DEEPCUT[1:0] bits are 00b, data in the Standby SRAM is retained in Deep Software Standby mode. When the DPSBYCR.DEEPCUT[1:0] bits are not 00b, data in the Standby SRAM is undefined in Deep Software Standby mode.

Note 9. Only VCOUT function is permitted. The VCOUT pin operates when ACMPHS uses no digital filter. For details on digital filter, see section 39, High-Speed Analog Comparator (ACMPHS).
Note 10. When using LVD in Deep Software Standby mode, DPSBYCR.DEEPCUT[1:0] bits must be 00b or 01b before entering Deep Software Standby mode.
Note 11. When the MCU enters Deep Software Standby mode with the DPSBYCR.DEEPCUT[1:0] bits set to 11b, the LVD circuit stops and the low-power function of the power-on reset circuit is enabled.
Note 12. AGT0 operation is possible when 100b (AGTLCLK) is selected by the AGT0.AGTMR1.TCK[2:0] bits. AGT1 operation is possible when 100b (AGTLCLK) or 101 (Underflow event signal from AGT0) is selected by the AGT1.AGTMR1.TCK[2:0] bits.
Note 13. When using the Programmable Gain Amplifiers, MSTPD16 must be set to 0. For details, see section 36.3.15. Programmable Gain Amplifier.

Table 10.3 Interrupt Source for canceling Snooze, Software Standby and Deep Software Standby Modes

Interrupt source	Name	Software Standby Mode	Snooze Mode	Deep Software Standby Mode
NMI		Yes	Yes	Yes
Port	PORT_IRQn (n = 0 to 15)	Yes	Yes	Yes*3
LVD	LVD_LVD1	Yes	Yes	Yes
	LVD_LVD2	Yes	Yes	Yes
IWDT	IWDT_NMIUNDF	Yes	Yes	No
KINT	KEY_INTKR	Yes	Yes	No
AGT1	AGT1_AGTI	Yes	Yes*2	No
	AGT1_AGTCMAI	Yes	Yes	No
	AGT1_AGTCMBI	Yes	Yes	No
IIC0	IIC0_WU	Yes	Yes	No
SCI0	SCI0_AM	No	Yes with SELSR0*1	No
DTC	DTC_COMPLETE	No	Yes with SELSR0*1 *2	No
DOC	DOC_DOPCI	No	Yes with SELSR0 *1	No

Note 1. To use the interrupt request as a trigger for exiting the Snooze mode, the request must be selected in SELSR0 . See section 12, Interrupt Controller Unit (ICU) for the setting of SELSR0. When a trigger selected in SELSR0 occurs after executing WFI instruction and during the transition from Normal mode to Software Standby mode, the request might or might not be accepted, depending on the timing of the occurrence.
Note 2. The event which is enabled by the SNZEDCR0 must not be used.
Note 3. IRQn-DS pin interrupt is available. IRQn pin interrupt is not available.



Note 1. When an interrupt that acts as a trigger for cancel is received during a transition to the program-stopped state after the execution of a WFI instruction, the MCU executes interrupt exception handling instead of transitioning to low power mode.
Note 2. The MOCO clock is the source of the operating clock following a transition from the reset state to Normal mode.
Note 3. The transition to Normal mode is made because of an interrupt from Sleep mode, Snooze mode, or Software Standby mode. The clock source is the same as before entering the low power mode.
Note 4. When an available interrupt request is generated, an internal reset (Deep Software Standby reset) is generated over a fixed period. Canceling of Deep Software Standby mode accompanies release from the internal reset state, and then the MCU transitions to Normal mode and executes reset exception processing with the MOCO clock as the source of the operating clock.
Note 5. See Table 10.3
Note 6. See Table 10.7
Note 7. See Table 10.9

Figure 10.1 Mode Transitions

10.2 Register Descriptions

10.2.1 LPMSAR : Low Power Mode Security Attribution Register

Base address: SYSC = 0x4001_E000

Offset address: 0x3C8

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	NONSEC9	NONSEC8	—	—	—	NONSEC4	—	NONSEC2	—	NONSEC0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	NONSEC0*1	Non Secure Attribute bit 0 Target register: OPCCR 0: Secure 1: Non Secure	R/W
1	—	This bit is read as 1. The write value should be 1.	R/W
2	NONSEC2	Non Secure Attribute bit 2 Target register: SBYCR 0: Secure 1: Non Secure	R/W
3	—	This bit is read as 1. The write value should be 1.	R/W
4	NONSEC4	Non Secure Attribute bit 4 Target register: SNZCR, SNZEDCR0, SNZREQCR0 0: Secure 1: Non Secure	R/W
7:5	—	These bits are read as 1. The write value should be 1.	R/W
8	NONSEC8	Non Secure Attribute bit 8 Target register: DPSBYCR 0: Secure 1: Non Secure	R/W
9	NONSEC9	Non Secure Attribute bit 9 Target register: DPSWCR 0: Secure 1: Non Secure	R/W
31:10	—	These bits are read as 1. The write value should be 1.	R/W

Note: Only secure access can write to this register. Both secure access and Non-secure read access are allowed but Non-secure write access is not allowed and TrustZone access error is not generated.

Note: This register is write-protected by PRCR register.

Note 1. It is recommended that these bits are configured as Non Secure when the device life cycle is NSECSD (DLMMON.DLMMON[3:0] = 0011b). See section 45.6.1. Restrictions on setting the security attribution for details.

The LPMSAR register controls the secure attribute of Low Power Mode registers.

NONSEC0 bit (Non Secure Attribute bit 0)

This bit controls the security attribute of OPCCR.

NONSEC2 bit (Non Secure Attribute bit 2)

This bit controls the security attribute of SBYCR.

NONSEC4 bit (Non Secure Attribute bit 4)

This bit controls the security attribute of SNZCR, SNZEDCR0, SNZREQCR0

NONSEC8 bit (Non Secure Attribute bit 8)

This bit controls the security attribute of DPSBYCR.

NONSEC9 bit (Non Secure Attribute bit 9)

This bit controls the security attribute of DPSWCR.

10.2.2 DPFSAR : Deep Software Standby Interrupt Factor Security Attribution Register

Base address: SYSC = 0x4001_E000

Offset address: 0x3E0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	DPFSA20	—	—	DPFSA17	DPFSA16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DPFSA15	DPFSA14	DPFSA13	DPFSA12	DPFSA11	DPFSA10	DPFSA9	DPFSA8	DPFSA7	DPFSA6	DPFSA5	DPFSA4	DPFSA3	DPFSA2	DPFSA1	DPFSA0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
7:0	DPFSA0 to DPFSA7	Deep Software Standby Interrupt Factor Security Attribute bit n (n = 0 to 7) Target register: DPSIER0.bn, DPSIFR0.bn, DPSIEGR0.bn (n = 0 to 7) Target factor : IRQn-DS Pin (n = 0 to 7) 0: Secure 1: Non Secure	R/W
15:8	DPFSA8 to DPFSA15	Deep Software Standby Interrupt Factor Security Attribute bit n (n = 8 to 15) Target register: DPSIER1.bn, DPSIFR1.bn, DPSIEGR1.bn (n = 0 to 7) Target factor : IRQn-DS Pin (n = 8 to 15) 0: Secure 1: Non Secure	R/W
16	DPFSA16	Deep Software Standby Interrupt Factor Security Attribute bit 16 Target register: DPSIER2.b0, DPSIFR2.b0, DPSIEGR2.b0 Target factor : LVD1 0: Secure 1: Non Secure	R/W
17	DPFSA17	Deep Software Standby Interrupt Factor Security Attribute bit 17 Target register: DPSIER2.b1, DPSIFR2.b1, DPSIEGR2.b1 Target factor : LVD2 0: Secure 1: Non Secure	R/W
19:18	—	These bits are read as 1. The write value should be 1.	R/W
20	DPFSA20	Deep Software Standby Interrupt Factor Security Attribute bit 20 Target register: DPSIER2.b4, DPSIFR2.b4, DPSIEGR2.b4 Target factor : NMI Pin 0: Secure 1: Non Secure	R/W
31:21	—	These bits are read as 1. The write value should be 1.	R/W

The DPFSAR register controls the secure attribute of Deep Software Standby Interrupt Factor control registers.

DPFSA bit (Deep Software Standby Interrupt Factor Security Attribute bit n (n = 0 to 7))

This bit controls the security attribute of DPSIER0.bn, DPSIFR0.bn, DPSIEGR0.bn (n = 0 to 7) .

Target factor is IRQn-DS Pin (n = 0 to 7).

DPFSA bit (Deep Software Standby Interrupt Factor Security Attribute bit n (n = 8 to 15))

This bit controls the security attribute of DPSIER1.bn, DPSIFR1.bn, DPSIEGR1.bn (n = 0 to 7) .

Target factor is IRQn-DS Pin (n = 8 to 15)

DPFSA16 bit (Deep Software Standby Interrupt Factor Security Attribute bit 16)

This bit controls the security attribute of DPSIER2.b0, DPSIFR2.b0, DPSIEGR2.b0 .

Target factor is LVD1.

DPFSA17 bits (Deep Software Standby Interrupt Factor Security Attribute bit 17)

This bit controls the security attribute of DPSIER2.b1, DPSIFR2.b1, DPSIEGR2.b1.

Target factor is LVD2.

DPFSA20 bit (Deep Software Standby Interrupt Factor Security Attribute bit 20)

This bit controls the security attribute of DPSIER2.b4, DPSIFR2.b4, DPSIEGR2.b4.

Target factor is NMI Pin.

10.2.3 SBYCR : Standby Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x00C

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

Bit	Symbol	Function	R/W
14:0	—	These bits are read as reset value.The write value should be reset value	R/W
15	SSBY	Software Standby Mode Select 0: Sleep mode 1: Software Standby mode.	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

SSBY bit (Software Standby Mode Select)

The SSBY bit specifies the transition destination after a WFI instruction is executed.

When the SSBY bit is set to 1, the MCU enters Software Standby mode after execution of a WFI instruction. When the MCU returns to Normal mode from Software Standby mode by an interrupt, the SSBY bit remains 1. The SSBY bit can be cleared by writing 0 to it.

While the OSTDCR.OSTDE bit is 1, setting of the SSBY bit is ignored. Even if SSBY bit is 1, the MCU enters Sleep mode on execution of a WFI instruction.

While the FENTRYR.FENTRYC bit is 1 setting of the SSBY bit is ignored. Even if SSBY bit is 1, the MCU enters Sleep mode on execution of a WFI instruction.

While the FENTRYR.FENTRYD bit is 1 setting of the SSBY bit is ignored. Even if SSBY bit is 1, the MCU enters Sleep mode on execution of a WFI instruction.

10.2.4 MSTPCRA : Module Stop Control Register A

Base address: MSTP = 0x4008_4000

Offset address: 0x000

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	MSTPA22	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	MSTPA7	—	—	—	—	—	—	MSTPA0
Value after reset:	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	0

Bit	Symbol	Function	R/W
0	MSTPA0	SRAM0 Module Stop Target module: SRAM0 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
6:1	—	These bits are read as 1. The write value should be 1.	R/W
7	MSTPA7	Standby SRAM Module Stop Target module: Standby SRAM 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
21:8	—	These bits are read as 1. The write value should be 1.	R/W
22	MSTPA22	DMA Controller/Data Transfer Controller Module Stop*1 Target module: DTC, DMAC 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
31:23	—	These bits are read as 1. The write value should be 1.	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note 1. When rewriting the MSTPA22 bit from 0 to 1, disable the DMAC and DTC before setting the MSTPA22 bit.

10.2.5 MSTPCRB : Module Stop Control Register B

Base address: MSTP = 0x4008_4000

Offset address: 0x004

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	MSTPB31	MSTPB30	MSTPB29	MSTPB28	MSTPB27	—	—	—	—	MSTPB22	—	—	MSTPB19	MSTPB18	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	MSTPB9	MSTPB8	—	—	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
7:0	—	This bit is read as 1. The write value should be 1.	R/W

Bit	Symbol	Function	R/W
8	MSTPB8	I2C Bus Interface 1 Module Stop*1 Target module: IIC1 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
9	MSTPB9	I2C Bus Interface 0 Module Stop*1 Target module: IIC0 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
17:10	—	These bits are read as 1. The write value should be 1.	R/W
18	MSTPB18	Serial Peripheral Interface 1 Module Stop*2 Target module: SPI1 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
19	MSTPB19	Serial Peripheral Interface 0 Module Stop*2 Target module: SPI0 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
21:20	—	These bits are read as 1. The write value should be 1.	R/W
22	MSTPB22	Serial Communication Interface 9 Module Stop*2 Target module: SCI9 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
26:23	—	These bits are read as 1. The write value should be 1.	R/W
27	MSTPB27	Serial Communication Interface 4 Module Stop*2 Target module: SCI4 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
28	MSTPB28	Serial Communication Interface 3 Module Stop*2 Target module: SCI3 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
29	MSTPB29	Serial Communication Interface 2 Module Stop*2 Target module: SCI2 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
30	MSTPB30	Serial Communication Interface 1 Module Stop*2 Target module: SCI1 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
31	MSTPB31	Serial Communication Interface 0 Module Stop*2 Target module: SCI0 0: Cancel the module-stop state 1: Enter the module-stop state	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note 1. The MSTPBi bit must be written while the oscillation of the clock controlled by this bit is stabilized. For entering Software Standby mode after writing the MSTPBi bit, wait for two IICCLK cycles after writing, and then execute a WFI instruction (i = 8, 9).

Note 2. The MSTPBi bit must be written while the oscillation of the clock controlled by this bit is stabilized. For entering Software Standby mode after writing the MSTPBi bit, wait for two SCISPICLK cycles after writing, and then execute a WFI instruction (i = 18, 19, 22, and 27 to 31).

10.2.6 MSTPCRC : Module Stop Control Register C

Base address: MSTP = 0x4008_4000

Offset address: 0x008

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	MSTPC31	—	—	—	MSTPC27	—	—	—	—	—	MSTPC21	MSTPC20	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	MSTPC14	MSTPC13	—	—	—	—	—	—	—	—	—	—	—	MSTPC1	MSTPC0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	MSTPC0	Clock Frequency Accuracy Measurement Circuit Module Stop*1 Target module: CAC 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
1	MSTPC1	Cyclic Redundancy Check Module Stop Target module: CRC 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
12:2	—	These bits are read as 1. The write value should be 1.	R/W
13	MSTPC13	Data Operation Circuit Module Stop Target module: DOC 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
14	MSTPC14	Event Link Controller Module Stop Target module: ELC 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
19:15	—	These bits are read as 1. The write value should be 1.	R/W
20	MSTPC20	Trigonometric Function Unit Module Stop Target module: TFU 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
21	MSTPC21	IIR Filter Accelerator Module Stop Target module: IIRFA 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
26:22	—	These bits are read as 1. The write value should be 1.	R/W
27	MSTPC27	CANFD Module Stop*2 Target module: CANFD 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
30:28	—	These bits are read as 1. The write value should be 1.	R/W
31	MSTPC31	Secure Cryptographic Engine Module Stop Target module: SCE5 0: Cancel the module-stop state 1: Enter the module-stop state	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note 1. The MSTPC0 bit must be written while the oscillation of the clock to be controlled by this bit is stable. To enter Software Standby mode after writing this bit, wait for 2 cycles of the slowest clock from the clocks output by the oscillators, then execute a WFI instruction.

Note 2. The MSTPC27 bit must be written while the oscillation of the clock controlled by this bit is stabilized. For entering Software Standby mode after writing the MSTPC27 bit, wait for two CANFD clock (CANFDCLK) cycles after writing, and then execute a WFI instruction.

10.2.7 MSTPCRD : Module Stop Control Register D

Base address: MSTP = 0x4008_4000

Offset address: 0x00C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	MSTPD28	MSTPD27	MSTPD26	MSTPD25	—	—	MSTPD22	—	MSTPD20	MSTPD19	—	—	MSTPD16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	MSTPD14	MSTPD13	MSTPD12	MSTPD11	—	—	—	—	—	—	—	MSTPD3	MSTPD2	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
1:0	—	These bits are read as 1. The write value should be 1.	R/W
2	MSTPD2	Low Power Asynchronous General Purpose Timer 1 Module Stop*1 Target module: AGT1 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
3	MSTPD3	Low Power Asynchronous General Purpose Timer 0 Module Stop*2 Target module: AGT0 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
10:4	—	These bits are read as 1. The write value should be 1.	R/W
11	MSTPD11	Port Output Enable for GPT Group D Module Stop Target module: POEGGD 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
12	MSTPD12	Port Output Enable for GPT Group C Module Stop Target module: POEGGC 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
13	MSTPD13	Port Output Enable for GPT Group B Module Stop Target module: POEGGB 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
14	MSTPD14	Port Output Enable for GPT Group A Module Stop Target module: POEGGA 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
15	—	This bit is read as 1. The write value should be 1.	R/W
16	MSTPD16	A/D Converter Module Stop Target module: ADC 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
18:17	—	These bits are read as 1. The write value should be 1.	R/W

Bit	Symbol	Function	R/W
19	MSTPD19	12-bit D/A Converter 1 Module Stop Target module: DAC121 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
20	MSTPD20	12-bit D/A Converter 0 Module Stop Target module: DAC120 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
21	—	This bit is read as 1. The write value should be 1.	R/W
22	MSTPD22	Temperature Sensor Module Stop Target module: TSN 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
24:23	—	These bits are read as 1. The write value should be 1.	R/W
25	MSTPD25	High-Speed Analog Comparator 3 Module Stop Target module: ACMPHS3 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
26	MSTPD26	High-Speed Analog Comparator 2 Module Stop Target module: ACMPHS2 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
27	MSTPD27	High-Speed Analog Comparator 1 Module Stop Target module: ACMPHS1 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
28	MSTPD28	High-Speed Analog Comparator 0 Module Stop Target module: ACMPHS0 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
31:29	—	These bits are read as 1. The write value should be 1.	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note 1. When the count source is LOCO, AGT1 counting does not stop even if MSTPD2 is set to 1. If the count source is LOCO, this bit must be set to 1 except when accessing the AGT1 registers.

Note 2. When the count source is LOCO, AGT0 counting does not stop even if MSTPD3 is set to 1. If the count source is LOCO, this bit must be set to 1 except when accessing the AGT0 registers.

10.2.8 MSTPCRE : Module Stop Control Register E

Base address: MSTP = 0x4008_4000

Offset address: 0x010

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	MSTPE31	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	MSTPE4	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 1. The write value should be 1.	R/W
4	MSTPE4	Key Interrupt Function Module Stop Target module: KINT 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
30:5	—	These bits are read as 1. The write value should be 1.	R/W
31	MSTPE31	General PWM Timer and PWM Delay Generation Circuit Module Stop*1 Target module: GPT, PDG 0: Cancel the module-stop state 1: Enter the module-stop state	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note 1. The MSTPE31 bit must be written while the oscillation of the clock controlled by this bit is stabilized. For entering Software Standby mode after writing the MSTPE31 bit, wait for two GPTCLK cycles after writing, and then execute a WFI instruction.

10.2.9 OPCCR : Operating Power Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x0A0

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	OPCMTSF	—	—	OPCM[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	OPCM[1:0]	Operating Power Control Mode Select 0 0: High-speed mode 0 1: Setting prohibited 1 0: Setting prohibited 1 1: Low-speed mode	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	OPCMTSF	Operating Power Control Mode Transition Status Flag 0: Transition completed 1: During transition	R
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

The OPCCR register is used to reduce power consumption in Normal and Sleep modes by specifying a lower operating frequency. For the procedure to change the operating power control modes, see section 10.5. Function for Lower Operating Power Consumption.

When transitioning from Software Standby mode to Normal or Snooze mode, the settings in the OPCCR.OPCM[1:0] bits are as follows, regardless of their settings before entering Software Standby mode:

- OPCCR.OPCM[1:0] = 00b (High-speed mode)

If Software Standby mode is canceled before the transition to Software Standby completes, the OPCCR.OPCM[1:0] bits retain their settings from before the WFI instruction is executed. If this causes any problem, set the MCU to High-speed mode during the exception handling procedure when canceling Software Standby mode.

OPCM[1:0] bits (Operating Power Control Mode Select)

The OPCM[1:0] bits select the operating power control mode in Normal and Sleep modes. Table 10.4 shows the relationship between the operating power control modes and the OPCM[1:0] settings.

OPCMTSF flag (Operating Power Control Mode Transition Status Flag)

The OPCMTSF flag indicates the switching control state when the operating power control mode is switched. This flag becomes 1 when the OPCM bit is written, and 0 when mode transition completes. Read this flag and confirm that it is 0 before proceeding.

Table 10.4 Operating power control mode

Operating power control mode	OPCM[1:0] bits	Power consumption
High-speed mode	00b	High
Low-speed mode	11b	Low

For details about the operating frequency range, see section 46, Electrical Characteristics.

Each operating power control mode is described below.

- High-speed mode
  After a reset cancellation, the MCU is activated in this mode.
- Low-speed mode
  The following constraints apply in low-speed mode:
  - Programming and erasure operations for the flash memory are prohibited
  - Using the PLL or PLL2 is prohibited. See section 10.10.1. Register Access

  In this mode, lower power consumption is possible than in High-speed mode when the same operation is performed under the same conditions, such as operating frequency.

10.2.10 SNZCR : Snooze Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x092

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

Bit	Symbol	Function	R/W
0	RXDREQEN	RXD0 Snooze Request Enable 0: Ignore RXD0 falling edge in Software Standby mode 1: Detect RXD0 falling edge in Software Standby mode	R/W
1	SNZDTCEN	DTC Enable in Snooze mode 0: Disable DTC operation 1: Enable DTC operation	R/W
6:2	—	These bits are read as 0. The write value should be 0.	R/W
7	SNZE	Snooze mode Enable 0: Disable Snooze mode 1: Enable Snooze mode	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

RXDREQEN bit (RXD0 Snooze Request Enable)

The RXDREQEN bit specifies whether to detect a falling edge of the RXD0 pin in Software Standby mode. This bit can be used only when SCI0 is operating in asynchronous mode. To detect a falling edge of the RXD0 pin, set this bit before entering Software Standby mode. When this bit is set to 1, a falling edge of the RXD0 pin in Software Standby mode causes the MCU to enter Snooze mode.

SNZDTCEN bit (DTC Enable in Snooze mode)

The SNZDTCEN bit specifies whether to use the DTC and SRAM in Snooze mode. To use the DTC and SRAM in Snooze mode, set this bit to 1 before entering Software Standby mode. When this bit is set to 1, the DTC can be activated by setting IELSRn register.

SNZE bit (Snooze mode Enable)

The SNZE bit specifies whether to enable a transition from Software Standby mode to Snooze mode. To use Snooze mode, set this bit to 1 before entering Software Standby mode. When this bit is set to 1, a trigger as shown in Table 10.7 in Software Standby mode causes the MCU to enter Snooze mode. After the MCU transitions from Software Standby mode or Snooze mode to Normal mode, set 0 to the SNZE bit once then set it before re-entering Software Standby mode. For details, see section 10.8. Snooze Mode.

10.2.11 SNZEDCR0 : Snooze End Control Register 0

Base address: SYSC = 0x4001_E000

Offset address: 0x094

Bit position:	7	6	5	4	3	2	1	0
Bit field:	SCI0UMTED	—	—	—	—	DTCNZRED	DTCZRED	AGTUNFED
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	AGTUNFED	AGT1 Underflow Snooze End Enable 0: Disable the snooze end request 1: Enable the snooze end request	R/W
1	DTCZRED	Last DTC Transmission Completion Snooze End Enable 0: Disable the snooze end request 1: Enable the snooze end request	R/W
2	DTCNZRED	Not Last DTC Transmission Completion Snooze End Enable 0: Disable the snooze end request 1: Enable the snooze end request	R/W
6:3	—	These bits are read as 0. The write value should be 0.	R/W
7	SCI0UMTED	SCI0 Address Mismatch Snooze End Enable 0: Disable the snooze end request 1: Enable the snooze end request	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

The SNZEDCR0 register controls the condition of switching from Snooze mode to Software Standby mode. In order to use a trigger shown in Table 10.8 as a condition to switch from Snooze mode to Software Standby mode, the corresponding bitin the SNZEDCR0 register must be set to 1.

The event that is used to return from snooze mode to normal mode as shown in Table 10.3 must not be enabled in the SNZEDCR0 register.

AGTUNFED bit (AGT1 Underflow Snooze End Enable)

The AGTUNFED bit specifies whether to enable a transition from Snooze mode to Software Standby mode on an AGT1 underflow. For details on the trigger conditions, see section 23, Low Power Asynchronous General Purpose Timer (AGTW).

DTCZRED bit (Last DTC Transmission Completion Snooze End Enable)

The DTCZRED bit specifies whether to enable a transition from Snooze mode to Software Standby mode on completion of the last DTC transmission, that is, when CRA or CRB registers in the DTC is 0. For details on the trigger conditions, see section 16, Data Transfer Controller (DTC).

DTCNZRED bit (Not Last DTC Transmission Completion Snooze End Enable)

The DTCNZRED bit specifies whether to enable a transition from Snooze mode to Software Standby mode on completion of each DTC transmission, that is, when CRA or CRB registers in the DTC is not 0. For details on the trigger conditions, see section 16, Data Transfer Controller (DTC).

SCI0UMTED bit (SCI0 Address Mismatch Snooze End Enable)

The SCI0UMTED bit specifies whether to enable a transition from Snooze mode to Software Standby mode on an SCI0 event when an address received in Software Standby mode does not match the expected data. For details on the trigger conditions, see section 26, Serial Communications Interface (SCI). Only set this bit to 1 when SCI0 operates in asynchronous mode.

10.2.12 SNZREQCR0 : Snooze Request Control Register 0

Base address: SYSC = 0x4001_E000

Offset address: 0x098

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	SNZREQEN30	SNZREQEN29	SNZREQEN28	—	—	—	—	—	—	—	—	—	—	SNZREQEN17	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SNZREQEN15	SNZREQEN14	SNZREQEN13	SNZREQEN12	SNZREQEN11	SNZREQEN10	SNZREQEN9	SNZREQEN8	SNZREQEN7	SNZREQEN6	SNZREQEN5	SNZREQEN4	SNZREQEN3	SNZREQEN2	SNZREQEN1	SNZREQEN0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SNZREQEN0	Enable IRQ0 pin snooze request 0: Disable the snooze request 1: Enable the snooze request	R/W
1	SNZREQEN1	Enable IRQ1 pin snooze request 0: Disable the snooze request 1: Enable the snooze request	R/W
2	SNZREQEN2	Enable IRQ2 pin snooze request 0: Disable the snooze request 1: Enable the snooze request	R/W
3	SNZREQEN3	Enable IRQ3 pin snooze request 0: Disable the snooze request 1: Enable the snooze request	R/W
4	SNZREQEN4	Enable IRQ4 pin snooze request 0: Disable the snooze request 1: Enable the snooze request	R/W
5	SNZREQEN5	Enable IRQ5 pin snooze request 0: Disable the snooze request 1: Enable the snooze request	R/W

Bit	Symbol	Function	R/W
6	SNZREQEN6	Enable IRQ6 pin snooze request 0: Disable the snooze request 1: Enable the snooze request	R/W
7	SNZREQEN7	Enable IRQ7 pin snooze request 0: Disable the snooze request 1: Enable the snooze request	R/W
8	SNZREQEN8	Enable IRQ8 pin snooze request 0: Disable the snooze request 1: Enable the snooze request	R/W
9	SNZREQEN9	Enable IRQ9 pin snooze request 0: Disable the snooze request 1: Enable the snooze request	R/W
10	SNZREQEN10	Enable IRQ10 pin snooze request 0: Disable the snooze request 1: Enable the snooze request	R/W
11	SNZREQEN11	Enable IRQ11 pin snooze request 0: Disable the snooze request 1: Enable the snooze request	R/W
12	SNZREQEN12	Enable IRQ12 pin snooze request 0: Disable the snooze request 1: Enable the snooze request	R/W
13	SNZREQEN13	Enable IRQ13 pin snooze request 0: Disable the snooze request 1: Enable the snooze request	R/W
14	SNZREQEN14	Enable IRQ14 pin snooze request 0: Disable the snooze request 1: Enable the snooze request	R/W
15	SNZREQEN15	Enable IRQ15 pin snooze request 0: Disable the snooze request 1: Enable the snooze request	R/W
16	—	This bit is read as 0. The write value should be 0.	R/W
17	SNZREQEN17	Enable Key Interrupt snooze request 0: Disable the snooze request 1: Enable the snooze request	R/W
27:18	—	These bits are read as 0. The write value should be 0.	R/W
28	SNZREQEN28	Enable AGT1 underflow snooze request 0: Disable the snooze request 1: Enable the snooze request	R/W
29	SNZREQEN29	Enable AGT1 compare match A snooze request 0: Disable the snooze request 1: Enable the snooze request	R/W
30	SNZREQEN30	Enable AGT1 compare match B snooze request 0: Disable the snooze request 1: Enable the snooze request	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

The SNZREQCR0 register controls which trigger causes the MCU to switch from Software Standby mode to Snooze mode. If a trigger is selected as a request to cancel Software Standby mode by setting the WUPENn register, see section 12, Interrupt Controller Unit (ICU), the MCU enters Normal mode when the trigger is generated while the associated bit of the SNZREQCR0 is 1. The setting of the WUPENn register always has higher priority than the setting of the SNZREQCR0 register. For details, see section 10.8. Snooze Mode and section 12, Interrupt Controller Unit (ICU).

10.2.13 DPSBYCR : Deep Software Standby Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x400

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DPSBY	IOKEEP	—	—	—	—	DEEPCUT[1:0]	
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
1:0	DEEPCUT[1:0]	Power-Supply Control 0 0: Power to the standby RAM and Low-speed on-chip oscillator is supplied in Deep Software Standby mode. 0 1: Power to the standby RAM and Low-speed on-chip oscillator is not supplied in Deep Software Standby mode. 1 0: Setting prohibited 1 1: Power to the standby RAM and Low-speed on-chip oscillator is not supplied in Deep Software Standby mode. In addition, LVD is disabled and the low power function in a power-on reset circuit is enabled.	R/W
5:2	—	These bits are read as 0. The write value should be 0.	R/W
6	IOKEEP	I/O Port Rentention 0: When the Deep Software Standby mode is canceled, the I/O ports are in the reset state. 1: When the Deep Software Standby mode is canceled, the I/O ports are in the same state as in the Deep Software Standby mode.	R/W
7	DPSBY	Deep Software Standby 0: Sleep mode (SBYCR.SSBY=0) / Software Standby mode (SBYCR.SSBY=1) 1: Sleep mode (SBYCR.SSBY=0) / Deep Software Standby mode (SBYCR.SSBY=1)	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

The DPSBYCR register controls the Deep Software Standby mode.

DPSBYCR is not initialized by the internal reset signal that is the source to cancel the Deep Software Standby mode. For details, see section 5, Resets.

DEEPCUT[1:0] bit (Power-Supply Control)

The DEEPCUT[1:0] bits control the internal power supply to the standby RAM and Low-speed on-chip oscillator in Deep Software Standby mode. In addition, these bits control the state of LVD and power-on reset circuit in Deep Software Standby mode.

When an LVD interrupt is used in Deep Software Standby mode, the DEEPCUT[1:0] bits must be set to 00b or 01b.

For lower power consumption, set the DEEPCUT[1:0] bits to 11b so that the LVD is stopped and the low power consumption function of the power-on reset circuit is enabled.

Regardless of the DEEPCUT[1:0] bit setting, during Deep Software Standby mode, internal power supply to SRAM other than standby SRAM is stopped.

When a Deep Software Standby mode is used, set DPSWCR.WTSTS bits depending on the value of DEEPCUT[1] before entering Deep Software Standby mode.

IOKEEP bit (I/O Port Rentention)

In Deep Software Standby mode, I/O ports keep the same states as in the Software Standby mode. The IOKEEP bit specifies whether to reset the state of the I/O ports or not when the Deep Software Standby mode is canceled.

DPSBY bit (Deep Software Standby)

The DPSBY bit controls transitions to Deep Software Standby mode.

When the WFI instruction is executed while SBYCR.SSBY bit and DPSBYCR.DPSBY bit are both 1, the MCU enters Deep Software Standby mode through Software Standby mode.

The DPSBY bit remains 1 when Deep Software Standby mode is canceled by certain pins which are sources of external pin interrupts (NMI, IRQn-DS (n = 0 to 15)) or a peripheral interrupt (voltage monitor 1, or voltage monitor 2). Write 0 to this bit to clear it.

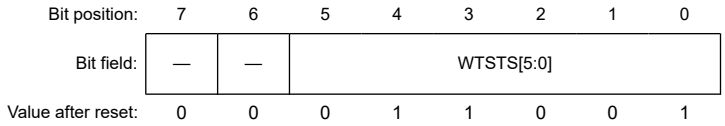
The DPSBY bit setting is invalid when OFS0.IWDTSTPCTL bit is 0 (counting continues) regardless of the setting in OFS0.IWDTSTRT bit. In that case, even when SBYCR.SSBY bit is 1 and the DPSBY bit is 1, the transition after the execution of a WFI instruction is to Software Standby mode.

The setting of the DPSBY bit is invalid when voltage monitor 1 reset is enabled (LVD1CR0.RI = 1) or when a voltage monitor 2 reset is enabled (LVD2CR0.RI = 1). In this case, even when the SBYCR.SSBY bit is 1 and the DPSBY bit is 1, the transition after the execution of a WFI instruction is to Software Standby mode.

10.2.14 DPSWCR : Deep Software Standby Wait Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x401



Bit	Symbol	Function	R/W
5:0	WTSTS[5:0]	Deep Software Wait Standby Time Setting Bit 0x0E: Wait cycle for fast recovery 0x19: Wait cycle for slow recovery Others: Setting prohibited	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

The DPSWCR register sets the wait stabilization time when a Deep Software Standby mode is canceled by certain pins that are the sources of external pin interrupts or a peripheral interrupt.

During a wait stabilization period set in this register, a Deep Software Standby reset occurs, and the MCU is initialized.

The DPSWCR register is not initialized with the internal reset signal by the cancellation of the Deep Software Standby mode. For details, see section 5, Resets.

When a Deep Software Standby mode is used, set DPSWCR.WTSTS bits according to the value of DPSBYCR.DEEPCUT[1] before entering Deep Software Standby mode.

When DPSBYCR.DEEPCUT[1]=0, you can set DPSWCR.WTSTS to the wait cycle for fast recovery.

When DPSBYCR.DEEPCUT[1]=1, you must set DPSWCR.WTSTS to the wait cycle for slow recovery.

10.2.15 DPSIER0 : Deep Software Standby Interrupt Enable Register 0

Base address: SYSC = 0x4001_E000

Offset address: 0x402

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DIRQ7E	DIRQ6E	DIRQ5E	DIRQ4E	DIRQ3E	DIRQ2E	DIRQ1E	DIRQ0E
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DIRQ0E	IRQ0-DS Pin Enable 0: Cancelling Deep Software Standby mode is disabled 1: Cancelling Deep Software Standby mode is enabled	R/W
1	DIRQ1E	IRQ1-DS Pin Enable 0: Cancelling Deep Software Standby mode is disabled 1: Cancelling Deep Software Standby mode is enabled	R/W
2	DIRQ2E	IRQ2-DS Pin Enable 0: Cancelling Deep Software Standby mode is disabled 1: Cancelling Deep Software Standby mode is enabled	R/W
3	DIRQ3E	IRQ3-DS Pin Enable 0: Cancelling Deep Software Standby mode is disabled 1: Cancelling Deep Software Standby mode is enabled	R/W
4	DIRQ4E	IRQ4-DS Pin Enable 0: Cancelling Deep Software Standby mode is disabled 1: Cancelling Deep Software Standby mode is enabled	R/W
5	DIRQ5E	IRQ5-DS Pin Enable 0: Cancelling Deep Software Standby mode is disabled 1: Cancelling Deep Software Standby mode is enabled	R/W
6	DIRQ6E	IRQ6-DS Pin Enable 0: Cancelling Deep Software Standby mode is disabled 1: Cancelling Deep Software Standby mode is enabled	R/W
7	DIRQ7E	IRQ7-DS Pin Enable 0: Cancelling Deep Software Standby mode is disabled 1: Cancelling Deep Software Standby mode is enabled	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

DPSIER0 is not initialized by the internal reset signal used as Deep Software Standby mode Cancelling source. For details, see section 5, Resets.

After the setting of DPSIER0 is modified, an edge may be internally generated depending on the state of the pin, resulting in DPSIFR0 being set to 1. Therefore, DPSIFR0 should be cleared to 0 before entering Deep Software Standby mode.

10.2.16 DPSIER1 : Deep Software Standby Interrupt Enable Register 1

Base address: SYSC = 0x4001_E000

Offset address: 0x403

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DIRQ15E	DIRQ14E	DIRQ13E	DIRQ12E	DIRQ11E	DIRQ10E	DIRQ9E	DIRQ8E
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DIRQ8E	IRQ8-DS Pin Enable 0: Cancelling Deep Software Standby mode is disabled 1: Cancelling Deep Software Standby mode is enabled	R/W
1	DIRQ9E	IRQ9-DS Pin Enable 0: Cancelling Deep Software Standby mode is disabled 1: Cancelling Deep Software Standby mode is enabled	R/W
2	DIRQ10E	IRQ10-DS Pin Enable 0: Cancelling Deep Software Standby mode is disabled 1: Cancelling Deep Software Standby mode is enabled	R/W
3	DIRQ11E	IRQ11-DS Pin Enable 0: Cancelling Deep Software Standby mode is disabled 1: Cancelling Deep Software Standby mode is enabled	R/W
4	DIRQ12E	IRQ12-DS Pin Enable 0: Cancelling Deep Software Standby mode is disabled 1: Cancelling Deep Software Standby mode is enabled	R/W
5	DIRQ13E	IRQ13-DS Pin Enable 0: Cancelling Deep Software Standby mode is disabled 1: Cancelling Deep Software Standby mode is enabled	R/W
6	DIRQ14E	IRQ14-DS Pin Enable 0: Cancelling Deep Software Standby mode is disabled 1: Cancelling Deep Software Standby mode is enabled	R/W
7	DIRQ15E	IRQ15-DS Pin Enable 0: Cancelling Deep Software Standby mode is disabled 1: Cancelling Deep Software Standby mode is enabled	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

DPSIER1 is not initialized by the internal reset signal used as Deep Software Standby mode Cancelling source. For details, see section 5, Resets.

After the setting of DPSIER1 is modified, an edge may be internally generated depending on the state of the pin, resulting in DPSIFR1 being set to 1. Therefore, DPSIFR1 should be cleared to 0 before entering Deep Software Standby mode.

10.2.17 DPSIER2 : Deep Software Standby Interrupt Enable Register 2

Base address: SYSC = 0x4001_E000

Offset address: 0x404

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	DNMIE	—	—	DLVD2IE	DLVD1IE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DLVD1IE	LVD1 Deep Software Standby Cancel Signal Enable 0: Cancelling Deep Software Standby mode is disabled 1: Cancelling Deep Software Standby mode is enabled	R/W
1	DLVD2IE	LVD2 Deep Software Standby Cancel Signal Enable 0: Cancelling Deep Software Standby mode is disabled 1: Cancelling Deep Software Standby mode is enabled	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
4	DNMIE	NMI Pin Enable 0: Cancelling Deep Software Standby mode is disabled 1: Cancelling Deep Software Standby mode is enabled	R/W*1
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.
Note 1. 1 can be written only once. Once 1 is written to this bit, subsequent write accesses are disabled.

DPSIER2 is not initialized by the internal reset signal used as Deep Software Standby mode Cancelling source. For details, see section 5, Resets.

After the setting of DPSIER2 is modified, an edge may be internally generated depending on the state of the pin, resulting in DPSIFR2 being set to 1. Therefore, DPSIFR2 should be cleared to 0 before entering Deep Software Standby mode.

10.2.18 DPSIFR0 : Deep Software Standby Interrupt Flag Register 0

Base address: SYSC = 0x4001_E000

Offset address: 0x406

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DIRQ7F	DIRQ6F	DIRQ5F	DIRQ4F	DIRQ3F	DIRQ2F	DIRQ1F	DIRQ0F
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DIRQ0F	IRQ0-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W
1	DIRQ1F	IRQ1-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W
2	DIRQ2F	IRQ2-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W
3	DIRQ3F	IRQ3-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W
4	DIRQ4F	IRQ4-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W
5	DIRQ5F	IRQ5-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W
6	DIRQ6F	IRQ6-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W
7	DIRQ7F	IRQ7-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note: Writing 0 clears the flag. Writing 1 is ignored.

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

Each flag is set to 1 when a cancel request specified by DPSIEGR0 is generated.

Each flag may be set to 1 when a cancel request is generated in any mode (not only in Deep Software Standby mode) or when the setting of DPSIER0 is modified. Therefore, a transition to Deep Software Standby mode should be made after DPSIFR0 is cleared to 0x00.

To clear DPSIFR0 to 0x00 after modifying DPSIER0, wait for at least 6 PCLKB cycles, read DPSIFR0, and then write 0 to DPSIFR0. Six or more PCLKB cycles can be secured, for example, by reading DPSIER0.

DPSIFR0 is not initialized by the internal reset signal used as Deep Software Standby mode Cancelling source. To clear DPSIFR0 to 0x00 after modifying DPSIER0, wait for at least 6 PCLKB cycles, read DPSIFR0, and then write 0 to DPSIFR0. Six or more PCLKB cycles can be secured, for example, by reading DPSIER0. For details, see section 5, Resets.

DIRQnF flag (IRQn-DS Pin Deep Software Standby Cancel Flag) (n = 0 to 7)

The DIRQnF flag indicates that a cancel request by the IRQn-DS pin has been generated.

[Setting condition]

A cancel request by the IRQn-DS pin specified by DPSIEGR0 is generated.

[Clearing condition]

Writing 0 to each flag after 1 is read.

10.2.19 DPSIFR1 : Deep Software Standby Interrupt Flag Register 1

Base address: SYSC = 0x4001_E000

Offset address: 0x407

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DIRQ15F	DIRQ14F	DIRQ13F	DIRQ12F	DIRQ11F	DIRQ10F	DIRQ9F	DIRQ8F
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DIRQ8F	IRQ8-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W
1	DIRQ9F	IRQ9-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W
2	DIRQ10F	IRQ10-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W
3	DIRQ11F	IRQ11-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W
4	DIRQ12F	IRQ12-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W
5	DIRQ13F	IRQ13-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W
6	DIRQ14F	IRQ14-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W
7	DIRQ15F	IRQ15-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W

Note: If the security attribution is configured as secure:

- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:

- Secure and Non-secure access are allowed.

Note: Writing 0 clears the flag. Writing 1 is ignored.
Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

Each flag is set to 1 when a cancel request specified by DPSIEGR1 is generated.

Each flag may be set to 1 when a cancel request is generated in any mode (not only in Deep Software Standby mode) or when the setting of DPSIER1 is modified. Therefore, a transition to Deep Software Standby mode should be made after DPSIFR1 is cleared to 0x00.

To clear DPSIFR1 to 0x00 after modifying DPSIER1, wait for at least 6 PCLKB cycles, read DPSIFR1, and then write 0 to DPSIFR1. Six or more PCLKB cycles can be secured, for example, by reading DPSIER1.

DPSIFR1 is not initialized by the internal reset signal used as Deep Software Standby mode Cancelling source. For details, see section 5, Resets.

DIRQnF flag (IRQn-DS Pin Deep Software Standby Cancel Flag) (n = 8 to 15)

The DIRQnF flag indicates that a cancel request by the IRQn-DS pin has been generated.

[Setting condition]

A cancel request by the IRQn-DS pin specified by DPSIEGR1 is generated.

[Clearing condition]

Writing 0 to each flag after 1 is read.

10.2.20 DPSIFR2 : Deep Software Standby Interrupt Flag Register 2

Base address: SYSC = 0x4001_E000

Offset address: 0x408

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	DNMIF	—	—	DLVD2IF	DLVD1IF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DLVD1IF	LVD1 Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W
1	DLVD2IF	LVD2 Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	DNMIF	NMI Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: If the security attribution is configured as secure:

- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:

- Secure and Non-secure access are allowed.

Note: Writing 0 clears the flag. Writing 1 is ignored.
Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

Each flag is set to 1 when a cancel request specified by DPSIEGR2 is generated.

Each flag may be set to 1 when a cancel request is generated in any mode (not only in Deep Software Standby mode) or when the setting of DPSIER2 is modified. Therefore, a transition to Deep Software Standby mode should be made after DPSIFR2 is cleared to 0x00.

To clear DPSIFR2 to 0x00 after modifying DPSIER2, wait for at least 6 PCLKB cycles, read DPSIFR2, and then write 0 to DPSIFR2. Six or more PCLKB cycles can be secured, for example, by reading DPSIER2.

DPSIFR2 is not initialized by the internal reset signal used as Deep Software Standby mode Cancelling source. For details, see section 5, Resets.

DLVDmIF flag (LVDm Deep Software Standby Cancel Flag) (m = 1 to 2)

The DLVDmIF flag indicates that a cancel request by the voltage monitor m signal has been generated.

[Setting condition]

A cancel request is generated by the voltage monitor m signal that is selected in DPSIEGR2.

[Clearing condition]

Writing 0 to each flag after 1 is read.

DNMIF flag (NMI Pin Deep Software Standby Cancel Flag)

This flag indicates that a cancel request by the NMI pin has been generated.

[Setting condition]

A cancel request by the NMI pin specified by DPSIEGR2 is generated

[Clearing condition]

Writing 0 to each flag after 1 is read.

10.2.21 DPSIEGR0 : Deep Software Standby Interrupt Edge Register 0

Base address: SYSC = 0x4001_E000

Offset address: 0x40A

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DIRQ7EG	DIRQ6EG	DIRQ5EG	DIRQ4EG	DIRQ3EG	DIRQ2EG	DIRQ1EG	DIRQ0EG
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DIRQ0EG	IRQ0-DS Pin Edge Select 0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
1	DIRQ1EG	IRQ1-DS Pin Edge Select 0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
2	DIRQ2EG	IRQ2-DS Pin Edge Select 0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
3	DIRQ3EG	IRQ3-DS Pin Edge Select 0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
4	DIRQ4EG	IRQ4-DS Pin Edge Select 0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
5	DIRQ5EG	IRQ5-DS Pin Edge Select 0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
6	DIRQ6EG	IRQ6-DS Pin Edge Select 0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W

Bit	Symbol	Function	R/W
7	DIRQ7EG	IRQ7-DS Pin Edge Select 0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

DPSIEGR0 is not initialized by the internal reset signal that is the source to cancel the Deep Software Standby mode. For details, see section 5, Resets.

10.2.22 DPSIEGR1 : Deep Software Standby Interrupt Edge Register 1

Base address: SYSC = 0x4001_E000

Offset address: 0x40B

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DIRQ15EG	DIRQ14EG	DIRQ13EG	DIRQ12EG	DIRQ11EG	DIRQ10EG	DIRQ9EG	DIRQ8EG
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DIRQ8EG	IRQ8-DS Pin Edge Select 0: A cancel request is generated at a falling edge. 1: A cancel request is generated at a rising edge.	R/W
1	DIRQ9EG	IRQ9-DS Pin Edge Select 0: A cancel request is generated at a falling edge. 1: A cancel request is generated at a rising edge.	R/W
2	DIRQ10EG	IRQ10-DS Pin Edge Select 0: A cancel request is generated at a falling edge. 1: A cancel request is generated at a rising edge	R/W
3	DIRQ11EG	IRQ11-DS Pin Edge Select 0: A cancel request is generated at a falling edge. 1: A cancel request is generated at a rising edge.	R/W
4	DIRQ12EG	IRQ12-DS Pin Edge Select 0: A cancel request is generated at a falling edge. 1: A cancel request is generated at a rising edge.	R/W
5	DIRQ13EG	IRQ13-DS Pin Edge Select 0: A cancel request is generated at a falling edge. 1: A cancel request is generated at a rising edge.	R/W
6	DIRQ14EG	IRQ14-DS Pin Edge Select 0: A cancel request is generated at a falling edge. 1: A cancel request is generated at a rising edge.	R/W
7	DIRQ15EG	IRQ15-DS Pin Edge Select 0: A cancel request is generated at a falling edge. 1: A cancel request is generated at a rising edge.	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

DPSIEGR1 is not initialized by the internal reset signal that is the source to cancel the Deep Software Standby mode. For details, see section 5, Resets.

10.2.23 DPSIEGR2 : Deep Software Standby Interrupt Edge Register 2

Base address: SYSC = 0x4001_E000

Offset address: 0x40C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	DNMIEG	—	—	DLVD2EG	DLVD1EG
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DLVD1EG	LVD1 Edge Select 0: A cancel request is generated when VCC < V_{det1} (fall) is detected 1: A cancel request is generated when VCC ≥ V_{det1} (rise) is detected	R/W
1	DLVD2EG	LVD2 Edge Select 0: A cancel request is generated when VCC < V_{det2} (fall) is detected 1: A cancel request is generated when VCC ≥ V_{det2} (rise) is detected	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	DNMIEG	NMI Pin Edge Select 0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

DPSIEGR2 is not initialized by the internal reset signal that is the source to cancel the Deep Software Standby mode. For details, see section 5, Resets.

10.2.24 SYOCDCR : System Control OCD Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x040E

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

Bit	Symbol	Function	R/W
0	DOCDF	Deep Software Standby OCD flag 0: DBIRQ is not generated 1: DBIRQ is generated	R/W*1
6:1	—	These bits are read as 0. The write value should be 0.	R/W
7	DBGEN	Debugger Enable bit Set to 1 first in on-chip debug mode. 0: On-chip debugger is disabled 1: On-chip debugger is enabled	R/W

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

Note 1. Writing 0 clears the flag. Writing 1 is ignored

This register is not controlled by any security attribute register (eg. LPMSAR, DPFSAR).

SYOCDCR can be written when DBGSTR.CDBGPWRUPREQ = 1 (the debugger is connected).

SYOCDCR is not initialized by the internal reset signal that is the source to cancel the Deep Software Standby mode.

DOCDF flag (Deep Software Standby OCD flag)

DOCDF flag indicates that a cancel request of Deep Software Standby mode by the MCUCTRL.DBIRQ bit has been generated. DOCDF flag is set to 1 when a cancel request is generated. This flag may be set to 1 when a cancel request is generated in any mode (not only in Deep Software Standby mode). Therefore, a transition to Deep Software Standby mode must be made after DOCDF flag is cleared to 0.

[Setting condition]

- A cancel request by the MCUCTRL.DBIRQ is generated

[Clearing condition]

- Writing 0 to the flag after reading the bit as 1
- When DBGEN bit is 0

DBGEN bit (Debugger Enable bit)

The DBGEN bit enables the on-chip debug mode. This bit must be set to 1 first in the on-chip debugger mode.

[Setting condition]

- Writing 1 to the bit when the debugger is connected.

[Clearing condition]

- Power-on reset is generated
- Writing 0 to the bit.

Note: Certain restrictions apply in terms of the MCU states in which the DBGEN bit can be set to 1. For details, see section 2.13.2. Restrictions on Connecting an OCD emulator.

10.3 Reducing Power Consumption by Switching Clock Signals

The clock frequency changes when the SCKDIVCR register is set.

For information on module and clock associations, see section 8.2.2. SCKDIVCR : System Clock Division Control Register.

10.4 Module-Stop Function

The module stop function can stop the clock supply set for each peripheral module.

When the MSTPmi bit (m = A to E, i = 31 to 0) in MSTPCRn (n = A to E) is set to 1, the specified module stops operating and enters the module-stop state, but the CPU continues to operate independently. Setting the MSTPmi bit to 0 cancels the module-stop state, allowing the module to resume operation at the end of the bus cycle.

After a reset is canceled, all modules other than the DMAC, DTC and SRAMn modules are placed in the module-stop state. Do not access the module while the corresponding MSTPmi bit is 1. Additionally, do not set 1 to the MSTPmi bit while the corresponding module is accessed.

When the PLL is selected as the clock source, MSTPmi bits must be changed only one bit at a time. In this case, wait at least 250 ns after changing each MSTPmi bit before starting subsequent processing if you change any of the following bits:

MSTPD16 (ADC), MSTPC31 (SCE5).

The recommended method to measure the wait time is through software. Be sure to consider the worst-case conditions to ensure that the required wait time elapses.

10.5 Function for Lower Operating Power Consumption

By selecting an appropriate operating power consumption control mode according to the operating frequency, power consumption can be reduced in Normal mode, Sleep mode, and Snooze mode.

10.5.1 Setting Operating Power Control Mode

Ensure the operating condition such as the frequency range is always within the specified range before and after switching the operating power control modes.

This section provides example procedures for switching operating power control modes.

Table 10.5 Available oscillators in each mode

Mode	Oscillator					
	PLL, PLL2	High-speed on-chip oscillator	Middle-speed on-chip oscillator	Low-speed on-chip oscillator	Main clock oscillator	IWDT-dedicated on-chip oscillator
High-speed	Available	Available	Available	Available	Available	Available
Low-speed	N/A	Available	Available	Available	Available	Available

(1) Switching from a higher power mode to a lower power mode

Example 1: From High-speed mode to Low-speed mode:

(Operation begins in High-speed mode)

1. Change the oscillator to what is used in Low-speed mode. Set the frequency of each clock lower than or equal to the maximum operating frequency in Low-speed mode.
2. Turn off the oscillator that is not required in Low-speed mode.
3. Confirm that the OPCCR.OPCMTSF flag is 0 (indicates transition completed).
4. Set the OPCCR.OPCM[1:0] bits to 11b (Low-speed mode).
5. Confirm that OPCCR.OPCMTSF flag is 0 (indicates transition completed).

(Operation is now in Low-speed mode)

(2) Switching from a lower power mode to a higher power mode

Example 1: From Low-speed mode to High-speed mode

(Operation begins in Low-speed mode)

1. Confirm that the OPCCR.OPCMTSF flag is 0 (indicates transition completed).
2. Set the OPCCR.OPCM[1:0] bits to 00b (High-speed mode).
3. Confirm that the OPCCR.OPCMTSF flag is 0 (indicates transition completed).
4. Turn on any required oscillator in High-speed mode.
5. Set the frequency of each clock lower than or equal to the maximum operating frequency for High-speed mode.

(Operation is now in High-speed mode)

10.6 Sleep Mode

10.6.1 Transitioning to Sleep Mode

When a WFI instruction is executed while SBYCR.SSBY bit is 0, the MCU enters Sleep mode. In this mode, the CPU stops operating but the contents of its internal registers are retained. Other peripheral functions do not stop. Available resets or interrupts in Sleep mode cause the MCU to cancel Sleep mode. All interrupt sources are available. If using an interrupt to cancel Sleep mode, you must set the associated IELSRn register before executing a WFI instruction. For details, see section 12, Interrupt Controller Unit (ICU).

Counting by IWDT stops when the MCU enters Sleep mode while the IWDT is in auto start mode and the OFS0.IWDTSTPCTL bit is 1 (IWDT stops in Sleep mode, Software Standby mode, or Snooze mode).

Counting by IWDT continues when the MCU enters Sleep mode while the IWDT is in auto start mode and the OFS0.IWDTSTPCTL bit is 0 (IWDT does not stop in Sleep mode, Software Standby mode, or Snooze mode).

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

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

10.6.2 Canceling Sleep Mode

Sleep mode is canceled by:

- An interrupt
- A RES pin reset
- A power-on reset
- A voltage monitor reset
- An SRAM parity error reset
- An SRAM ECC error reset
- A bus master MPU error reset
- A TrustZone error reset
- A reset caused by an IWDT or a WDT underflow

The operations are as follows:

1. Canceling by an interrupt
   When an interrupt request is generated, Sleep mode is canceled and the MCU starts the interrupt handling.
2. Canceling by RES pin reset
   When the RES pin is driven low, the MCU enters the reset state. Be sure to keep the RES pin low for the time period specified in section 46, Electrical Characteristics. When the RES pin is driven high after the specified time period, the CPU starts the reset exception handling.
3. Canceling by IWDT reset
   - Sleep mode is canceled by an internal reset generated by an IWDT underflow and the MCU starts the reset exception handling. However, IWDT stops in Sleep mode and an internal reset for canceling Sleep mode is not generated in the following conditions:
   - OFS0.IWDTSTRT = 0 and OFS0.IWDTSTPCTL = 1.
4. Canceling by WDT reset
   Sleep mode is canceled by an internal reset generated by a WDT underflow and the MCU starts the reset exception handling. However, WDT stops in Sleep mode even when counting in Normal mode and an internal reset for canceling Sleep mode is not generated in the following conditions:
   - OFS0.WDTSTRT = 0 (auto start mode) and OFS0.WDTSTPCTL = 1
   - OFS0.WDTSTRT = 1 (register start mode) and WDTCSTPR.SLCSTP = 1.
5. Canceling by other resets available in Sleep mode
   Sleep mode is canceled by other resets and the MCU starts the reset exception handling.

Note: For details on proper setting of the interrupts, see section 12, Interrupt Controller Unit (ICU).

10.7 Software Standby Mode

10.7.1 Transitioning to Software Standby Mode

When a WFI instruction is executed while SBYCR.SSBY bit is 1 and DPSBYCR.DPSBY bit is 0, the MCU enters Software Standby mode. In this mode, the CPU, most of the on-chip peripheral functions and the oscillators stop. However, the contents of the CPU internal registers and SRAM data, the states of on-chip peripheral functions and the I/O ports are retained. Software Standby mode allows significant reduction in power consumption because most of the oscillators stops in this mode. Table 10.2 shows the status of each on-chip peripheral functions and oscillators. Available resets or interrupts in Software Standby mode make the MCU to cancel Software Standby mode. See Table 10.3 for available interrupt sources and section 12.2.17. WUPEN0 : Wake Up Interrupt Enable Register 0 for information on waking up the MCU from

Software Standby mode. If using an interrupt to cancel an interrupt, you must set the associated IELSRn register before executing a WFI instruction. For details, see section 12, Interrupt Controller Unit (ICU).

Clear DMAST.DMST bit and DTCST.DTCST bit to 0 before executing WFI instruction except when using DTC in Snooze mode. If DTC is required in Snooze mode, set DTCST.DTCST bit to 1 before executing a WFI instruction.

Counting by the IWDT stops if the MCU enters Software Standby mode while the IWDT is in auto start mode and the OFS0.IWDTSTPCTL bit is 1 (IWDT stops in Sleep, Software Standby or Snooze mode).

Counting by the IWDT continues if the MCU enters Software Standby mode while the IWDT is in auto start mode and the OFS0.IWDTSTPCTL bit is 0 (IWDT does not stop in Sleep, Software Standby or Snooze mode).

WDT stops counting when the MCU enters Software Standby mode because the PCLKB stops.

Do not enter Software Standby mode while OSTDCR.OSTDE = 1 (oscillation stop detection function is enabled). To enter Software Standby mode, execute a WFI instruction after disabling the oscillation stop detection function (OSTDCR.OSTDE = 0). In case of executing WFI instruction while OSTDCR.OSTDE = 1, the MCU enters Sleep mode even if SBYCR.SSBY = 1.

Do not enter Software Standby mode while the flash memory is programming or erasing. To enter Software Standby mode, execute a WFI instruction after programming or erasing procedure completes.

When the PLL is selected as the clock source, set the following modules to the module-stop state before executing a WFI instruction:

ADC, SCE5.

In this case, wait for least 750 ns and if the ICLK frequency before executing the WFI instruction exceeds 120 MHz, it is necessary to set the ICLK frequency division ratio to 1/2 and wait 5 µs. The recommended method to measure the wait time is through software. Be sure to consider the worst-case conditions to ensure that the required wait time elapses.

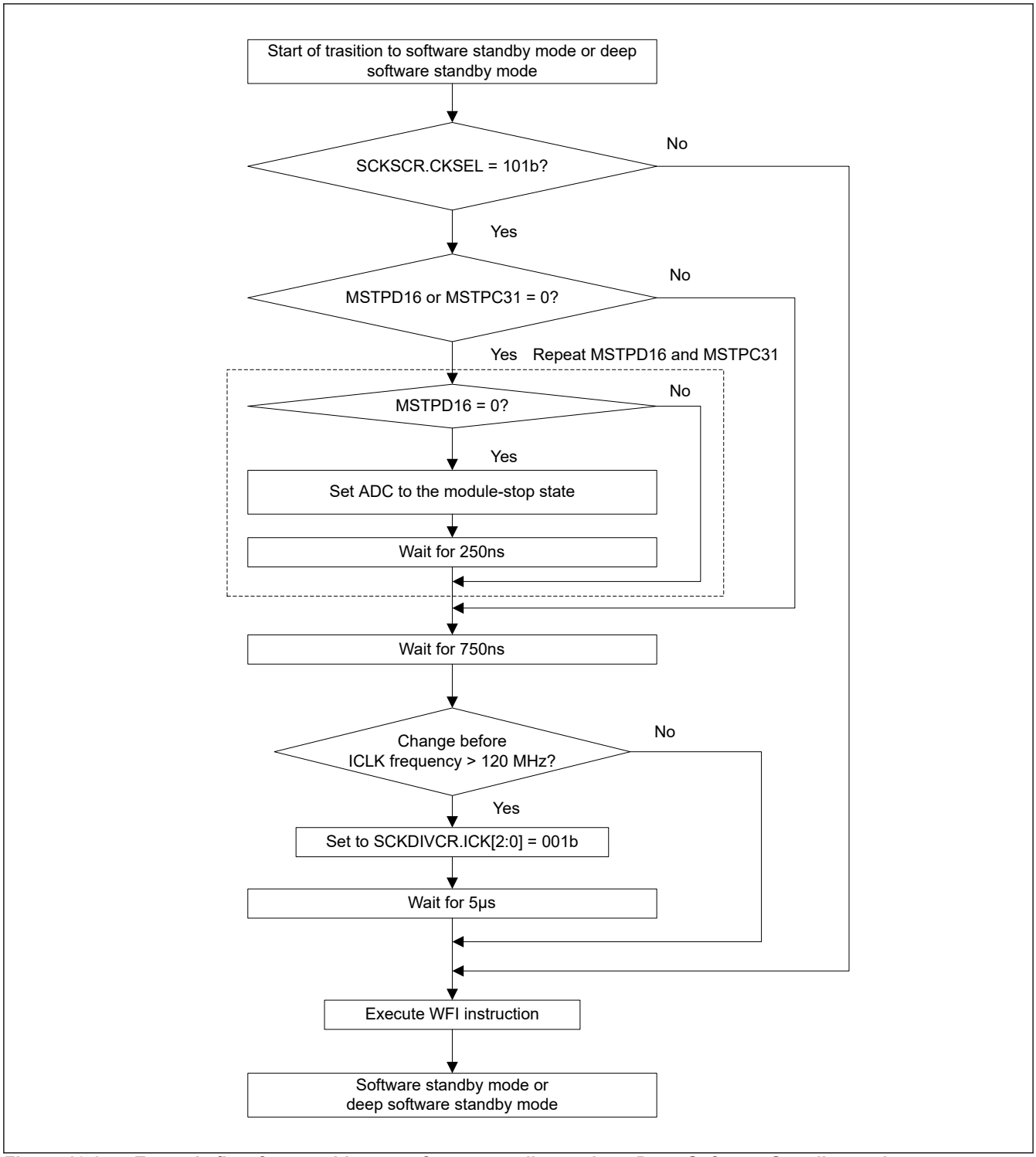


Figure 10.2 Example flow for transition to software standby mode or Deep Software Standby mode

Table 10.6 shows the setting of the related control bits and the modes to enter after executing WFI instruction.

Table 10.6 Bit settings that affect modes when executing a WFI instruction (1 of 2)

		SBYCR.SSBY and PSBYCR.DPSBY bit settings			
		SSBY = 0, DPSBY = 0	SSBY = 0, DPSBY = 1	SSBY = 1, DPSBY = 0	SSBY = 1, DPSBY = 1
OSTDCR.OSTDE	0	Sleep	Sleep	Software Standby	Deep Software Standby
	1			Sleep	Sleep

Table 10.6 Bit settings that affect modes when executing a WFI instruction (2 of 2)

		SBYCR.SSBY and PSBYCR.DPSBY bit settings			
		SSBY = 0, DPSBY = 0	SSBY = 0, DPSBY = 1	SSBY = 1, DPSBY = 0	SSBY = 1, DPSBY = 1
FENTRYR.FENTRYC FENTRYR.FENTRYD	0	Sleep	Sleep	Software Standby	Deep Software Standby
	1			Sleep	Sleep
OFS0.IWDTSTPCTL	0	Sleep	Sleep	Software Standby	Software Standby
	1				Deep Software Standby
LVD1CR0.RI	0	Sleep	Sleep	Software Standby	Deep Software Standby
	1				Software Standby
LVD2CR0.RI	0	Sleep	Sleep	Software Standby	Deep Software Standby
	1				Software Standby

10.7.2 Canceling Software Standby Mode

Software Standby mode is canceled by:

- An available interrupt shown in Table 10.3
- A RES pin reset
- A power-on reset
- A voltage monitor reset
- A reset caused by an IWDT underflow.

On exiting Software Standby mode, the oscillators that operate before the transition to the mode restart. After all the oscillators are stabilized, the MCU returns to Normal mode from Software Standby mode. See section 12.2.17. WUPEN0 : Wake Up Interrupt Enable Register 0 for information on how to wake up the MCU from Software Standby mode.

You can cancel Software Standby mode in any of the following ways:

1. Canceling by an interrupt
   When an available interrupt request (see Table 10.3) is generated, an oscillator that operates before the transition to Software Standby mode restarts. After all the oscillators are stabilized, the MCU returns to Normal mode from Software Standby mode and starts the interrupt handling.
   When the PLL is selected as the clock source, you must insert a wait time of at least 250 ns at the beginning of the interrupt handling. The recommended method to measure the wait time is through software.
   Be sure to consider the worst-case conditions to ensure that the required wait time elapses.

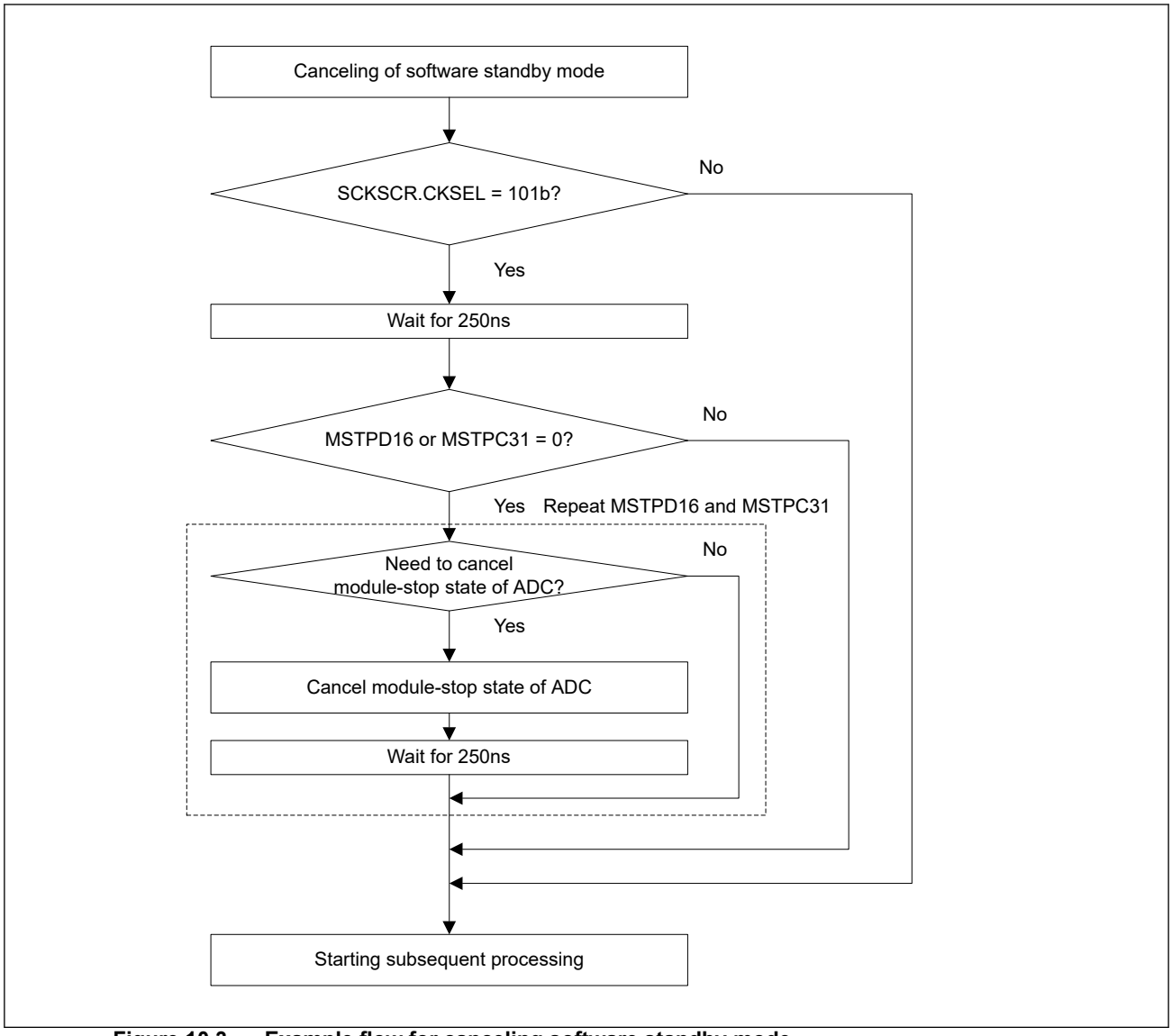


Figure 10.3 Example flow for canceling software standby mode

2. Canceling by a RES pin reset
   When the RES pin is driven low, the MCU enters the reset state, and the oscillators whose default status is operating, start the oscillation. Be sure to keep the RES pin low for the time period specified in section 46, Electrical Characteristics. When the RES pin is driven high after the specified time period, the CPU starts the reset exception handling.
3. Canceling by a power-on reset
   Software Standby mode is canceled by a power-on reset and the MCU starts the reset exception handling.
4. Canceling by a voltage monitor reset
   Software Standby mode is canceled by a voltage monitor reset from the voltage detection circuit and the MCU starts the reset exception handling.
5. Canceling by IWDT reset
   Software Standby mode is canceled by an internal reset generated by an IWDT underflow and the MCU starts the reset exception handling. However, IWDT stops in Software Standby mode and an internal reset for canceling Software Standby mode is not generated in the following condition:
   - OFS0.IWDTSTRT = 0 and OFS0.IWDTSTPCTL = 1.

10.7.3 Example of Software Standby Mode Application

Figure 10.4 shows an example of entry to Software Standby mode on detection of a falling edge of the IRQn pin, and exit from Software Standby mode by a rising edge of the IRQn pin.

In this example, an IRQn pin interrupt is accepted with the IRQCRi.IRQMD[1:0] bits of the ICU set to 00b (falling edge) in Normal mode, and the IRQCRi.IRQMD[1:0] bits are set to 01b (rising edge). After that, the SBYCR.SSBY bit is set to 1 and a WFI instruction is executed. As a result, entry to Software Standby mode completes and exit from Software Standby mode is initiated by a rising edge of the IRQn pin.

Setting the ICU is also required to exit Software Standby mode. For details, see section 12, Interrupt Controller Unit (ICU). The oscillation stabilization time in Figure 10.4 is specified in section 46, Electrical Characteristics.

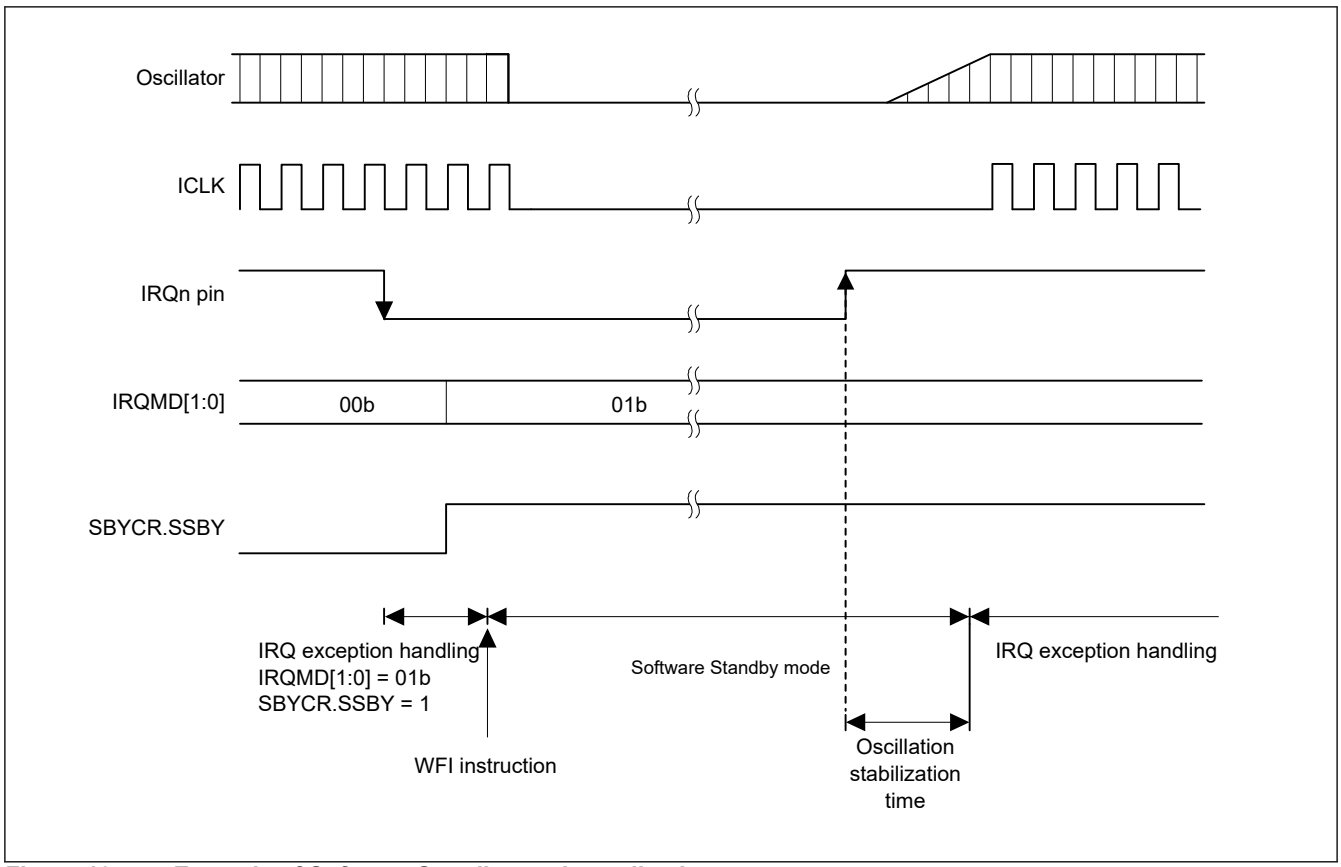


Figure 10.4 Example of Software Standby mode application

10.8 Snooze Mode

10.8.1 Transition to Snooze Mode

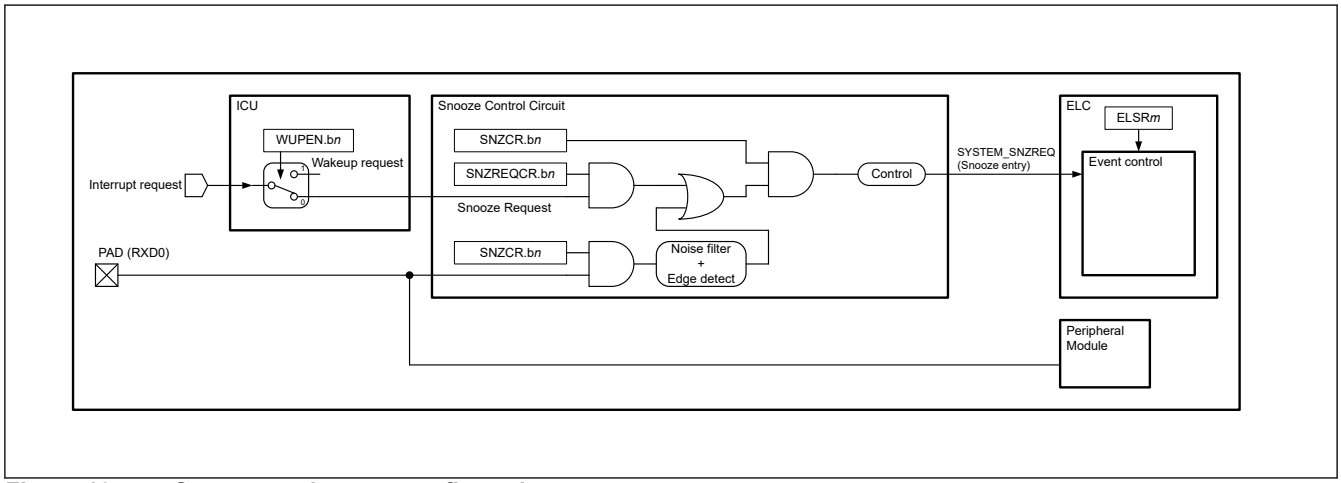


Figure 10.5 Snooze mode entry configuration

When the snooze control circuit accepts an available snooze request in Software Standby mode, the MCU transfers to Snooze mode. In this mode, some peripheral modules operates without waking the CPU. The peripheral modules that can operate in Snooze mode are shown in Table 10.2. Also, DTC operation in Snooze mode can be selected by the setting of SNZCR.SNZDTCEN bit.

Table 10.7 shows the Snooze requests that switch the MCU from Software Standby mode to Snooze mode. To use the listed Snooze requests as a trigger to switch to Snooze mode, the corresponding SNZREQENn bit of the SNZREQCR0 register or RXDREQEN bit of SNZCR register must be set before entering Software Standby mode.

Table 10.7 Available snooze requests to switch to Snooze mode

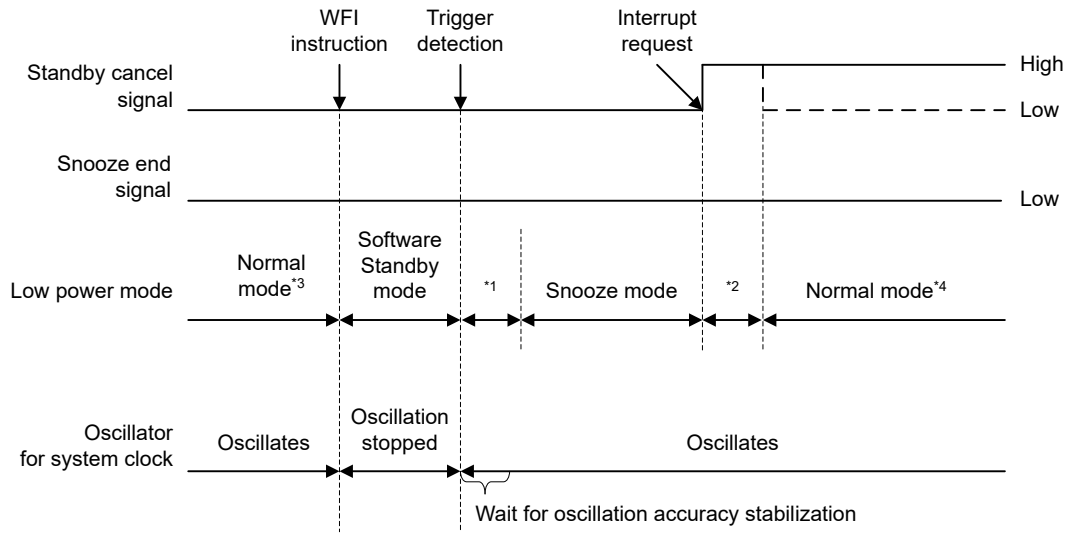
Snooze request	Control Register	
	Register	Bit*1
PORT_IRQn (n = 0 to 15)	SNZREQCR0	SNZREQENn (n = 0 to 15)
KEY_INTKR	SNZREQCR0	SNZREQEN17
AGT1_AGTI	SNZREQCR0	SNZREQEN28
AGT1_AGTCMAI	SNZREQCR0	SNZREQEN29
AGT1_AGTCMBI	SNZREQCR0	SNZREQEN30
RXD0 falling edge	SNZCR	RXDREQEN*2

Note 1. Do not enable multiple snooze requests at the same time.
Note 2. Do not set the RXDREQEN bit to 1 except in asynchronous mode.

Clear the DMAST.DMST and DTCST.DTCST bits to 0 before executing a WFI instruction, except when using the DTC in Snooze mode. If the DTC is required in Snooze mode, set the DTCST.DTCST bit to 1 before executing a WFI instruction.

10.8.2 Canceling Snooze Mode

Snooze mode is canceled by an interrupt request that is available in Software Standby mode or a reset. Table 10.3 shows the requests that can be used to exit each mode. After canceling the Snooze mode, the MCU enters Normal mode and proceeds with exception processing for the given interrupt or reset. The action triggered by the interrupt requests, selected in SELSR0, cancels Snooze mode. Interrupt canceling Snooze mode must be selected in IELSRn to link to the NVIC for the corresponding interrupt handling. See section 12, Interrupt Controller Unit (ICU) for information on SELSR0 and IELSRn registers.



Note 1. Transition time from Software Standby mode to Snooze mode.
Note 2. Transition time from Snooze mode to Normal mode.
Note 3. Enable Snooze mode (SNZCR.SNZE = 1) immediately before switching to Software Standby mode.
Note 4. Disable Snooze mode (SNZCR.SNZE = 0) immediately after returning to Normal mode.

Figure 10.6 Canceling of Snooze mode when an interrupt request signal is generated

10.8.3 Returning from Snooze Mode to Software Standby Mode

Table 10.8 shows the snooze end request that can be used as triggers to return to Software Standby mode. The snooze end requests are available only in Snooze mode. If the requests are generated when the MCU is not in Snooze mode, they are ignored. When multiple requests are selected, each of the requests invokes transition to Software Standby mode from Snooze mode.

Table 10.9 shows the snooze end conditions that consist of the snooze end requests and the conditions of the peripheral modules. The SCI0 and DTC modules can keep the MCU in Snooze mode until they complete the operation. However, an AGTn (n = 1) underflow as a trigger to return to Software Standby mode cancels Snooze mode without waiting for the completion of SCI0 operation.

Figure 10.7 shows the timing diagram for the transition from Snooze mode to Software Standby mode. This mode transition occurs according to which snooze end requests are set in the SNZEDCR0 register. A snooze request is cleared automatically after returning to Software Standby mode.

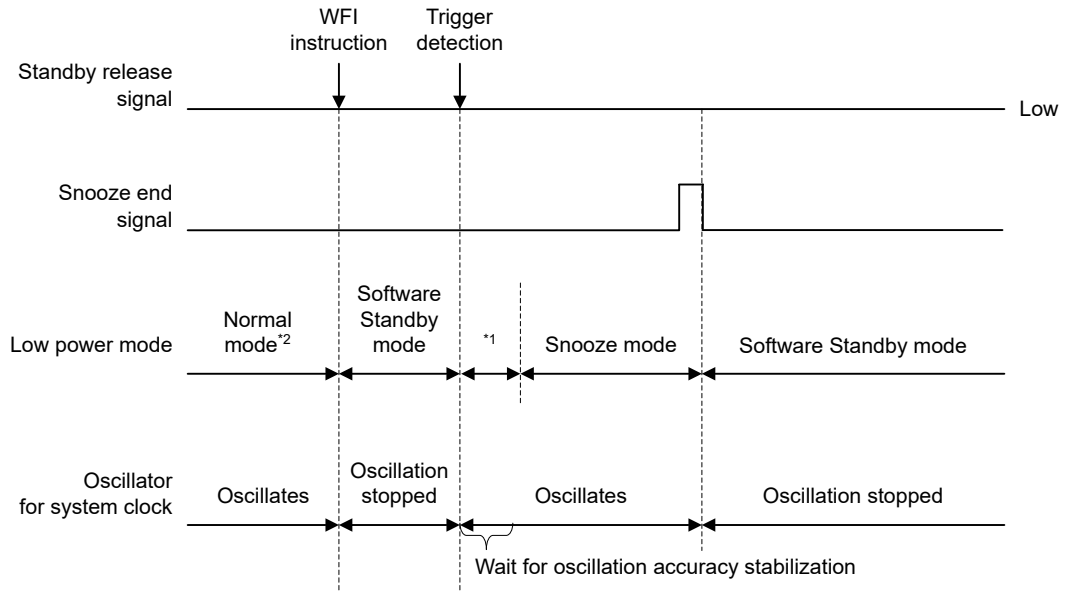
Table 10.8 Available snooze end requests (triggers to return to Software Standby mode)

Peripheral Module	Snooze end request	Enable/Disable Control	
		Register	Bit
AGT1	AGT1 underflow (AGT1_AGTI)	SNZEDCR0	AGTUNFED
DTC	Last DTC transmission completion (DTC_COMPLETE)	SNZEDCR0	DTCZRED
DTC	Not Last DTC Transmission Completion (DTC_TRANSFER)	SNZEDCR0	DTCNZRED
SCI0	SCI0 address mismatch (SCI0_DCUF)	SNZEDCR0	SCI0UMTED

Table 10.9 Snooze end conditions

Operating module when a snooze end request occurs	Snooze end request	
	AGT1 underflow	Other than AGT1 underflow
DTC	The MCU transitions to the Software Standby mode after DTC complete operation.	The MCU transitions to the Software Standby mode after DTC and SCI0 complete the operation.
SCI0	The MCU transitions to the Software Standby mode immediately after the snooze end request is generated.	
Other than specified	The MCU transitions to the Software Standby mode immediately after a snooze end request is generated.	

Note: If the DTC is used to activate the SCI, the MCU transitions to Software Standby mode immediately after a snooze end request is generated.

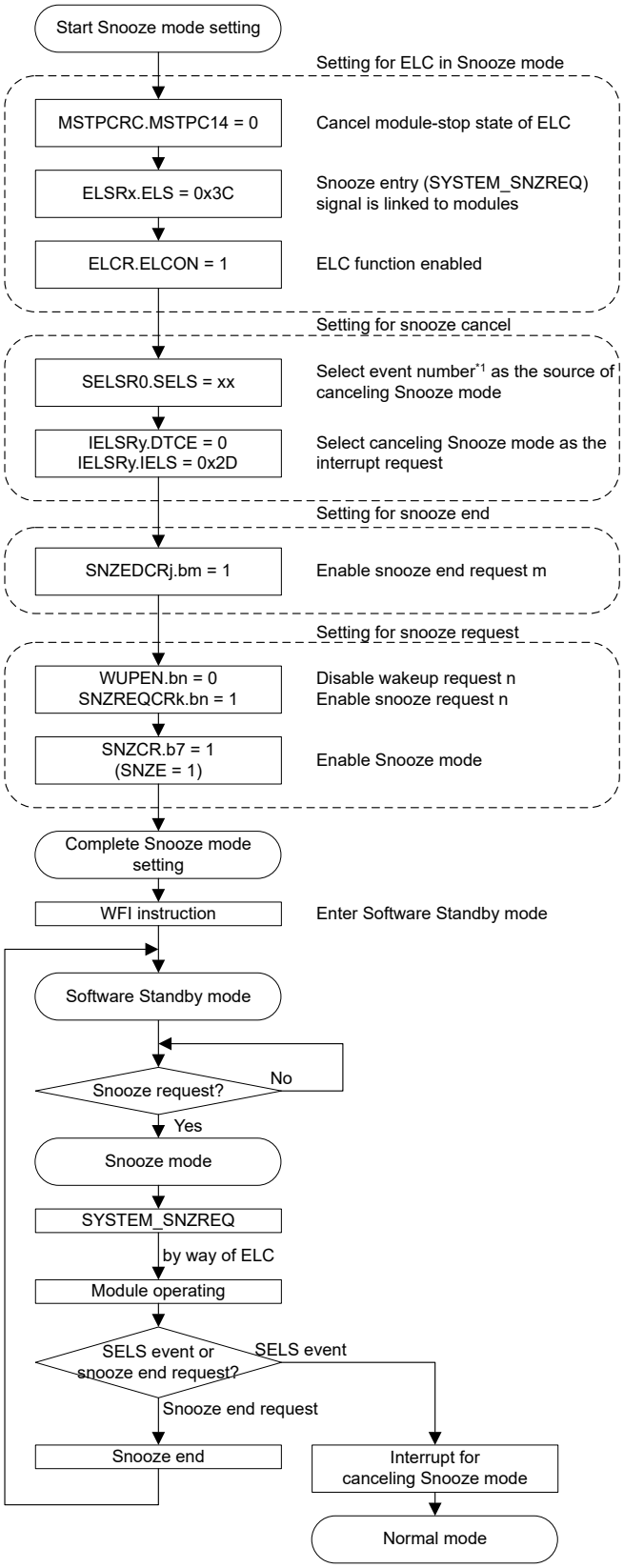


Note 1. Transition time from Software Standby mode to Snooze mode.
Note 2. Enable Snooze mode (SNZCR.SNZE = 1) immediately before switching to Software Standby mode.

Figure 10.7 Canceling of Snooze mode when an interrupt request signal is not generated

10.8.4 Snooze Operation Example

Figure 10.8 shows an example setting for using ELC in Snooze mode.



Note 1. See Table 12.4

Figure 10.8 Setting example of using ELC in Snooze mode

The MCU can transmit and receive data in SCI0 asynchronous mode without CPU intervention. When using the SCI0 in Snooze mode, use either High-speed mode or Low-speed mode.

Table 10.10 shows the maximum transfer rate of SCI0 in Snooze mode.

Table 10.10 HOCO: ± 1.4% (Ta = -20°C to 105°C) (Unit: bps)

Maximum division ratio of ICLK, PCLKA, PCLKB, PCLKC, PCLKD, FCLK, and TRCLK	HOCO frequency					
	LOCO is not operating			LOCO is operating		
	16 MHz	18 MHz	20 MHz	16 MHz	18 MHz	20 MHz
1	2400			4800		
2						
4						
8						
16						
32	1200			2400		
64						

When using SCI0 in Snooze mode, use the following setting: BGDM = 0, ABCS = 0, ABCSE = 0. See section 26, Serial Communications Interface (SCI) for information on these bits.

Figure 10.9 shows a setting example for using SCI0 in Snooze mode entry.

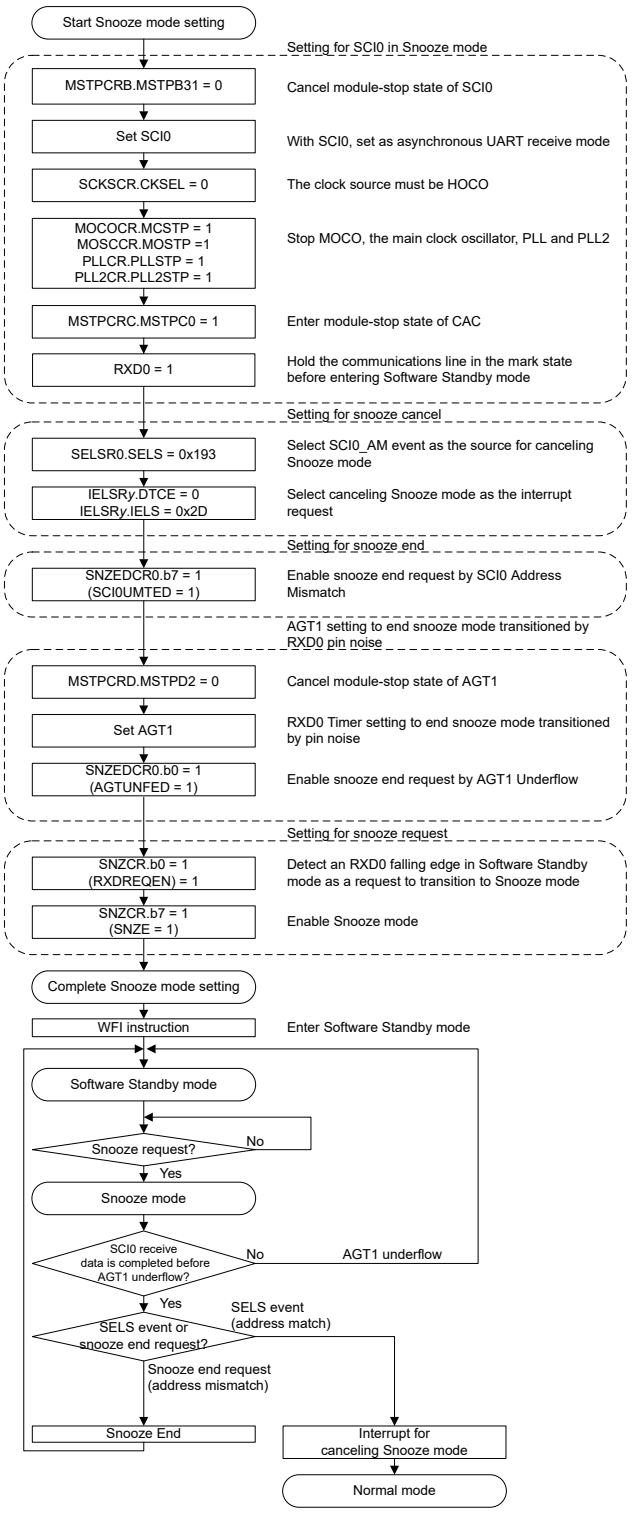


Figure 10.9 Setting example of using SCI0 in Snooze mode entry

10.9 Deep Software Standby Mode

10.9.1 Transitioning to Deep Software Standby Mode

When a WFI instruction is executed with the SBYCR.SSBY and DPSBYCR.DPSBY bits set to 1, the MCU enters Deep Software Standby mode. See Table 10.6 for the setting of the related control bits. In this mode, the CPU, on-chip peripheral functions, SRAM (except for standby RAM), and all oscillators (except for Low-speed on-chip oscillator) are stopped. Also because the internal power supply to these modules is stopped, power consumption is remarkably reduced. The contents of all CPU registers and internal peripheral modules become undefined.

Data in the standby SRAM are preserved if the setting of the DEEPCUT[1:0] bits are 00b. If the setting of the DEEPCUT[1:0] bits are 01b, the internal power supply to the standby SRAM is cut off, reducing power consumption. Data in the standby SRAM becomes undefined at this time. If the setting of the DEEPCUT[1:0] bits are 11b, the internal power supply to the standby SRAM is cut off, the LVD is stopped, and the low-power-consumption function of the power-on reset circuit is enabled, so power consumption is further reduced. For details, see section 46, Electrical Characteristics.

When the MCU enters Deep Software Standby mode while the IWDT is in auto start mode and the OFS0.IWDTSTPCTL bit is 1, power supply to the IWDT-dedicated clock and the IWDT is cut off, and counting by the IWDT stops.

When OFS0.IWDTSTPCTL bit is 0, the MCU enters Software Standby mode instead of Deep Software Standby mode, regardless of the setting of OFS0.IWDTSTRT bit or DPSBYCR.DPSBY bit. If OFS0.IWDTSTPCTL bit is 0 while OFS0.IWDTSTRT bit is 0 (auto start mode), IWDT-dedicated clock and IWDT continues the operation.

When LVD1CR0.RI = 1 (voltage monitor 1 reset selected) or LVD2CR0.RI = 1 (voltage monitor 2 reset selected), the MCU enters Software Standby mode instead of Deep Software Standby mode. The I/O port states are the same as in Software Standby mode.

When the PLL is selected as the clock source, set the following modules to the module-stop state before executing a WFI instruction:

ADC, SCE5.

In this case, wait for least 750 ns and if the ICLK frequency before executing the WFI instruction exceeds 120 MHz, it is necessary to set the ICLK frequency division ratio to 1/2 and wait 5 µs. Measurement of the waiting time, it is recommended the measurement by the software. If you use the timer, regardless of the use conditions, ensure that the waiting time has elapsed.

See Figure 10.2 for Example flow for transition to software standby mode or Deep Software Standby mode.

Note: Conditions on the DTC, DMAC, and IWDT for transitioning to Software Standby mode should be met before the WFI instruction is executed. For details, see section 10.7. Software Standby Mode.

10.9.2 Cancelling Deep Software Standby Mode

Deep Software Standby mode is canceled by:

- An interrupt shown in Table 10.3
- A RES pin reset
- A power-on reset
- A voltage monitor 0 reset.

(1) Cancelling by an interrupt

Cancelling by interrupts is controlled by DPSIERn (n = 0 to 2) and DPSIFRn (n = 0 to 2). When a Deep Software Standby Cancelling interrupt is generated, the corresponding flag in DPSIFRn is set to 1. If the interrupt is enabled in DPSIERn, Deep Software Standby mode is canceled. Rising edge or falling edge can be selected by DPSIEGRn (n = 0 to 2). The interrupts for which an edge can be selected are the NMI, IRQn-DS (n = 0 to 15), voltage monitor 1, and voltage monitor 2 interrupts. When a Deep Software Standby mode canceling request occurs, the internal power is supplied and MOCO starts oscillating, and an internal reset (Deep Software Standby reset) is generated for the entire MCU.

The stable MOCO clock is supplied to the entire MCU and Deep Software Standby reset is canceled. The MCU starts reset exception handling.

When Deep Software Standby mode is canceled by an external interrupt pin or internal interrupt signal, the RSTSR0.DPSRSTF flag is set to 1.

(2) Cancelling by RES pin reset

When the RES pin is driven low, the MCU cancels Deep Software Standby mode and enters the reset state. Keep the RES pin low for the time specified in section 46, Electrical Characteristics. When RES pin is driven high after the specified time period, the CPU starts the reset exception handling.

(3) Cancelling by a power-on reset

Deep Software Standby mode is canceled by a power-on reset and the MCU starts the reset exception handling.

(4) Cancelling by a voltage monitor 0 reset

Deep Software Standby mode is canceled by a voltage monitor 0 reset from the voltage detection circuit and the MCU starts the reset exception handling.

10.9.3 Pin States when Deep Software Standby mode is Canceled

In Deep Software Standby mode, the I/O ports retain the same states from Software Standby mode. The MCU is initialized by an internal reset generated when Deep Software Standby mode is canceled, and reset exception handling starts immediately. The DPSBYCR.IOKEEP bit setting determines whether to initialize the I/O ports or to retain the I/O ports states for Software Standby mode. The following is the state of the I/O ports for each bit setting:

- When the DPSBYCR.IOKEEP bit = 0
  I/O ports are initialized by an internal reset generated when Deep Software Standby mode is canceled.
- When the DPSBYCR.IOKEEP bit = 1
  Although the MCU is initialized by an internal reset generated when Deep Software Standby mode is canceled, the I/O ports retain their states from Software Standby mode regardless of the MCU internal state. The I/O ports states remain unchanged from Software Standby mode even when settings are made to the I/O ports or peripheral modules. The retained I/O ports states are released by clearing the DPSBYCR.IOKEEP bit to 0, and the MCU operates according to the internal state. The DPSBYCR.IOKEEP bit is not initialized by any internal reset generated when Deep Software Standby mode is canceled.

10.9.4 Example of Deep Software Standby Mode Application

(1) Entering and exiting Deep Software Standby mode

Figure 10.10 shows an example where a transition to Deep Software Standby mode is made at the falling edge of the IRQn-DS pin, and exiting Deep Software Standby mode is made at the rising edge of the IRQn-DS pin. In this example, an IRQn interrupt is accepted with the IRQCRi.IRQMD[1:0] bits of the ICU set to 00b (falling edge). After the DPSIEGRy.DIRQnEG (y = 0 or 1, n = 0 to 15) bit is set to 1 (rising edge) and the SBYCR.SSBY bit and DPSBYCR.DPSBY bit are both set to 1, the WFI instruction is executed. As a result, the MCU transitions to Deep Software Standby mode. Deep Software Standby mode is then canceled on the rising edge of the IRQn-DS pin.

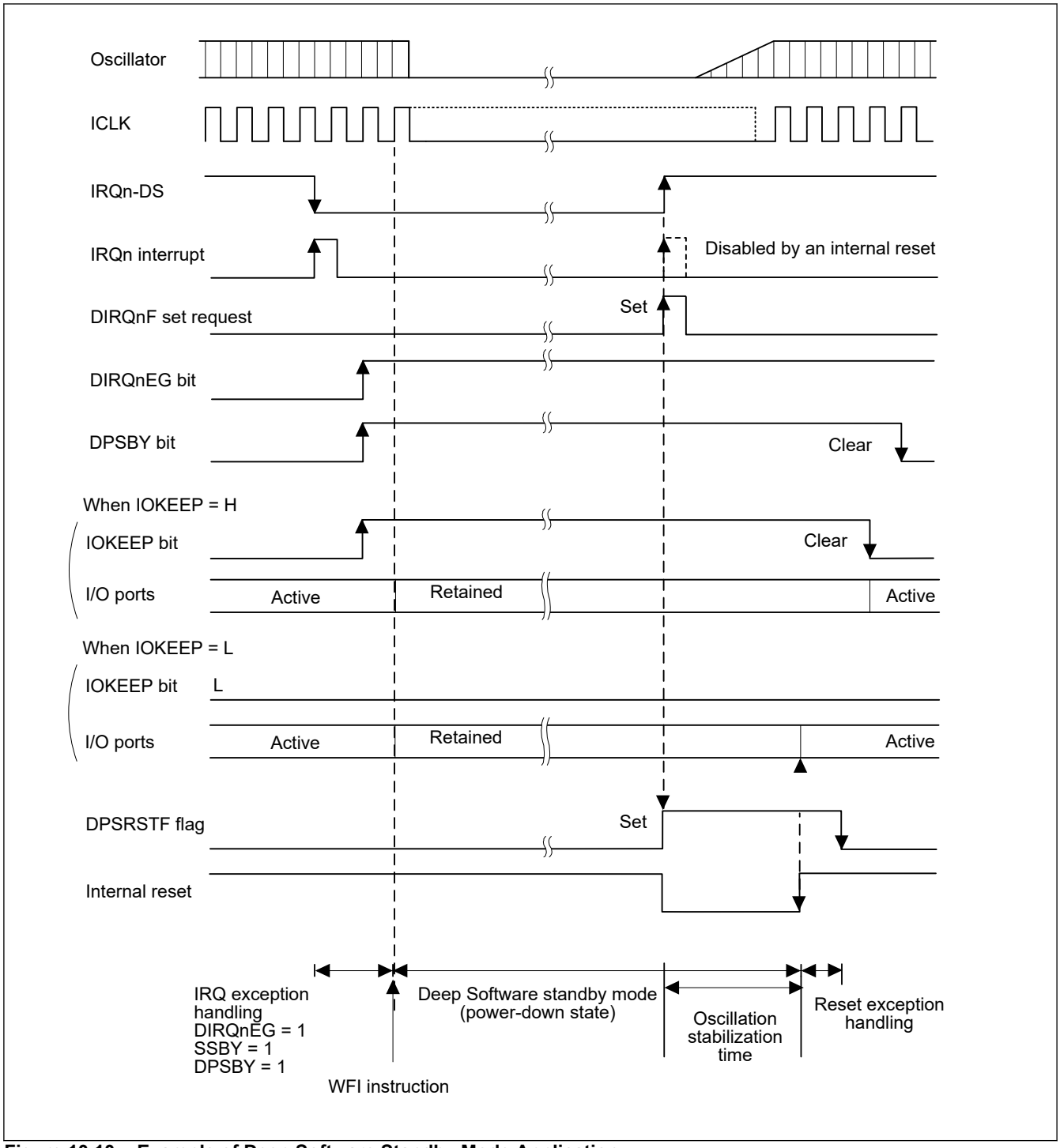


Figure 10.10 Example of Deep Software Standby Mode Application

10.9.5 Usage Flow for Deep Software Standby Mode

Figure 10.11 shows an example flow for using Deep Software Standby mode.

In this example, the RSTSR0.DPSRSTF flag of the reset function is read after the reset exception handling to determine whether a reset was generated by the RES pin or by the cancellation of Deep Software Standby mode.

For a reset by the RES pin, the MCU transitions to Deep Software Standby mode after the required register settings are made.

For a reset by cancellation of Deep Software Standby mode, the DPSBYCR.IOKEEP bit is cleared to 0 after the I/O port settings are made.

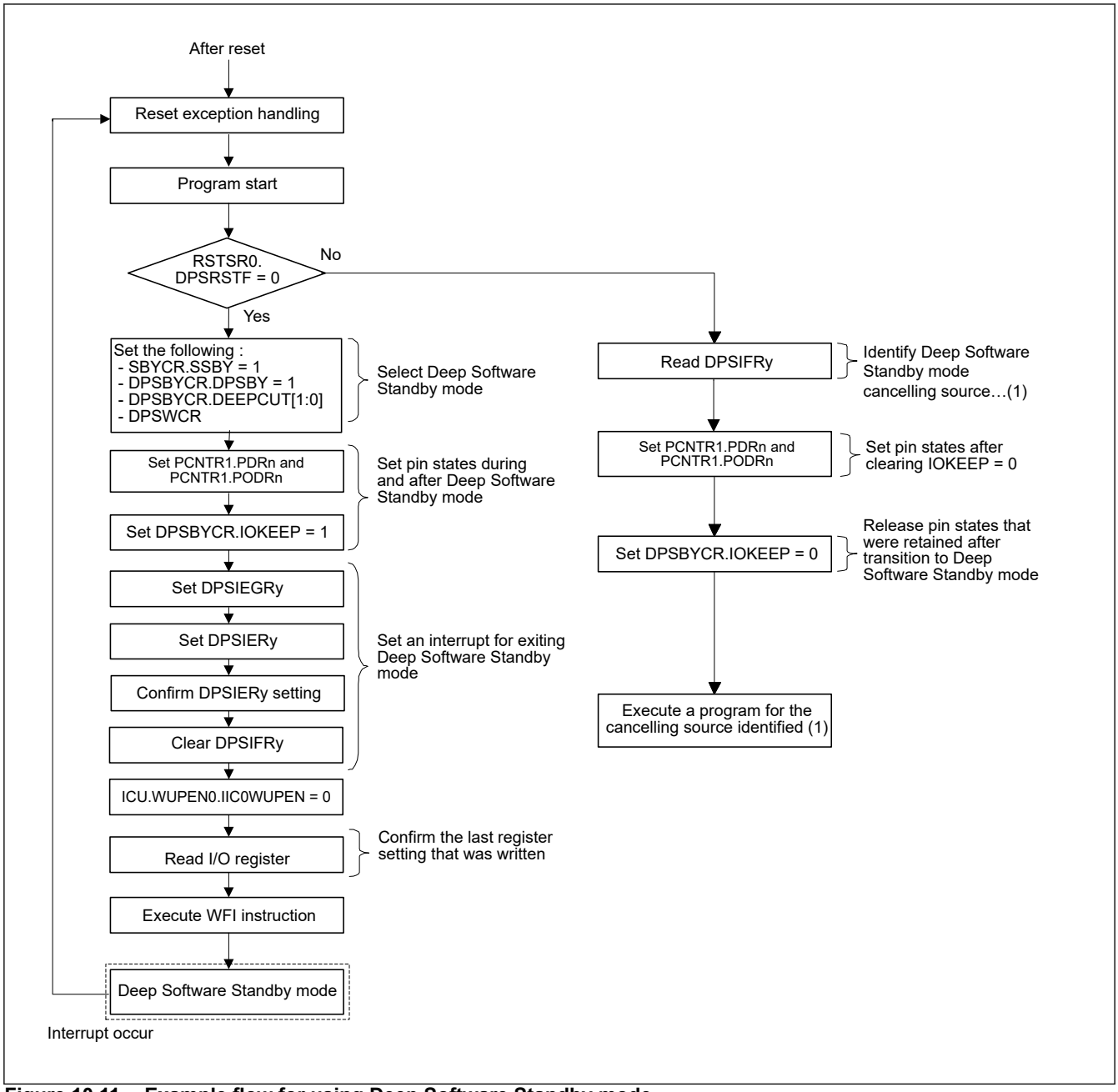


Figure 10.11 Example flow for using Deep Software Standby mode

10.10 Usage Notes

10.10.1 Register Access

(1) Invalid register write accesses during specific modes or transitions

Do not write to registers under any of the conditions listed in this section.

[Registers]

- All registers with a peripheral name of SYSTEM.

[Conditions]

- OPCCR.OPCMTSF = 1 (during transition of the operating power control mode)
- During the time period from executing a WFI instruction to returning to Normal mode
- FENTRYR.FENTRYC = 1 or FENTRYR.FENTRYD = 1 (flash P/E mode, data flash P/E mode)

(2) Valid setting for the clock-related registers

Table 10.11 and Table 10.12 show the valid settings of the clock-related registers in each operating power control mode. Do not write any value other than the valid setting. Each register has certain prohibited settings under conditions other than those related to the operating power control modes. See section 8, Clock Generation Circuit for another condition of each register.

Table 10.11 Valid settings for the clock-related registers (1)

Mode	Valid settings							
	SCKSCR. CKSEL[2:0] CKOCR. CKOSEL[2:0]	SCKDIVCR. FCK[2:0] ICK[2:0]	PLLCR. PLLSTP	PLL2CR. PLL2STP	HOCOCR. HCSTP	MOCOCR. MCSTP	LOCOCR. LCSTP	MOSCCR. MOSTP
High-speed	000b (HOCO) 001b (MOCO) 010b (LOCO) 011b (Main clock) 101b (PLL) *1	000b (1/1) 001b (1/2) 010b (1/4) 011b (1/8) 100b (1/16) 101b (1/32) 110b (1/64)	0 (operating) 1 (stop)	0 (operating) 1 (stop)	0 (operating) 1 (stopped)	0 (operating) 1 (stopped)	0 (operating) 1 (stopped)	0 (operating) 1 (stopped)
Low-speed	000b (HOCO) 001b (MOCO) 010b (LOCO) 011b (Main clock)		1 (stop)	1 (stop)				

Note 1. SCKSCR.CKSEL[2:0] only

Table 10.12 Valid settings for the clock-related registers (2)

Operating oscillator	Valid settings
	OPCCR.OPCM[1:0]
PLL, PLL2	00b
High-speed on-chip oscillator	00b, 11b
Middle-speed on-chip oscillator	
Main clock oscillator	
Low-speed on-chip oscillator	
IWDT-dedicated on-chip oscillator	

(3) Invalid register write accesses by the DTC or DMAC

Do not write to registers listed in this section by the DTC or DMAC.

[Registers]

- MSTPCRA, MSTPCRB, MSTPCRC, MSTPCRD, MSTPCRE

(4) Invalid register write accesses in Snooze mode

Do not write to registers listed in this section in Snooze mode. They must be set before entering Software Standby mode.

[Registers]

- SNZCR, SNZEDCR0, SNZREQCR0.

(5) Invalid write access when PRCR.PRC1 is 0

Do not write to registers listed in this section when the PRCR.PRC1 bit is 0.

[Registers]

- SBYCR, SNZCR, SNZEDCR0, SNZREQCR0, OPCCR, DPSBYCR, DPSWCR, DPSIERn, DPSIFRn, DPSIEGRn, SYOCDCR

(6) Invalid write access when when PRCR.PRC4 bit is 0

Do not write to registers listed in this section when the PRCR.PRC4 bit is 0.

[Registers]

- LPMSAR, DPFSAR

10.10.2 I/O Port pin states

The I/O port pin states in Software Standby mode, Deep Software Standby and Snooze mode, unless modifying in Snooze mode, are the same before entering the modes. Therefore, power consumption is not reduced while the output signals are held high.

10.10.3 Module-Stop State of DTC, DMAC

Before writing 1 to MSTPCRA.MSTPA22, clear the DMAST.DMST bit of the DMAC and the DTCST.DTCST bit of the DTC to 0. For details, see section 15, DMA Controller (DMAC) and section 16, Data Transfer Controller (DTC).

10.10.4 Internal Interrupt Sources

Interrupts do not operate in the module-stop state. If setting the module-stop bit while an interrupt request is generated, a CPU interrupt source or a DTC or DMAC startup source cannot be cleared. Always disable the associated interrupts before setting the module-stop bits.

10.10.5 Input Buffer Control by DIRQnE Bit

Setting the DPSIERy.DIRQnE (y = 0 or 1, n = 0 to 15) bit to 1 enables the associated input buffer of the IRQn-DS (n = 0 to 15) pins. Although inputs to these pins are sent to the DPSIFRy.DIRQnF (y = 0 or 1, n = 0 to 15) bits, they are not sent to the interrupt controller (ICU), peripheral modules, and I/O ports.

10.10.6 Transitioning to Low Power Modes

Because the MCU does not support wakeup by events, do not enter the low power modes such as Sleep mode, Software Standby mode or Deep Software Standby Mode by executing a WFE instruction. Also, do not set the SLEEPDEEP bit of the System Control Register in the Cortex-M33 core because the MCU does not support low power modes by SLEEPDEEP.

10.10.7 Timing of WFI Instruction

It is possible for the WFI instruction to be executed before I/O register write is completed, in which case operation might not be as intended. This can happen if the WFI is placed immediately after a write to an I/O register. To avoid this problem, read back the register that was written to confirm that the write completed.

10.10.8 Writing to the WDT/IWDT Registers by DTC or DMAC in Sleep Mode or Snooze Mode

Do not write to the WDT or IWDT registers by the DTC or DMAC while WDT or IWDT is stopped after entering Sleep mode or Snooze mode.

10.10.9 Oscillators in Snooze Mode

Oscillators that stop on entering Software Standby mode automatically restart when a trigger for switching to Snooze mode is generated. The MCU does not enter Snooze mode until all the oscillators stabilize. If in Snooze mode, you must disable oscillators that are not required in Snooze mode before entering Software Standby mode. Otherwise, the transition from Software Standby mode to Snooze mode takes longer.

10.10.10 Snooze Mode Entry by RXD0 Falling Edge

When the SNZCR.RXDREQEN bit is 1, the falling edge of RXD0 pin is used to switch MCU from Software Standby mode to Snooze mode when using UART of SCI0 in Snooze mode. In this case an interrupt such as SCI0_AM or an address mismatch event is used as the source for canceling Snooze mode. However noise on the RXD0 pin might cause the MCU to transfer from Software Standby mode to Snooze mode unexpectedly. In this case if the MCU does not receive RXD0 data after the noise, an interrupt such as SCI0_AM, or an address mismatch event is not generated and the MCU stays in Snooze mode. This can be avoided by using AGTn (n = 1) underflow interrupt to return to Software Standby mode or Normal mode unless otherwise UART receive data is completed before AGTn (n = 1) underflow. However, do not use the AGTn (n = 1) underflow as a source to return to Software Standby mode during an UART communication. This causes the UART to stop the operation in a half-finished state.

10.10.11 Using UART of SCI0 in Snooze Mode

When using UART in Snooze mode, ensure that the snooze request (RXD0 falling edge) does not conflict with the wakeup requests set by the WUPEN register, otherwise UART cannot be guaranteed.

When using UART in Snooze mode, the following conditions must be satisfied:

- The clock source must be HOCO
- MOCO, PLL, PLL2, and the main clock oscillator must be stopped before entering Software Standby mode
- The RXD0 pin must be kept high before entering Software Standby mode
- A transition to Software Standby mode must not occur during an SCI0 communication
- The MSTPCRC.MSTPC0 bit must be 1 before entering Software Standby mode.

10.10.12 ELC Events in Snooze Mode

This section lists available ELC events in Snooze mode. Do not use any other events. If starting peripheral modules for the first time after entering Snooze mode, the Event Link Setting Register (ELSRn) must set a Snooze mode entry event (SYSTEM_SNZREQ) as the trigger.

- Snooze mode entry (SYSTEM_SNZREQ)
- DTC transfer end (DTC_DTCEND)
- Data operation circuit interrupt (DOC_DOPCI).

10.10.13 Module-Stop Bit Write Timing

It is possible that access to I/O register may be executed before the corresponding module-stop bit write completed. In this case, access to I/O register may not proceed as intended. To avoid this issue, before accessing I/O register, read back the module-stop bit that was written to confirm that the write completed.

11. Register Write Protection

11.1 Overview

The register write protection function protects important registers from being overwritten due to software errors. The registers to be protected are set with the Protect Register (PRCR).

Table 11.1 lists the association between the bits in the PRCR register and the registers to be protected.

Table 11.1 Association between the bits in the PRCR register and registers to be protected

PRCR bit	Register to be protected
PRC0	• Registers related to the clock generation circuit: SCKDIVCR, SCKSCR, PLLCCR, PLLCR, MOSCCR, HOCOCR, MOCOCR, CKOCR, OSTDCR, OSTDSR, PLL2CCR, PLL2CR, MOCOUTCR, HOCOUTCR, SCISPICKDIVCR, CANFDCKDIVCR, GPTCKDIVCR, IICCKDIVCR, SCISPICKCR, CANFDCKCR, GPTCKCR, IICCKCR, MOSCWTCR, MOMCR, LOCOCR, LOCOUTCR
PRC1	• Registers related to the low power modes: SBYCR, SNZCR, SNZEDCR0, SNZREQCR0, OPCCR, DPSBYCR, DPSWCR, DPSIER0-2, DPSIFR0-2, DPSIEGR0-2, SYOCDCR
PRC3	• Registers related to the LVD: LVD1CR1, LVD1SR, LVD2CR1, LVD2SR, LVD1CMPCR, LVD2CMPCR, LVD1CR0, LVD2CR0
PRC4	• Registers related to the security function: CGFSAR, RSTSAR, LPMSAR, LVDSAR, DPFSAR, CSAR, SRAMSAR, STBRAMSAR, DTCSAR, DMACSAR, ICUSARx, BUSSARx, MMPUSARx, TZFSAR, CPUDSAR, FSAR, PSARx, MSSAR, PmSAR, ELCSARx

11.2 Register Descriptions

11.2.1 PRCR : Protect Register

Base address: SYSC = 0x4001_E000

Offset address: 0x3FE

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PRKEY[7:0]								—	—	—	PRC4	PRC3	—	PRC1	PRC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PRC0	Enable writing to the registers related to the clock generation circuit 0: Disable writes 1: Enable writes	R/W
1	PRC1	Enable writing to the registers related to the low power modes 0: Disable writes 1: Enable writes	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W
3	PRC3	Enable writing to the registers related to the LVD 0: Disable writes 1: Enable writes	R/W
4	PRC4	Enables writing to the registers related to the security function 0: Disable writes 1: Enable writes	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
15:8	PRKEY[7:0]	PRC Key Code These bits control the write access to the PRCR register. To modify the PRCR register, write 0xA5 to the upper 8 bits and the target value to the lower 8 bits as a 16-bit unit.	W

PRCn bits (Protect bit n) (n = 0, 1, 3, 4)

The PRCn bits enable or disable writing to the protected registers listed in Table 11.1. Setting the PRCn bits to 1 or 0 enables or disables writing, respectively.

The register controlled by PRC4 may not reflect the PRC4 change when PRCR and its controlled registers are continuously written access. Avoid continuous write access or read the PRCR after PRC4 change, and then write access the PRC4-controlled register.

12. Interrupt Controller Unit (ICU)

12.1 Overview

The Interrupt Controller Unit (ICU) controls which event signals are linked to the Nested Vector Interrupt Controller (NVIC), the DMA Controller (DMAC), and the Data Transfer Controller (DTC) modules. The ICU also controls non-maskable interrupts.

Table 12.1 lists the ICU specifications, Figure 12.1 shows a block diagram, and Table 12.2 lists the I/O pins.

Table 12.1 ICU specifications

Parameter		Description
Maskable interrupts	Peripheral function interrupts	• Interrupts from peripheral modules Number of sources: 265 (select factor within event list numbers 17 to 511)
	External pin interrupts	• Interrupt detection on low level*4, falling edge, rising edge, rising and falling edges. One of these detection methods can be set for each source • Digital filter function supported • 16 sources, with interrupts from IRQi (i = 0 to 15) pins.
	Interrupt requests to CPU (NVIC)	• 96 interrupt requests are output to NVIC.
	DMAC control	• The DMAC can be activated using interrupt sources*1 • The target interrupt source can be selected individually for every DMAC channels.
	DTC control	• The DTC can be activated using interrupt sources*1 • The method for selecting an interrupt source is the same as that of the interrupt request to NVIC.
Non-maskable interrupts*2	NMI pin interrupt	• Interrupt from the NMI pin • Interrupt detection on falling edge or rising edge • Digital filter function supported
	WDT underflow/refresh error*3	Interrupt on an underflow of the down-counter or occurrence of a refresh error
	IWDT underflow/refresh error*3	Interrupt on an underflow of the down-counter or occurrence of a refresh error
	Low voltage detection 1*3	Voltage monitor 1 interrupt of the voltage monitor 1 circuit (LVD_LVD1)
	Low voltage detection 2*3	Voltage monitor 2 interrupt of the voltage monitor 2 circuit (LVD_LVD2)
	RPEST*5	Interrupt on SRAM parity error
	RECCST*5	Interrupt on SRAM ECC error
	TZFST*5	Interrupt on TrustZone Filter error
	CPEST*5	Interrupt on Cache RAM Parity error
	Oscillation stop detection interrupt*3	Interrupt on detecting that the main oscillation has stopped
	Bus master MPU error*5	Interrupt on bus master MPU error
Low power modes		• Sleep mode: return is initiated by non-maskable interrupts or any other interrupt source • Software Standby mode: return is initiated by non-maskable interrupts. Interrupt can be selected in the WUPEN register. • Snooze mode: return is initiated by non-maskable interrupts. Interrupt can be selected in the SELSR0 and WUPEN registers. See section 12.2.16. SELSR0 : SYS Event Link Setting Register and section 12.2.17. WUPEN0 : Wake Up Interrupt Enable Register 0.
TrustZone Filter		Available

Note 1. For the DMAC and DTC activation sources, see Table 12.4.
Note 2. Non-maskable interrupts can be enabled only once after a reset release.
Note 3. These non-maskable interrupts can also be used as maskable interrupts. When used as maskable interrupts, do not change the value of the NMIER register from the reset state. To enable voltage monitor 1 and voltage monitor 2 interrupts, set the LVD1CR1.IRQSEL and LVD2CR1.IRQSEL bits to 1.
Note 4. Low level: interrupt detection is not canceled if you do not clear it after a detection.
Note 5. These non-maskable interrupt sources cannot be recovered if the request source clock is stopped during low power mode.

Figure 12.1 shows the ICU block diagram.

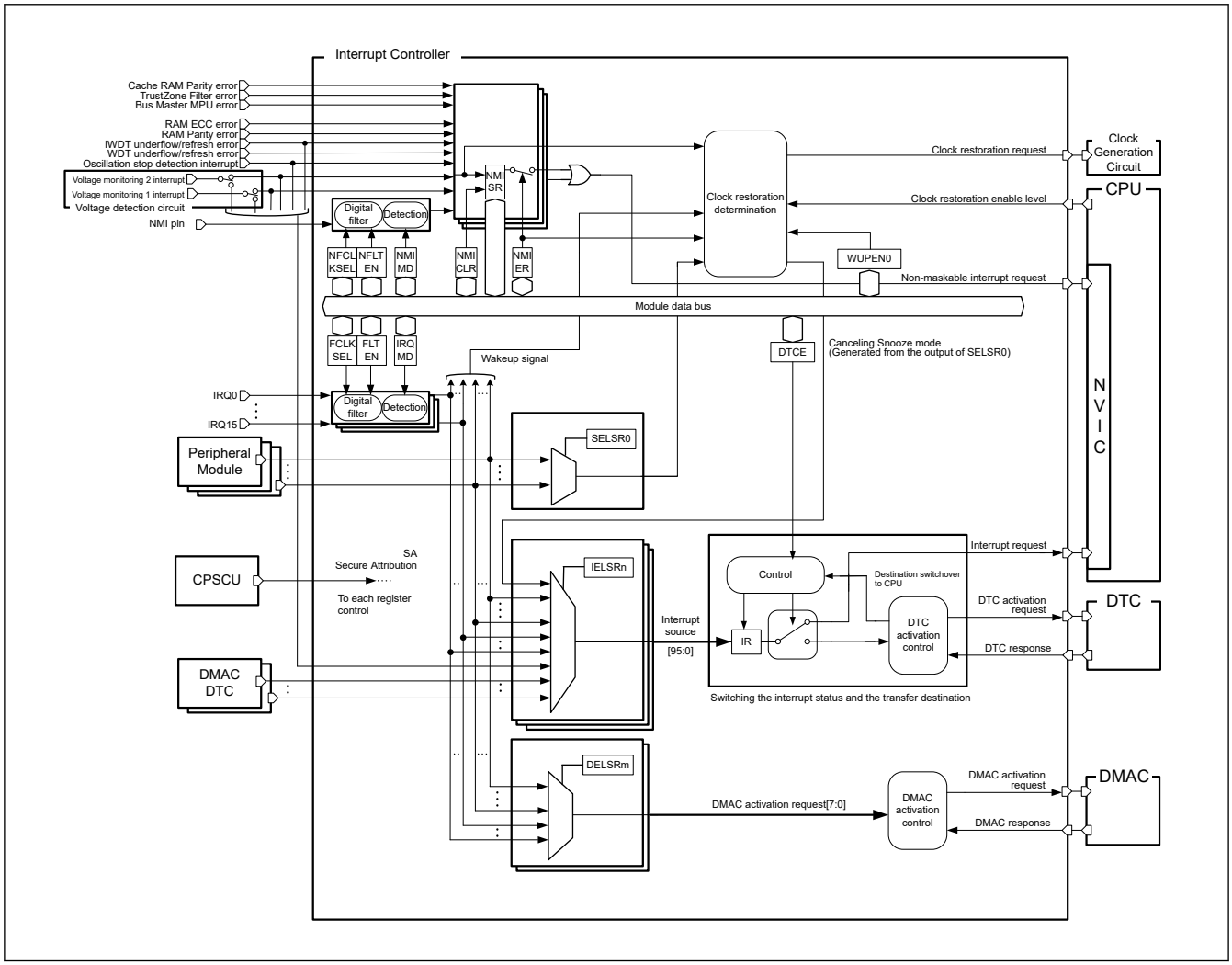


Figure 12.1 ICU block diagram

Table 12.2 lists the ICU input/output pins.

Table 12.2 ICU I/O pins

Pin name	I/O	Description
NMI	Input	Non-maskable interrupt request pin
IRQi (i = 0 to 15)	Input	External interrupt request pins

12.2 Register Descriptions

This chapter does not describe the Arm® NVIC internal registers. For information about these registers, see ARM Limited., ARM® Cortex®-M33 Processor Technical Reference Manual (ARM 100230).

12.2.1 ICUSARA : Interrupt Controller Unit Security Attribution Register A

Base address: CPSCU = 0x4000_8000

Offset address: 0x40

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SAIRQ CR15	SAIRQ CR14	SAIRQ CR13	SAIRQ CR12	SAIRQ CR11	SAIRQ CR10	SAIRQ CR9	SAIRQ CR8	SAIRQ CR7	SAIRQ CR6	SAIRQ CR5	SAIRQ CR4	SAIRQ CR3	SAIRQ CR2	SAIRQ CR1	SAIRQ CR0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
15:0	SAIRQCR15 to SAIRQCR0	Security attributes of registers for the IRQCRn register 0: Secure 1: Non-secure	R/W
31:16	—	These bits are read as 1. The write value should be 1.	R/W

Note: Only secure access can write to this register. Both secure access and Non-secure read access are allowed but Non-secure write access is not allowed and TrustZone access error is not generated.

Note: This register is write-protected by PRCR register.

SAIRQCRn bits (Security attributes of registers for the IRQCRn register)

The target registers are as follows:

- IRQCR0 to IRQCR15 registers
- WUPEN0.IRQWUPEN[15:0] bits

12.2.2 ICUSARB : Interrupt Controller Unit Security Attribution Register B

Base address: CPSCU = 0x4000_8000

Offset address: 0x44

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

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

Bit	Symbol	Function	R/W
0	SANMI	Security attributes of registers for nonmaskable interrupt 0: Secure 1: Non-secure	R/W
31:1	—	These bits are read as 1. The write value should be 1.	R/W

Note: Only secure access can write to this register. Both secure access and Non-secure read access are allowed but Non-secure write access is not allowed and TrustZone access error is not generated.

Note: This register is write-protected by PRCR register.

SANMI bit (Security attributes of registers for nonmaskable interrupt)

Security attributes of registers for non-maskable interrupt. The target registers are as follows:

- NMIER

- NMICLR
- NMICR

The value of AIRCR.BFHFNMINS bit [13] in Application Interrupt and Reset Control Register of ARM CPU should be the same as the value of security attribution. The initial values of AIRCR.BFHFNMINS and the SANMI bits are different. AIRCR.BFHFNMINS is secure and SANMI is non-secure. Polarity has the same meaning so program these to match.

Note: Only one of Secure and Non-Secure can set security attribution for non-maskable interrupt-related registers. If you program the Secure attribute as secure, it always goes to the Secure interrupt handler. To release any of the non-maskable interrupt sources to the non-secure user, write a function to execute a nonsecure program from the interrupt handler for Secure.

12.2.3 ICUSARC : Interrupt Controller Unit Security Attribution Register C

Base address: CPSCU = 0x4000_8000

Offset address: 0x48

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	SADMAC7	SADMAC6	SADMAC5	SADMAC4	SADMAC3	SADMAC2	SADMAC1	SADMAC0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
7:0	SADMAC7 to SADMAC0	Security attributes of registers for DMAC channel 0: Secure 1: Non-secure	R/W
31:8	—	These bits are read as 1. The write value should be 1.	R/W

Note: Only secure access can write to this register. Both secure access and Non-secure read access are allowed but Non-secure write access is not allowed and TrustZone access error is not generated.

Note: This register is write-protected by PRCR register.

SADMACn bits (Security attributes of registers for DMAC channel)

Security attributes of registers for DMAC channel. This register is referred to as the security attribute of the ICU and DMAC registers.

The controlled ICU register is:

- DELSRn

The controlled DMAC registers are:

- DMACn.DMSAR
- DMACn.DMSRR
- DMACn.DMDAR
- DMACn.DMDRR
- DMACn.DMCRA
- DMACn.DMCRB
- DMACn.DMTMD
- DMACn.DMINT
- DMACn.DMAMD
- DMACn.DMOFR

- DMACn.DMCNT
- DMACn.DMREQ
- DMACn.DMSTS
- DMACn.DMSBS
- DMACn.DMDBS

For details on DMAC registers, see section 15, DMA Controller (DMAC).

12.2.4 ICUSARD : Interrupt Controller Unit Security Attribution Register D

Base address: CPSCU = 0x4000_8000

Offset address: 0x4C

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SASELSR0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	SASELSR0	Security attributes of registers for SELSR0 0: Secure 1: Non-secure	R/W
31:1	—	These bits are read as 1. The write value should be 1.	R/W

Note: Only secure access can write to this register. Both secure access and Non-secure read access are allowed but Non-secure write access is not allowed and TrustZone access error is not generated.

Note: This register is write-protected by PRCR register.

12.2.5 ICUSARE : Interrupt Controller Unit Security Attribution Register E

Base address: CPSCU = 0x4000_8000

Offset address: 0x50

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	SAIIC0WUP	SAAGT1CBWUP	SAAGT1CAWUP	SAAGT1UDWUP	—	—	—	—	—	—	—	—	SALVD2WUP	SALVD1WUP	SAKEYWUP	SAIWDTWUP
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

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

Bit	Symbol	Function	R/W
15:0	—	These bits are read as 1. The write value should be 1.	R/W
16	SAIWDTWUP	Security attributes of registers for WUPEN0.b16 0: Secure 1: Non-secure	R/W
17	SAKEYWUP	Security attributes of registers for WUPEN0.b17 0: Secure 1: Non-secure	R/W

Bit	Symbol	Function	R/W
18	SALVD1WUP	Security attributes of registers for WUPEN0.b18 0: Secure 1: Non-secure	R/W
19	SALVD2WUP	Security attributes of registers for WUPEN0.b19 0: Secure 1: Non-secure	R/W
27:20	—	These bits are read as 1. The write value should be 1.	R/W
28	SAAGT1UDWUP	Security attributes of registers for WUPEN0.b28 0: Secure 1: Non-secure	R/W
29	SAAGT1CAWUP	Security attributes of registers for WUPEN0.b29 0: Secure 1: Non-secure	R/W
30	SAAGT1CBWUP	Security attributes of registers for WUPEN0.b30 0: Secure 1: Non-secure	R/W
31	SAIIC0WUP	Security attributes of registers for WUPEN0.b31 0: Secure 1: Non-secure	R/W

Note: Only secure access can write to this register. Both secure access and Non-secure read access are allowed but Non-secure write access is not allowed and TrustZone access error is not generated.
Note: This register is write-protected by PRCR register.

12.2.6 ICUSARG : Interrupt Controller Unit Security Attribution Register G

Base address: CPSCU = 0x4000_8000

Offset address: 0x70

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	SAIELSR31	SAIELSR30	SAIELSR29	SAIELSR28	SAIELSR27	SAIELSR26	SAIELSR25	SAIELSR24	SAIELSR23	SAIELSR22	SAIELSR21	SAIELSR20	SAIELSR19	SAIELSR18	SAIELSR17	SAIELSR16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SAIELSR15	SAIELSR14	SAIELSR13	SAIELSR12	SAIELSR11	SAIELSR10	SAIELSR9	SAIELSR8	SAIELSR7	SAIELSR6	SAIELSR5	SAIELSR4	SAIELSR3	SAIELSR2	SAIELSR1	SAIELSR0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
31:0	SAIELSR31 to SAIELSR0	Security attributes of registers for IELSR31 to IELSR0 0: Secure 1: Non-secure	R/W

Note: Only secure access can write to this register. Both secure access and Non-secure read access are allowed but Non-secure write access is not allowed and TrustZone access error is not generated.
Note: This register is write-protected by PRCR register.

SAIELSRn bits (Security attributes of registers for IELSR31 to IELSR0)

The Secure Attribute managed within the Arm CPU NVIC must match the security attribution of IELSEn (n = 0 to 31). NVIC internal registers are in NVIC_ITNS0[31:0]. The initial values of NVIC_ITNS0 and ICUSARG are different. NVIC_ITNS0 is secure and ICUSARG is non-secure. Polarity has the same meaning so program these to match.

12.2.7 ICUSARH : Interrupt Controller Unit Security Attribution Register H

Base address: CPSCU = 0x4000_8000

Offset address: 0x74

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	SAIELSR63	SAIELSR62	SAIELSR61	SAIELSR60	SAIELSR59	SAIELSR58	SAIELSR57	SAIELSR56	SAIELSR55	SAIELSR54	SAIELSR53	SAIELSR52	SAIELSR51	SAIELSR50	SAIELSR49	SAIELSR48
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SAIELSR47	SAIELSR46	SAIELSR45	SAIELSR44	SAIELSR43	SAIELSR42	SAIELSR41	SAIELSR40	SAIELSR39	SAIELSR38	SAIELSR37	SAIELSR36	SAIELSR35	SAIELSR34	SAIELSR33	SAIELSR32
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
31:0	SAIELSR63 to SAIELSR32	Security attributes of registers for IELSR63 to IELSR32 0: Secure 1: Non-secure	R/W

Note: Only secure access can write to this register. Both secure access and Non-secure read access are allowed but Non-secure write access is not allowed and TrustZone access error is not generated.

Note: This register is write-protected by PRCR register.

SAIELSRn bits (Security attributes of registers for IELSR63 to IELSR32)

The Secure Attribute managed within the ARM CPU NVIC must match the security attribution of IELSEn (n = 32 to 63). NVIC internal registers are in NVIC_ITNS1[31:0]. The initial values of NVIC_ITNS1 and ICUSARH are different. NVIC_ITNS1 is secure and ICUSARH is non-secure. Polarity has the same meaning so program these to match.

12.2.8 ICUSARI : Interrupt Controller Unit Security Attribution Register I

Base address: CPSCU = 0x4000_8000

Offset address: 0x78

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	SAIELSR95	SAIELSR94	SAIELSR93	SAIELSR92	SAIELSR91	SAIELSR90	SAIELSR89	SAIELSR88	SAIELSR87	SAIELSR86	SAIELSR85	SAIELSR84	SAIELSR83	SAIELSR82	SAIELSR81	SAIELSR80
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SAIELSR79	SAIELSR78	SAIELSR77	SAIELSR76	SAIELSR75	SAIELSR74	SAIELSR73	SAIELSR72	SAIELSR71	SAIELSR70	SAIELSR69	SAIELSR68	SAIELSR67	SAIELSR66	SAIELSR65	SAIELSR64
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
31:0	SAIELSR95 to SAIELSR64	Security attributes of registers for IELSR95 to IELSR64 0: Secure 1: Non-secure	R/W

Note: Only secure access can write to this register. Both secure access and Non-secure read access are allowed but Non-secure write access is not allowed and TrustZone access error is not generated.

Note: This register is write-protected by PRCR register.

SAIELSRn bits (Security attributes of registers for IELSR95 to IELSR64)

The Secure Attribute managed within the ARM CPU NVIC must match the security attribution of IELSEn (n = 64 to 95). NVIC internal registers are in NVIC_ITNS2[31:0]. The initial values of NVIC_ITNS2 and ICUSARI are different. NVIC_ITNS2 is secure and ICUSARI is non-secure. Polarity has the same meaning so program these to match.

12.2.9 IRQCRi : IRQ Control Register i (i = 0 to 15)

Base address: ICU = 0x4000_6000

Offset address: 0x000 + 0x1 × i

Bit position:	7	6	5	4	3	2	1	0
Bit field:	FLTEN	—	FCLKSEL[1:0]		—	—	IRQMD[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	IRQMD[1:0]	IRQi Detection Sense Select 0 0: Falling edge 0 1: Rising edge 1 0: Rising and falling edges 1 1: Low level	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
5:4	FCLKSEL[1:0]	IRQi Digital Filter Sampling Clock Select 0 0: PCLKB 0 1: PCLKB/8 1 0: PCLKB/32 1 1: PCLKB/64	R/W
6	—	This bit is read as 0. The write value should be 0.	R/W
7	FLTEN	IRQi Digital Filter Enable 0: Digital filter is disabled 1: Digital filter is enabled.	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

IRQCRi register changes must satisfy the following conditions:

- For a CPU interrupt or DTC trigger:
  Change the IRQCRi register value before setting the target IELSRn register (n = 0 to 95).
  The register value should be changed only when the value of the target IELSRn register is 0x0000.
- For a DMAC trigger:
  Change the IRQCRi register value before setting the target DELSRn register (n = 0 to 7).
  The register value should be changed only when the value of the target DELSRn register is 0x0000.
- For a wakeup enable signal:
  Change the IRQCRi register setting before setting the target WUPEN0.IRQWUPEN[n] (n = 0 to 15). The register value should be changed when the target WUPEN0.IRQWUPEN[n] is 0.

IRQMD[1:0] bits (IRQi Detection Sense Select)

The IRQMD[1:0] bits set the detection sensing method for the IRQi external pin interrupt sources. For setting method when using external pin interrupt, see section 12.5.6. External Pin Interrupts.

FCLKSEL[1:0] bits (IRQi Digital Filter Sampling Clock Select)

The FCLKSEL[1:0] bits select the digital filter sampling clock for the IRQi external pin interrupt request pins, selectable to:

- PCLKB (every cycle)
- PCLKB/8 (once every 8 cycles)
- PCLKB/32 (once every 32 cycles)
- PCLKB/64 (once every 64 cycles)

For details of the digital filter, see section 12.5.5. Digital Filter.

FLTEN bit (IRQi Digital Filter Enable)

The FLTEN bit enables the digital filter used for the IRQi external pin interrupt sources. The digital filter is enabled when the IRQCRi.FLTEN bit is 1 and disabled when the IRQCRi.FLTEN bit is 0. The IRQi pin level is sampled at the clock cycle specified in the IRQCRi.FCLKSEL[1:0] bits. When the sampled level matches three times, the output level from the digital filter changes. For details of the digital filter, see section 12.5.5. Digital Filter.

12.2.10 NMISR : Non-Maskable Interrupt Status Register

Base address: ICU = 0x4000_6000

Offset address: 0x140

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CPEST	—	TZFST	—	BUSMST	—	RECCST	RPEST	NMIST	OSTST	—	—	LVD2ST	LVD1ST	WDTST	IWDTST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	IWDTST	IWDT Underflow/Refresh Error Interrupt Status Flag 0: Interrupt not requested 1: Interrupt requested	R
1	WDTST	WDT Underflow/Refresh Error Interrupt Status Flag 0: Interrupt not requested 1: Interrupt requested	R
2	LVD1ST	Voltage Monitor 1 Interrupt Status Flag 0: Interrupt not requested 1: Interrupt requested	R
3	LVD2ST	Voltage Monitor 2 Interrupt Status Flag 0: Interrupt not requested 1: Interrupt requested	R
5:4	—	These bits are read as 0.	R
6	OSTST	Main Clock Oscillation Stop Detection Interrupt Status Flag 0: Interrupt not requested for main clock oscillation stop 1: Interrupt requested for main clock oscillation stop	R
7	NMIST	NMI Pin Interrupt Status Flag 0: Interrupt not requested 1: Interrupt requested	R
8	RPEST	SRAM Parity Error Interrupt Status Flag 0: Interrupt not requested 1: Interrupt requested	R
9	RECCST	SRAM ECC Error Interrupt Status Flag 0: Interrupt not requested 1: Interrupt requested	R
10	—	This bit is read as 0.	R
11	BUSMST	Bus Master MPU Error Interrupt Status Flag 0: Interrupt not requested 1: Interrupt requested	R
12	—	This bit is read as 0.	R
13	TZFST	TrustZone Filter Error Interrupt Status Flag 0: Interrupt not requested 1: Interrupt requested	R
14	—	This bit is read as 0.	R
15	CPEST	Cache RAM Parity Error Interrupt Status Flag 0: Interrupt not requested 1: Interrupt requested	R

The NMISR register monitors the status of non-maskable interrupt sources. Writes to the NMISR register are ignored. The setting in the Non-Maskable Interrupt Enable Register (NMIER) does not affect the status flags in this register. Before the end of the non-maskable interrupt handler, check that all of the bits in this register are set to 0 to confirm that no other NMI requests are generated during handler processing.

IWDTST flag (IWDT Underflow/Refresh Error Interrupt Status Flag)

The IWDTST flag indicates an IWDT underflow/refresh error interrupt request. It is read-only and cleared by the NMICLR.IWDTCLR bit.

[Setting condition]

When the IWDT underflow/refresh error interrupt is generated and this interrupt source is enabled.

[Clearing condition]

When 1 is written to the NMICLR.IWDTCLR bit.

WDTST flag (WDT Underflow/Refresh Error Interrupt Status Flag)

The WDTST flag indicates a WDT underflow/refresh error interrupt request. It is read-only and cleared by the NMICLR.WDTCLR bit.

[Setting condition]

When the WDT underflow/refresh error interrupt is generated.

[Clearing condition]

When 1 is written to the NMICLR.WDTCLR bit.

LVD1ST flag (Voltage Monitor 1 Interrupt Status Flag)

The LVD1ST flag indicates a request for voltage monitor 1 interrupt. It is read-only and cleared by the NMICLR.LVD1CLR bit.

[Setting condition]

When the voltage monitor 1 interrupt is generated and this interrupt source is enabled.

[Clearing condition]

When 1 is written to the NMICLR.LVD1CLR bit.

LVD2ST flag (Voltage Monitor 2 Interrupt Status Flag)

The LVD2ST flag indicates a request for voltage monitor 2 interrupt. It is read-only and cleared by the NMICLR.LVD2CLR bit.

[Setting condition]

When the voltage monitor 2 interrupt is generated and this interrupt source is enabled.

[Clearing condition]

When 1 is written to the NMICLR.LVD2CLR bit.

OSTST flag (Main Clock Oscillation Stop Detection Interrupt Status Flag)

The OSTST flag indicates a main clock oscillation stop detection interrupt request. It is read-only and cleared by the NMICLR.OSTCLR bit.

[Setting condition]

When the main clock oscillation stop detection interrupt is generated.

[Clearing condition]

When 1 is written to the NMICLR.OSTCLR bit.

NMIST flag (NMI Pin Interrupt Status Flag)

The NMIST flag indicates an NMI pin interrupt request. It is read-only and cleared by the NMICLR.NMICLR bit.

[Setting condition]

When an edge specified by the NMICR.NMIMD bit is input to the NMI pin.

[Clearing condition]

When 1 is written to the NMICLR.NMICLR bit.

RPEST flag (SRAM Parity Error Interrupt Status Flag)

The RPEST flag indicates an SRAM parity error interrupt request.

[Setting condition]

When an interrupt is generated in response to an SRAM parity error.

[Clearing condition]

When 1 is written to the NMICLR.RPECLR bit.

RECCST flag (SRAM ECC Error Interrupt Status Flag)

The RECCST flag indicates an SRAM ECC error interrupt request.

[Setting condition]

When an interrupt is generated in response to an SRAM ECC error.

[Clearing condition]

When 1 is written to the NMICLR.RECCCLR bit.

BUSMST flag (Bus Master MPU Error Interrupt Status Flag)

The BUSMST flag indicates a bus master error interrupt request.

[Setting condition]

When an interrupt is generated in response to a bus master error.

[Clearing condition]

When 1 is written to the NMICLR.BUSMCLR bit.

TZFST flag (TrustZone Filter Error Interrupt Status Flag)

This flag indicates the TrustZone Filter error interrupt request.

[Setting condition]

When an interrupt is generated in response to a TrustZone Filter error

[Clearing condition]

When 1 is written to the NMICLR.TZFCLR bit

CPEST flag (Cache RAM Parity Error Interrupt Status Flag)

This flag indicates the Cache RAM Parity error interrupt request.

[Setting condition]

When an interrupt is generated in response to an Cache RAM Parity error

[Clearing condition]

When 1 is written to the NMICLR.CPECLR bit

12.2.11 NMIER : Non-Maskable Interrupt Enable Register

Base address: ICU = 0x4000_6000

Offset address: 0x120

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CPEEN	—	TZFEN	—	BUSMEN	—	RECCEN	RPEEN	NMIEN	OSTEN	—	—	LVD2EN	LVD1EN	WDTEN	IWDTEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	IWDTEN	IWDT Underflow/Refresh Error Interrupt Enable 0: Disabled 1: Enabled.	R/W*1 *2
1	WDTEN	WDT Underflow/Refresh Error Interrupt Enable 0: Disabled 1: Enabled	R/W*1 *2
2	LVD1EN	Voltage monitor 1 Interrupt Enable 0: Disabled 1: Enabled	R/W*1 *2
3	LVD2EN	Voltage monitor 2 Interrupt Enable 0: Disabled 1: Enabled	R/W*1 *2
5:4	—	These bits are read as 0. The write value should be 0.	R/W
6	OSTEN	Main Clock Oscillation Stop Detection Interrupt Enable 0: Disabled 1: Enabled	R/W*1 *2
7	NMIEN	NMI Pin Interrupt Enable 0: Disabled 1: Enabled	R/W*1
8	RPEEN	SRAM Parity Error Interrupt Enable 0: Disabled 1: Enabled	R/W*1
9	RECCEN	SRAM ECC Error Interrupt Enable 0: Disabled 1: Enabled	R/W*1
10	—	This bit is read as 0. The write value should be 0.	R/W
11	BUSMEN	Bus Master MPU Error Interrupt Enable 0: Disabled 1: Enabled	R/W*1
12	—	This bit is read as 0. The write value should be 0.	R/W
13	TZFEN	TrustZone Filter Error Interrupt Enable 0: Disabled 1: Enabled	R/W*1
14	—	This bit is read as 0. The write value should be 0.	R/W
15	CPEEN	Cache RAM Parity Error Interrupt Enable 0: Disabled 1: Enabled	R/W*1

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note 1. You can write 1 to this bit only once after reset. Subsequent write accesses are invalid. Writing 0 to this bit is invalid.
Note 2. Do not write 1 to this bit when the source is used as an event signal.

IWDTEN bit (IWDT Underflow/Refresh Error Interrupt Enable)

The IWDTEN bit enables IWDT underflow/refresh error interrupt as an NMI trigger.

WDTEN bit (WDT Underflow/Refresh Error Interrupt Enable)

The WDTEN bit enables WDT underflow/refresh error interrupt as an NMI trigger.

LVD1EN bit (Voltage monitor 1 Interrupt Enable)

The LVD1EN bit enables voltage monitor 1 interrupt as an NMI trigger.

LVD2EN bit (Voltage monitor 2 Interrupt Enable)

The LVD2EN bit enables voltage monitor 2 interrupt as an NMI trigger.

OSTEN bit (Main Clock Oscillation Stop Detection Interrupt Enable)

The OSTEN bit enables main clock oscillation stop detection interrupt as an NMI trigger.

NMIEN bit (NMI Pin Interrupt Enable)

The NMIEN bit enables NMI pin interrupt as an NMI trigger.

RPEEN bit (SRAM Parity Error Interrupt Enable)

The RPEEN bit enables SRAM parity error interrupt as an NMI trigger.

RECCEN bit (SRAM ECC Error Interrupt Enable)

The RECCEN bit enables SRAM ECC error interrupt as an NMI trigger.

BUSMEN bit (Bus Master MPU Error Interrupt Enable)

The BUSMEN bit enables bus master error interrupt as an NMI trigger.

TZFEN bit (TrustZone Filter Error Interrupt Enable)

TZFEN bit enables the TrustZone Filter error interrupt as an NMI trigger.

CPEEN bit (Cache RAM Parity Error Interrupt Enable)

CPEEN bit enables the Cache RAM Parity error interrupt as an NMI trigger.

12.2.12 NMICLR : Non-Maskable Interrupt Status Clear Register

Base address: ICU = 0x4000_6000

Offset address: 0x130

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CPECLR	—	TZFCLR	—	BUSMCLR	—	RECCCLR	RPECLR	NMICLR	OSTCLR	—	—	LVD2CLR	LVD1CLR	WDTCLR	IWDTCLR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	IWDTCLR	IWDT Underflow/Refresh Error Interrupt Status Flag Clear 0: No effect 1: Clear the NMISR.IWDTST flag	R/W*1
1	WDTCLR	WDT Underflow/Refresh Error Interrupt Status Flag Clear 0: No effect 1: Clear the NMISR.WDTST flag	R/W*1
2	LVD1CLR	Voltage Monitor 1 Interrupt Status Flag Clear 0: No effect 1: Clear the NMISR.LVD1ST flag	R/W*1
3	LVD2CLR	Voltage Monitor 2 Interrupt Status Flag Clear 0: No effect 1: Clear the NMISR.LVD2ST flag.	R/W*1
5:4	—	These bits are read as 0. The write value should be 0.	R/W
6	OSTCLR	Oscillation Stop Detection Interrupt Status Flag Clear 0: No effect 1: Clear the NMISR.OSTST flag	R/W*1
7	NMICLR	NMI Pin Interrupt Status Flag Clear 0: No effect 1: Clear the NMISR.NMIST flag	R/W*1
8	RPECLR	SRAM Parity Error Interrupt Status Flag Clear 0: No effect 1: Clear the NMISR.RPEST flag	R/W*1

Bit	Symbol	Function	R/W
9	RECCCLR	SRAM ECC Error Interrupt Status Flag Clear 0: No effect 1: Clear the NMISR.RECCST flag	R/W*1
10	—	This bit is read as 0. The write value should be 0.	R/W
11	BUSMCLR	Bus Master MPU Error Interrupt Status Flag Clear 0: No effect 1: Clear the NMISR.BUSMST flag	R/W*1
12	—	This bit is read as 0. The write value should be 0.	R/W
13	TZFCLR	TrustZone Filter Error Interrupt Status Flag Clear 0: No effect 1: Clear the NMISR.TZFCLR flag	R/W*1
14	—	This bit is read as 0. The write value should be 0.	R/W
15	CPECLR	Cache RAM Parity Error Interrupt Status Flag Clear 0: No effect 1: Clear the NMISR.CPECLR flag	R/W*1

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note 1. Only write 1 to this bit.

IWDTCLR bit (IWDT Underflow/Refresh Error Interrupt Status Flag Clear)

Writing 1 to the IWDTCLR bit clears the NMISR.IWDTST flag. This bit is read as 0.

WDTCLR bit (WDT Underflow/Refresh Error Interrupt Status Flag Clear)

Writing 1 to the WDTCLR bit clears the NMISR.WDTST flag. This bit is read as 0.

LVD1CLR bit (Voltage Monitor 1 Interrupt Status Flag Clear)

Writing 1 to the LVD1CLR bit clears the NMISR.LVD1ST flag. This bit is read as 0.

LVD2CLR bit (Voltage Monitor 2 Interrupt Status Flag Clear)

Writing 1 to the LVD2CLR bit clears the NMISR.LVD2ST flag. This bit is read as 0.

OSTCLR bit (Oscillation Stop Detection Interrupt Status Flag Clear)

Writing 1 to the OSTCLR bit clears the NMISR.OSTST flag. This bit is read as 0.

NMICLR bit (NMI Pin Interrupt Status Flag Clear)

Writing 1 to the NMICLR bit clears the NMISR.NMIST flag. This bit is read as 0.

RPECLR bit (SRAM Parity Error Interrupt Status Flag Clear)

Writing 1 to the RPECLR bit clears the NMISR.RPEST flag. This bit is read as 0.

RECCCLR bit (SRAM ECC Error Interrupt Status Flag Clear)

Writing 1 to the RECCCLR bit clears the NMISR.RECCST flag. This bit is read as 0.

BUSMCLR bit (Bus Master MPU Error Interrupt Status Flag Clear)

Writing 1 to the BUSMCLR bit clears the NMISR.BUSMST flag. This bit is read as 0.

TZFCLR bit (TrustZone Filter Error Interrupt Status Flag Clear)

Writing 1 to the TZFCLR bit clears the NMISR.TZFST flag. This bit is read as 0.

CPECLR bit (Cache RAM Parity Error Interrupt Status Flag Clear)

Writing 1 to the CPECLR bit clears the NMISR.CPEST flag. This bit is read as 0.

12.2.13 NMICR : NMI Pin Interrupt Control Register

Base address: ICU = 0x4000_6000

Offset address: 0x100

Bit position:	7	6	5	4	3	2	1	0
Bit field:	NFLTEN	—	NFCLKSEL[1:0]		—	—	—	NMIMD
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	NMIMD	NMI Detection Set 0: Falling edge 1: Rising edge	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
5:4	NFCLKSEL[1:0]	NMI Digital Filter Sampling Clock Select 0 0: PCLKB 0 1: PCLKB/8 1 0: PCLKB/32 1 1: PCLKB/64	R/W
6	—	This bit is read as 0. The write value should be 0.	R/W
7	NFLTEN	NMI Digital Filter Enable 0: Disabled. 1: Enabled.	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Change the NMICR register settings before enabling NMI pin interrupts, that is, before setting NMIER.NMIEN to 1.

NMIMD bit (NMI Detection Set)

The NMIMD bit selects the detection sensing method for the NMI pin interrupts.

NFCLKSEL[1:0] bits (NMI Digital Filter Sampling Clock Select)

The NFCLKSEL[1:0] bits select the digital filter sampling clock for the NMI pin interrupts, selectable to:

- PCLKB (every cycle)
- PCLKB/8 (once every 8 cycles)
- PCLKB/32 (once every 32 cycles)
- PCLKB/64 (once every 64 cycles)

For details of the digital filter, see section 12.5.5. Digital Filter.

NFLTEN bit (NMI Digital Filter Enable)

The NFLTEN bit enables the digital filter used for NMI pin interrupts. The filter is enabled when NFLTEN is 1, and disabled when NFLTEN is 0. The NMI pin level is sampled at the clock cycle specified in NFCLKSEL[1:0]. When the sampled level matches three times, the output level from the digital filter changes. For details of the digital filter, see section 12.5.5. Digital Filter.

12.2.14 IELSRn : ICU Event Link Setting Register n (n = 0 to 95)

Base address: ICU = 0x4000_6000

Offset address: 0x300 + 0x4 × n

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

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

Bit	Symbol	Function	R/W
8:0	IELS[8:0]	ICU Event Link Select 0x00: Disable interrupts to the associated NVIC or DTC module Others: Event signal number to be linked. For details, see section 12.3.2. Event Number.	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W
16	IR	Interrupt Status Flag 0: No interrupt request generated. 1: An interrupt request is generated.	R/W*1
23:17	—	These bits are read as 0. The write value should be 0.	R/W
24	DTCE	DTC Activation Enable 0: DTC activation is disabled. 1: DTC activation is enabled.	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note: This register requires halfword or word access.

Note 1. Writing 1 to the IR flag is prohibited.

The IELSRn register selects the IRQi source used by the NVIC. For details, see Table 12.3. IELSRn corresponds to the NVIC IRQ input source number, where n = 0 to 95.

IELS[8:0] bits (ICU Event Link Select)

The IELS[8:0] bits link an event signal to the associated NVIC or DTC module. Event options are classified into 8 groups (groups 0 to 7). For details, see Table 12.3 and Table 12.4.

IR flag (Interrupt Status Flag)

The IR status flag indicates an individual interrupt request from the event specified in IELS[8:0].

[Setting condition]

When an interrupt request is received from the associated peripheral module or IRQi pin.

[Clearing condition]

- The IR flag is cleared to 0 by writing 0.
- In the case of DTC.DISEL = 0. At the time other than the final transfer end in DTC transfer during DTCE = 1, IR flag repeat set and cleared by Hardware.
- In the case of DTC.DISEL = 1. For DTC transfers during DTCE = 1, the hardware does not clear the IR flag. Should be cleared by the CPU writing 0.

When DTC transfer except last transfer is completed (DTCE bit is changed from 1 to 0).

During DTCE = 1, write 0 to IR register is prohibited.

In the case of level detection, clear of the IR flag should follow the steps below.

1. Negate the input interrupt signal.
2. Read access the peripheral once and wait for 2 clock cycles of the target module clock.
3. Clear the IR flag by writing 0.

DTCE bit (DTC Activation Enable)

When the DTCE bit is set to 1, the associated event is selected as the source for DTC activation.

[Setting condition]

- When 1 is written to the DTCE bit.

[Clearing condition]

- When the specified number of transfers is complete. For chain transfers, when the specified number of transfers for the last chain transfer is complete.
- When 0 is written to the DTCE bit.

Note: Error during DTC transfer

If an error response occurs during DTC transfer, the DTC notifies the ICU that an error has occurred. ICU clears all bits of the target IELSRn (n = 0 to 95). IELSRn that is not the target is not cleared.

Note: DTC transfer error in snooze mode

When an error occurs in DTC transfer in Snooze mode, ICU issues a wakeup request. However, interrupt requests are not issued automatically. See section 16, Data Transfer Controller (DTC) for how to set the interrupt when a DTC error occurs.

12.2.15 DELSRn : DMAC Event Link Setting Register n (n = 0 to 7)

Base address: ICU = 0x4000_6000

Offset address: 0x280 + 0x4 × n

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

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

Bit	Symbol	Function	R/W
8:0	DELS[8:0]	DMAC Event Link Select 0x00: Disable interrupts to the associated DMAC module. Others: Event signal number to be linked. For details, see Table 12.4.	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W
16	IR	DMAC Activation Request Status Flag 0: No DMAC activation request occurred. 1: DMAC activation request occurred.	R/W*1
31:17	—	These bits are read as 0. The write value should be 0.	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note 1. Writing 1 to the IR flag is prohibited.

DELS[8:0] bit (DMAC Event Link Select)

The DELS[8:0] bits link an event signal to the associated DMAC module. Do not set the same event number in multiple DELSRn registers.

IR flag (DMAC Activation Request Status Flag)

The IR flag is the status flag of a DMAC activation request. This flag is associated with the DELS[8:0] bits of this register.

[Setting condition]

The flag is set to 1 when a DMAC activation request is generated from the associated peripheral module or IRQi pin.

[Clearing conditions]

- When 0 is written to the IR flag.
- At the start of a DMA transfer after the DMAC activation request is issued.

Note: The IR flag is automatically cleared after completion of a DMA transfer. Therefore, do not write 0 unless an abort occurs. When 0 is written, DMA transfer operation cannot be guaranteed.

Note: Error during DMAC transfer

If an error response occurs during a DMAC transfer, the DMAC notifies the ICU that an error has occurred.

The ICU clears all bits of the target channel of DELSRn (n = 0 to 7). DELSRn that is not the target channel is not cleared.

12.2.16 SELSR0 : SYS Event Link Setting Register

Base address: ICU = 0x4000_6000

Offset address: 0x200

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

Bit	Symbol	Function	R/W
8:0	SELS[8:0]	SYS Event Link Select 0x00: Disable event output to the associated low-power mode module Others: Event signal number to be linked. For details, see Table 12.4.	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

The SELSR0 register selects the events that wake up the CPU from Snooze mode. You can use only the events listed in Table 12.4 checked as “Canceling Snooze mode”. When ICU_SNZCANCEL is selected in the IELSRn.IELS[8:0] bits, an interrupt is generated that cancels snooze mode.

Caution: For security attribution added to parts related to a series of actions, make sure to match all security attribution so that security holes cannot be created.

About security attribution to be matched

- **Event source to be set to SELSR0.**
- **SELSR0**
- **IELSRn (n = 0 to 95) to receive event No. 45 (ICU_SNZCANCEL).**
- **NVIC internal registers in the CPU of the interrupt specified in the previous item.**
- **Interrupt Handler.**

12.2.17 WUPEN0 : Wake Up Interrupt Enable Register 0

Base address: ICU = 0x4000_6000

Offset address: 0x1A0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	IIC0WUPEN	AGT1CBWUPEN	AGT1CAWUPEN	AGT1UDWUPEN	—	—	—	—	—	—	—	—	LVD2WUPEN	LVD1WUPEN	KEYWUPEN	IWDTWUPEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
15:0	IRQWUPEN[15:0]	IRQn Interrupt Software Standby/Snooze Mode Returns Enable bit (n = 0 to 15) 0: Software Standby/Snooze Mode returns by IRQn interrupt is disabled 1: Software Standby/Snooze Mode returns by IRQn interrupt is enabled*1	R/W
16	IWDTWUPEN	IWDT Interrupt Software Standby/Snooze Mode Returns Enable bit 0: Software Standby/Snooze Mode returns by IWDT interrupt is disabled 1: Software Standby/Snooze Mode returns by IWDT interrupt is enabled	R/W
17	KEYWUPEN	Key interrupt S/W standby returns enable bit 0: S/W standby returns by KEY interrupt is disabled 1: S/W standby returns by KEY interrupt is enabled	R/W
18	LVD1WUPEN	LVD1 Interrupt Software Standby/Snooze Mode Returns Enable bit 0: Software Standby/Snooze Mode returns by LVD1 interrupt is disabled 1: Software Standby/Snooze Mode returns by LVD1 interrupt is enabled	R/W
19	LVD2WUPEN	LVD2 Interrupt Software Standby/Snooze Mode Returns Enable bit 0: Software Standby/Snooze Mode returns by LVD2 interrupt is disabled 1: Software Standby/Snooze Mode returns by LVD2 interrupt is enabled	R/W
27:20	—	These bits are read as 0. The write value should be 0.	R/W
28	AGT1UDWUPEN	AGT1 Underflow Interrupt Software Standby/Snooze Mode Returns Enable bit 0: Software Standby/Snooze Mode returns by AGT1 underflow interrupt is disabled 1: Software Standby/Snooze Mode returns by AGT1 underflow interrupt is enabled	R/W
29	AGT1CAWUPEN	AGT1 Compare Match A Interrupt Software Standby/Snooze Mode Returns Enable bit 0: Software Standby/Snooze Mode returns by AGT1 compare match A interrupt is disabled 1: Software Standby/Snooze Mode returns by AGT1 compare match A interrupt is enabled	R/W
30	AGT1CBWUPEN	AGT1 Compare Match B Interrupt Software Standby/Snooze Mode Returns Enable bit 0: Software Standby/Snooze Mode returns by AGT1 compare match B interrupt is disabled 1: Software Standby/Snooze Mode returns by AGT1 compare match B interrupt is enabled	R/W
31	IIC0WUPEN	IIC0 Address Match Interrupt Software Standby/Snooze Mode Returns Enable bit 0: Software Standby/Snooze Mode returns by IIC0 address match interrupt is disabled 1: Software Standby/Snooze Mode returns by IIC0 address match interrupt is enabled	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note 1. Description is a description of each bit.

The security attribution of this register is set for each wakeup event.

To avoid the occurrence of a security hole, the target event of a wakeup and the security attribution added to this bit must match.

12.3 Vector Table

The ICU detects maskable and non-maskable interrupts. Interrupt priorities are set up in the Arm NVIC. For information about these registers, see section 12.9. Reference.

12.3.1 Interrupt Vector Table

Table 12.3 describes the interrupt vector table. The interrupt vector addresses conform to the NVIC specifications.

Table 12.3 Interrupt vector table (1 of 3)

Exception number	IRQ number	Vector offset	Source	Description
0	—	0x000	Arm	Initial stack pointer
1	—	0x004	Arm	Initial program counter (reset vector)
2	—	0x008	Arm	Non-Maskable Interrupt (NMI)
3	—	0x00C	Arm	Hard Fault
4	—	0x010	Arm	MemManage fault
5	—	0x014	Arm	BusFault
6	—	0x018	Arm	UsageFault
7	—	0x01C	Arm	SecureFault
8	—	0x020	Arm	Reserved
9	—	0x024	Arm	Reserved
10	—	0x028	Arm	Reserved
11	—	0x02C	Arm	Supervisor Call (SVCall)
12	—	0x030	Arm	DebugMonitor
13	—	0x034	Arm	Reserved
14	—	0x038	Arm	Pendable request for system service (PendableSrvReq)
15	—	0x03C	Arm	System Tick Timer (SysTick)
16	0	0x040	ICU.IELSR0	Event selected in the ICU.IELSR0 register
17	1	0x044	ICU.IELSR1	Event selected in the ICU.IELSR1 register
18	2	0x048	ICU.IELSR2	Event selected in the ICU.IELSR2 register
19	3	0x04C	ICU.IELSR3	Event selected in the ICU.IELSR3 register
20	4	0x050	ICU.IELSR4	Event selected in the ICU.IELSR4 register
21	5	0x054	ICU.IELSR5	Event selected in the ICU.IELSR5 register
22	6	0x058	ICU.IELSR6	Event selected in the ICU.IELSR6 register
23	7	0x05C	ICU.IELSR7	Event selected in the ICU.IELSR7 register
24	8	0x060	ICU.IELSR8	Event selected in the ICU.IELSR8 register
25	9	0x064	ICU.IELSR9	Event selected in the ICU.IELSR9 register
26	10	0x068	ICU.IELSR10	Event selected in the ICU.IELSR10 register
27	11	0x06C	ICU.IELSR11	Event selected in the ICU.IELSR11 register
28	12	0x070	ICU.IELSR12	Event selected in the ICU.IELSR12 register
29	13	0x074	ICU.IELSR13	Event selected in the ICU.IELSR13 register
30	14	0x078	ICU.IELSR14	Event selected in the ICU.IELSR14 register
31	15	0x07C	ICU.IELSR15	Event selected in the ICU.IELSR15 register

Table 12.3 Interrupt vector table (2 of 3)

Exception number	IRQ number	Vector offset	Source	Description
32	16	0x080	ICU.IELSR16	Event selected in the ICU.IELSR16 register
33	17	0x084	ICU.IELSR17	Event selected in the ICU.IELSR17 register
34	18	0x088	ICU.IELSR18	Event selected in the ICU.IELSR18 register
35	19	0x08C	ICU.IELSR19	Event selected in the ICU.IELSR19 register
36	20	0x090	ICU.IELSR20	Event selected in the ICU.IELSR20 register
37	21	0x094	ICU.IELSR21	Event selected in the ICU.IELSR21 register
38	22	0x098	ICU.IELSR22	Event selected in the ICU.IELSR22 register
39	23	0x09C	ICU.IELSR23	Event selected in the ICU.IELSR23 register
40	24	0x0A0	ICU.IELSR24	Event selected in the ICU.IELSR24 register
41	25	0x0A4	ICU.IELSR25	Event selected in the ICU.IELSR25 register
42	26	0x0A8	ICU.IELSR26	Event selected in the ICU.IELSR26 register
43	27	0x0AC	ICU.IELSR27	Event selected in the ICU.IELSR27 register
44	28	0x0B0	ICU.IELSR28	Event selected in the ICU.IELSR28 register
45	29	0x0B4	ICU.IELSR29	Event selected in the ICU.IELSR29 register
46	30	0x0B8	ICU.IELSR30	Event selected in the ICU.IELSR30 register
47	31	0x0BC	ICU.IELSR31	Event selected in the ICU.IELSR31 register
48	32	0x0C0	ICU.IELSR32	Event selected in the ICU.IELSR32 register
49	33	0x0C4	ICU.IELSR33	Event selected in the ICU.IELSR33 register
50	34	0x0C8	ICU.IELSR34	Event selected in the ICU.IELSR34 register
51	35	0x0CC	ICU.IELSR35	Event selected in the ICU.IELSR35 register
52	36	0x0D0	ICU.IELSR36	Event selected in the ICU.IELSR36 register
53	37	0x0D4	ICU.IELSR37	Event selected in the ICU.IELSR37 register
54	38	0x0D8	ICU.IELSR38	Event selected in the ICU.IELSR38 register
55	39	0x0DC	ICU.IELSR39	Event selected in the ICU.IELSR39 register
56	40	0x0E0	ICU.IELSR40	Event selected in the ICU.IELSR40 register
57	41	0x0E4	ICU.IELSR41	Event selected in the ICU.IELSR41 register
58	42	0x0E8	ICU.IELSR42	Event selected in the ICU.IELSR42 register
59	43	0x0EC	ICU.IELSR43	Event selected in the ICU.IELSR43 register
60	44	0x0F0	ICU.IELSR44	Event selected in the ICU.IELSR44 register
61	45	0x0F4	ICU.IELSR45	Event selected in the ICU.IELSR45 register
62	46	0x0F8	ICU.IELSR46	Event selected in the ICU.IELSR46 register
63	47	0x0FC	ICU.IELSR47	Event selected in the ICU.IELSR47 register
64	48	0x100	ICU.IELSR48	Event selected in the ICU.IELSR48 register
65	49	0x104	ICU.IELSR49	Event selected in the ICU.IELSR49 register
66	50	0x108	ICU.IELSR50	Event selected in the ICU.IELSR50 register
67	51	0x10C	ICU.IELSR51	Event selected in the ICU.IELSR51 register
68	52	0x110	ICU.IELSR52	Event selected in the ICU.IELSR52 register
69	53	0x114	ICU.IELSR53	Event selected in the ICU.IELSR53 register
70	54	0x118	ICU.IELSR54	Event selected in the ICU.IELSR54 register
71	55	0x11C	ICU.IELSR55	Event selected in the ICU.IELSR55 register
72	56	0x120	ICU.IELSR56	Event selected in the ICU.IELSR56 register

Table 12.3 Interrupt vector table (3 of 3)

Exception number	IRQ number	Vector offset	Source	Description
73	57	0x124	ICU.IELSR57	Event selected in the ICU.IELSR57 register
74	58	0x128	ICU.IELSR58	Event selected in the ICU.IELSR58 register
75	59	0x12C	ICU.IELSR59	Event selected in the ICU.IELSR59 register
76	60	0x130	ICU.IELSR60	Event selected in the ICU.IELSR60 register
77	61	0x134	ICU.IELSR61	Event selected in the ICU.IELSR61 register
78	62	0x138	ICU.IELSR62	Event selected in the ICU.IELSR62 register
79	63	0x13C	ICU.IELSR63	Event selected in the ICU.IELSR63 register
80	64	0x140	ICU.IELSR64	Event selected in the ICU.IELSR64 register
81	65	0x144	ICU.IELSR65	Event selected in the ICU.IELSR65 register
82	66	0x148	ICU.IELSR66	Event selected in the ICU.IELSR66 register
83	67	0x14C	ICU.IELSR67	Event selected in the ICU.IELSR67 register
84	68	0x150	ICU.IELSR68	Event selected in the ICU.IELSR68 register
85	69	0x154	ICU.IELSR69	Event selected in the ICU.IELSR69 register
86	70	0x158	ICU.IELSR70	Event selected in the ICU.IELSR70 register
87	71	0x15C	ICU.IELSR71	Event selected in the ICU.IELSR71 register
88	72	0x160	ICU.IELSR72	Event selected in the ICU.IELSR72 register
89	73	0x164	ICU.IELSR73	Event selected in the ICU.IELSR73 register
90	74	0x168	ICU.IELSR74	Event selected in the ICU.IELSR74 register
91	75	0x16C	ICU.IELSR75	Event selected in the ICU.IELSR75 register
92	76	0x170	ICU.IELSR76	Event selected in the ICU.IELSR76 register
93	77	0x174	ICU.IELSR77	Event selected in the ICU.IELSR77 register
94	78	0x178	ICU.IELSR78	Event selected in the ICU.IELSR78 register
95	79	0x17C	ICU.IELSR79	Event selected in the ICU.IELSR79 register
96	80	0x180	ICU.IELSR80	Event selected in the ICU.IELSR80 register
97	81	0x184	ICU.IELSR81	Event selected in the ICU.IELSR81 register
98	82	0x188	ICU.IELSR82	Event selected in the ICU.IELSR82 register
99	83	0x18C	ICU.IELSR83	Event selected in the ICU.IELSR83 register
100	84	0x190	ICU.IELSR84	Event selected in the ICU.IELSR84 register
101	85	0x194	ICU.IELSR85	Event selected in the ICU.IELSR85 register
102	86	0x198	ICU.IELSR86	Event selected in the ICU.IELSR86 register
103	87	0x19C	ICU.IELSR87	Event selected in the ICU.IELSR87 register
104	88	0x1A0	ICU.IELSR88	Event selected in the ICU.IELSR88 register
105	89	0x1A4	ICU.IELSR89	Event selected in the ICU.IELSR89 register
106	90	0x1A8	ICU.IELSR90	Event selected in the ICU.IELSR90 register
107	91	0x1AC	ICU.IELSR91	Event selected in the ICU.IELSR91 register
108	92	0x1B0	ICU.IELSR92	Event selected in the ICU.IELSR92 register
109	93	0x1B4	ICU.IELSR93	Event selected in the ICU.IELSR93 register
110	94	0x1B8	ICU.IELSR94	Event selected in the ICU.IELSR94 register
111	95	0x1BC	ICU.IELSR95	Event selected in the ICU.IELSR95 register

12.3.2 Event Number

The following table lists heading details for Table 12.4, which describes each event number.

Heading	Description
Interrupt request source	Name of the source generating the interrupt request
Name	Name of the interrupt
Connect to NVIC	" ✓" indicates the interrupt can be used as a CPU interrupt
Invoke DTC	" ✓" indicates the interrupt can be used to request DTC activation
Invoke DMAC	"✓" indicates the interrupt can be used to request DMAC activation
Canceling Snooze	" ✓" indicates the interrupt can be used to request a return from Snooze mode
Canceling Software Standby	" ✓" indicates the interrupt can be used to request a return from Software Standby mode
Canceling Deep Software Standby	"✓" indicates the interrupt can be used to request a return from Deep Software Standby mode

Table 12.4 Event table (1 of 9)

Event number	Interrupt request source	Name	IELSRn		DELSRn	Canceling Snooze	Canceling Software Standby	Canceling Deep Software Standby
			Connect to NVIC	Invoke DTC	Invoke DMAC			
0x001	Port	PORT_IRQ0	✓	✓	✓	✓	✓	✓
0x002		PORT_IRQ1	✓	✓	✓	✓	✓	✓
0x003		PORT_IRQ2	✓	✓	✓	✓	✓	✓
0x004		PORT_IRQ3	✓	✓	✓	✓	✓	✓
0x005		PORT_IRQ4	✓	✓	✓	✓	✓	✓
0x006		PORT_IRQ5	✓	✓	✓	✓	✓	✓
0x007		PORT_IRQ6	✓	✓	✓	✓	✓	✓
0x008		PORT_IRQ7	✓	✓	✓	✓	✓	✓
0x009		PORT_IRQ8	✓	✓	✓	✓	✓	✓
0x00A		PORT_IRQ9	✓	✓	✓	✓	✓	✓
0x00B		PORT_IRQ10	✓	✓	✓	✓	✓	✓
0x00C		PORT_IRQ11	✓	✓	✓	✓	✓	✓
0x00D		PORT_IRQ12	✓	✓	✓	✓	✓	✓
0x00E		PORT_IRQ13	✓	✓	✓	✓	✓	✓
0x00F		PORT_IRQ14	✓	✓	✓	✓	✓	✓
0x010		PORT_IRQ15	✓	✓	✓	✓	✓	✓
0x011	IIRFA	IIRFA_ORDY0	✓	—	—	—	—	—
0x012		IIRFA_ORDY1	✓	—	—	—	—	—
0x013		IIRFA_ORDY2	✓	—	—	—	—	—
0x014		IIRFA_ORDY3	✓	—	—	—	—	—
0x015		IIRFA_CPRCF0	✓	—	—	—	—	—
0x016		IIRFA_CPRCF1	✓	—	—	—	—	—
0x017		IIRFA_CPRCF2	✓	—	—	—	—	—
0x018		IIRFA_CPRCF3	✓	—	—	—	—	—
0x019		IIRFA_ERR	✓	—	—	—	—	—
0x020	DMAC0	DMAC0_INT	✓	✓	—	—	—	—
0x021	DMAC1	DMAC1_INT	✓	✓	—	—	—	—
0x022	DMAC2	DMAC2_INT	✓	✓	—	—	—	—

Table 12.4 Event table (2 of 9)

Event number	Interrupt request source	Name	IELSRn		DELSRn	Canceling Snooze	Canceling Software Standby	Canceling Deep Software Standby
			Connect to NVIC	Invoke DTC	Invoke DMAC			
0x023	DMAC3	DMAC3_INT	✓	✓	—	—	—	—
0x024	DMAC4	DMAC4_INT	✓	✓	—	—	—	—
0x025	DMAC5	DMAC5_INT	✓	✓	—	—	—	—
0x026	DMAC6	DMAC6_INT	✓	✓	—	—	—	—
0x027	DMAC7	DMAC7_INT	✓	✓	—	—	—	—
0x029	DTC	DTC_COMPLETE	✓	—	—	✓*1	—	—
0x02B	DMAC/DTC	DMA_TRANSERR	✓	—	—	✓	—	—
0x02D	ICU	ICU_SNZCANCEL	✓	—	—	✓	—	—
0x030	FCU	FCU_FIFERR	✓	—	—	—	—	—
0x031		FCU_FRDYI	✓	—	—	—	—	—
0x038	LVD	LVD_LVD1	✓	—	—	✓	✓	✓
0x039		LVD_LVD2	✓	—	—	✓	✓	✓
0x03B	MOSC	MOSC_STOP	✓	—	—	—	—	—
0x03C	LPW	SYSTEM_SNZREQ	—	✓	—	—	—	—
0x040	AGT0	AGT0_AGTI	✓	✓	✓	—	—	—
0x041		AGT0_AGTCMAI	✓	✓	✓	—	—	—
0x042		AGT0_AGTCMBI	✓	✓	✓	—	—	—
0x043	AGT1	AGT1_AGTI	✓	✓	✓	✓	✓	—
0x044		AGT1_AGTCMAI	✓	✓	✓	✓	✓	—
0x045		AGT1_AGTCMBI	✓	✓	✓	✓	✓	—
0x052	IWDT	IWDT_NMIUNDF	✓	—	—	✓	✓	—
0x053	WDT	WDT_NMIUNDF	✓	—	—	—	—	—
0x059	CANFD	CAN_RXF	✓	—	—	—	—	—
0x05A		CAN_GLERR	✓	—	—	—	—	—
0x05B		CAN_RF_DMAREQ0	✓	✓	✓	—	—	—
0x05C		CAN_RF_DMAREQ1	✓	✓	✓	—	—	—
0x063		CAN0_TX	✓	—	—	—	—	—
0x064		CAN0_CHERR	✓	—	—	—	—	—
0x065		CAN0_COMFRX	✓	—	—	—	—	—
0x066		CAN0_CF_DMAREQ	✓	✓	✓	—	—	—
0x067		CAN0_RXMB	✓	—	—	—	—	—
0x08F	ACMPHS	ACMP_HS0	✓	—	—	—	—	—
0x090		ACMP_HS1	✓	—	—	—	—	—
0x091		ACMP_HS2	✓	—	—	—	—	—
0x092		ACMP_HS3	✓	—	—	—	—	—
0x09D	KINT	KEY_INTKR	✓	—	—	✓*2	✓*2	—
0x09E	CAC	CAC_FERRI	✓	—	—	—	—	—
0x09F		CAC_MENDI	✓	—	—	—	—	—
0x0A0		CAC_OVFI	✓	—	—	—	—	—

Table 12.4 Event table (3 of 9)

Event number	Interrupt request source	Name	IELSRn		DELSRn	Canceling Snooze	Canceling Software Standby	Canceling Deep Software Standby
			Connect to NVIC	Invoke DTC	Invoke DMAC			
0x0B1	PORT	IOPORT_GROUPB	✓	✓*3	✓*3	—	—	—
0x0B2		IOPORT_GROUPC	✓	✓*3	✓*3	—	—	—
0x0B3		IOPORT_GROUPD	✓	✓*3	✓*3	—	—	—
0x0B4		IOPORT_GROUPE	✓	✓*3	✓*3	—	—	—
0x0B5	ELC	ELC_SWEVT0	✓*4	✓	—	—	—	—
0x0B6		ELC_SWEVT1	✓*4	✓	—	—	—	—
0x0B7	POEG	POEG_GROUPA	✓	—	—	—	—	—
0x0B8		POEG_GROUPB	✓	—	—	—	—	—
0x0B9		POEG_GROUPC	✓	—	—	—	—	—
0x0BA		POEG_GROUPD	✓	—	—	—	—	—
0x0C0	GPT0	GPT0_CCMPA	✓	✓	✓	—	—	—
0x0C1		GPT0_CCMPB	✓	✓	✓	—	—	—
0x0C2		GPT0_CMPC	✓	✓	✓	—	—	—
0x0C3		GPT0_CMPD	✓	✓	✓	—	—	—
0x0C4		GPT0_CMPE	✓	✓	✓	—	—	—
0x0C5		GPT0_CMPF	✓	✓	✓	—	—	—
0x0C6		GPT0_OVF	✓	✓	✓	—	—	—
0x0C7		GPT0_UDF	✓	✓	✓	—	—	—
0x0C8		GPT0_PC	✓	✓	✓	—	—	—
0x0CA		GPT0_ADTRGA	✓	✓	✓	—	—	—
0x0CB		GPT0_ADTRGB	✓	✓	✓	—	—	—
0x0CC	GPT1	GPT1_CCMPA	✓	✓	✓	—	—	—
0x0CD		GPT1_CCMPB	✓	✓	✓	—	—	—
0x0CE		GPT1_CMPC	✓	✓	✓	—	—	—
0x0CF		GPT1_CMPD	✓	✓	✓	—	—	—
0x0D0		GPT1_CMPE	✓	✓	✓	—	—	—
0x0D1		GPT1_CMPF	✓	✓	✓	—	—	—
0x0D2		GPT1_OVF	✓	✓	✓	—	—	—
0x0D3		GPT1_UDF	✓	✓	✓	—	—	—
0x0D4		GPT1_PC	✓	✓	✓	—	—	—
0x0D6		GPT1_ADTRGA	✓	✓	✓	—	—	—
0x0D7		GPT1_ADTRGB	✓	✓	✓	—	—	—

Table 12.4 Event table (4 of 9)

Event number	Interrupt request source	Name	IELSRn		DELSRn	Canceling Snooze	Canceling Software Standby	Canceling Deep Software Standby
			Connect to NVIC	Invoke DTC	Invoke DMAC			
0x0D8	GPT2	GPT2_CCMPA	✓	✓	✓	—	—	—
0x0D9		GPT2_CCMPB	✓	✓	✓	—	—	—
0x0DA		GPT2_CMPC	✓	✓	✓	—	—	—
0x0DB		GPT2_CMPD	✓	✓	✓	—	—	—
0x0DC		GPT2_CMPE	✓	✓	✓	—	—	—
0x0DD		GPT2_CMPF	✓	✓	✓	—	—	—
0x0DE		GPT2_OVF	✓	✓	✓	—	—	—
0x0DF		GPT2_UDF	✓	✓	✓	—	—	—
0x0E0		GPT2_PC	✓	✓	✓	—	—	—
0x0E2		GPT2_ADTRGA	✓	✓	✓	—	—	—
0x0E3		GPT2_ADTRGB	✓	✓	✓	—	—	—
0x0E4	GPT3	GPT3_CCMPA	✓	✓	✓	—	—	—
0x0E5		GPT3_CCMPB	✓	✓	✓	—	—	—
0x0E6		GPT3_CMPC	✓	✓	✓	—	—	—
0x0E7		GPT3_CMPD	✓	✓	✓	—	—	—
0x0E8		GPT3_CMPE	✓	✓	✓	—	—	—
0x0E9		GPT3_CMPF	✓	✓	✓	—	—	—
0x0EA		GPT3_OVF	✓	✓	✓	—	—	—
0x0EB		GPT3_UDF	✓	✓	✓	—	—	—
0x0EC		GPT3_PC	✓	✓	✓	—	—	—
0x0EE		GPT3_ADTRGA	✓	✓	✓	—	—	—
0x0EF		GPT3_ADTRGB	✓	✓	✓	—	—	—
0x0F0	GPT4	GPT4_CCMPA	✓	✓	✓	—	—	—
0x0F1		GPT4_CCMPB	✓	✓	✓	—	—	—
0x0F2		GPT4_CMPC	✓	✓	✓	—	—	—
0x0F3		GPT4_CMPD	✓	✓	✓	—	—	—
0x0F4		GPT4_CMPE	✓	✓	✓	—	—	—
0x0F5		GPT4_CMPF	✓	✓	✓	—	—	—
0x0F6		GPT4_OVF	✓	✓	✓	—	—	—
0x0F7		GPT4_UDF	✓	✓	✓	—	—	—
0x0FA		GPT4_ADTRGA	✓	✓	✓	—	—	—
0x0FB		GPT4_ADTRGB	✓	✓	✓	—	—	—

Table 12.4 Event table (5 of 9)

Event number	Interrupt request source	Name	IELSRn		DELSRn	Canceling Snooze	Canceling Software Standby	Canceling Deep Software Standby
			Connect to NVIC	Invoke DTC	Invoke DMAC			
0x0FC	GPT5	GPT5_CCMPA	✓	✓	✓	—	—	—
0x0FD		GPT5_CCMPB	✓	✓	✓	—	—	—
0x0FE		GPT5_CMPC	✓	✓	✓	—	—	—
0x0FF		GPT5_CMPD	✓	✓	✓	—	—	—
0x100		GPT5_CMPE	✓	✓	✓	—	—	—
0x101		GPT5_CMPF	✓	✓	✓	—	—	—
0x102		GPT5_OVF	✓	✓	✓	—	—	—
0x103		GPT5_UDF	✓	✓	✓	—	—	—
0x106		GPT5_ADTRGA	✓	✓	✓	—	—	—
0x107		GPT5_ADTRGB	✓	✓	✓	—	—	—
0x108	GPT6	GPT6_CCMPA	✓	✓	✓	—	—	—
0x109		GPT6_CCMPB	✓	✓	✓	—	—	—
0x10A		GPT6_CMPC	✓	✓	✓	—	—	—
0x10B		GPT6_CMPD	✓	✓	✓	—	—	—
0x10C		GPT6_CMPE	✓	✓	✓	—	—	—
0x10D		GPT6_CMPF	✓	✓	✓	—	—	—
0x10E		GPT6_OVF	✓	✓	✓	—	—	—
0x10F		GPT6_UDF	✓	✓	✓	—	—	—
0x112		GPT6_ADTRGA	✓	✓	✓	—	—	—
0x113		GPT6_ADTRGB	✓	✓	✓	—	—	—
0x114	GPT7	GPT7_CCMPA	✓	✓	✓	—	—	—
0x115		GPT7_CCMPB	✓	✓	✓	—	—	—
0x116		GPT7_CMPC	✓	✓	✓	—	—	—
0x117		GPT7_CMPD	✓	✓	✓	—	—	—
0x118		GPT7_CMPE	✓	✓	✓	—	—	—
0x119		GPT7_CMPF	✓	✓	✓	—	—	—
0x11A		GPT7_OVF	✓	✓	✓	—	—	—
0x11B		GPT7_UDF	✓	✓	✓	—	—	—
0x11E		GPT7_ADTRGA	✓	✓	✓	—	—	—
0x11F		GPT7_ADTRGB	✓	✓	✓	—	—	—
0x120	GPT8	GPT8_CCMPA	✓	✓	✓	—	—	—
0x121		GPT8_CCMPB	✓	✓	✓	—	—	—
0x122		GPT8_CMPC	✓	✓	✓	—	—	—
0x123		GPT8_CMPD	✓	✓	✓	—	—	—
0x124		GPT8_CMPE	✓	✓	✓	—	—	—
0x125		GPT8_CMPF	✓	✓	✓	—	—	—
0x126		GPT8_OVF	✓	✓	✓	—	—	—
0x127		GPT8_UDF	✓	✓	✓	—	—	—
0x12A		GPT8_ADTRGA	✓	✓	✓	—	—	—
0x12B		GPT8_ADTRGB	✓	✓	✓	—	—	—

Table 12.4 Event table (6 of 9)

Event number	Interrupt request source	Name	IELSRn		DELSRn	Canceling Snooze	Canceling Software Standby	Canceling Deep Software Standby
			Connect to NVIC	Invoke DTC	Invoke DMAC			
0x12C	GPT9	GPT9_CCMPA	✓	✓	✓	—	—	—
0x12D		GPT9_CCMPB	✓	✓	✓	—	—	—
0x12E		GPT9_CMPC	✓	✓	✓	—	—	—
0x12F		GPT9_CMPD	✓	✓	✓	—	—	—
0x130		GPT9_CMPE	✓	✓	✓	—	—	—
0x131		GPT9_CMPF	✓	✓	✓	—	—	—
0x132		GPT9_OVF	✓	✓	✓	—	—	—
0x133		GPT9_UDF	✓	✓	✓	—	—	—
0x136		GPT9_ADTRGA	✓	✓	✓	—	—	—
0x137		GPT9_ADTRGB	✓	✓	✓	—	—	—
0x138	GPT	GPT_UVWEDGE	✓	—	—	—	—	—
0x140	IIC0	IIC0_RX	✓	✓	✓	—	—	—
0x141		IIC0_TX	✓	✓	✓	—	—	—
0x142		IIC0_TEND	✓	—	—	—	—	—
0x143		IIC0_EEI	✓	—	—	—	—	—
0x144		IIC0_WU	✓	—	—	✓	✓	—
0x146	IIC1	IIC1_RX	✓	✓	✓	—	—	—
0x147		IIC1_TX	✓	✓	✓	—	—	—
0x148		IIC1_TEND	✓	—	—	—	—	—
0x149		IIC1_EEI	✓	—	—	—	—	—

Table 12.4 Event table (7 of 9)

Event number	Interrupt request source	Name	IELSRn		DELSRn	Canceling Snooze	Canceling Software Standby	Canceling Deep Software Standby
			Connect to NVIC	Invoke DTC	Invoke DMAC			
0x157	ADC	ADC_LIMCLPI	✓	—	—	—	—	—
0x158		ADC_FIFOOVF	✓	—	—	—	—	—
0x159		ADC_ADI0	✓	✓	✓	—	—	—
0x15A		ADC_ADI1	✓	✓	✓	—	—	—
0x15B		ADC_ADI2	✓	✓	✓	—	—	—
0x15C		ADC_CMPI0	✓	—	—	—	—	—
0x15D		ADC_CMPI1	✓	—	—	—	—	—
0x15E		ADC_CCMPM0	✓	✓	✓	—	—	—
0x160		ADC_ERR0	✓	—	—	—	—	—
0x161		ADC_RESOVF0	✓	—	—	—	—	—
0x163		ADC_CALEND0	✓	—	—	—	—	—
0x164		ADC_FIFOREQ0	✓	✓	✓	—	—	—
0x165		ADC_FIFOREQ1	✓	✓	✓	—	—	—
0x166		ADC_FIFOREQ2	✓	✓	✓	—	—	—
0x167		ADC_ADI3	✓	✓	✓	—	—	—
0x168		ADC_ADI4	✓	✓	✓	—	—	—
0x169		ADC_ADI5678	✓	✓	✓	—	—	—
0x16A		ADC_CMPI2	✓	—	—	—	—	—
0x16B		ADC_CMPI3	✓	—	—	—	—	—
0x16C		ADC_CCMPM1	✓	✓	✓	—	—	—
0x16E		ADC_ERR1	✓	—	—	—	—	—
0x16F		ADC_RESOVF1	✓	—	—	—	—	—
0x171		ADC_CALEND1	✓	—	—	—	—	—
0x172		ADC_FIFOREQ3	✓	✓	✓	—	—	—
0x173		ADC_FIFOREQ4	✓	✓	✓	—	—	—
0x174		ADC_FIFOREQ5678	✓	✓	✓	—	—	—
0x18D	SCI0	SCI0_RXI	✓	✓	✓	—	—	—
0x18E		SCI0_TXI	✓	✓	✓	—	—	—
0x18F		SCI0_TEI	✓	—	—	—	—	—
0x190		SCI0_ERI	✓	—	—	—	—	—
0x191		SCI0_AED	✓	—	—	—	—	—
0x192		SCI0_BFD	✓	—	—	—	—	—
0x193		SCI0_AM	✓	—	—	✓*1	—	—
0x195	SCI1	SCI1_RXI	✓	✓	✓	—	—	—
0x196		SCI1_TXI	✓	✓	✓	—	—	—
0x197		SCI1_TEI	✓	—	—	—	—	—
0x198		SCI1_ERI	✓	—	—	—	—	—
0x199		SCI1_AED	✓	—	—	—	—	—
0x19A		SCI1_BFD	✓	—	—	—	—	—
0x19B		SCI1_AM	✓	—	—	—	—	—

Table 12.4 Event table (8 of 9)

Event number	Interrupt request source	Name	IELSRn		DELSRn	Canceling Snooze	Canceling Software Standby	Canceling Deep Software Standby
			Connect to NVIC	Invoke DTC	Invoke DMAC			
0x19C	SCI2	SCI2_RXI	✓	✓	✓	—	—	—
0x19D		SCI2_TXI	✓	✓	✓	—	—	—
0x19E		SCI2_TEI	✓	—	—	—	—	—
0x19F		SCI2_ERI	✓	—	—	—	—	—
0x1A0		SCI2_AED	✓	—	—	—	—	—
0x1A1		SCI2_BFD	✓	—	—	—	—	—
0x1A2		SCI2_AM	✓	—	—	—	—	—
0x1A3	SCI3	SCI3_RXI	✓	✓	✓	—	—	—
0x1A4		SCI3_TXI	✓	✓	✓	—	—	—
0x1A5		SCI3_TEI	✓	—	—	—	—	—
0x1A6		SCI3_ERI	✓	—	—	—	—	—
0x1A7		SCI3_AED	✓	—	—	—	—	—
0x1A8		SCI3_BFD	✓	—	—	—	—	—
0x1A9		SCI3_AM	✓	—	—	—	—	—
0x1AA	SCI4	SCI4_RXI	✓	✓	✓	—	—	—
0x1AB		SCI4_TXI	✓	✓	✓	—	—	—
0x1AC		SCI4_TEI	✓	—	—	—	—	—
0x1AD		SCI4_ERI	✓	—	—	—	—	—
0x1AE		SCI4_AED	✓	—	—	—	—	—
0x1AF		SCI4_BFD	✓	—	—	—	—	—
0x1B0		SCI4_AM	✓	—	—	—	—	—
0x1B1	SCI9	SCI9_RXI	✓	✓	✓	—	—	—
0x1B2		SCI9_TXI	✓	✓	✓	—	—	—
0x1B3		SCI9_TEI	✓	—	—	—	—	—
0x1B4		SCI9_ERI	✓	—	—	—	—	—
0x1B5		SCI9_AED	✓	—	—	—	—	—
0x1B6		SCI9_BFD	✓	—	—	—	—	—
0x1B7		SCI9_AM	✓	—	—	—	—	—
0x1C4	SPI0	SPI0_SPRI	✓	✓	✓	—	—	—
0x1C5		SPI0_SPTI	✓	✓	✓	—	—	—
0x1C6		SPI0_SPII	✓	—	—	—	—	—
0x1C7		SPI0_SPEI	✓	—	—	—	—	—
0x1C8		SPI0_SPCEND	✓	—	—	—	—	—
0x1C9	SPI1	SPI1_SPRI	✓	✓	✓	—	—	—
0x1CA		SPI1_SPTI	✓	✓	✓	—	—	—
0x1CB		SPI1_SPII	✓	—	—	—	—	—
0x1CC		SPI1_SPEI	✓	—	—	—	—	—
0x1CD		SPI1_SPCEND	✓	—	—	—	—	—
0x1D0	CANFD ECC	CAN_MRAM_ERI	✓	—	—	—	—	—

Table 12.4 Event table (9 of 9)

Event number	Interrupt request source	Name	IELSRn		DELSRn	Canceling Snooze	Canceling Software Standby	Canceling Deep Software Standby
			Connect to NVIC	Invoke DTC	Invoke DMAC			
0x1DB	DOC	DOC_DOPCI	✓	—	—	✓*1	—	—

Note 1. Using SELSR0.
Note 2. Only supported when KRCTL.KRMD = 1.
Note 3. Only the first edge detection is valid.
Note 4. Only interrupts after DTC transfer are supported.

12.4 Interrupt Operation

The ICU performs the following functions:

- Detecting interrupts
- Enabling and disabling interrupts
- Selecting interrupt request destinations such as CPU interrupt, DTC activation, or DMAC activation.

12.4.1 Detecting Interrupts

The ICU selects an event source input from a peripheral function interrupt or an external pin interrupt with IELSRn.IELS [8:0].

The accepted interrupt source sets the IELSRn.IR to 1 and sends an interrupt request to the NVIC.

External pin interrupt requests are detected by either:

- Edges (falling edge, rising edge, or rising and falling edges)
- Level (low level) of the interrupt signal.

Set the IRQCRi.IRQMD[1:0] bits to select the detection mode for the IRQi pins. For interrupt sources associated with peripheral modules, see Table 12.3 and Table 12.4. Events must be accepted by the NVIC before an interrupt occurs and is accepted by the CPU.

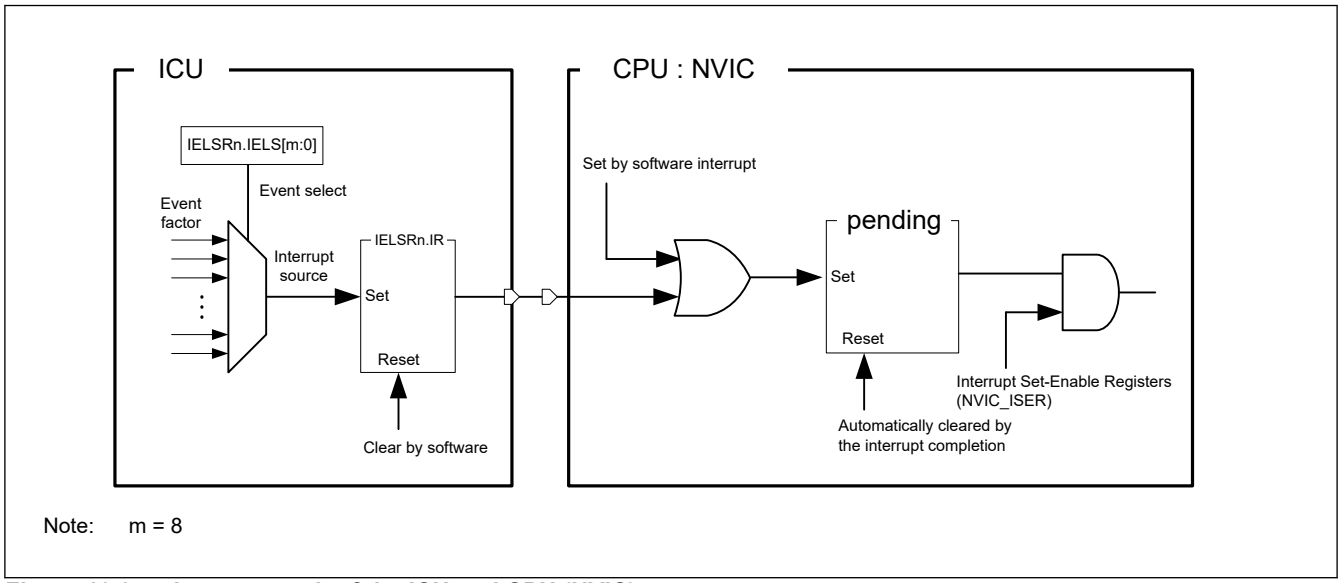


Figure 12.2 Interrupt path of the ICU and CPU (NVIC)

12.5 Interrupt setting procedure

12.5.1 Enabling Interrupt Requests

The procedure for enabling an interrupt request is as follows:

1. Set the Interrupt Set-Enable register (NVIC_ISER).
2. Set the IELSRn.IELS[8:0] bits as the interrupt source.
3. Specify the operation settings for the event source, such as DMAC activation (DELSRn.DELS[8:0]), snooze mode cancellation (SELSR0.SELS[8:0]), software standby mode cancellation (WUPEN register setting).

12.5.2 Disabling Interrupt Requests

The procedure to disable the interrupt request is as follows:

1. Disable the operation settings for the event source, such as DMAC activation (DELSRn.DELS[8:0]), snooze mode cancellation (SELSR0.SELS[8:0]), software standby mode cancellation (WUPEN register setting).
2. Clear the interrupt source setting (IELSRn.IELS[8:0] = 0x00).
3. Clear the interrupt status flag (IELSRn.IR = 0).
4. Clear the interrupt Clear-Enable register (NVIC_ICER) and interrupt Clear-Pending register (NVIC_ICPR).

12.5.3 Polling for interrupts

The procedure for polling for interrupt requests is as follows:

1. Set the Interrupt Clear-Enable register (NVIC_ICER).
2. Set the IELSRn.IELS[8:0] bits as the interrupt source.
3. Specify the operation settings for the event source, such as DMAC activation (DELSRn.DELS[8:0]), snooze mode cancellation (SELSR0.SELS[8:0]), software standby mode cancellation (WUPEN register setting).
4. Poll the interrupt Set-Pending register (NVIC_ISPR).

12.5.4 Selecting Interrupt Request Destinations

The available destinations are fixed for each interrupt, as described in Table 12.3, Table 12.4.

The interrupt output destination, CPU, DMAC, or DTC can be independently selected for each interrupt source.

Use an interrupt request destination setting that is indicated by a "✓" in the event list (see section 12.3.2. Event Number).

Note: Setting the same interrupt source for IELSRn and DELSRn is prohibited.

If the DMAC or DTC is selected as the destination for requests from an IRQi pin, you must set the IRQCRi.IRQMD[1:0] bits for that interrupt to select edge detection.

12.5.4.1 CPU interrupt request

When IELSRn.DTCE = 0, the event specified in the IELSRn register is output to the NVIC. Set the IELSRn.IELS[8:0] bits to the target event and set the IELSRn.DTCE bit to 0.

12.5.4.2 DTC activation

When IELSRn.DTCE = 1, the event specified in the IELSRn register is output to the DTC. Use the following procedure:

1. Set the IELSRn.IELS[8:0] bits to the target event and set the IELSRn.DTCE bit to 1.
2. Set the DTC Module Start bit (DTCST.DTCST) to 1.

Table 12.5 shows operation when the DTC is the interrupt request destination.

Table 12.5 Operation when DTC becomes interrupt request destination

Interrupt request destination	DISEL*1	Remaining transfer operations	Operation per request	IR*2	Interrupt request destination after transfer
DTC*3	1	≠ 0	DTC transfer → CPU interrupt	Cleared on interrupt acceptance by the CPU	DTC
		= 0	DTC transfer → CPU interrupt	Cleared on interrupt acceptance by the CPU	CPU (IELSRn.DTCE bit is automatically cleared)
	0	≠ 0	DTC transfer	Cleared at the start of DTC data transfer after reading DTC transfer data	DTC
		= 0	DTC transfer → CPU interrupt	Cleared on interrupt acceptance by the CPU	CPU (IELSRn.DTCE bit is automatically cleared)

Note 1. DTC.MRB.DISEL bit controles the interrupt generates timing from DTC to CPU.
Note 2. When the IELSRn.IR flag is 1, an interrupt request (DTC activation request) that occurs again is ignored.
Note 3. For chain transfers, DTC transfer continues until the last chain transfer ends. The DISEL bit state and the remaining transfer count determine whether a CPU interrupt occurs, the IELSRn.IR flag clear timing, and the interrupt request destination after transfer. See Table 16.2 in section 16, Data Transfer Controller (DTC).

Note: Error during DTC transfer

If an error response occurs during DTC transfer, the DTC notifies the ICU that an error has occurred. ICU clears all bits of the target IELSRn (n = 0 to 95). IELSRn that is not the target is not cleared.

Note: DTC transfer error in snooze mode

When an error occurs in DTC transfer in Snooze mode, ICU issues a Wake Up request. However, interrupt requests are not issued automatically. See section 16, Data Transfer Controller (DTC) chapter for information on how to set the interrupt when a DTC error occurs.

12.5.4.3 DMAC Activation

Events specified in the DELSRn registers are output to the DMAC.

To set the interrupt source for DMAC, use the following procedure:

1. Set the DELSRn.DELS[8:0] bits to the event to activate the DMAC.
2. When using interrupts to CPU, set the IELSRn.IELS bit to factor of DMAC interrupt and IELSRn.DTCE bit to 0.
3. Set the activation source for the target DMAC channel (DMACm.DMTMD.DCTG[1:0]) to 01b (interrupt module detection).
4. Set the DMAC transfer enable bit for the target DMAC channel (DMACm.DMCNT.DTE) to 1.
5. Set the DMAC operation enable bit (DMAST.DMST) to 1.

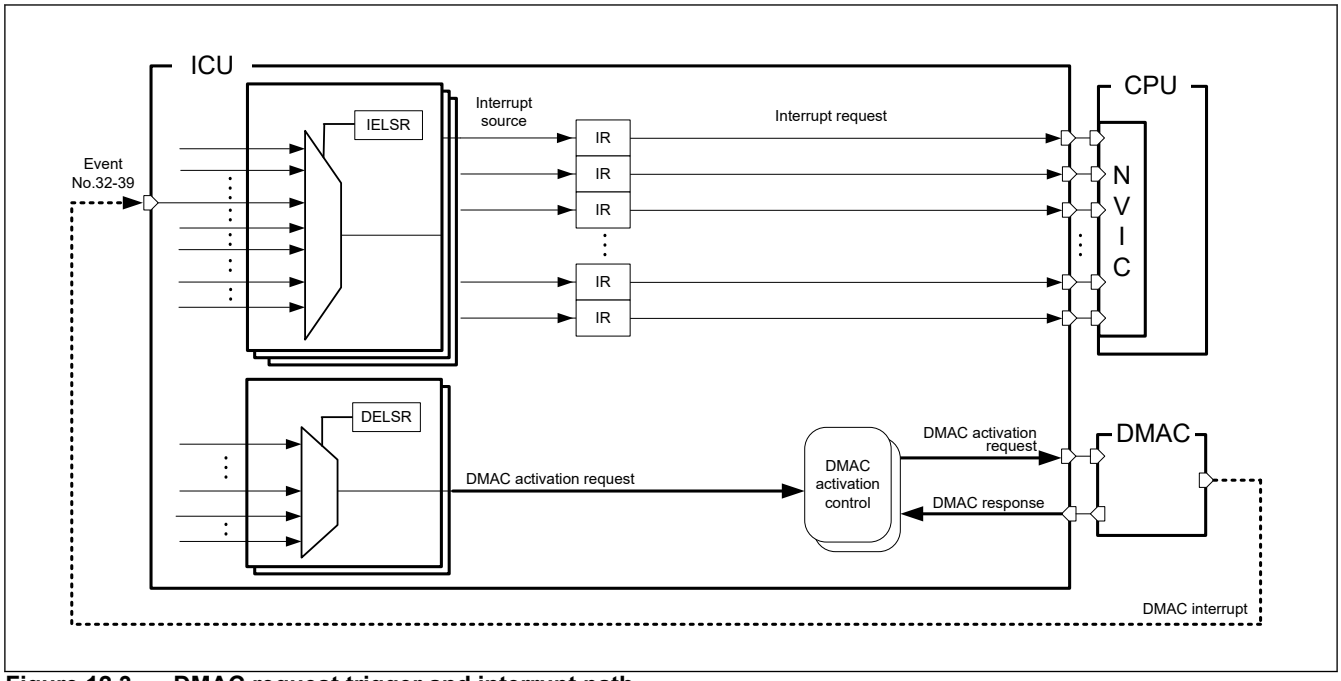


Figure 12.3 DMAC request trigger and interrupt path

Note: Error during DMAC transfer

If an error response occurs during DMAC transfer, the DMAC notifies the ICU that an error has occurred.

The ICU clears all bits of the target channel of DELSRn (n = 0 to 7). DELSRn that is not the target channel is not cleared.

12.5.5 Digital Filter

A digital filter function is provided for the external interrupt request pins IRQi, (i = 0 to 15) and the NMI pin interrupt. It samples input signals on the filter PCLKB sampling clock and removes any signal with a pulse width less than 3 sampling cycles.

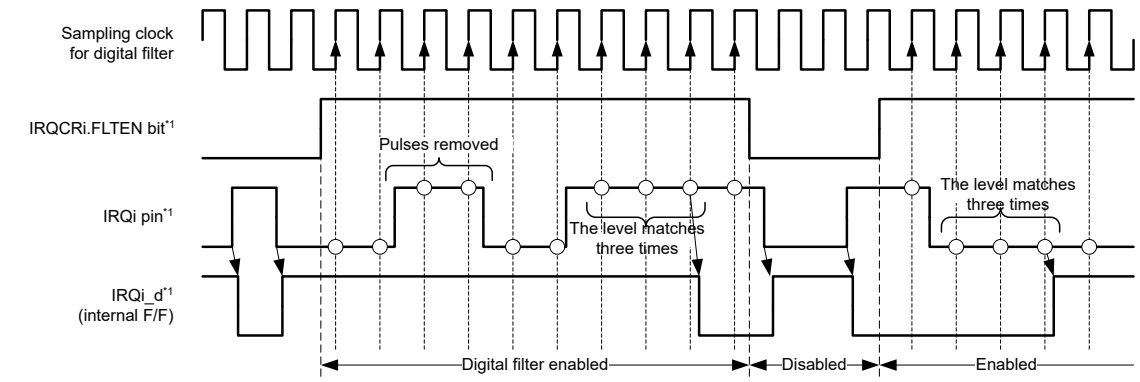
To use the digital filter for an IRQi pin:

1. Set the sampling clock cycle to PCLKB, PCLKB/8, PCLKB/32, or PCLKB/64 in the IRQCRi.FCLKSEL[1:0] bits (i = 0 to 15).
2. Set the IRQCRi.FLTEN bit (i = 0 to 15) to 1 (digital filter enabled).

To use the digital filter for an NMI pin:

1. Set the sampling clock cycle to PCLKB, PCLKB/8, PCLKB/32, or PCLKB/64 in the NMICR.NFCLKSEL[1:0] bits.
2. Set the NMICR.NFLTEN bit to 1 (digital filter enabled).

Figure 12.4 shows an example of digital filter operation.



Note 1. i = 0 to 7

Figure 12.4 Digital filter operation example

Before entering Software Standby mode, disable the digital filters by clearing the IRQCRi.FLTEN and NMICR.NFLTEN bits. The ICU clock stops in Software Standby mode.

On exiting Software Standby mode, the circuit detects the edge by comparing the state before standby to the state after standby release. If the input changes during Software Standby mode, an incorrect edge might be detected. The digital filters can be enabled again after exiting Software Standby mode.

12.5.6 External Pin Interrupts

To use external pin interrupts:

1. Configure I/O ports settings.
2. Clear the IRQCRi.FLTEN bit (i = 0 to 15) to 0 (digital filter disabled).
3. Set the IRQMD[1:0] bits of the given IRQCRi register (i = 0 to 15) to select the senses of detection.
4. Set the FCLKSEL[1:0] bits, and the FLTEN bit of the IRQCRi register.
5. Select the IRQ pin as follows:
   - If the IRQ pin is to be used for CPU interrupt requests, set the IELSRn.IELS[8:0] bits and the IELSRn.DTCE bit to 0.
   - If the IRQ pin is to be used for DTC activation, set the IELSRn.IELS[8:0] bits and the IELSRn.DTCE bit to 1.
   - If the IRQ pin is to be used for DMAC activation, set the DELSRn.DELS[8:0] bits.

12.6 Non-Maskable Interrupt Operation

The following sources can trigger a non-maskable interrupt:

- NMI pin interrupt
- Oscillation stop detection interrupt
- WDT underflow/refresh error interrupt
- IWDT underflow/refresh error interrupt
- Voltage monitor 1 interrupt
- Voltage monitor 2 interrupt
- SRAM parity error interrupt
- SRAM ECC error interrupt

- Bus master MPU error interrupt
- TrustZone filter error interrupt
- Cache RAM parity error interrupt.

Non-maskable interrupts can only be used with the CPU, not to activate the DTC or DMAC. Non-maskable interrupts take precedence over all other interrupts. The non-maskable interrupt states can be verified in the Non-Maskable Interrupt Status Register (NMISR). Confirm that all bits in the NMISR are 0 before returning from the NMI handler.

Non-maskable interrupts are disabled by default. To use non-maskable interrupts:

1. Clear the NMICR.NFLTEN bit to 0 (digital filter disabled).
2. Set the NMIMD bit, NFCLKSEL[1:0] bits, and NFLTEN bit of NMICR register.
3. Write 1 to the NMICLR.NMICLR bit to clear the NMISR.NMIST flag to 0.
4. Enable the non-maskable interrupt by writing 1 to the associated bit in the Non-Maskable Interrupt Enable Register (NMIER).

After 1 is written to the NMIER register, subsequent write access to the NMIEN bit in NMIER is ignored. An NMI cannot be disabled when enabled, except by a reset.

The secure attribution managed within the Application Interrupt and Reset Control Register (AIRCR) of the Arm CPU must match the security attribution of NMI.

The NMI secure of the CPU is changed by AIRCR.BFHFNMINS. It is managed by software developers who manage Secure program.

12.6.1 Correspondence to TrustZone-M by NMI

The NMI security is set by AIRCR.BFHFNMINS.

Although there is only one NMI as a CPU, multiple factors can be set.

This section describes the procedure for mixing secure and non-secure factors of NMI.

When mixing secure and non-secure, NMI related registers in the CPU should be set to Secure.

NMI-related registers:

- NMIER
- NMICLR
- NMICR

Figure 12.5 shows the flow.

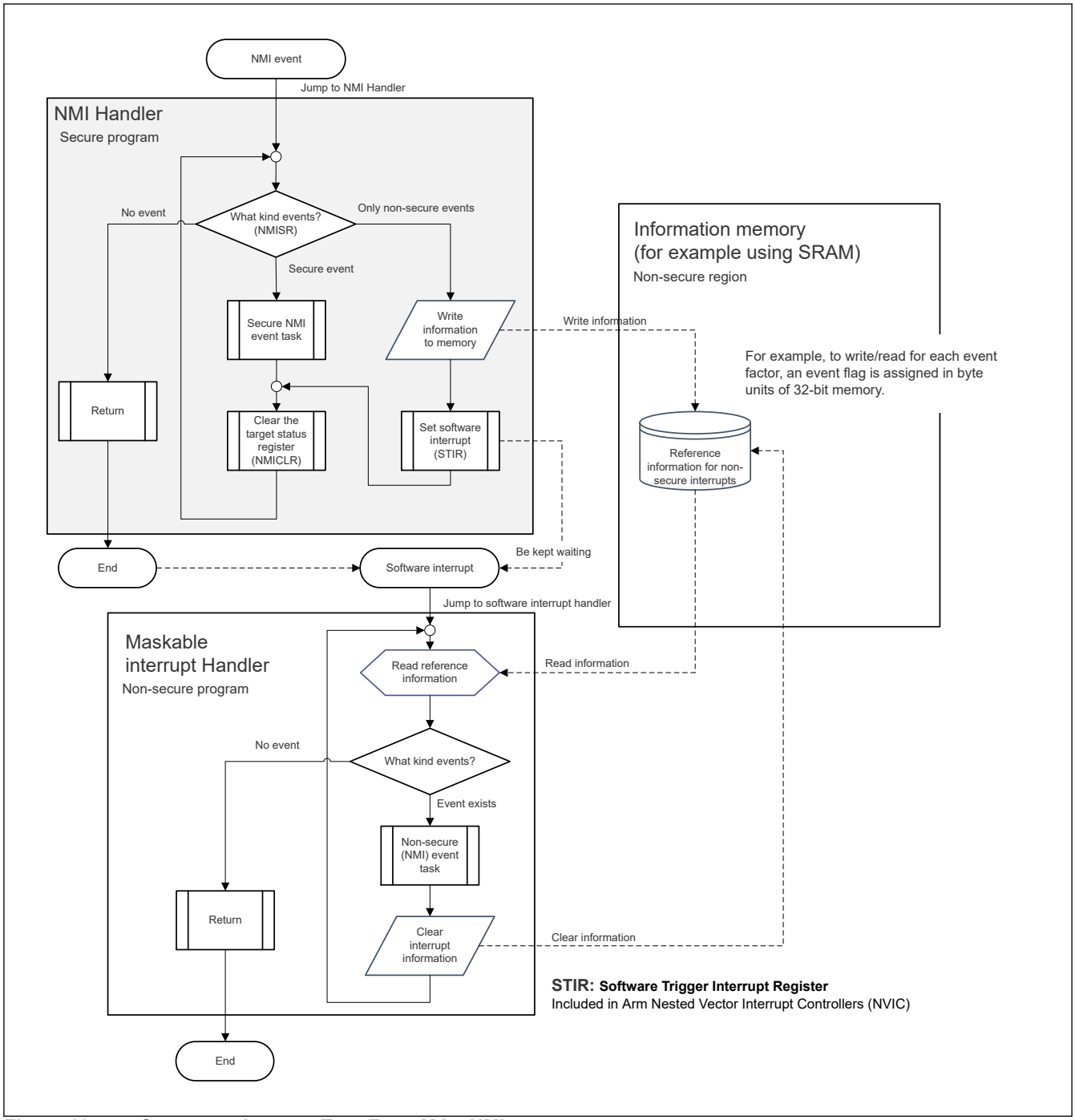


Figure 12.5 Correspondence to TrustZone-M by NMI

See the Arm documentation for details on how to move between secure and non-secure programs.

12.7 Return from Low Power Modes

Table 12.4 lists the interrupt sources that can be used to exit Sleep, Snooze, or Software Standby mode. For more information, see section 10, Low Power Modes.

12.7.1 Return from Sleep Mode

To return from Sleep mode in response to an interrupt:

non-maskable interrupt

- Use the NMIER register to enable the target interrupt request.

maskable interrupt

- Select the CPU as the interrupt request destination.
- Enable the interrupt in the NVIC.

12.7.2 Return from Software Standby Mode

The ICU returns from Software Standby mode using a non-maskable interrupt or a maskable interrupt. For maskable interrupt of canceling source, see Table 12.4.

To return from Software Standby mode:

1. Select the interrupt source that enables return from Software Standby:
   - For non-maskable interrupts, use the NMIER register to enable the target interrupt request
   - For maskable interrupts, use the WUPEN register to enable the target interrupt request.
2. Select the CPU as the interrupt request destination
3. Enable the interrupt in the NVIC.

Interrupt requests through the IRQn pins that do not satisfy these conditions are not detected while the clock is stopped in Software Standby mode.

Similarly, request for a non-maskable interrupt from a request source whose clock is stopped in Software Standby mode cannot be detected.

Transition to/from Software Standby mode

1. Before Software Standby mode is entered, disable the digital filter for the interrupt source as a return target (IRQCRi.FLTEN = 0, NMICR.NFLTEN = 0).
2. To use the digital filter again after returning from Software Standby mode, enable the digital filter (IRQCRi.FLTEN = 1, NMICR.NFLTEN = 1).

12.7.3 Return from Snooze Mode

The ICU can return to Normal mode from Snooze mode using the interrupts provided for this mode.

To return to Normal mode from Snooze mode:

1. Set the number of the required interrupt request in SELSR0.SELS[8:0]
2. Set the value 0x02D (ICU_SNZCANCEL) in IELSRn.IELS[8:0] (n = 0 to 95).
3. Select the CPU as the interrupt request destination.
4. Enable the interrupt in the NVIC.

Interrupt requests through the non-maskable that do not satisfy the above conditions are not detected while the clock is stopped in snooze mode.

Note: In Snooze mode, a clock is supplied to the ICU. If an event selected in IELSRn is detected, the CPU acknowledges the interrupt after returning to Normal mode from Software Standby mode. If an event selected in DELSRn is detected, the DMAC can acknowledge the interrupt after returning to Normal mode from Software Standby mode.

12.8 Using the WFI Instruction with Non-Maskable Interrupts

Whenever a WFI instruction is executed, confirm that all status flags in the NMISR register are 0.

12.9 Reference

- ARM Limited., ARM® Cortex®-M33 Processor Technical Reference Manual (ARM 100230)

13. Buses

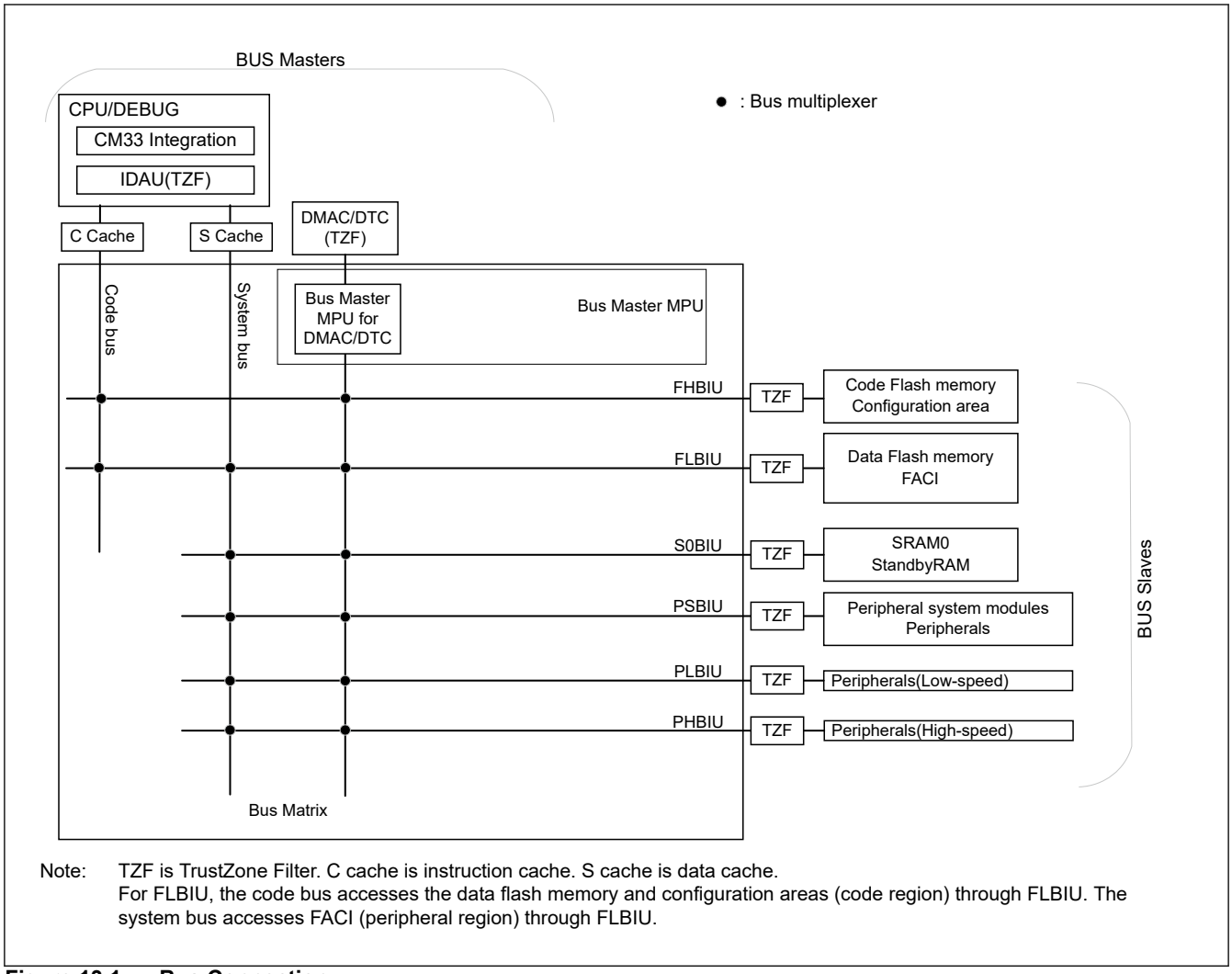
13.1 Overview

The buses consists of 32 bits AHB bus matrix. Table 13.1 lists the bus masters and bus slaves and Figure 13.1 shows the bus configuration.

Table 13.1 Bus Specifications

Classification	Bus Master/Slave name	Bus I/F Max Freq	Sync Clock	Specifications
Bus Masters	Code bus (Cortex-M33)	240 MHz	ICLK	Connected to the CPU Instruction Cache for instructions and operands
	System bus (Cortex-M33)	240 MHz	ICLK	Connected to the CPU Data Cache for system
	DMAC / DTC	240 MHz	ICLK	Connected to the DMAC/DTC
Bus Slaves	FHBIU	240 MHz	ICLK	Connected to Code Flash memory and Configuration area
	FLBIU	60 MHz	FCLK	Connected to Data Flash memory, FACI
	S0BIU	240 MHz	ICLK	Connected to SRAM0 (Standby RAM)
	PSBIU	240 MHz	ICLK	• Connected to peripheral system modules (DTC, DMAC, ICU, Flash, MPU, SRAM, Debug/Trace module, System controller and BUS controller) • Connected to peripheral modules (IIRFA, TFU, and IO ports)
	PLBIU	60 MHz	PCLKB	Connected to peripheral modules (CAC, ELC, POEG, WDT, IWDT, AGT, CANFD, TSN, ACMPHS, and KINT)
	PHBIU	120 MHz	PCLKA	Connected to peripheral modules (GPT, SCI, SPI, CRC, DOC, ADC, DAC12, CNECC, IIC, SCE5, and PDG)

Note: FHBIU: Flash High speed Bus Interface Unit.
FLBIU: Flash Low speed Bus Interface Unit.
S0BIU: SRAM0 Bus Interface Unit.
PSBIU: Peripheral System Bus Interface Unit.
PLBIU: Peripheral Low speed Bus Interface Unit.
PHBIU: Peripheral High speed Bus Interface Unit.



Note: TZF is TrustZone Filter. C cache is instruction cache. S cache is data cache.
For FLBIU, the code bus accesses the data flash memory and configuration areas (code region) through FLBIU. The system bus accesses FACI (peripheral region) through FLBIU.

Figure 13.1 Bus Connection

13.2 Description of Buses

13.2.1 Arbitration

For arbitration between masters in each slave, fixed-priority and round-robin methods can be selected for each master. For details, see section 13.3.3. BUSSCNT<slave> : Slave Bus Control Register (<slave> = FHBIU, FLBIU, S0BIU), section 13.3.4. BUSSCNT<slave> : Slave Bus Control Register (<slave> = PSBIU, PLBIU, PHBIU).

13.2.2 Parallel Operation

Parallel operation is possible when different bus-master modules are requesting access to different slave modules. For example, if the CPU is fetching an instruction from Code Flash and an operand from SRAM0, the DMAC can handle transfer between a peripheral module at the same time.

Figure 13.2 shows an example of parallel operations. In this example, the CPU uses code bus and system bus for simultaneous access to FHBIU and S0BIU, respectively. Furthermore, the DMAC/DTC simultaneously accesses the peripheral bus during access to FHBIU and S0BIU by the CPU.

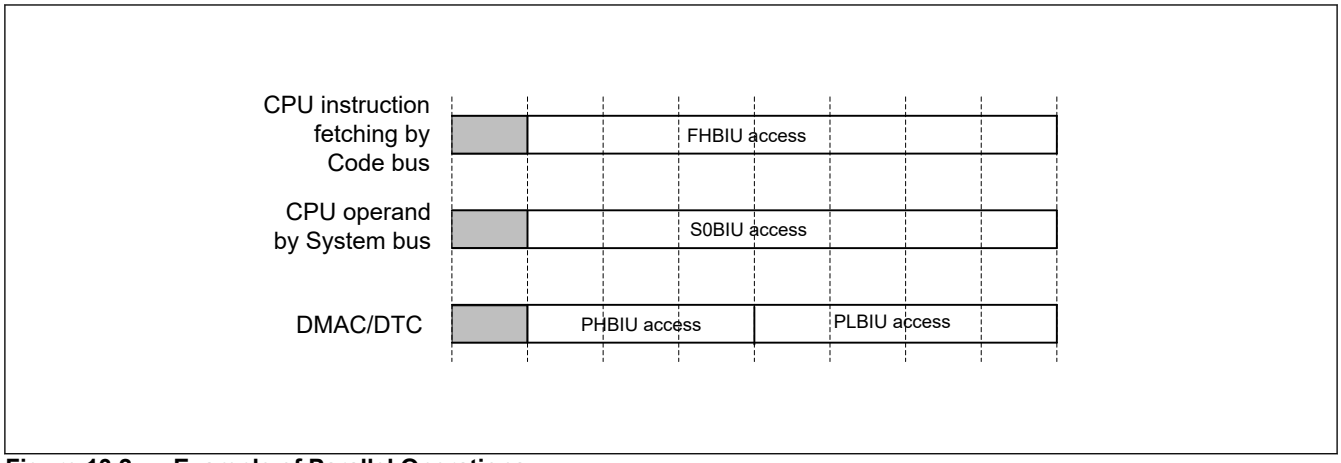


Figure 13.2 Example of Parallel Operations

13.2.3 Restrictions

(1) Restriction on Endian

Memory space must be little-endian in order to execute Cortex code.

(2) Bufferable write access

When CPU perform Bufferable Write access to PLBIU or PHBIU, if an STZF error occurs then the error response is invalidated. So there is no error flag will be set and no NMI / RESET request is generated.

When CPU perform Bufferable Write access to PHBIU, if a Slave BUS error occurs then the error response will become invalid and the error flag will not be set.

If error response is required, set the bus master to non-bufferable access.

(3) Access to reserved area of FLBIU and S0BIU

Access to the reserved area of FLBIU and S0BIU is prohibited. Operation is not guaranteed if accessed.

(4) Clock setting

The clock division ratio prohibits setting changes during slave bus access to FLBIU, PLBIU, and PHBIU.

13.3 Register Descriptions

13.3.1 BUSSARA : BUS Security Attribution Register A

Base address: CPSCU = 0x4000_8000

Offset address: 0x0100

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BUSSA0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	BUSSA0	BUS Security Attribution A0 0: Secure 1: Non-Secure	R/W

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

Note: Only secure access can write to this register. Both secure access and Non-secure read access are allowed but Non-secure write access is not allowed and TrustZone access error is not generated.
Note: This register is write-protected by PRCR register.

BUSSA0 bit (BUS Security Attribution A0)

The correspondence between register and BIU name is as follows

Connection (BUSSCNT<slave> = FHBIU/FLBIU/S0BIU/PSBIU/PLBIU/PHBIU)

See to Figure 13.1 for connection between BIU and BUS

- BUSSCNTFHBIU
- BUSSCNTFLBIU
- BUSSCNTS0BIU
- BUSSCNTPSBIU
- BUSSCNTPLBIU
- BUSSCNTPHBIU

13.3.2 BUSSARB : BUS Security Attribution Register B

Base address: CPSCU = 0x4000_8000

Offset address: 0x0104

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BUSSB0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	BUSSB0	BUS Security Attribution B0 0: Secure 1: Non-Secure	R/W
31:1	—	These bits are read as 1.	R

Note: Only secure access can write to this register. Both secure access and Non-secure read access are allowed but Non-secure write access is not allowed and TrustZone access error is not generated.
Note: This register is write-protected by PRCR register.

BUSSB0 bit (BUS Security Attribution B0)

The BUSSB0 bit specifies the security attributes of registers for Bus Error Clear registers and DMAC/DTC Error Clear register.

BUS1ERRCLR: Code bus

BUS2ERRCLR: System bus

BUS3ERRCLR: DMAC/DTC

DMACDTCERRCLR: DMAC/DTC (Master-TZF)

See Figure 13.1 for connection of each BUS.

13.3.3 BUSSCNT<slave> : Slave Bus Control Register (<slave> = FHBIU, FLBIU, S0BIU)

Base address: BUS = 0x4000_3000

Offset address: 0x1100 (BUSSCNTFHBIU)
0x1104 (BUSSCNTFLBIU)
0x1110 (BUSSCNTS0BIU)

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

Bit	Symbol	Function	R/W
1:0	ARBS[1:0]	Arbitration Select for two masters Specify the priority between bus master. > : Fixed Priority ↔: Round-Robin 0 0: DMAC/DTC > CPU 0 1: DMAC/DTC ↔ CPU 1 0: Setting prohibited 1 1: Setting prohibited	R/W
15:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note:
- BUSSCNT<slave> : <slave> is bus interface unit name for Slave
- The change is prohibited from initial value (0) to reserved bit. Operation when changing is not guaranteed.

ARBS[1:0] bits (Arbitration Select for two masters)

The ARBS bits sets the arbitration method of each master.

13.3.4 BUSSCNT<slave> : Slave Bus Control Register (<slave> = PSBIU, PLBIU, PHBIU)

Base address: BUS = 0x4000_3000

Offset address: 0x1120 (BUSSCNTPSBIU)
0x1130 (BUSSCNTPLBIU)
0x1134 (BUSSCNTPHBIU)

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

Bit	Symbol	Function	R/W
0	ARBS	Arbitration Select for two masters Specify the priority between bus master. > : Fixed Priority ↔: Round-Robin 0: DMAC/DTC > CPU 1: DMAC/DTC ↔ CPU	R/W
15:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note:
- BUSSCNT<slave> : <slave> is bus interface unit name for Slave
- The change is prohibited from initial value (0) to reserved bit. Operation when changing is not guaranteed.

ARBS bit (Arbitration Select for two masters)

The ARBS bits sets the arbitration method of each master.

13.3.5 BUSnERRADD : BUS Error Address Register (n = 1 to 3)

Base address: BUS = 0x4000_3000

Offset address: 0x1800 + 0x10 × (n - 1)

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

Bit	Symbol	Function	R/W
31:0	BERAD[31:0]	Bus Error Address When a bus error occurs, these bits store the error address	R

This register is cleared by reset other than MPU- and TZF-related resets which are Bus Master MPU Error Reset and TrustZone Filter Error Reset.

For detail of MPU- and TZF-related resets, see section 5, Resets, section 14, Memory Protection Unit (MPU) and section 45.2. Arm TrustZone Security .

The following bus errors correspond to the master bus:

BUS1ERRADD : Code bus

BUS2ERRADD : System bus

BUS3ERRADD : DMAC/DTC

BERAD[31:0] bits (Bus Error Address)

The BERAD[31:0] bits indicate the address when an error occurs on the associated bus. For detail of error that occurs by bus, see section 13.3.9. BUSnERRSTAT : BUS Error Status Register n (n = 1 to 3) and section 13.4. Bus Error Monitoring Section.

When an error occurs on the bus, the corresponding bit of ILERRSTAT, MMERRSTAT, SLERRSTAT in BUSnERRSTAT (n = 1 to 3) is set to 1, at the same time, the BERAD[31:0] bits store the address of the bus error access.

BERAD[31:0] bits are only valid when ILERRSTAT, MMERRSTAT, and SLERRSTAT in BUSnERRSTAT (n = 1 to 3) are set to 1.

13.3.6 BUSnERRRW : BUS Error Read Write Register (n = 1 to 3)

Base address: BUS = 0x4000_3000

Offset address: 0x1804 + 0x10 × (n - 1)

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

Bit	Symbol	Function	R/W
0	RWSTAT	Error Access Read/Write Status The status at the time of the error 0: Read access 1: Write access	R
7:1	—	These bits are read as 0. The write value should be 0.	R/W

This register is cleared by reset other than MPU- and TZF-related resets which are Bus Master MPU Error Reset and TrustZone Filter Error Reset.

For detail of MPU related reset, see section 5, Resets and section 14, Memory Protection Unit (MPU).

The following bus errors correspond to the master bus:

BUS1ERRRW : Code bus

BUS2ERRRW : System bus

BUS3ERRRW : DMAC/DTC

RWSTAT bit (Error Access Read/Write Status)

The RWSTAT bit indicates the access status, (write access or read access) when an error occurs on the associated bus. For detail of error that occurs by bus, see section 13.3.9. BUSnERRSTAT : BUS Error Status Register n (n = 1 to 3) and section 13.4. Bus Error Monitoring Section.

When an error occurs on the bus, the corresponding bit of ILERRSTAT, MMERRSTAT, SLERRSTAT in BUSnERRSTAT (n = 1 to 3) is set to 1, at the same time, the RWSTAT bits store the read/write status of the bus error access.

RWSTAT bit is only valid when ILERRSTAT, MMERRSTAT, and SLERRSTAT in BUSnERRSTAT (n = 1 to 3) are set to 1.

13.3.7 BTZFnERRADD : BUS TZF Error Address Register (n = 1 to 3)

Base address: BUS = 0x4000_3000

Offset address: 0x1900 + 0x10 × (n - 1)

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

Bit	Symbol	Function	R/W
31:0	BTZFERAD[31:0]	Bus TrustZone Filter Error Address When a bus error occurs, these bits store the error address	R

This register is cleared by reset other than MPU- and TZF-related resets which are Bus Master MPU Error Reset and TrustZone Filter Error Reset.

For detail of MPU- and TZF-related resets, see section 5, Resets, section 14, Memory Protection Unit (MPU) and section 45.2. Arm TrustZone Security .

The following bus errors correspond to the master bus:

BTZF1ERRADD : Code bus

BTZF2ERRADD : System bus

BTZF3ERRADD : DMAC/DTC

See Figure 13.1 for connection of each BUS.

BTZFERAD[31:0] bits (Bus TrustZone Filter Error Address)

The BTZFERAD[31:0] bits indicate the address, when an error occurs on the associated bus. For detail of error that occurs by bus, see section 13.3.9. BUSnERRSTAT : BUS Error Status Register n (n = 1 to 3) and section 13.4. Bus Error Monitoring Section.

When an error occurs on the bus, the corresponding bit of STERRSTAT in BUSnERRSTAT (n = 1 to 3) is set to 1, at the same time, the BTZFERAD[31:0] bits store the address of the bus error access.

BTZFERAD[31:0] bits are only valid when STERRSTAT in BUSnERRSTAT (n = 1 to 3) is set to 1.

13.3.8 BTZFnERRRW : BUS TZF Error Read Write Register (n = 1 to 3)

Base address: BUS = 0x4000_3000

Offset address: 0x1904 + 0x10 × (n - 1)

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

Bit	Symbol	Function	R/W
0	TRWSTAT	TrustZone filter error access Read/Write Status The status at the time of the error 0: Read access 1: Write access	R
7:1	—	These bits are read as 0. The write value should be 0.	R/W

This register is cleared by reset other than MPU- and TZF-related resets which are Bus Master MPU Error Reset and TrustZone Filter Error Reset.

For detail of MPU- and TZF-related resets, see section 5, Resets, section 14, Memory Protection Unit (MPU) and section 45.2. Arm TrustZone Security .

The following bus errors correspond to the master bus:

BTZF1ERRRW : Code bus

BTZF2ERRRW : System bus

BTZF3ERRRW : DMAC/DTC

See Figure 13.1 for connection of each BUS.

TRWSTAT bit (TrustZone filter error access Read/Write Status)

The TRWSTAT bit indicates the access status, (write access or read access), when an error occurs on the associated bus. For detail of error that occurs by bus, see section 13.3.9. BUSnERRSTAT : BUS Error Status Register n (n = 1 to 3) and section 13.4. Bus Error Monitoring Section.

When an error occurs on the bus, the corresponding bit of STERRSTAT in BUSnERRSTAT (n = 1 to 3) is set to 1, at the same time, the TRWSTAT bits store the read/write status of the bus error access. The TRWSTAT bit is only valid when STERRSTAT in BUSnERRSTAT (n = 1 to 3) is set to 1.

13.3.9 BUSnERRSTAT : BUS Error Status Register n (n = 1 to 3)

Base address: BUS = 0x4000_3000

Offset address: 0x1A00 + 0x10 × (n - 1)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	ILERRSTAT	MMERRSTAT	—	STERRSTAT	SLERRSTAT
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SLERRSTAT	Slave bus Error Status 0: No error occurred 1: Error occurred	R
1	STERRSTAT	Slave TrustZone filter Error Status 0: No error occurred 1: Error occurred	R
2	—	This bit is read as 0.	R

Bit	Symbol	Function	R/W
3	MMERRSTAT	Master MPU Error Status 0: No error occurred 1: Error occurred	R
4	ILERRSTAT	Illegal address access Error Status 0: No error occurred 1: Error occurred	R
7:5	—	These bits are read as 0.	R

This register is cleared by reset other than MPU- and TZF-related resets which are Bus Master MPU Error Reset and TrustZone Filter Error Reset.

For detail of MPU- and TZF-related resets, see section 5, Resets, section 14, Memory Protection Unit (MPU) and section 45.2. Arm TrustZone Security .

The following bus errors correspond to the master bus:

BUS1ERRSTAT : Code bus

BUS2ERRSTAT : System bus

BUS3ERRSTAT : DMAC/DTC

See Figure 13.1 for connection of each BUS

When an illegal access error, master MPU error, and slave bus error all occurred at the same time, the STAT bit is only valid in the following order of priority. The left side has higher priority.

Master MPU Error > Illegal access error, slave bus error

Note: Illegal access error and slave bus error do not occur at the same time.

If one of ILERRSTAT, MMERRSTAT or SLERRSTAT is set, these bits are not renewed until it is cleared.

SLERRSTAT bit (Slave bus Error Status)

When slave error occurs by bus, BUSnERRSTAT.SLERRSTAT (n = 1 to 3) is set to 1. Clear condition is reset or set BUSnERRCLR.SLERRCLR (n = 1 to 3) to 1. Slave error is an error that occurs on a slave such as a timeout. For detail of slave error that occurs by bus, see section 13.4. Bus Error Monitoring Section.

STERRSTAT bit (Slave TrustZone filter Error Status)

When slave TrustZone filter error occurs by bus, BUSnERRSTAT.STERRSTAT (n = 1 to 3) is set to 1. Clear condition is reset or set BUSnERRCLR.STERRCLR (n = 1 to 3) to 1. The STERRSTAT bit is not set when the debugger accesses the security area. For detail of slave TrustZone filter error that occurs by bus, see section 45, Security Features .

MMERRSTAT bit (Master MPU Error Status)

When master MPU error occurs by bus, BUSnERRSTAT.MMERRSTAT (n = 1 to 3) is set to 1. Clear condition is reset or set BUSnERRCLR.MMERRCLR (n = 1 to 3) to 1. For detail of master MPU error that occurs by bus, see section 14, Memory Protection Unit (MPU).

Note: At master MPU error is occur in DMAC or DTC access, if error address value is not as master MPU area, Illegal address access error or slave error is occurring before DMAC or DTC access. Decide the what error was happened by referring the error address value.

ILERRSTAT bit (Illegal address access Error Status)

When illegal address access error occurs by bus, BUSnERRSTAT.ILERRSTAT (n = 1 to 3) is set to 1. Clear condition is reset or set BUSnERRCLR.ILERRCLR (n = 1 to 3) to 1. For detail of illegal address access error that occurs by bus, see section 13.4. Bus Error Monitoring Section.

13.3.10 DMACDTCERRSTAT : DMAC/DTC Error Status Register

Base address: BUS = 0x4000_3000

Offset address: 0x1A24

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

Bit	Symbol	Function	R/W
0	MTERRSTAT	Master TrustZone Filter Error Status 0: No error occurred 1: Error occurred	R
7:1	—	These bits are read as 0.	R

This register is cleared by reset other than MPU- and TZF-related resets which are Bus Master MPU Error Reset and TrustZone Filter Error Reset.

For detail of MPU- and TZF-related resets, see section 5, Resets, section 14, Memory Protection Unit (MPU) and section 45.2. Arm TrustZone Security .

MTERRSTAT bit (Master TrustZone Filter Error Status)

When a master TrustZone filter error occurs by DMAC or DTC, DMACDTCERRSTAT.MTERRSTAT is set to 1. Clear condition is reset or set DMACDTCERRCLR.MTERRCLR to 1.

For detail of master TrustZone filter error that occurs by DMAC or DTC, see section 15, DMA Controller (DMAC) and section 16, Data Transfer Controller (DTC)

13.3.11 BUSnERRCLR : BUS Error Clear Register n (n = 1 to 3)

Base address: BUS = 0x4000_3000

Offset address: 0x1A08 + 0x10 × (n - 1)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	ILERRCLR	MMERRCLR	—	STERRCLR	SLERRCLR
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SLERRCLR	Slave bus Error Clear Writing 1 to the SLERRCLR bit clears the BUSnERRSTAT.SLERRSTAT (n = 1 to 3)	R/W*1
1	STERRCLR	Slave TrustZone filter Error Clear Writing 1 to the STERRCLR bit clears the BUSnERRSTAT.STERRSTAT (n = 1 to 3)	R/W*1
2	—	This bit is read as 0. The write value should be 0.	R/W
3	MMERRCLR	Master MPU Error Clear Writing 1 to the MMERRCLR bit clears the BUSnERRSTAT.MMERRSTAT (n = 1 to 3)	R/W*1
4	ILERRCLR	Illegal Address Access Error Clear Writing 1 to the ILERRCLR bit clears the BUSnERRSTAT.ILERRSTAT (n = 1 to 3)	R/W*1
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note 1. Only 1 can be written to this bit. This bit is read as 0. Writing 0 to this bit has no effect.

The following bus errors correspond to the master bus:

BUS1ERRCLR : Code bus

BUS2ERRCLR : System bus

BUS3ERRCLR : DMAC/DTC

When writing 1 to BUSnERRCLR (n = 1 to 3), stop the bus access that causes an error in the corresponding bus master.

13.3.12 DMACDTCERRCLR : DMAC/DTC Error Clear Register

Base address: BUS = 0x4000_3000

Offset address: 0x1A2C

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

Bit	Symbol	Function	R/W
0	MTERRCLR	Master TrustZone filter Error Clear Writing 1 to this bit clears the DMACDTCERRSTAT.MTERRSTAT flag.	R/W*1
7:1	—	This bit is read as 0. The write value should be 0.	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note 1. Only 1 can be written to this bit. This bit is read as 0. Writing 0 to this bit has no effect.

When writing 1 to DMACDTCERRCLR, stop the bus access that causes an error in DMAC/DTC.

13.4 Bus Error Monitoring Section

The bus error monitoring system monitors each individual area, and when an error is detected, an error is returned to the requesting master IP using the AHB-Lite error response protocol.

13.4.1 Bus Error Types

The following types of errors can occur on each bus:

- Illegal address access
- Bus master MPU error
- TrustZone Filter error
- Bus error transmitted from each slave IP

Table 13.2 lists the address ranges where access leads to illegal address access errors. The reserved area in the slave does not trigger an illegal address access error. For more information on the bus master MPU, see section 14, Memory Protection Unit (MPU).

13.4.2 Operations When a Bus Error Occurs

When a bus error occurs, operation is not guaranteed and the error is returned to the requesting master IP.

Figure 13.3 shows operation from each error detection to user notification on the bus.

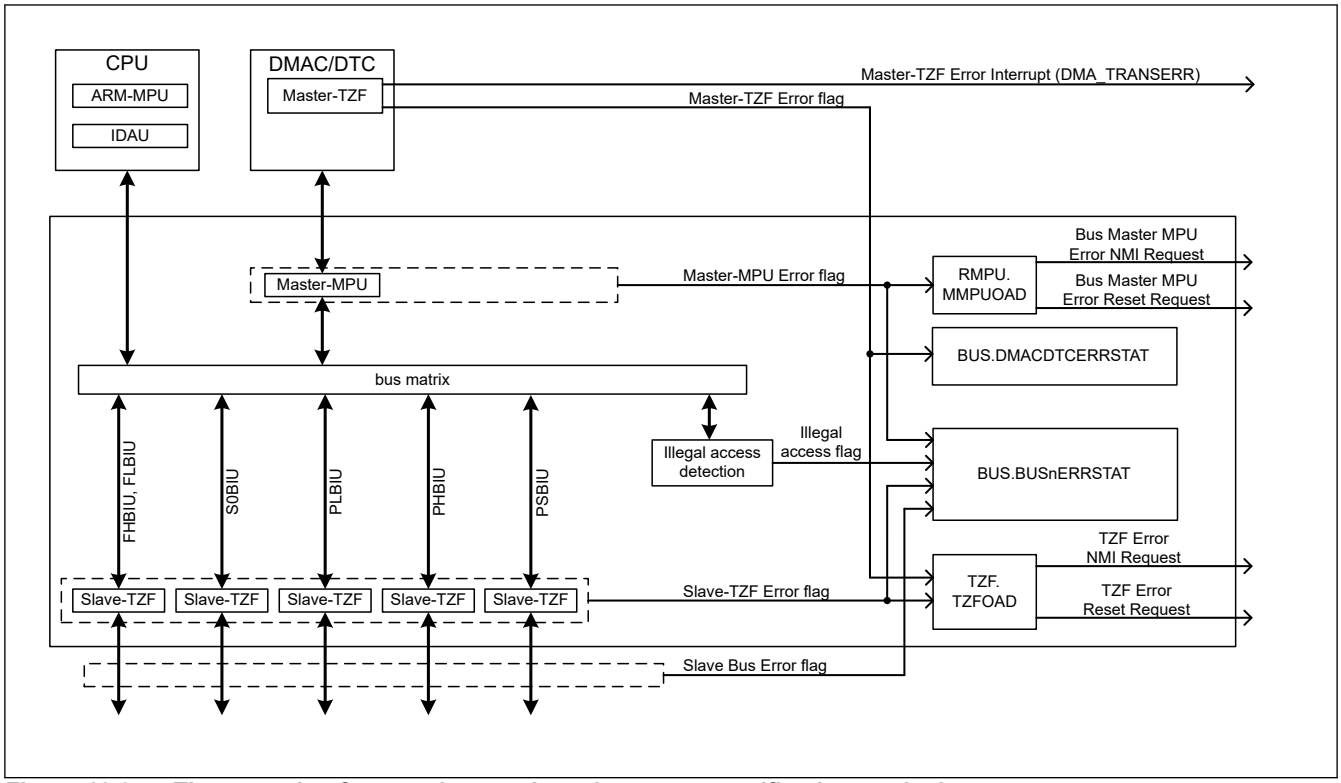


Figure 13.3 The operation from each error detection to user notification on the bus

(1) Bus Master MPU Error

The bus master of DMAC/DTC has a master MPU for access control of the set address area. The CPU does not have a master MPU because it has an Arm MPU. When a bus master MPU error is detected, an Error response is returned to the master. At the same time, perform the following steps:

1. Store the address of the error in BUSnERRADD (n = 3).
2. Store the read/write information of the error in BUSnERRRW (n = 3).
3. Set 1 to MMERRSTAT bit of BUSnERRSTAT (n = 3).

An NMI request or a reset request is generated according to the MMPUOAD.OAD setting (see section 14, Memory Protection Unit (MPU)). Since BUSnERRADD (n = 3), BUSnERRRW (n = 3), and BUSnERRSTAT (n = 3) are held until reset other than MPU- and TZF-related resets or cleared by BUSnERRCLR (n = 3), they can be verified in the NMI handler or after reset.

An NMI request is generated only on the first bus master MPU error after a reset or clearing of BUSnERRSTAT.MMERRSTAT (n = 3) bit by BUSnERRCLR (n = 3).

(2) Illegal Access Error

section 13.4.3. Conditions Leading to Illegal Address Access Errors, describes illegal access errors. When an illegal access error is detected, an Error response is returned to the master. At the same time, perform the following steps:

1. Store the address of the error in BUSnERRADD (n = 1 to 3).
2. Store the read/write information of the error in BUSnERRRW (n = 1 to 3).
3. Set 1 to ILERRSTAT bit of BUSnERRSTAT (n = 1 to 3).

NMI request and reset request are not generated. Since BUSnERRADD (n = 1 to 3), BUSnERRRW (n = 1 to 3), BUSnERRSTAT (n = 1 to 3) are held until reset other than MPU- and TZF-related resets or cleared by BUSnERRCLR (n = 1 to 3), they can be confirmed in the Bus Fault handler or the interrupt handler.

(3) Master-TZF Error

As described in section 45, Security Features , DMAC/DTC has Master-TZF errors. When a Master-TZF error is detected, 1 is set to MTERRSTAT bit of DMACDTCERRSTAT, and because the DMAC/DTC does not perform bus access, no bus error information is stored in BTZF3ERRADD and BTZF3ERRRW.

An NMI request or reset request is generated according to the setting of TZFOAD.OAD. See section 15, DMA Controller (DMAC), section 16, Data Transfer Controller (DTC) for details on Master-TZF errors. Because DMACDTCERRSTAT is held until reset other than MPU- and TZF-related resets or cleared by DMACDTCERRCLR, they can be verified in the NMI handler or after reset.

An NMI request is generated only on the first Master-TZF error after reset or clearing of the DMACDTCERRSTAT.MTERRSTAT bit by DMACDTCERRCLR.

(4) Slave-TZF Error

As described in section 45, Security Features , FHBIU (code flash), FLBIU (data flash), S0BIU (SRAM), PSBIU, PHBIU and PLBIU have Slave-TZF errors. See section 45.2.3. Peripheral Security Attribution for TZF error occurrence of each peripheral module. When a Slave-TZF error is detected, perform the following steps:

1. Store the address of the error in BTZFnERRADD (n = 1 to 3).
2. Store the read/write information of the error in BTZFnERRRW (n = 1 to 3).
3. Set 1 to STERRSTAT bit of BUSnERRSTAT (n = 1 to 3).

NMI request or reset request is generated according to the setting in TZFOAD.OAD. Since BTZFnERRADD (n = 1 to 3), BTZFnERRRW (n = 1 to 3), and BUSnERRSTAT (n = 1 to 3) are held until reset other than MPU- and TZF-related resets or cleared by BUSnERRCLR (n = 1 to 3), they can be verified in the NMI handler or after reset.

An NMI request is generated only on the first Slave-TZF error after a reset or clearing of the BUSnERRSTAT.STERRSTAT (n = 1 to 3) bit by BUSnERRCLR (n = 1 to 3).

(5) Slave Bus Error

Slave Bus Error occurs in the slave. When Slave Bus Error is detected, an Error response is returned to the master. At the same time, perform the following steps:

1. Store the address of the error in BUSnERRADD (n = 1 to 3)
2. Store the read/write information of the error in BUSnERRRW (n = 1 to 3)
3. Set 1 to SLERRSTAT bit of BUSnERRSTAT (n = 1 to 3).

An NMI request and reset request are not generated. Since BUSnERRADD (n = 1 to 3), BUSnERRRW (n = 1 to 3), and BUSnERRSTAT (n = 1 to 3) are held until reset other than MPU- and TZF-related resets or cleared by BUSnERRCLR (n = 1 to 3), they can be verified in the Bus Fault handler or interrupt handler.

13.4.3 Conditions Leading to Illegal Address Access Errors

Table 13.2 lists the address spaces for each bus that trigger illegal address access errors.

Table 13.2 Conditions leading to illegal address access errors (1 of 2)

Address	Slave bus	Master bus		
		CPU		DMA
		Code	System	
0x0000_0000 to 0x01FF_FFFF	FHBIU	—		—
0x0200_0000 to 0x07FF_FFFF	Reserved	E		E
0x0800_0000 to 0x0803_FFFF	FLBIU	—		—
0x0804_0000 to 0x0FFF_FFFF	Reserved	E		E
0x1000_0000 to 0x100F_FFFF	Reserved	—		E
0x1010_0000 to 0x1FFF_FFFF	Reserved	E		E

Table 13.2 Conditions leading to illegal address access errors (2 of 2)

Address	Slave bus	Master bus		
		CPU		DMA
		Code	System	
0x2000_0000 to 0x2800_FFFF	S0BIU		—	—
0x2801_0000 to 0x3FFF_FFFF	Reserved		E	E
0x4000_0000 to 0x4007_FFFF	PSBIU		—	—
0x4008_0000 to 0x400F_FFFF	PLBIU		—	—
0x4010_0000 to 0x4017_FFFF	PHBIU		—	—
0x4018_0000 to 0x407D_FFFF	Reserved		E	E
0x407E_0000 to 0x407F_FFFF	FLBIU		—	—
0x4080_0000 to 0x5FFF_FFFF	Reserved		E	E
0x6000_0000 to 0xDFFF_FFFF	Reserved		E	E
0xE000_0000 to 0xFFFF_FFFF	System for Cortex®-M33			E

Note: "E" : A bus error occurs.
" " : Transfer does not occur.
"—" : A bus error has not occurred. Even if there has reserved area, a bus error has not occurred. Do not access reserved area in FLBIU and S0BIU. If accessed, a slave TZF error might occur.

13.4.4 Time-out

For some peripheral modules, a timeout error occurs with the module-stop function. When there is no response from the slave for a certain period of time, a timeout error is detected. A timeout error is returned to the requesting master IP using the AHB-Lite error response protocol.

13.5 References

1. ARM Limited, *ARM v8-M Architecture Reference Manual* (ARM DDI0553B.g)
2. ARM Limited, *ARM Cortex-M33 Processor Technical Reference Manual Revision:r0p4* (ARM 100230_0004_00_en)
3. ARM Limited, *ARM AMBA 5 AHB Protocol Specification AHB5, AHB-Lite* (ARM IHI 0033B.b)
4. ARM Limited, *ARM AMBA AXI and ACE Protocol Specification AXI3, AXI4, and AXI4-Lite, ACE and ACE-Lite* (ARM IHI 0022D)
5. ARM Limited, *ARM AMBA APB Protocol Specification Version: 2.0* (ARM IHI 0024C)

13.6 Cache

13.6.1 Overview

There are two types of caches:
- C-cache on code bus
- S-cache on system bus.

Table 13.3 lists the specifications of the cache, Figure 13.4 shows a block diagram of the cache, and Figure 13.5 shows the cache structure.

Table 13.3 Cache specifications (1 of 2)

Parameter	C-cache	S-cache
Capacity	4 KB	4 KB
Way	4-way set associative	4-way set associative
Line size	32/64 bytes	32/64 bytes
Number of entry	32/16 entry/way	32/16 entry/way

Table 13.3 Cache specifications (2 of 2)

Parameter	C-cache	S-cache
Write way	No write	Write-through, non-write allocate
Replace way	4-way: Full LRU (least recently used)	4-way: Full LRU (least recently used)
Cache support area	0x0000_0000 to 0x1FFF_FFFF	0x2000_0000 – 0xDFFF_FFFF*1 except Standby SRAM area (0x2800_0000 to 0x2FFF_FFFF)

Note 1. Peripheral area 0x4000_0000 to 0x5FFF_FFFF must not have the cacheable attribution in the Arm MPU.

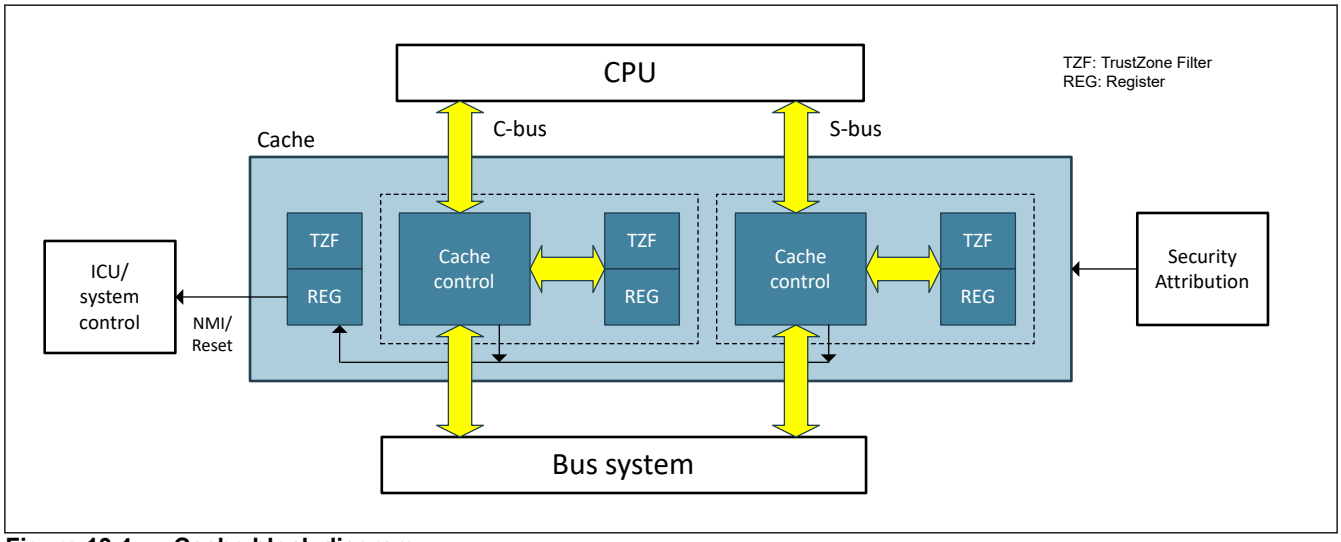


Figure 13.4 Cache block diagram

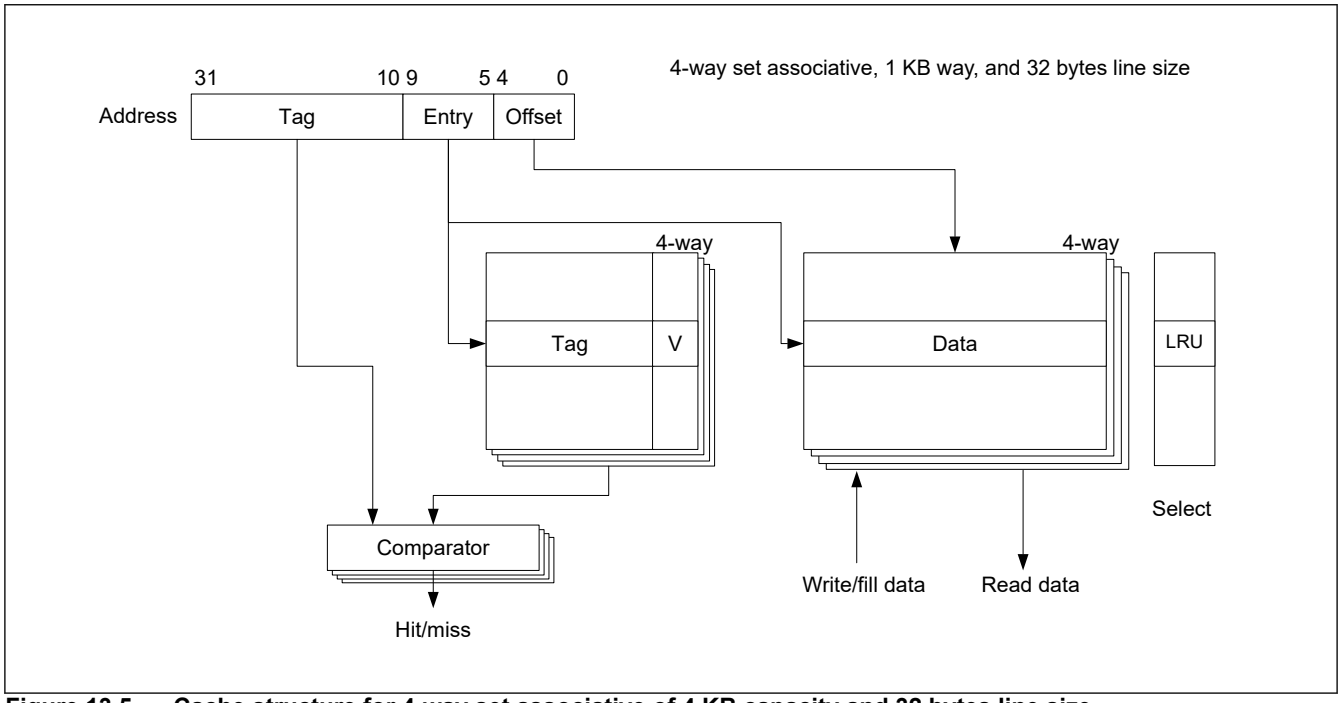


Figure 13.5 Cache structure for 4-way set associative of 4 KB capacity and 32 bytes line size

13.6.2 Register Description

13.6.2.1 CSAR : Cache Security Attribution Register

Base address: CPSCU = 0x4000_8000

Offset address: 0x000

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

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

Bit	Symbol	Function	R/W
0	CACHESA	Security Attributes of Registers for Cache Control 0: Secure 1: Non-secure	R/W
1	CACHELSA	Security Attributes of Registers for Cache Line Configuration 0: Secure 1: Non-secure	R/W
2	CACHEESA	Security Attributes of Registers for Cache Error 0: Secure 1: Non-secure	R/W
31:3	—	These bits are read as 1.	R

Note: Only secure access can write to this register. Both secure access and Non-secure read access are allowed but Non-secure write access is not allowed and TrustZone access error is not generated.
Note: This register is write-protected by PRCR register.
Note: When CACHESA = 0 (secure), cache maintenance operation cannot be performed after updated the program. Therefore, when debugging, software breakpoints cannot be used for the cache target area.

CACHESA bit (Security Attributes of Registers for Cache Control)

The CACHESA bit indicates the security attributes of registers for cache control. The target registers are:

- CCACTL
- CCAFCT
- SCACTL
- SCAFCT.

CACHELSA bit (Security Attributes of Registers for Cache Line Configuration)

The CACHELSA bit indicates the security attributes of registers for cache line configuration. The target registers are:

- CCALCF
- SCALCF.

CACHEESA bit (Security Attributes of Registers for Cache Error)

The CACHEESA bit indicates the security attributes of registers for cache error.

- CAPOAD
- CAPRCR.

13.6.2.2 CCACTL : C-Cache Control Register

Base address: CACHE = 0x4000_7000

Offset address: 0x000

Bit position:	31																													0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ENC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ENC	C-Cache Enable Set the C-cache enable: 0: Disable C-cache 1: Enable C-cache	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

ENC bit (C-Cache Enable)

The ENC bit controls the cache enable of C-cache. When the ENC bit changes from 0 to 1, the Valid bit of C-cache is cleared.

13.6.2.3 CCAFCT : C-Cache Flush Control Register

Base address: CACHE = 0x4000_7000

Offset address: 0x004

Bit position:	31																													0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	FC	C-Cache Flush Set the C-cache line flush: 0: No action 1: C-cache line flush (all lines invalidated)	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

FC bit (C-Cache Flush)

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

[Setting condition]

- When writing 1 to this bit.
- When setting CCACTL.ENC bit from 0 to 1.

[Clearing condition]

- This bit is cleared automatically when cache flush is performed.

13.6.2.4 CCALCF : C-Cache Line Configuration Register

Base address: CACHE = 0x4000_7000

Offset address: 0x008

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

Bit	Symbol	Function	R/W
1:0	CC[1:0]	C-Cache Line Size Set the C-cache line size: 0 0: Prohibited 0 1: Cache line size 32 bytes 1 0: Cache line size 64 bytes 1 1: Prohibited	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

CC[1:0] bits (C-Cache Line Size)

The CC[1:0] bits control the cache line size of C-cache. This bit can be written when the CCACTL.ENC bit is 0. Otherwise, this bit cannot be written.

13.6.2.5 SCACTL : S-Cache Control Register

Base address: CACHE = 0x4000_7000

Offset address: 0x040

Bit position:	31																														0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ENS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ENS	S-Cache Enable Set the S-cache enable: 0: Disable S-cache 1: Enable S-cache	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

ENS bit (S-Cache Enable)

The ENS bit controls the cache enable of S-cache. When the ENS bit changes from 0 to 1, the Valid bit of S-cache is cleared.

13.6.2.6 SCAFCT : S-Cache Flush Control Register

Base address: CACHE = 0x4000_7000

Offset address: 0x044

Bit position:	31																													0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	FS	S-Cache Flush Set the S-cache line flush: 0: No action 1: S-cache line flush (all lines invalidated)	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

FS bit (S-Cache Flush)

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

[Setting condition]

When writing 1 to this bit.

When setting SCACTL.ENS bit from 0 to 1.

[Clearing condition]

This bit is cleared automatically when cache flush is performed.

13.6.2.7 SCALCF : S-Cache Line Configuration Register

Base address: CACHE = 0x4000_7000

Offset address: 0x048

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

Bit	Symbol	Function	R/W
1:0	CS[1:0]	S-Cache Line Size Set the S-cache line size: 0 0: Prohibited 0 1: Cache line size 32 bytes 1 0: Cache line size 64 bytes 1 1: Prohibited	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

CS[1:0] bits (S-Cache Line Size)

The CS[1:0] bits control the cache line size of S-cache. This bit can be written when the SCACTL.ENS bit is 0. Otherwise, this bit cannot be written.

[Setting condition]

This bit can be written when the CACTL.ENS bit is 0. Otherwise, this bit cannot be written.

13.6.2.8 CAPOAD : Cache Parity Error Operation After Detection Register

Base address: CACHE = 0x4000_7000

Offset address: 0x200

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

Bit	Symbol	Function	R/W
0	OAD	Operation after Detection Select the operation after detection: 0: Non-maskable interrupt 1: Reset	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

OAD bit (Operation after Detection)

The OAD bit generates either a reset or a non-maskable interrupt when parity error is detected.

The protection register (CAPRCR) protects this register against writes. Modify the valid bit in the protection register (CAPRCR) to enable before writing to this bit. Do not write to CAPOAD while access to cache is in progress.

13.6.2.9 CAPRCR : Cache Protection Register

Base address: CACHE = 0x4000_7000

Offset address: 0x204

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

Bit	Symbol	Function	R/W
0	PRCR	Register Write Control Set the register write control: 0: Disable writes to protected registers 1: Enable writes to protected registers	R/W
7:1	KW[6:0]	Write key code These bits enable or disable writes to the PRCR bit.	R/W

Bit	Symbol	Function	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

PRCR bit (Register Write Control)

The PRCR bit controls the write mode of the CAPOAD register. When this bit is set to 1, writing to the CAPOAD register is enabled. When writing to this bit, write 0x78 to the KW[6:0] bits simultaneously.

KW[6:0] bits (Write key code)

The KW[6:0] bits enable or disable writes to the PRCR bit. When writing to the PRCR bit, write 0x78 to the KW[6:0] bits simultaneously. When a value other than 0x78 is written to KW[6:0] bits, the PRCR bit is not updated. The KW[6:0] bits are always read as 0x00.

13.6.3 Operation

13.6.3.1 S-Cache

Figure 13.6 shows the access flow from CPU to S-cache.

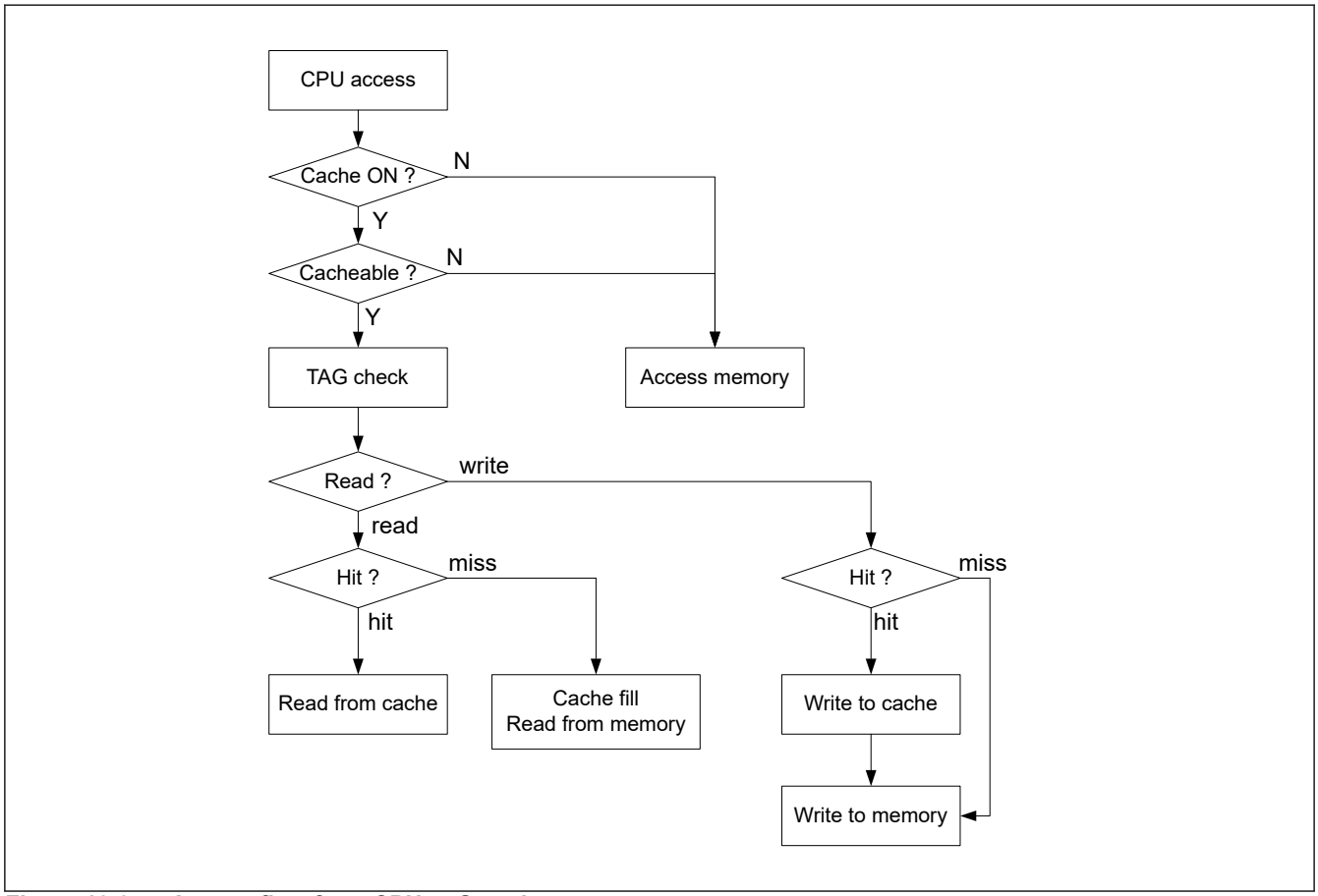


Figure 13.6 Access flow from CPU to S-cache

The cache function works when cache is enabled (CACTL.ENS = 1) and cacheable access is from CPU. The cache checks the address of CPU access request and request in cache tag, then determines whether the CPU access is a hit or a miss-hit.

Read miss

The cache reads one cache line data from memory and stores it into the cache data. The cache then returns the required data to CPU.

Read hit

The cache reads required data from the cache data and returns it to CPU. Access cycle then determines it is a hit because of the 0 wait cycle.

Write miss

The cache processes only a write cycle to memory. No affect to cache data.

Write hit

The cache processes both a write cycle to cache data and a write cycle to memory.

13.6.3.2 C-Cache

Figure 13.7 shows the access flow from CPU to C-cache.

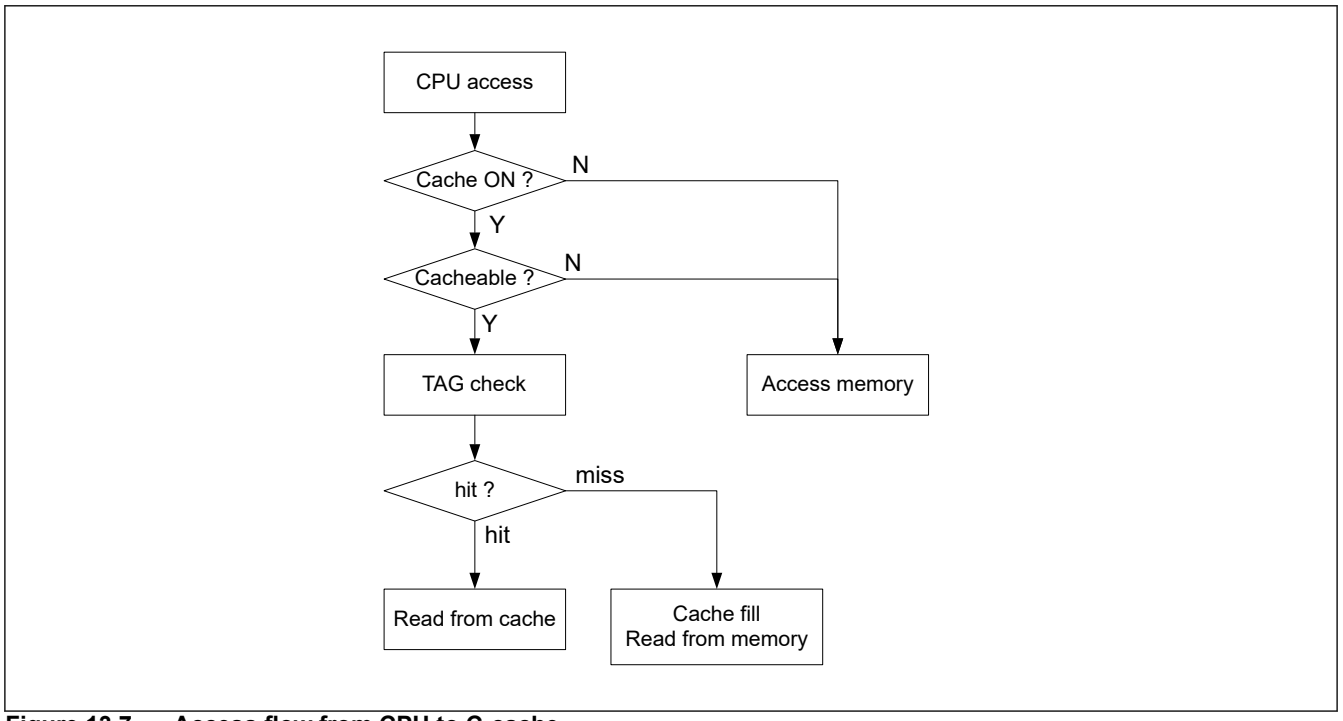


Figure 13.7 Access flow from CPU to C-cache

The cache function works when cache is enabled (CACTL.ENC = 1) and cacheable access is from CPU. The cache checks the address of CPU access request and request in cache tag, then determines whether the CPU access is a hit or a miss-hit.

Read miss

The cache reads one cache line data from memory and stores it into the cache data. The cache then returns the required data to CPU.

Read hit

The cache reads required data from the cache data and returns it to CPU. Access cycle then determines it is a hit because of the 0 wait cycle.

Because C-cache does not function in the ROM area of C-cache, therefore it operates in read-only access.

13.6.3.3 Cache Flush

The Valid bit is cleared with the CAFCT register. However, tag and cache data are not affected by the CAFCT register.

The valid bit is also cleared when CACTL is set from 0 to 1.

Note: After changing the cacheable attribute by the Arm MPU, clear the valid bit using the CAFCT register.

13.6.3.4 LRU and Replace

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

The algorithm for a 4-way Full-LRU shows the order each way, for example way 0 to way 3.

13.6.3.5 Parity Check

The cache has a parity check function for cache RAM that is stored as cache fill data. The cache has 4-bit parities for 32-bit data, that is when data is read, a parity bit is added to every 8-bit data of 32-bit data width. When the cache reads data with a hit status, it checks for parity errors. When a parity error occurs, a parity error notification is generated.

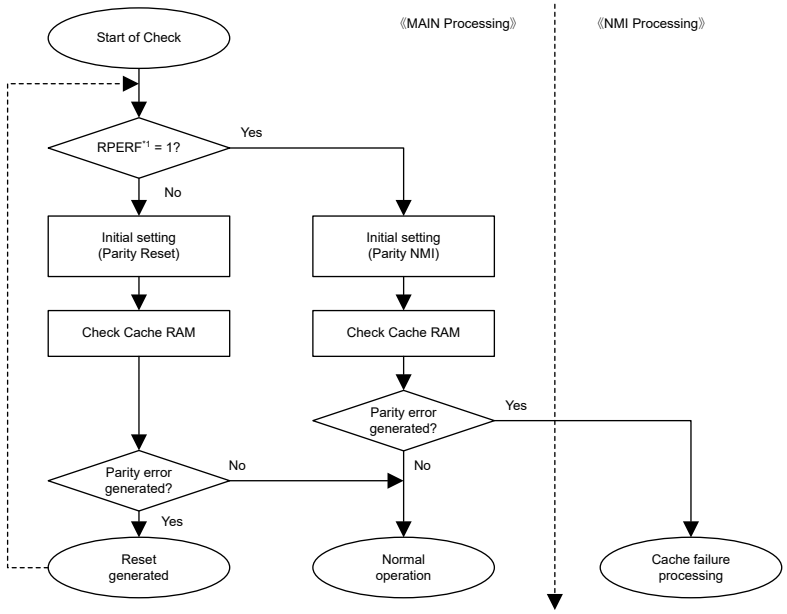
The cache reads 32-bit data even when the CPU requests a byte read or half-word read.

Note: A parity error might occur even though it is caused by a non validated-data byte of which the CPU does not request.

Parity error notification can be specified as a non-maskable interrupt or a reset request in the CAPOAD register. However, if the debug mode requests to suppress the parity error notification, then notification is not generated.

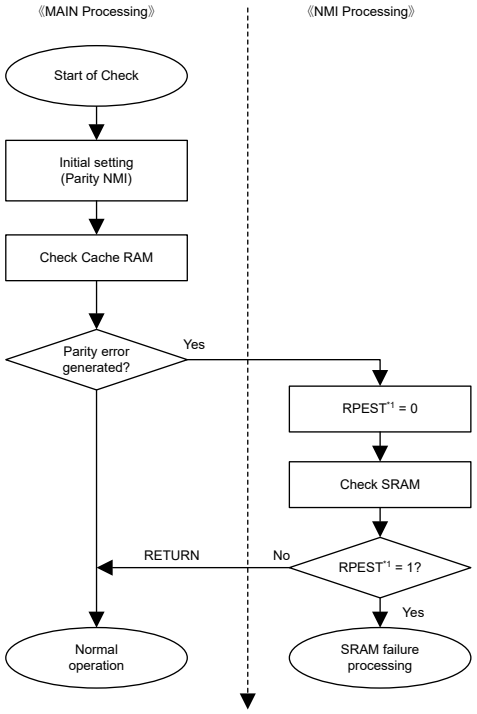
When a parity error occurs, the cache does not perform a cache flush and does not respond to the CPU with a bus error.

Parity errors often occur due to noise. To confirm whether the cause of the parity error is noise or corruption, see the flows for cache parity check in Figure 13.8 and Figure 13.9.



Note 1. RPERF: Cache Parity Error Reset Detect Flag (RSTSR1.RPERF)

Figure 13.8 Flow of cache parity check when parity reset is enabled



Note 1. RPEST: Cache Parity Error Interrupt Detect Flag (NMISR.RPEST)

Figure 13.9 Flow of cache parity check when parity interrupt is enabled

13.6.3.5.1 Cache RAM Check

Parity error of cache RAM occurs at a read access by CPU with a cache status of read-hit. With a read-hit status, some conditions are required before performing a cache RAM check. For S-cache check, execute the check program in flash memory. For C-cache check, execute the check program in SRAM.

(1) Cache RAM check flow

1. Flush all valid bits in cache to clear the cache enable bit.
2. Reserve a 4-KB work memory such as SRAM for S-cache. Because each cache in the MCU is a 4-way set associative with 1 KB RAM per way, a total of 4 KB is required for S-cache. The target address should not be used as reserved area.
3. Set the cache enable to 1.
4. Read data from the target word address of 4 KB using the CPU. The status of cache should be a read-miss with the result stored as cache fill data.
5. Read data in another cache way whose address is calculated by adding the address in step 4. with 1 KB address. The status of cache should be a read-miss with the result stored as cache fill data in another way.
6. Read data in another cache way whose address is calculated by adding the address in step 5. with 1 KB address. The status of cache should be a read-miss with the result stored as cache fill data in another way.
7. Read data in another cache way whose address is calculated by adding the address in step 6. with 1 KB address. The status of cache should be a read-miss with the result stored as cache fill data in another way. Cache RAM check for a write/read-hit status is now complete.
8. Write test data to the target word address in steps 4., 5., 6., and 7.. The status of cache in steps 4., 5., 6., and 7. should be a write-hit with the results written to the cache RAM.
9. Read from the target word address in steps 4., 5., 6., and 7. again. The status of cache in steps 4., 5., 6., and 7. should be a read-hit. Parity check for a word data is now complete.
10. Go to step 1. to continue parity check for different target addresses.

13.6.3.6 Bus Error

The association from a bus slave to a bus error is described in the sections that follow.

In cache off

The cache returns a bus error to the CPU.

For non-cacheable access

The cache returns a bus error to the CPU.

During read accesses for cache fill

For the first data that corresponds to a CPU access request, the cache returns a bus error to the CPU. For other read data while filling the cache line, the cache cannot return a bus error to the CPU except for read data with early forwarding. The cache enable bit clears the cache line if the cache accepts a bus error response from the slave.

For write-hit status

The cache cannot return a bus error to the CPU because the cache enable bit does not clear the cache line.

For write-miss status

The cache cannot return a bus error to the CPU.

13.6.3.7 Early Forwarding Function

While filling data in the cache, if the address of the CPU read request and the address of the cache fill request are the same, the cache returns the data to CPU. Table 13.4 shows an example.

Table 13.4 Example of early forwarding

Operation	Access sequence								
Address of CPU read request	0x04	0x08	0x0C	0x14	→	0x10	→	→	→
Address of cache fill	0x04	0x08	0x0C	0x10	0x14	0x18	0x1C	0x00	—
CPU access status	Read (0x04)	Read (0x08)	Read (0x0C)	—	Read (0x14)	—	—	—	Read (0x10)

When the CPU requests read accesses and the addresses are 0x04, 0x08, 0x0C, 0x14 and 0x10 sequentially, the first read access to address 0x04 is of a miss-hit status and the cache starts to fill data into cache. The early forwarding function allows a return of read data to CPU when accesses are to addresses 0x08, 0x0C and 0x14 while the cache is filling the cache line. On the other hand, access to address 0x10 must wait for the completion of filling the cache line. The cache then returns data for address 0x10 when it finished filling the cache line.

13.6.4 Usage Notes

13.6.4.1 Cache Line Configuration Register

Writes to the Cache Line Configuration Register are allowed when the status is cache off (CACTL.ENS = 0 for S-cache, CACTL.ENC = 0 for C-cache).

13.6.4.2 Coherency

The coherency between the cache and the internal SRAM must be guaranteed by software.

When allocating shared memory between the CPU and a bus master such as DMAC in the cache support area, invalidate the cache data as necessary.

14. Memory Protection Unit (MPU)

14.1 Overview

The MCU has one Memory Protection Unit (MPU).

Table 14.1 lists the MPU specifications, and Table 14.2 shows the behavior on detection of each MPU error.

Table 14.1 MPU specifications

Classification	Module/Function	Specifications
Illegal memory access	Arm® Cortex®-M33 CPU	• Arm CPU has a default memory map. If the CPU makes an illegal access, an exception interrupt occurs • The MPU can change a default memory map.
Memory protection	Arm MPU	Memory protection function for the CPU: • (8+8) region MPU with sub regions and background region for secure and non-secure.
	Bus master MPU	Memory protection function for each bus master except for the CPU: • DMAC/DTC: 8 regions

Table 14.2 Behavior on MPU error detection

MPU type	Notification type	Error Response by HRESP signal of AHB I/F	Bus Access on error detection	Storing of error access information
Arm MPU	• Hard fault	Not supported	• Does not correctly write access • Does not correctly read access	Stored in the Cortex-M33 processor
Bus Master MPU	• Reset or Non-maskable interrupts • Hard fault	Supported	• Write access ignore • Read access is read as 0	Stored

For information on error access for the Arm MPU, see section 14.4. References. For information on error access for other MPUs, see section 13.3. Register Descriptions and section 13.4. Bus Error Monitoring Section in section 13, Buses.

14.2 Arm MPU

The Arm MPU monitors the addresses accessed by the CPU across the entire address space (0x0000_0000 to 0xFFFF_FFFF) and provides support for:

- (8 + 8) protected regions
- When memory regions overlap, the processor generates a fault if a core access hits the overlapping regions
- Setting access permissions to protected region (Read, Write, Execution)
- Export of memory attributes to the system.

Arm MPU mismatches and permission violations invoke the programmable-priority MemManage fault (Hard Fault) handler. For details, see section 14.4. References.

14.3 Bus Master MPU

The bus master MPU monitors the addresses accessed by the bus masters in the entire address space (0x0000_0000 to 0xFFFF_FFFF). Access-control information can be set up to 8 regions in DMAC/DTC and monitor for access to each region is in accord with this information.

If access to a protected region is detected, the bus master MPU generates an internal reset or a non-maskable interrupt. For information on error access, see section 13.3. Register Descriptions and section 13.4. Bus Error Monitoring Section in section 13, Buses.

The access control information for each area consists of protected/not-protected to read or write.

Table 14.3 lists the specifications of the bus master MPU.

Table 14.3 Bus master MPU specifications

Parameter	Description
Protected master groups	• DMAC, DTC
Protected regions	0x0000_0000 to 0xFFFF_FFFF
Number of regions	• DMAC/DTC: 8 regions
Address specification for individual regions	• Specifying start and end address for individual regions
Enable or disable setting for memory protection in individual regions	• Enabling or disabling setting for the associated region
Access-control settings for individual regions	• Permission for read and write
Operation on error detection	• Reset or non-maskable interrupts
Register protection	• Protecting registers from illegal writes
TrustZone Filter	• DMAC: Security attribution can be set for each regions

14.3.1 Register Descriptions

Bus access must be stopped before writing to MPU registers.

14.3.1.1 MMPUSARA : Master Memory Protection Unit Security Attribution Register A

Base address: CPSCU = 0x4000_8000

Offset address: 0x130

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	MMPUASA7	MMPUASA6	MMPUASA5	MMPUASA4	MMPUASA3	MMPUASA2	MMPUASA1	MMPUASA0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
7:0	MMPUASAn	MMPUA Security Attribution (n = 0 to 7) 0: Secure 1: Non-Secure	R/W
31:8	—	These bits are read as 1.	R*1

Note: Only secure access can write to this register. Both secure access and Non-secure read access are allowed but Non-secure write access is not allowed and TrustZone access error is not generated.

Note: This register is write-protected by PRCR register.

Note 1. This bit is read only.

MMPUASAn bits (MMPUA Security Attribution (n = 0 to 7))

The MMPUASAn bits specify the security attributes of registers for the Bus Master MPU Region Setting register. The target registers are:

- MMPUSDMACn (n = 0 to 7)
- MMPUEDMACn (n = 0 to 7)
- MMPUACDMACn (n = 0 to 7)

14.3.1.2 MMPUSARB : Master Memory Protection Unit Security Attribution Register B

Base address: CPSCU = 0x4000_8000

Offset address: 0x134

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MMPUBSA0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	MMPUBSA0	MMPUB Security Attribution 0: Secure 1: Non-Secure	R/W
31:1	—	These bits are read as 1.	R*1

Note: Only secure access can write to this register. Both secure access and Non-secure read access are allowed but Non-secure write access is not allowed and TrustZone access error is not generated.
Note: This register is write-protected by PRCR register.
Note 1. This bit is read-only.

MMPUBSA0 bit (MMPUB Security Attribution)

The MMPUBSA0 bit specifies the security attributes of registers for the Bus Master MPU Region Setting register, Protect register, and OAD register. The target registers are:

- MMPUENDMAC
- MMPUENPTDMAC
- MMPURPTDMAC
- MMPURPTDMAC_SEC
- MMPUOAD
- MMPUOADPT

The Secure user provides a Secure API to Non-secure user for the modification of the MMPURPTDMAC value when MMPUBSA0 bit is set to 0 (Secure).

14.3.1.3 MMPUSDMACn : MMPU Start Address Register for DMAC (n = 0 to 7)

Base address: RMPU = 0x4000_0000

Offset address: 0x0204 + 0x010 × n

Bit position:	31	5	4	3	2	1	0
Bit field:	MMPUS[31:5]	—	—	—	—	—	
Value after reset:	x x x x x x x x x x x x x x x x x x x x x x x x x x x	0	0	0	0	0	

Bit	Symbol	Function	R/W
4:0	—	These bits are read as 0. The write value should be 0.	R/W
31:5	MMPUS[31:5]	Region start address register Address where the region starts, for use in region determination	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

The MMPUSDMACn (n = 0 to 7) register specifies the start address where the region starts.

This register requires word access. Byte access and half word access is prohibited. When byte access and half word access is executed, operation is not guaranteed.

Regions set by MMPUSDMACn (n = 0 to 7), MMPUEDMACn (n = 0 to 7) and MMPUACDMACn (n = 0 to 7) registers, can be set for a secure access or a non-secure access with the MMPUSARA register. If the corresponding MMPUSARA.MMPUASAn (n = 0 to 7) bit is set to 1, only non-secure access is permitted for that region. On the other hand, if the corresponding MMPUSARA.MMPUASAn (n = 0 to 7) bit is set to 0, only secure access is permitted for that region.

14.3.1.4 MMPUEDMACn : MMPU End Address Register for DMAC (n = 0 to 7)

Base address: RMPU = 0x4000_0000

Offset address: 0x0208 + 0x010 × n

Bit position:	31																							5					0
Bit field:	MMPUE[31:5]																							—	—	—	—	—	
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	1	1	1	1	1	

Bit	Symbol	Function	R/W
4:0	—	These bits are read as 1. The write value should be 1.	R/W
31:5	MMPUE[31:5]	Region end address register Address where the region end, for use in region determination	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

The MMPUEDMAC (n = 0 to 7) register specifies the end address where the region ends.

This register requires word access. Byte access and half word access is prohibited. When byte access and half word access is executed, operation is not guaranteed.

14.3.1.5 MMPUACDMACn : MMPU Access Control Register for DMAC (n = 0 to 7)

Base address: RMPU = 0x4000_0000

Offset address: 0x0200 + 0x010 × n

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

Bit	Symbol	Function	R/W
0	ENABLE	Region enable 0: DMAC Region n unit is disabled 1: DMAC Region n unit is enabled	R/W
1	RP	Read protection 0: Read permission 1: Read protection	R/W
2	WP	Write protection 0: Write permission 1: Write protection	R/W
15:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Region n unit sets the ENABLE bit, the RP bit, and the WP bit each.

ENABLE bit (Region enable)

The ENABLE bit controls the enable or disable of DMAC/DTC region n (n = 0 to 7) unit.

When the ENABLE bit is set to 1, the RP bit and the WP bit control access permission for read and write protection to MMPUSDMACn (n = 0 to 7) and MMPUEDMACn (n = 0 to 7).

When the ENABLE bit is set to 0, access to DMAC region n (n = 0 to 7) is the outside region.

RP bit (Read protection)

The RP bit enables or disables read protection of DMAC/DTC region n (n = 0 to 7).

When the ENABLE bit is set to 1, the RP bit is available.

WP bit (Write protection)

The WP bit enables or disables write protection of DMAC/DTC region n (n = 0 to 7).

When the ENABLE bit is set to 1, the WP bit is available.

Table 14.4 Function of Region Control Circuit for DMAC

MMPUACDMACn (n = 0 to 7)			Access	Region	Output of DMAC Region n unit (n = 0 to 7)
ENABLE	RP	WP			
0	—	—	Read	—	Outside region
			Write		Outside region
1	0	0	Read	Inside	Permitted region
				Outside	Outside region
			Write	Inside	Permitted region
				Outside	Outside region
	0	1	Read	Inside	Permitted region
				Outside	Outside region
			Write	Inside	Protection region
				Outside	Outside region
	1	0	Read	Inside	Protection region
				Outside	Outside region
			Write	Inside	Permitted region
				Outside	Outside region
	1	1	Read	Inside	Protection region
				Outside	Outside region
			Write	Inside	Protection region
				Outside	Outside region

Note: Each regions of DMAC / DTC are set for secure access and non-secure access by MMPUSARA register. In this case, Non-Secure regions in secure access and secure regions in Non-Secure access are outside regions.

Table 14.5 Function of Master Control Circuit for DMAC

MMPUENDMAC ENABLE	Output of DMAC Region 0 unit	Output of DMAC Region 1 unit	Output of DMAC Region 2-7 unit	Function of DMAC
1	Protected region	Don't care	Don't care	Generate error
	Don't care	Protected region	Don't care	Generate error
	Don't care	Don't care	Protected region	Generate error
	Outside region	Outside region	Outside region	Generate error
Other cases				No error

A master MPU error occurs on the following conditions:

1. MMPUENDMAC.ENABLE = 1, and output of one or more Region n unit is protected region.
2. MMPUENDMAC.ENABLE = 1, and output of all Region n unit are outside region.

Other cases are handled as permitted region.

14.3.1.6 MMPUENDMAC : MMPU Enable Register for DMAC

Base address: RMPU = 0x4000_0000

Offset address: 0x0100

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

Bit	Symbol	Function	R/W
0	ENABLE	Bus Master MPU of DMAC enable 0: Bus Master MPU of DMAC is disabled. 1: Bus Master MPU of DMAC is enabled.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code These bits enable or disable writes to the ENABLE bit.	W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note: It is necessary to write by half word access.
Byte write access is prohibited. When byte write access is executed, operation is not guaranteed.

ENABLE bit (Bus Master MPU of DMAC enable)

The ENABLE bit controls enable or disable of the bus master MPU function of each master group.

When the ENABLE bit is set to 1, MMPUACDMACn (n = 0 to 7) is valid. When the ENABLE bit is set to 0, MMPUACDMACn (n = 0 to 7) is invalid for all regions. The bus master MPU function sets the ENABLE bit of each master group. When the ENABLE bit is set, write 0xA5 in KEY[7:0] at the same time.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits enable or disable writing to the ENABLE bit. When writing to the ENABLE bit, write 0xA5 in KEY[7:0] bits at the same time. When values other than 0xA5 are written to KEY[7:0] bits, the ENABLE bit is not updated. The KEY[7:0] bits are always read as 0x00.

14.3.1.7 MMPUENPTDMAC : MMPU Enable Protect Register for DMAC

Base address: RMPU = 0x4000_0000

Offset address: 0x0104

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

Bit	Symbol	Function	R/W
0	PROTECT	Protection of register 0: MMPUENDMAC register writes are possible. 1: MMPUENDMAC register writes are protected. Read is possible.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code These bits enable or disable writes to the PROTECT bit.	W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note: It is necessary to write by half word access.
Byte write access is prohibited. When byte write access is executed, operation is not guaranteed.

PROTECT bit (Protection of register)

The PROTECT bit enables or disables writes to the MMPUENDMAC register.

When writing to the PROTECT bit, write 0xA5 in KEY[7:0] at the same time.

KEY[7:0] bits (Key Code)

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

14.3.1.8 MMPURPTDMAC : MMPU Regions Protect Register for DMAC

Base address: RMPU = 0x4000_0000

Offset address: 0x0108

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

Bit	Symbol	Function	R/W
0	PROTECT	Protection of register 0: Bus Master MPU register for DMAC writing is possible. 1: Bus Master MPU register for DMAC writing is protected. Read is possible.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code These bits enable or disable writes to the PROTECT bit.	W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:

- Secure and Non-secure access are allowed.

Note: It is necessary to write by half word access.
Byte write access is prohibited. When byte write access is executed, operation is not guaranteed.

PROTECT bit (Protection of register)

The PROTECT bit enables or disables writes to the associated registers to be protected.

MMPURPTDMAC.PROTECT controls the following registers:

- MMPUSDMACn (n = 0 to 7) of Non-Secure program
- MMPUEDMACn (n = 0 to 7) of Non-Secure program
- MMPUACDMACn (n = 0 to 7) of Non-Secure program

When writing to the PROTECT bit, write 0xA5 simultaneously to the KEY[7:0] bits, using halfword access.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits enable or disable writes to the PROTECT bit. When writing to the PROTECT bit, write 0xA5 simultaneously to the KEY[7:0] bits. When other values are written, the PROTECT bit is not updated.

The KEY[7:0] bits always read as 0x00.

14.3.1.9 MMPURPTDMAC_SEC : MMPU Regions Protect register for DMAC Secure

Base address: RMPU = 0x4000_0000

Offset address: 0x010C

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

Bit	Symbol	Function	R/W
0	PROTECT	Protection of register 0: Bus master MPU register for DMAC secure writes are possible. 1: Bus master MPU register for DMAC secure writes are protected. Read is possible.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code These bits enable or disable writes to the PROTECT bit.	W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note: It is necessary to write by half word access.
Byte write access is prohibited. When byte write access is executed, operation is not guaranteed.

PROTECT bit (Protection of register)

The PROTECT bit enables or disables writes to the associated registers to be protected.

MMPURPTDMAC_SEC.PROTECT controls the following registers:

- MMPUSDMACn (n = 0 to 7) of Secure program
- MMPUEDMACn (n = 0 to 7) of Secure program
- MMPUACDMACn (n = 0 to 7) of Secure program

When writing to the PROTECT bit, write 0xA5 simultaneously to the KEY[7:0] bits, using halfword access.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits enable or disable writes to the PROTECT bit. When writing to the PROTECT bit, write 0xA5 simultaneously to the KEY[7:0] bits. When other values are written, the PROTECT bit is not updated.

The KEY[7:0] bits are always read as 0x00.

14.3.1.10 MMPUOAD : MMPU Operation After Detection Register

Base address: RMPU = 0x4000_0000

Offset address: 0x0000

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

Bit	Symbol	Function	R/W
0	OAD	Operation after detection 0: Non-maskable interrupt 1: Reset	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code This bit enables or disables writes to the OAD bit.	W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note: It is necessary to write by half word access.
Byte write access is prohibited. When byte write access is executed, operation is not guaranteed.

OAD bit (Operation after detection)

The OAD bit is specified to generate either reset or non-maskable interrupt when the access to the protect region is detected by the BUS Master MPU.

When writing to the OAD bit, write 0xA5 simultaneously to the KEY[7:0] bits using halfword access.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits enable or disable writing to the OAD bit. When writing to the OAD bit, write 0xA5 simultaneously to the KEY[7:0] bits. When other values are written, the OAD bit is not updated.

The KEY[7:0] bits always read as 0x00.

14.3.1.11 MMPUOADPT : MMPU Operation After Detection Protect Register

Base address: RMPU = 0x4000_0000

Offset address: 0x0004

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

Bit	Symbol	Function	R/W
0	PROTECT	Protection of register 0: MMPUOAD register writes are possible. 1: MMPUOAD register writes are protected. Read is possible.	R/W

Bit	Symbol	Function	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key code These bits enable or disable writes to the PROTECT bit.	W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note: It is necessary to write by half word access.
Byte write access is prohibited. When byte write access is executed, operation is not guaranteed.

PROTECT bit (Protection of register)

The PROTECT bit enables or disables writes to the associated registers to be protected.

MMPUOADPT.PROTECT controls the following register:

- MMPUOAD

When the PROTECT bit is set simultaneously, write 0xA5 to the KEY[7:0] bits using half word access.

KEY[7:0] bits (Key code)

The KEY[7:0] bits enable or disable writes to the PROTECT bit. When writing to the PROTECT bit simultaneously, write 0xA5 to the KEY[7:0] bits. When other values are written to the KEY[7:0] bits, the PROTECT bit is not updated. The KEY[7:0] bits are always read as 0x00.

14.3.2 Operation

14.3.2.1 Memory protection

The bus master MPU monitors memory access using control settings made individually for the access control regions. If access to a protected region is detected, the bus master MPU generates a memory protection error.

Bus Master MPU can be set for up to 8 protection regions. It is protection region when set up of permission region and protection region overlaps. It is protection region when set up of two protection region overlaps.

Bus Master MPU has master groups of DMAC/DTC.

Memory protection checks the address of the bus which the master group unified. Therefore, all the access of a master group is detected by memory protection.

The region setting registers of the Bus Master MPU for DMAC/DTC can be set for secure access and Non-Secure access using the MMPUSARA register. Make secure access and Non-Secure access settings the same for each DMAC/DTC channel and the corresponding region setting registers of the Bus Master MPU.

Bus Master MPU is permission of all regions after reset. All region is protected by setting MMPUENDMAC.ENABLE = 1.

Each region sets up a permission region on the protection region. If access to the protected region is detected, Bus Master MPU will generate an error.

Figure 14.1 shows the use case of a bus master MPU.

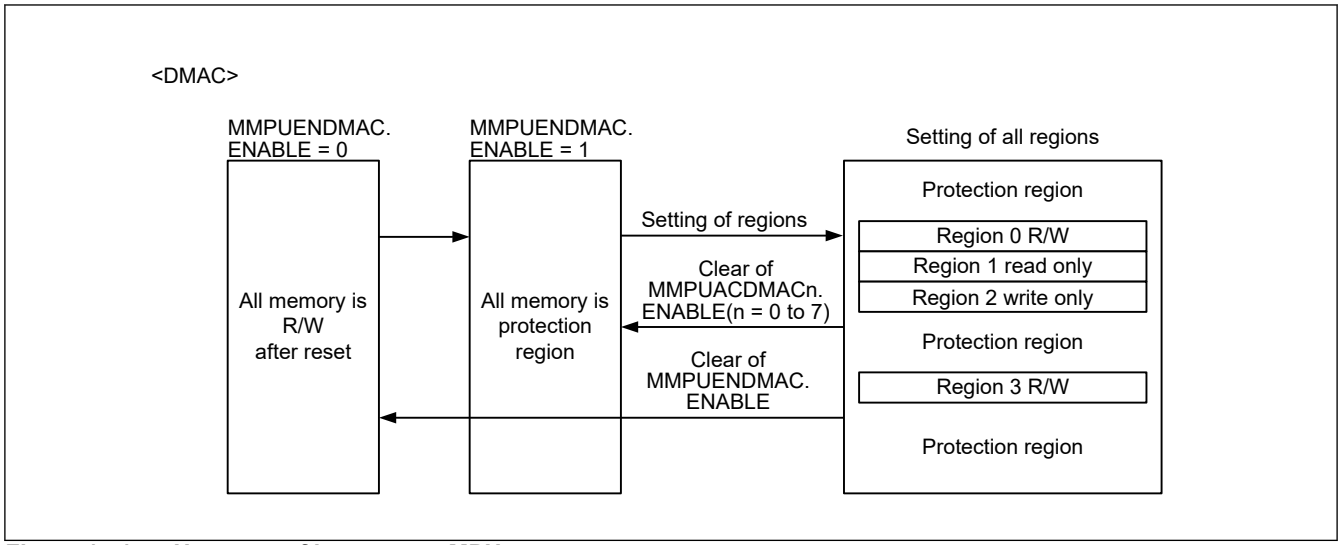


Figure 14.1 Use case of bus master MPU

Figure 14.2 shows the access permission or protection by the overlapping bus master MPU regions.

Access control for the overlapping regions is as follows:

- The region is handled as a protected region when output of one or more region units is a protected region
- The region is handled as a protected region when output of all region units is outside of the regions
- Other cases are handled as permitted regions.

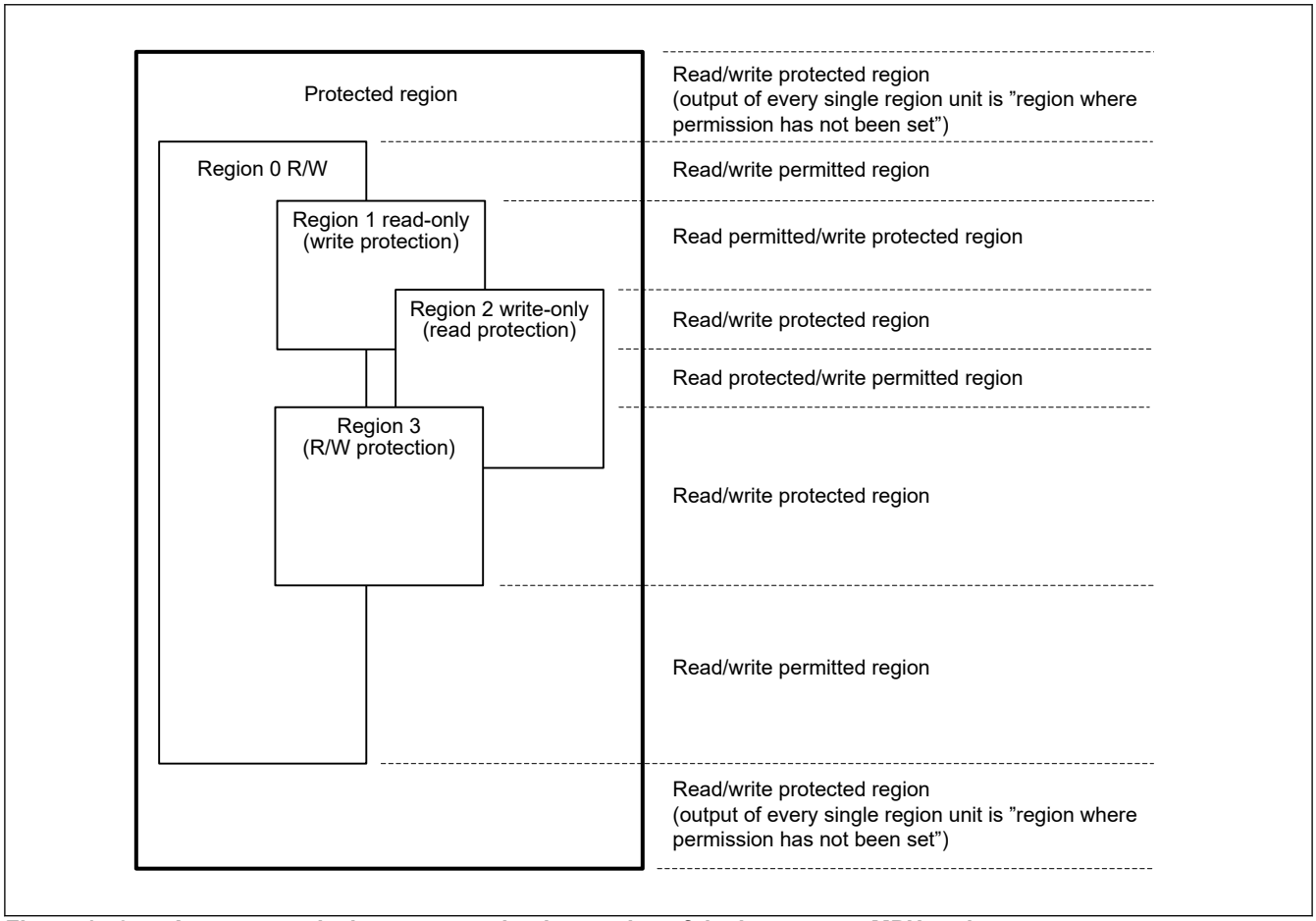


Figure 14.2 Access permission or protection by overlap of the bus master MPU regions

Figure 14.3 shows the register setting flow after reset. During this register setting, stop all bus masters except the CPU.

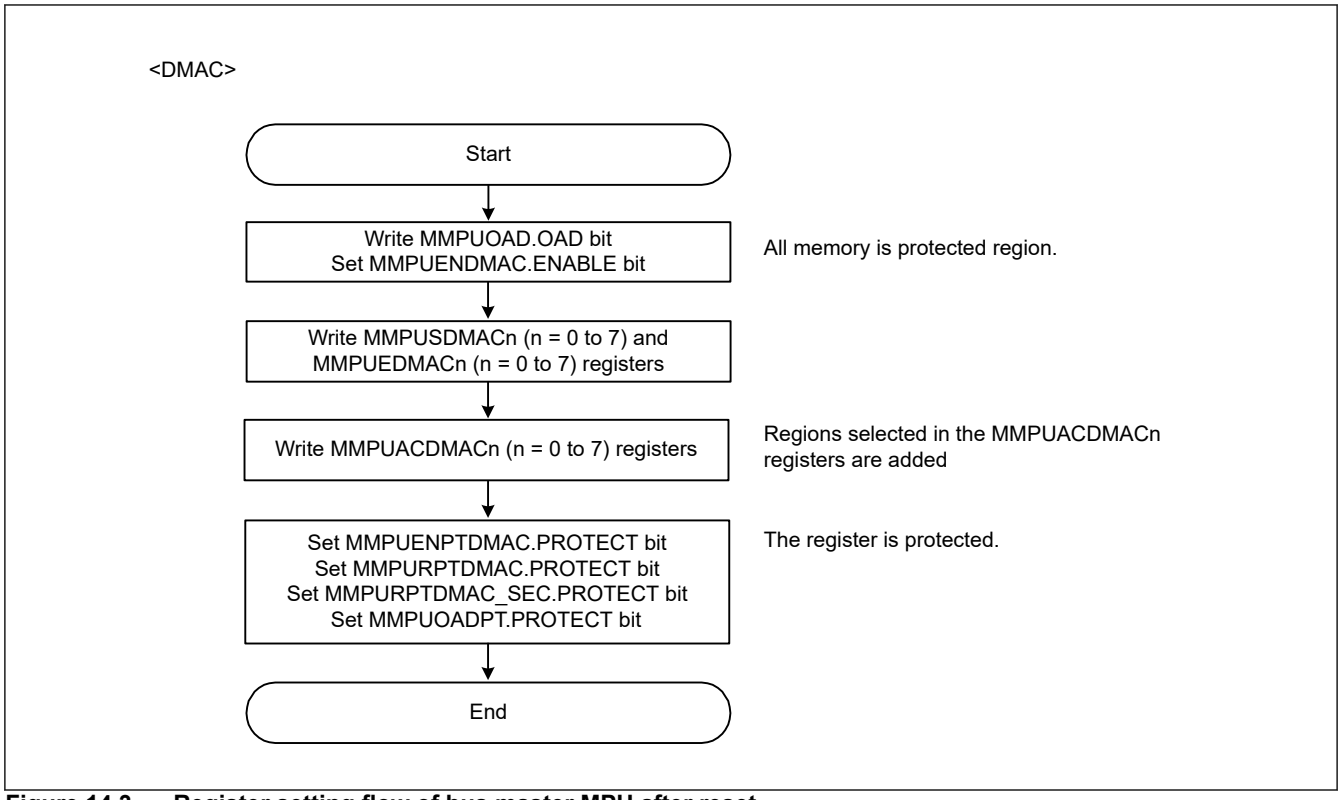


Figure 14.3 Register setting flow of bus master MPU after reset

Figure 14.4 shows the register setting flow for adding regions. During this register setting, stop all masters except the CPU.

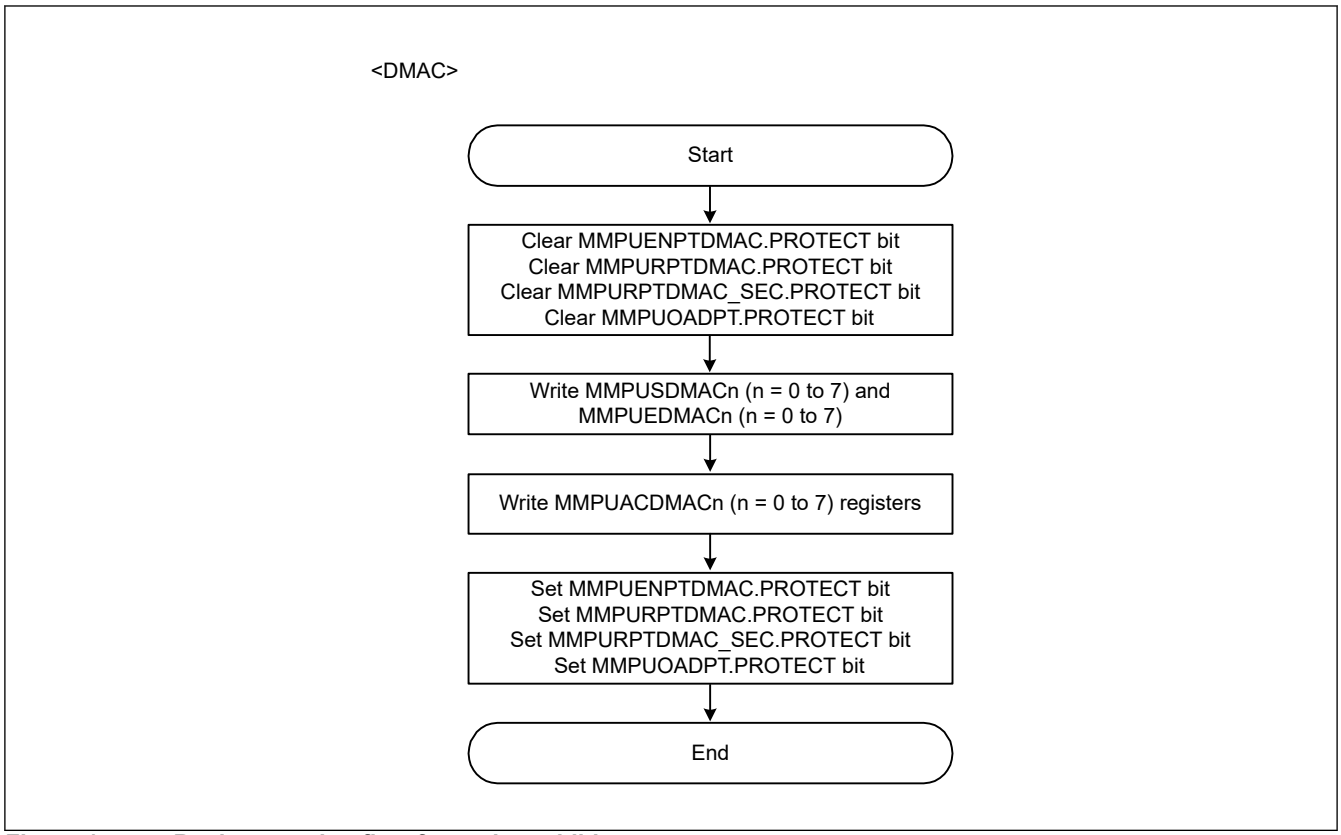


Figure 14.4 Register setting flow for region addition

14.3.2.2 Protecting the registers

Registers related to the Bus Master MPU can be protected with the PROTECT bit in the MMPUENPTDMAC, MMPURPTDMAC, MMPURPTDMAC_SEC, and MMPUOADPT registers.

Table 14.6 PROTECT bit and Protected target registers

PROTECT bit	Protect target registers
MMPUENPTDMAC.PROTECT	MMPUENDMAC
MMPURPTDMAC.PROTECT	The following registers set to Non-Secure by MMPUSARA.MMPUASAn (n = 0 to 7). MMPUSDMACn (n = 0 to 7) MMPUEDMACn (n = 0 to 7) MMPUACDMACn (n = 0 to 7)
MMPURPTDMAC_SEC.PROTECT	The following registers set to Secure by MMPUSARA.MMPUASAn (n = 0 to 7). MMPUSDMACn (n = 0 to 7) MMPUEDMACn (n = 0 to 7) MMPUACDMACn (n = 0 to 7)
MMPUOADPT.PROTECT	MMPUOAD

14.3.2.3 Memory protection error

If access to a protected region is detected, the bus master MPU generates an error. Set the OAD bit to select whether the error is reported as a non-maskable interrupt or a reset.

The non-maskable interrupt status is indicated in ICU.NMISR.BUSMST. For details, see section 12, Interrupt Controller Unit (ICU). The reset status is indicated in SYSTEM.RSTSR1.BUSMRF. For details, see section 5, Resets.

14.4 References

1. *Arm®v8-M Architecture Reference Manual* (ARM DDI0553B.g)
2. *Arm® Cortex®-M33 Processor Technical Reference Manual* (ARM 100230_0004_00_en)

15. DMA Controller (DMAC)

15.1 Overview

The MCU includes an 8-channel direct memory access controller (DMAC) that can transfer data without intervention from the CPU. When a DMA transfer request is generated, the DMAC transfers data stored at the transfer source address to the transfer destination address.

Table 15.1 lists the DMAC specifications, and Figure 15.1 shows a block diagram of the DMAC.

Table 15.1 DMAC specifications (1 of 2)

Item		Description
Number of channels		8 channels (DMACn (n = 0 to 7))
Transfer space		4 GB (0x00000000 to 0xFFFFFFFF excluding reserved areas)
Maximum transfer volume		64 M data (Maximum number of transfers in block transfer mode: 1,024 data/block × 65,536 blocks)
DMAC activation source		Selectable for each channel: • Software trigger • Interrupt requests from peripheral modules or trigger from external interrupt input pins.*1
Channel priority		Channel 0 > Channel 1 > Channel 2 > Channel 3… > Channel 7 (Channel 0: Highest)
Transfer data	Single data	Bit length: 8, 16, 32 bits
	Block size	Number of data: 1 to 1,024
Transfer mode	Normal transfer mode	• One data transfer by one DMA transfer request • Free-running function (setting in which total number of data transfers is not specified) settable
	Repeat transfer mode	• One data transfer by one DMA transfer request • Program returns to the transfer start address on completion of the repeat size of data transfer specified for the transfer source or destination. • Maximum settable repeat size: 1,024 • Selectable free-running function
	Repeat-block transfer mode	• One block data transfer by one DMA transfer request • Maximum settable block size: 1,024 • Block transfer can be repeated • Maximum settable repeat size: 64K • Selectable free-running function
	Block transfer mode	• One block data transfer by one DMA transfer request • Maximum settable block size: 1,024 data • Selectable free-running function
Selective functions	Extended repeat area function	• Function in which data can be transferred by repeating the address values in the specified range with the upper bit values in the transfer address register fixed. • Area of 2 bytes to 128 Mbytes separately settable as extended repeat area for transfer source and destination.
Processing on DMAC transfer error		• When a DMAC transfer error occurs, the transfer on the channel that caused the error is stopped. • A request to clear the register for activation request of DMAC error channel is sent to ICU.
Interrupt (DMACn_INT)	Transfer end interrupt	Generated on completion of transferring data volume specified by the transfer counter.
	Transfer escape end interrupt	• Generated when the repeat size of data transfer is completed. • Generated when the source address extended repeat area overflows. • Generated when the destination address extended repeat area overflows.
Interrupt (DMA_TRANSERR)	Error response detection interrupt	Generated when the DMAC transfer error occurs.
Event link activation (DMACn_INT)		An event link request is generated after each data transfer (for block transfer, after each block is transferred).

Table 15.1 DMAC specifications (2 of 2)

Item	Description
Master TrustZone Filter	TrustZone violation area of Flash and SRAM is detected before a non-secure channel access the bus.
Power consumption reduction function	Module-stop state can be set.
TrustZone Filter	Security attribution can be set for each channels

Note: Security attribution Register of DMAC channel is described in ICU.ICUSARC
Note 1. For details on DMAC activation sources, see Table 12.4 in section 12, Interrupt Controller Unit (ICU).

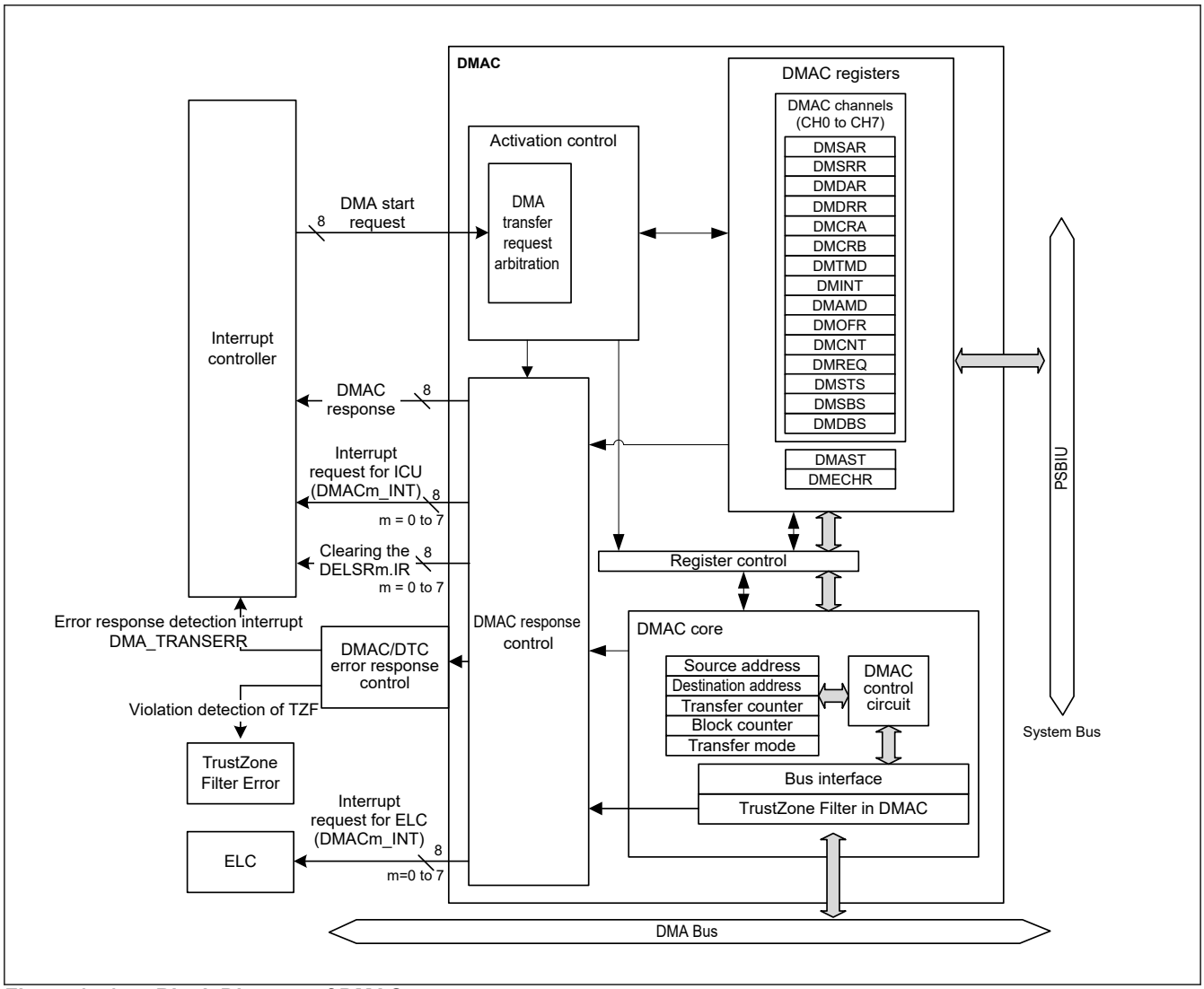


Figure 15.1 Block Diagram of DMAC

15.2 Register Descriptions

15.2.1 DMACSAR : DMAC Controller Security Attribution Register

Base address: CPSCU = 0x4000_8000

Offset address: 0x34

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

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

Bit	Symbol	Function	R/W
0	DMASTSA	DMAST Security Attribution 0: Secure 1: Non-secure	R/W
31:1	—	These bits are read as 1.	R/W

Note: Only secure access can write to this register. Both secure access and Non-secure read access are allowed but Non-secure write access is not allowed and TrustZone access error is not generated.

Note: This register is write-protected by PRCR register.

For DMAC, security attribution is set for each channel. However, this register only sets the DMAST register security attribute. The security attribution setting of each channel is described in the section 12.2.3. ICUSARC : Interrupt Controller Unit Security Attribution Register C.

DMASTSA bit (DMAST Security Attribution)

Security attributes of registers for DMAST. Do not write to DMASTSA bit while DMA transfer is enabled or a bus master is writing to the DMA registers.

15.2.2 DMSAR : DMA Source Address Register

Base address: DMACn = 0x4000_5000 + 0x0040 × n (n = 0 to 7)

Offset address: 0x00

Bit position:	31	0
Bit field:		
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
31:0	n/a	Specifies the transfer source start address Setting range is 0x0000_0000 to 0xFFFF_FFFF (4 Gbytes).	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

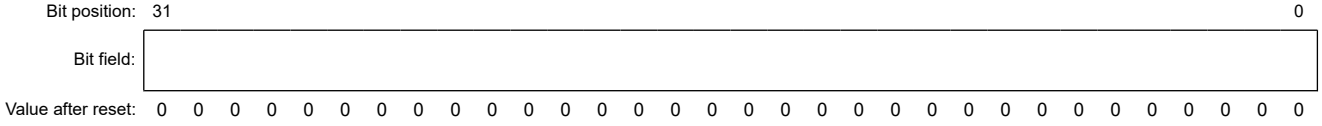
Set DMSAR while DMAC activation is disabled (DMAST.DMST = 0) or DMA transfer is disabled (DMCNT.DTE = 0).

Note: Address alignment in this register must match the Transfer Data Size value selected in the DMTMD.SZ bits.

15.2.3 DMSRR : DMA Source Reload Address Register

Base address: DMACn = 0x4000_5000 + 0x0040 × n (n = 0 to 7)

Offset address: 0x20



Bit	Symbol	Function	R/W
31:0	n/a	Specifies the transfer source reload address Setting range is 0x0000_0000 to 0xFFFF_FFFF (4 Gbytes).	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Set DMSRR while DMAC activation is disabled (DMAST.DMST = 0) or DMA transfer is disabled (DMCNT.DTE = 0).

DMSRR is the initial value of DMSAR. In repeat-block transfer mode, DMSAR reloads the value of DMSRR after the specified transfer is finished.

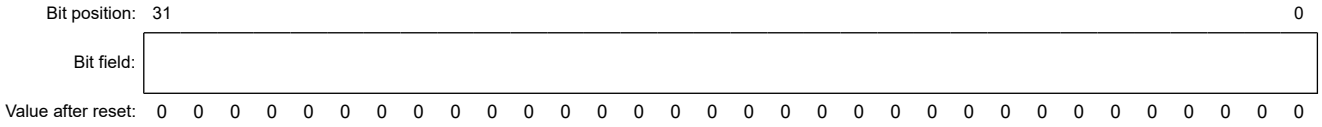
In normal transfer mode, repeat transfer mode, and block transfer mode DMSRR is not used. The setting is invalid.

Note: Address alignment in this register must match the Transfer Data Size value selected in the DMTMD.SZ bits.

15.2.4 DMDAR : DMA Destination Address Register

Base address: DMACn = 0x4000_5000 + 0x0040 × n (n = 0 to 7)

Offset address: 0x04



Bit	Symbol	Function	R/W
31:0	n/a	Specifies the transfer destination start address Setting range is 0x0000_0000 to 0xFFFF_FFFF (4 Gbytes).	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

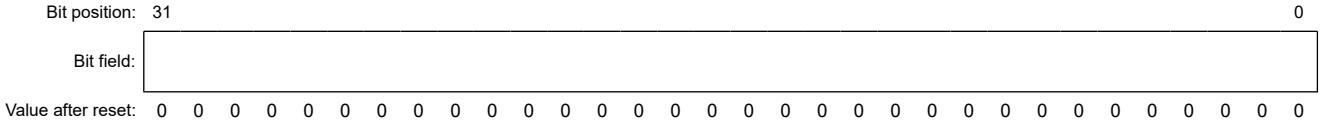
Set DMDAR while DMAC activation is disabled (DMAST.DMST = 0) or DMA transfer is disabled (DMCNT.DTE = 0).

Note: Address alignment in this register must match the Transfer Data Size value selected in the DMTMD.SZ bits.

15.2.5 DMDRR : DMA Destination Reload Address Register

Base address: DMACn = 0x4000_5000 + 0x0040 × n (n = 0 to 7)

Offset address: 0x24



Bit	Symbol	Function	R/W
31:0	n/a	Specifies the transfer destination reload address Setting range is 0x0000_0000 to 0xFFFF_FFFF (4 Gbytes).	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Set DMDRR while DMAC activation is disabled (DMAST.DMST = 0) or DMA transfer is disabled (DMCNT.DTE = 0).

DMDRR is the initial value of DMDAR. In repeat-block transfer mode, DMDAR reloads the value of DMDRR after the specified transfer is finished.

In normal transfer mode, repeat transfer mode and block transfer mode, DMDRR is not used. The setting is invalid.

Note: Address alignment in this register must match the Transfer Data Size value selected in the DMTMD.SZ bits.

15.2.6 DMCRA : DMA Transfer Count Register

Base address: DMACn = 0x4000_5000 + 0x0040 × n (n = 0 to 7)

Offset address: 0x08

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	DMCRAH[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
15:0	DMCRAL[15:0]	Lower bits of transfer count Specifies the number of transfer operations.	R/W
25:16	DMCRAH[9:0]	Upper bits of transfer count Specifies the number of transfer operations.	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Set the same value for DMCRAH and DMCRAL in repeat transfer mode, block transfer mode, and repeat-block transfer mode. Bits 15 to 10 are fixed to 0 in repeat transfer mode, block transfer mode, and repeat-block transfer mode.

(1) Normal Transfer Mode (DMTMD.MD[1:0] = 00b)

DMCRAL functions as a 16-bit transfer counter.

The number of transfer operations is one when the setting is 0x0001, and 65,535 when it is 0xFFFF. The value is decremented by one each time data is transferred.

When the setting is 0x0000, no specific number of transfer operations is set; data transfer is performed with the transfer counter stopped (free running function).

Free running function is not selected by DMTMD.TKP bit in normal transfer mode.

DMCRAH is not used in normal transfer mode. Write 0x0000 to DMCRAH.

(2) Repeat Transfer Mode (DMTMD.MD[1:0] = 01b)

DMCRAH specifies the repeat size and DMCRAL functions as a 10-bit transfer counter.

The number of transfer operations is one when the setting is 0x001, 1023 when it is 0x3FF, and 1024 when it is 0x000. In repeat transfer mode, a value in the range of 0x000 to 0x3FF (1 to 1024) can be set for DMCRAH and DMCRAL.

Setting bits 15 to 10 in DMCRAL is invalid. Write 0 to these bits.

The value in DMCRAL is decremented by one each time data is transferred until it reaches 0x000, at which the value in DMCRAH is loaded into DMCRAL.

(3) Block Transfer Mode (DMTMD.MD[1:0] = 10b)

DMCRAH specifies the block size and DMCRAL functions as a 10-bit block size counter.

The block size is one when the setting is 0x001, 1023 when it is 0x3FF, and 1024 when it is 0x000. In block transfer mode, a value in the range of 0x000 to 0x3FF can be set for DMCRAH and DMCRAL.

Setting bits 15 to 10 in DMCRAL is invalid. Write 0 to these bits.

The value in DMCRAL is decremented by one each time data is transferred until it reaches 0x000, at which the value in DMCRAH is loaded into DMCRAL.

(4) Repeat-Block Transfer Mode (DMTMD.MD[1:0] = 11b)

DMCRAH specifies the block size and DMCRAL functions as a 10-bit block size counter.

The block size is one when the setting is 0x001, 1023 when it is 0x3FF, and 1024 when it is 0x000. In repeat-block transfer mode, a value in the range of 0x000 to 0x3FF can be set for DMCRAH and DMCRAL.

Setting bits 15 to 10 in DMCRAL is invalid. Write 0 to these bits.

The value in DMCRAL is decremented by one each time data is transferred until it reaches 0x000, at which the value in DMCRAH is loaded into DMCRAL.

15.2.7 DMCRB : DMA Block Transfer Count Register

Base address: DMACn = 0x4000_5000 + 0x0040 × n (n = 0 to 7)

Offset address: 0x0C

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

Bit	Symbol	Function	R/W
15:0	DMCRBL[15:0]	Functions as a number of block, repeat or repeat-block transfer counter. 0x0001 to 0xFFFF (1 to 65535) 0x0000 (65536)	R/W
31:16	DMCRBH[15:0]	Specifies the number of block, repeat or repeat-block transfer operations. 0x0001 to 0xFFFF (1 to 65535) 0x0000 (65536)	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Set the same value for DMCRBH and DMCRBL in repeat transfer mode, block transfer mode and repeat-block transfer mode.

DMCRBH specifies the number of block, repeat and repeat-block transfer operations, and DMCRBL functions as a 16-bit the number of block counter in block, repeat, and repeat-block transfer mode, respectively.

The number of transfer operations is one when the setting is 0x0001, 65535 when it is 0xFFFF, and 65536 when it is 0x0000.

In repeat transfer mode, the value is decremented by one when the final data of one repeat size is transferred.

In block transfer mode and repeat-block transfer mode, the value is decremented by one when the final data of one block size is transferred.

In normal transfer mode, DMCRB is not used. The setting is invalid.

When DMTMD.TKP is 1 and the final data of one repeat size or one block size is transferred, DMCRBL reloads the value of DMCRBH automatically.

15.2.8 DMTMD : DMA Transfer Mode Register

Base address: DMACn = 0x4000_5000 + 0x0040 × n (n = 0 to 7)

Offset address: 0x10

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

Bit	Symbol	Function	R/W
1:0	DCTG[1:0]	Transfer Request Source Select 0 0: Software request 0 1: Hardware request*1 1 0: Setting prohibited 1 1: Setting prohibited	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
9:8	SZ[1:0]	Transfer Data Size Select 0 0: 8 bits 0 1: 16 bits 1 0: 32 bits 1 1: Setting prohibited	R/W
10	TKP	Transfer Keeping 0: Transfer is stopped by completion of specified total number of transfer operations. 1: Transfer is not stopped by completion of specified total number of transfer operations (free-running).	R/W
11	—	This bit is read as 0. The write value should be 0.	R/W
13:12	DTS[1:0]	Repeat Area Select 0 0: The destination is specified as the repeat area or block area. 0 1: The source is specified as the repeat area or block area. 1 0: The repeat area or block area is not specified. 1 1: Setting prohibited.	R/W
15:14	MD[1:0]	Transfer Mode Select 0 0: Normal transfer 0 1: Repeat transfer 1 0: Block transfer 1 1: Repeat-block transfer	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note 1. To select the DMAC activation source, use the DELSRn registers of the ICU. For details on DMAC activation sources, see Table 12.4 in section 12, Interrupt Controller Unit (ICU).

DTS[1:0] bits (Repeat Area Select)

DTS[1:0] select either the source or destination as the repeat area in repeat or block transfer mode. In normal or repeat-block transfer mode, setting these bits is invalid.

TKP bit (Transfer Keeping)

TKP selects either stopping transfer or keeping transfer by completion of specified total number of transfer operations in repeat, block or repeat-block transfer mode. In normal transfer mode, setting this bit is invalid.

15.2.9 DMINT : DMA Interrupt Setting Register

Base address: DMACn = 0x4000_5000 + 0x0040 × n (n = 0 to 7)

Offset address: 0x13

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	DTIE	ESIE	RPTIE	SARIE	DARIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DARIE	Destination Address Extended Repeat Area Overflow Interrupt Enable 0: Disables an interrupt request for an extended repeat area overflow on the destination address. 1: Enables an interrupt request for an extended repeat area overflow on the destination address.	R/W
1	SARIE	Source Address Extended Repeat Area Overflow Interrupt Enable 0: Disables an interrupt request for an extended repeat area overflow on the source address. 1: Enables an interrupt request for an extended repeat area overflow on the source address.	R/W
2	RPTIE	Repeat Size End Interrupt Enable 0: Disables the repeat size end interrupt request. 1: Enables the repeat size end interrupt request.	R/W
3	ESIE	Transfer Escape End Interrupt Enable 0: Disables the transfer escape end interrupt request. 1: Enables the transfer escape end interrupt request.	R/W
4	DTIE	Transfer End Interrupt Enable 0: Disables the transfer end interrupt request. 1: Enables the transfer end interrupt request.	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

DARIE bit (Destination Address Extended Repeat Area Overflow Interrupt Enable)

When an extended repeat area overflow on the destination address occurs while DARIE bit is set to 1, the DMCNT.DTE bit is cleared to 0. At the same time, the DMSTS.ESIF flag is set to 1 to indicate that an interrupt by an extended repeat area overflow on the destination address is requested.

When block transfer mode is used with the extended repeat area function, an interrupt is requested after completion of a 1-block size transfer. When setting 1 in the DMCNT.DTE bit of the channel for which a transfer has been stopped, the transfer is resumed from the state when the transfer is stopped.

When the extended repeat area is not specified for the destination address, this bit is ignored.

When set to repeat-block transfer mode, do not use this bit.

SARIE bit (Source Address Extended Repeat Area Overflow Interrupt Enable)

When an extended repeat area overflow on the source address occurs while SARIE bit is set to 1, the DMCNT.DTE bit is cleared to 0. At the same time, the DMSTS.ESIF flag is set to 1 to indicate that an interrupt by an extended repeat area overflow on the source address is requested.

When block transfer mode is used with the extended repeat area function, an interrupt is requested after completion of a 1-block size transfer. When setting 1 in the DMCNT.DTE bit of the channel for which a transfer has been stopped, the transfer is resumed from the state when the transfer is stopped.

When the extended repeat area is not specified for the source address, this bit is ignored.

When set to repeat-block transfer mode, do not use this bit.

RPTIE bit (Repeat Size End Interrupt Enable)

When RPTIE bit is set to 1 in repeat transfer mode, the DMCNT.DTE bit is cleared to 0 after completion of a 1-repeat size data transfer. At the same time, the DMSTS.ESIF flag is set to 1 to indicate that the repeat size end interrupt request has been generated. The repeat size end interrupt request can be generated even when the DMTMD.DTS[1:0] bits are 10b (= repeat area or block area is not specified).

When this bit is set to 1 in block transfer mode, the DMCNT.DTE bit is cleared to 0 after completion of a 1-block data transfer in the same way as repeat transfer mode. At the same time, the DMSTS.ESIF flag is set to 1 to indicate that the repeat size end interrupt request has been generated. The repeat size end interrupt request can be generated even when the DMTMD.DTS[1:0] bits are 10b (= repeat area or block area is not specified).

When set to repeat-block transfer mode, do not use this bit.

ESIE bit (Transfer Escape End Interrupt Enable)

ESIE bit enables or disables the transfer escape end interrupt requests (repeat size end interrupt request and extended repeat area overflow interrupt request) that are generated during DMA transfer.

The transfer escape end interrupt is generated when the DMSTS.ESIF flag is set to 1 with this bit set to 1. The transfer escape end interrupt is cleared by clearing this bit or the DMSTS.ESIF flag to 0.

DTIE bit (Transfer End Interrupt Enable)

DTIE bit enables or disables the transfer end interrupt request to be generated on completion of a specified number of data transfers.

The transfer end interrupt is generated when the DMSTS.DTIF flag is set to 1 with this bit set to 1. The transfer end interrupt is cleared by clearing this bit or the DMSTS.DTIF flag to 0.

15.2.10 DMAMD : DMA Address Mode Register

Base address: DMACn = 0x4000_5000 + 0x0040 × n (n = 0 to 7)

Offset address: 0x14

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SM[1:0]		SADR	SARA[4:0]					DM[1:0]		DADR	DARA[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	DARA[4:0]	Destination Address Extended Repeat Area Specifies the extended repeat area on the destination address. For details on the settings, see Table 15.2.	R/W
5	DADR	Destination Address Update Select After Reload 0: Only reloading. 1: Add index after reloading.	R/W
7:6	DM[1:0]	Destination Address Update Mode 0 0: Destination address is fixed. 0 1: Offset addition. 1 0: Destination address is incremented. 1 1: Destination address is decremented.	R/W
12:8	SARA[4:0]	Source Address Extended Repeat Area Specifies the extended repeat area on the source address. For details on the settings, see Table 15.2.	R/W

Bit	Symbol	Function	R/W
13	SADR	Source Address Update Select After Reload 0: Only reloading. 1: Add index after reloading.	R/W
15:14	SM[1:0]	Source Address Update Mode 0 0: Source address is fixed. 0 1: Offset addition. 1 0: Source address is incremented. 1 1: Source address is decremented.	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

DARA[4:0] bits (Destination Address Extended Repeat Area)

DARA[4:0] bits specify the extended repeat area on the destination address. The extended repeat area function is realized by updating the specified lower address bits with the remaining upper address bits fixed. The size of the extended repeat area can be any power of two between 2 bytes and 128 Mbytes.

When the lower address overflows the extended repeat area by address increment, the start address of the extended repeat area is set. Similarly, when the lower address underflows the extended repeat area by address decrement, the end address of the extended repeat area is set.

When the repeat area or block area is specified as a transfer destination, do not specify the extended repeat area on the destination address. When repeat transfer or block transfer is selected, and when DMTMD.DTS[1:0] = 00b (the transfer destination is specified as the repeat area or block area), write 00000b in the DARA[4:0] bits.

In repeat-block transfer mode, write 00000b in the DARA[4:0] bits.

An interrupt can be requested when an overflow or underflow occurs in the extended repeat area with the DMINT.DARIE bit set to 1. Table 15.2 lists the settings and the corresponding extended repeat areas.

DADR bits (Destination Address Update Select After Reload)

In repeat-block transfer mode, this bit specifies the behavior of DMDAR after reloading DMDRR.

When this bit is set to 1, an index value ((DMDBSH-DMDBSL) × DataSize) is added to DMDAR after reloading DMDRR.

When this bit is set to 0, DMDAR only reloads DMDRR. This behavior is described in Table 15.13.

In normal, repeat or block transfer mode, this bit is ignored.

DM[1:0] bits (Destination Address Update Mode)

DM[1:0] bits select the mode of updating the destination address.

When increment is selected and the DMTMD.SZ[1:0] bits are set to 00b, 01b, or 10b, the destination address is incremented by 1, 2, or 4, respectively.

When decrement is selected and the DMTMD.SZ[1:0] bits are set to 00b, 01b, or 10b, the destination address is decremented by 1, 2, or 4, respectively.

When offset addition is selected, the offset specified by the DMOFR register is added to the address.

SARA[4:0] bits (Source Address Extended Repeat Area)

SARA[4:0] bits specify the extended repeat area on the source address. The extended repeat area function is realized by updating the specified lower address bits with the remaining upper address bits fixed. The size of the extended repeat area can be any power of two between 2 bytes and 128 Mbytes.

When the lower address overflows the extended repeat area by address increment, the start address of the extended repeat area is set. Similarly, when the lower address underflows the extended repeat area by address decrement, the end address of the extended repeat area is set.

When the repeat area or block area is specified as a transfer source, do not specify the extended repeat area on the source address. When repeat transfer or block transfer is selected, and when DMTMD.DTS[1:0] = 01b (the transfer source is specified as the repeat area or block area), write 00000b in the SARA[4:0] bits.

In repeat-block transfer mode, write 00000b in the SARA[4:0] bits.

An interrupt can be requested when an overflow or underflow occurs in the extended repeat area with the DMINT.SARIE bit set to 1. Table 15.2 lists the settings and the corresponding extended repeat areas.

SADR bits (Source Address Update Select After Reload)

In repeat-block transfer mode, this bit specifies the behavior of DMSAR after reloading DMSRR.

When this bit is set to 1, an index value ((DMSBSH-DMSBSL) × DataSize) is added to DMSAR after reloading DMSRR.

When this bit is set to 0, DMSAR only reloads DMSRR. This behavior is described in Table 15.12.

In normal, repeat or block transfer mode, this bit is ignored.

SM[1:0] bits (Source Address Update Mode)

SM[1:0] bits select the mode of updating the source address.

When increment is selected and the DMTMD.SZ[1:0] bits are set to 00b, 01b, or 10b, the source address is incremented by 1, 2, or 4, respectively.

When decrement is selected and the DMTMD.SZ[1:0] bits are set to 00b, 01b, or 10b, the source address is decremented by 1, 2, or 4, respectively.

When offset addition is selected, the offset specified by the DMOFR register is added to the address.

Table 15.2 SARA[4:0] or DARA[4:0] settings and corresponding repeat areas (1 of 2)

SARA[4:0] or DARA[4:0] settings	Extended repeat area
00000b	Not specified
00001b	2 bytes specified as extended repeat area by the lower 1 bit of the address
00010b	4 bytes specified as extended repeat area by the lower 2 bits of the address
00011b	8 bytes specified as extended repeat area by the lower 3 bits of the address
00100b	16 bytes specified as extended repeat area by the lower 4 bits of the address
00101b	32 bytes specified as extended repeat area by the lower 5 bits of the address
00110b	64 bytes specified as extended repeat area by the lower 6 bits of the address
00111b	128 bytes specified as extended repeat area by the lower 7 bits of the address
01000b	256 bytes specified as extended repeat area by the lower 8 bits of the address
01001b	512 bytes specified as extended repeat area by the lower 9 bits of the address
01010b	1 Kbyte specified as extended repeat area by the lower 10 bits of the address
01011b	2 Kbytes specified as extended repeat area by the lower 11 bits of the address
01100b	4 Kbytes specified as extended repeat area by the lower 12 bits of the address
01101b	8 Kbytes specified as extended repeat area by the lower 13 bits of the address
01110b	16 Kbytes specified as extended repeat area by the lower 14 bits of the address
01111b	32 Kbytes specified as extended repeat area by the lower 15 bits of the address
10000b	64 Kbytes specified as extended repeat area by the lower 16 bits of the address
10001b	128 Kbytes specified as extended repeat area by the lower 17 bits of the address
10010b	256 Kbytes specified as extended repeat area by the lower 18 bits of the address
10011b	512 Kbytes specified as extended repeat area by the lower 19 bits of the address
10100b	1 Mbyte specified as extended repeat area by the lower 20 bits of the address
10101b	2 Mbytes specified as extended repeat area by the lower 21 bits of the address
10110b	4 Mbytes specified as extended repeat area by the lower 22 bits of the address
10111b	8 Mbytes specified as extended repeat area by the lower 23 bits of the address
11000b	16 Mbytes specified as extended repeat area by the lower 24 bits of the address
11001b	32 Mbytes specified as extended repeat area by the lower 25 bits of the address

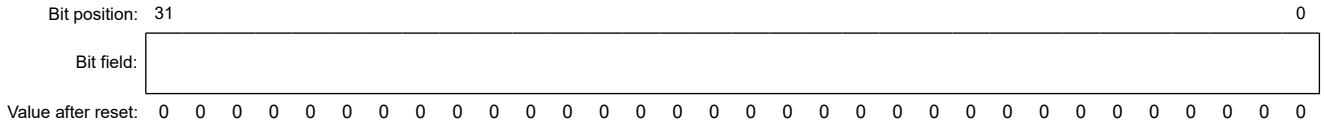
Table 15.2 SARA[4:0] or DARA[4:0] settings and corresponding repeat areas (2 of 2)

SARA[4:0] or DARA[4:0] settings	Extended repeat area
11010b	64 Mbytes specified as extended repeat area by the lower 26 bits of the address
11011b	128 Mbytes specified as extended repeat area by the lower 27 bits of the address
11100b to 11111b	Setting prohibited.

15.2.11 DMOFR : DMA Offset Register

Base address: DMACn = 0x4000_5000 + 0x0040 × n (n = 0 to 7)

Offset address: 0x18



Bit	Symbol	Function	R/W
31:0	n/a	Specifies the offset when offset addition is selected as the address update mode for transfer source or destination. 0x00000000 to 0x00FFFFFF (0 bytes to (16 M – 1) bytes) 0xFF000000 to 0xFFFFFFFF (–16 Mbytes to –1 byte)	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Write to this register while the DMAC operation is stopped or DMA transfer is disabled (not during data transfer).

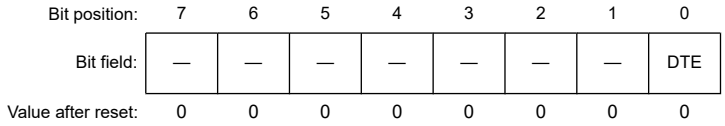
Setting bits 31 to 25 is invalid; a value of bit 24 is extended to bits 31 to 25. Reading DMOFR returns the extended value.

In repeat-block transfer mode, the offset is not specified by DMOFR when offset addition is selected, write 0 to DMOFR.

15.2.12 DMCNT : DMA Transfer Enable Register

Base address: DMACn = 0x4000_5000 + 0x0040 × n (n = 0 to 7)

Offset address: 0x1C



Bit	Symbol	Function	R/W
0	DTE	DMA Transfer Enable 0: Disables DMA transfer. 1: Enables DMA transfer.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

DTE bit (DMA Transfer Enable)

When the DMAST.DMST bit is set to 1 (DMAC activation is enabled) and this bit is set to 1 (DMA transfer is enabled), DMA transfer can be started for the corresponding channel.

[Setting condition]

- When 1 is written to this bit.

[Clearing conditions]

- When 0 is written to this bit.
- When the specified total volume of data transfer is completed.
- When DMA transfer is stopped by the repeat size end interrupt.
- When DMA transfer is stopped by the extended repeat area overflow interrupt.
- When DMA transfer is stopped by the access error occurs. See section 15.5. Processing on DMA Transfer Error.

15.2.13 DMREQ : DMA Software Start Register

Base address: DMACn = 0x4000_5000 + 0x0040 × n (n = 0 to 7)

Offset address: 0x1D

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

Bit	Symbol	Function	R/W
0	SWREQ	DMA Software Start 0: DMA transfer is not requested. 1: DMA transfer is requested.	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
4	CLRS	DMA Software Start Bit Auto Clear Select 0: SWREQ bit is cleared after DMA transfer is started by software. 1: SWREQ bit is not cleared after DMA transfer is started by software.	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

SWREQ bit (DMA Software Start)

When 1 is written to SWREQ bit, a DMA transfer request is generated. After DMA transfer is started in response to the request, this bit is cleared to 0 if the CLRS bit is set to 0. This bit is not cleared to 0 while the CLRS bit is set to 1. In this case, a DMA transfer request can be issued again after completion of a transfer.

Note that, however, setting this bit is valid and DMA transfer by software is enabled only when the DMTMD.DCTG[1:0] bits are set to 00b (DMAC activation source is software).

Setting this bit is invalid when the DMTMD.DCTG[1:0] bits are set to a value other than 00b.

To start DMA transfer by software with the CLRS bit being 0, ensure that the SWREQ bit is 0, and then write 1 to the SWREQ bit.

[Setting condition]

- When 1 is written to this bit.

[Clearing conditions]

- When a DMA transfer request by software is accepted and DMA transfer is started while the CLRS bit is set to 0 (the SWREQ bit is cleared after DMA transfer is started by software).
- When 0 is written to this bit.

CLRS bit (DMA Software Start Bit Auto Clear Select)

CLRS bit specifies whether to clear the SWREQ bit to 0 after DMA transfer is started in response to the DMA transfer request generated by setting the SWREQ bit to 1. With this bit set to 0, the SWREQ bit is cleared to 0 after DMA transfer is started. With this bit set to 1, the SWREQ bit is not cleared to 0. In this case, a DMA transfer request can be issued again after completion of a transfer.

15.2.14 DMSTS : DMA Status Register

Base address: DMACn = 0x4000_5000 + 0x0040 × n (n = 0 to 7)

Offset address: 0x1E

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

Bit	Symbol	Function	R/W
0	ESIF	Transfer Escape End Interrupt Flag 0: A transfer escape end interrupt has not been generated. 1: A transfer escape end interrupt has been generated.	R/W*1
3:1	—	These bits are read as 0. The write value should be 0.	R/W
4	DTIF	Transfer End Interrupt Flag 0: A transfer end interrupt has not been generated. 1: A transfer end interrupt has been generated.	R/W*1
6:5	—	These bits are read as 0. The write value should be 0.	R/W
7	ACT	DMAC Active Flag 0: DMAC is in the idle state. 1: DMAC is operating.	R

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note 1. Only 0 can be written to clear the flag.

ESIF flag (Transfer Escape End Interrupt Flag)

This flag indicates that the transfer escape end interrupt has been generated.

[Setting conditions]

- When 1-repeat size data transfer is completed in repeat transfer mode with the DMINT.RPTIE bit set to 1.
- When 1-block data transfer is completed in block transfer mode with the DMINT.RPTIE bit set to 1.
- When an extended repeat area overflow on the source address occurs while the DMINT.SARIE bit is set to 1 and the DMAMD.SARA[4:0] bits are set to a value other than 00000b (extended repeat area is specified on the transfer source address).
- When an extended repeat area overflow on the destination address occurs while the DMINT.DARIE bit is set to 1 and the DMAMD.DARA[4:0] bits are set to a value other than 00000b (extended repeat area is specified on the transfer destination address).

[Clearing conditions]

- When 0 is written to this bit.
- When 1 is written to the DMCNT.DTE bit.

DTIF flag (Transfer End Interrupt Flag)

This flag indicates that the transfer end interrupt has been generated.

[Setting conditions]

- When the specified number of unit-transfers are completed in normal transfer mode (the value of DMCRAL becoming 0 on completion of transfer).
- When the specified number of repeat transfer operations are completed in repeat transfer mode (the value of DMCRBL becoming 0 on completion of transfer with DMTMD.TKP = 0 or the value of DMCRBL reloading DMCRBH with DMTMD.TKP = 1).
- When the specified number of blocks have been transferred in block transfer mode and repeat-block transfer mode (the value of DMCRBL becoming 0 on completion of transfer with DMTMD.TKP = 0 or the value of DMCRBL reloading DMCRBH with DMTMD.TKP = 1).

[Clearing conditions]

- When 0 is written to this bit.
- When 1 is written to the DMCNT.DTE bit.

ACT flag (DMAC Active Flag)

This flag indicates whether the DMAC is in the idle or active state.

[Setting condition]

- When the DMAC starts data transfer operation.

[Clearing condition]

- When data transfer in response to one transfer request is completed.

15.2.15 DMSBS : DMA Source Buffer Size Register

Base address: DMACn = 0x4000_5000 + 0x0040 × n (n = 0 to 7)

Offset address: 0x28

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	DMSBSH[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
15:0	DMSBSL[15:0]	Functions as data transfer counter in repeat-block transfer mode See Table 15.3 for available settings.	R/W
31:16	DMSBSH[15:0]	Specifies the repeat-area size in repeat-block transfer mode See Table 15.3 for available settings.	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Set the same value for DMSBSH and DMSBSL in repeat-block transfer mode. Write 0x00000000 to DMSBS in normal, repeat and block transfer mode.

DMSBSH specifies buffer size and DMSBSL functions as a 16-bit buffer size counter in repeat-block transfer mode. In repeat-block transfer mode, source repeat area is specified by DMSBSH.

When address update mode is incremented address or decremented address, this register means the numbers of data of whole buffer. When address update mode is offset addition, this register means the numbers of data of an individual buffer. In offset addition, setting DMSBSH and DMSBSL to 0x0000 is prohibited. When final data of one buffer size is transferred, DMSBSL reloads value of DMSBSH. When address update mode is fixed address, this register is ignored. Table 15.3 shows

the setting values of DMA Source Buffer Size Register corresponding to Transfer Data Size in Source Address Update Mode.

Table 15.3 Available setting for DMSBS register in repeat-block transfer mode

Source address update mode (DMAMD.SM)	Transfer data size (DMTMD.SZ)	Available setting for DMSBSH and DMSBSL bits
Source address is fixed (SM = 00b)	Don't care	0x0000 (DMSBS is not used)
Offset addition (SM = 01b)	8 bits (SZ = 00b)	0x0001 to 0xFFFF (1 to 65535)
	16 bits (SZ = 01b)	0x0001 to 0x7FFF (1 to 32767)
	32 bits (SZ = 10b)	0x0001 to 0x3FFF (1 to 16383)
Source address is incremented or decremented (SM = 1xb)	Don't care	0x0000 (infinite) 0x0001 to 0xFFFF (1 to 65535)

In normal, repeat and block transfer mode, DMSBS is not used. The setting is invalid.

15.2.16 DMDBS : DMA Destination Buffer Size Register

Base address: DMACn = 0x4000_5000 + 0x0040 × n (n = 0 to 7)

Offset address: 0x2C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	DMDBSH[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
15:0	DMDBSL[15:0]	Functions as data transfer counter in repeat-block transfer mode. See Table 15.4 for available settings.	R/W
31:16	DMDBSH[15:0]	Specifies the repeat-area size in repeat-block transfer mode. See Table 15.4 for available settings.	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Set the same value for DMDBSH and DMDBSL in repeat-block transfer mode. Write 0x00000000 to DMDBS in normal, repeat and block transfer mode.

DMDBSH specifies buffer size and DMDBSL functions as a 16-bit buffer size counter in repeat-block transfer mode. In repeat-block transfer mode, destination repeat area is specified by DMDBSH.

When address update mode is incremented address or decremented address, this register means the numbers of data of whole buffer. When address update mode is offset addition, this register means the numbers of data of an individual buffer. In offset addition, setting DMDBSH and DMDBSL to 0x0000 is prohibited. When final data of one buffer size is transferred, DMDBSL reloads value of DMDBSH. When address update mode is fixed address, this register is ignored. Table 15.4 shows the setting values of Destination Buffer Size Register corresponding to Transfer Data Size in Destination Address Update Mode.

Table 15.4 Available setting for DMDBS register in repeat-block transfer mode (1 of 2)

Destination address update mode (DMAMD.DM)	Transfer data size (DMTMD.SZ)	Available setting for DMDBSH and DMDBSL bits
Destination address is fixed (DM = 00b)	Don't care	0x0000 (DMDBS is not used)

Table 15.4 Available setting for DMDBS register in repeat-block transfer mode (2 of 2)

Destination address update mode (DMAMD.DM)	Transfer data size (DMTMD.SZ)	Available setting for DMDBSH and DMDBSL bits
Offset addition (DM = 01b)	8 bits (SZ = 00b)	0x0001 to 0xFFFF (1 to 65535)
	16 bits (SZ = 01b)	0x0001 to 0x7FFF (1 to 32767)
	32 bits (SZ = 10b)	0x0001 to 0x3FFF (1 to 16383)
Destination address is incremented or decremented (DM = 1xb)	Don't care	0x0000 (infinite) 0x0001 to 0xFFFF (1 to 65535)

In normal, repeat and block transfer mode, DMDBS is not used. The setting is invalid.

15.2.17 DMAST : DMA Module Activation Register

Base address: DMA = 0x4000_5200

Offset address: 0x00

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

Bit	Symbol	Function	R/W
0	DMST	DMAC Operation Enable 0: DMAC activation is disabled. 1: DMAC activation is enabled.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

DMST bit (DMAC Operation Enable)

Setting the DMAST.DMST to 1 enables DMAC activation for all channels. When the DMST bit is set to 1 (DMAC activation is enabled), and 1 is written to the DMCNT.DTE bit (DMA transfer is enabled) for multiple channels, all associated channels can be placed in the transfer request ready state at the same time.

When the DMST bit clears to 0 during DMA transfer, DMA transfer is suspended after the current data transfer associated with a single transfer request completes. To resume DMA transfer, set the DMST bit to 1 again.

[Setting condition]

- When 1 is written to this bit.

[Clearing condition]

- When 0 is written to this bit.

15.2.18 DMECHR : DMAC Error Channel Register

Base address: DMA = 0x4000_5200

Offset address: 0x40

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

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

Bit	Symbol	Function	R/W
2:0	DMECH	DMAC Error channel Indicates the channel number causing the error 0 0 0: Error occurred on Channel 0 0 0 1: Error occurred on Channel 1 0 1 0: Error occurred on Channel 2 ⋮ ⋮ 1 1 1: Error occurred on Channel 7	R
7:3	—	These bits are read as 0. The write value should be 0.	R
8	DMECHSAM	DMAC Error channel Security Attribution Monitor Indicates the security attribution of a channel causing the error 0: secure channel 1: non-secure channel	R
15:9	—	These bits are read as 0. The write value should be 0.	R
16	DMESTA	DMAC Error Status 0: No DMA transfer error occurred 1: DMA transfer error occurred	R/W*1
31:17	—	These bits are read as 0. The write value should be 0.	R

Note 1. Writing to DMESTA depends on the value of DMECHSAM

DMECH[2:0] bits (DMAC Error channel)

When a transfer error due to DMA transfer occurs, the DMECH[2:0] bits store the violating DMAC channel.

When reset is selected in MPU.MMPUOAD.OAD and TZF.TZFOAD.OAD, this register is also reset. Select NMI when you want to debug the program.

[Set condition]

- When the DMAC transfer error occurs and DMESTA = 0.

[Clearing condition]

- When 1 is written to DMESTA.

DMECHSAM bit (DMAC Error channel Security Attribution Monitor)

When a transfer error due to DMA transfer occurs, the DMECHSAM bit indicates the security attribution of the violating DMAC channel.

When reset is selected in MPU.MMPUOAD.OAD and TZF.TZFOAD.OAD, this register is also reset. Select NMI when you want to debug the program.

[Set condition]

- When the DMAC transfer error occurs and DMESTA = 0.

[Clearing condition]

- When 1 is written to DMESTA.

DMESTA bit (DMAC Error Status)

The DMESTA bit indicates whether or not a DMA transfer error occurred.

DMECH, DMECHSAM, DMESTA are cleared by writing 1 to DMESTA. Writing 0 to DMESTA is ignored.

When reset is selected in MPU.MMPUOAD.OAD and TZF.TZFOAD.OAD, this register is also reset. Select NMI when you want to debug the program.

[Set condition]

- When the DMAC transfer error occurs.

[Clearing condition]

- When 1 is written to DMESTA.

Note: When DMECHSAM = 1, it can be cleared in the secure state and non-secure state. DMECHSAM = 0, it cannot be cleared in the non-secure state.

15.3 Operation

15.3.1 Transfer Mode

15.3.1.1 Normal Transfer Mode

In normal transfer mode, one data is transferred by one transfer request. A maximum of 65535 can be set as the number of transfer operations using the DMCRAL register. When these bits are set to 0x0000, no specific number of transfer operations is set; data transfer is performed with the transfer counter stopped (free-running function). Setting DMCRB register is invalid in normal transfer mode. Except in free-running function, a transfer end interrupt request can be generated after completion of the specified number of transfer operations.

Table 15.5 summarizes the register update operation in normal transfer mode, and Figure 15.2 shows the operation in normal transfer mode.

Table 15.5 Register update operation in normal transfer mode

Register	Function	Update operation after completion of a transfer by one transfer request
DMSAR	Transfer source address	Increment/decrement/fixed/offset addition
DMDAR	Transfer destination address	Increment/decrement/fixed/offset addition
DMCRAL	Transfer count	Decremented by one/not updated (in free running function)
DMCRAH	—	Not updated (Not used in normal transfer mode)
DMCRB	—	Not updated (Not used in normal transfer mode)

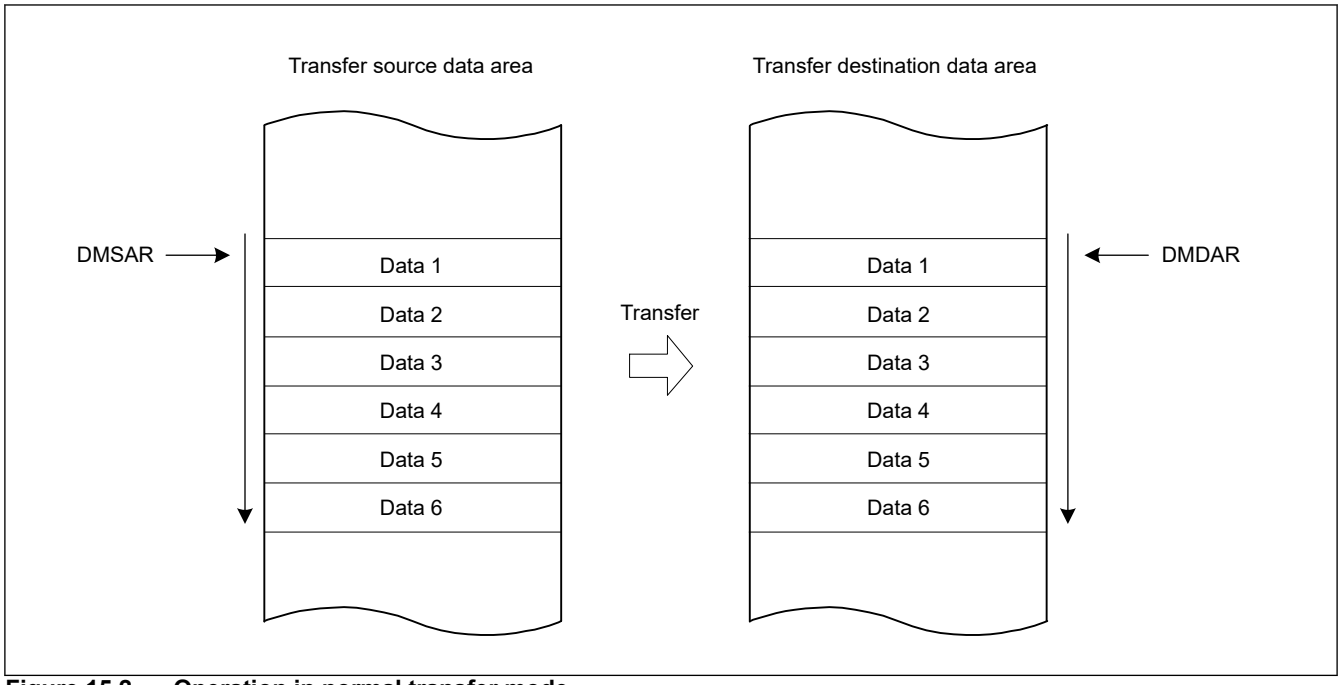


Figure 15.2 Operation in normal transfer mode

15.3.1.2 Repeat Transfer Mode

In repeat transfer mode, one data is transferred by one transfer request.

A maximum of 1K data can be set as a total repeat transfer size using DMCRA register.

A maximum of 64K can be set as the number of repeat transfer operations using DMCRB register; therefore, a maximum of 64M data (1K data × 64K counts of repeat transfer operations) can be set as a total data transfer size.

Either the transfer source or transfer destination can be specified as a repeat area. When transfer of the repeat size data is completed, the address of the specified repeat area (DMSAR or DMDAR) returns to the transfer start address. When data of the specified repeat size has all been transferred in repeat transfer mode, DMA transfer can be stopped, and the repeat size end interrupt can be requested. DMA transfer can be resumed by writing 1 to the DMCNT.DTE bit in the repeat size end interrupt handling.

A transfer end interrupt request can be generated after completion of the specified number of repeat transfer operations.

Table 15.6 summarizes the register update operation in repeat transfer mode, and Figure 15.3 shows the operation in repeat transfer mode.

Table 15.6 Register update operation in repeat transfer mode (1 of 2)

Register	Function	Update operation after completion of a transfer by one transfer request	
		When DMCRAL register is not 1	When DMCRAL register is 1 (Transfer of the last data in repeat size)
DMSAR	Transfer source address	Increment/decrement/fixed/offset addition	• DMTMD.DTS[1:0] = 00b Increment/decrement/fixed/offset addition • DMTMD.DTS[1:0] = 01b Initial value of DMSAR • DMTMD.DTS[1:0] = 10b Increment/decrement/fixed/offset addition
DMDAR	Transfer destination address	Increment/decrement/fixed/offset addition	• DMTMD.DTS[1:0] = 00b Initial value of DMDAR • DMTMD.DTS[1:0] = 01b Increment/decrement/fixed/offset addition • DMTMD.DTS[1:0] = 10b Increment/decrement/fixed/offset addition
DMCRAH	Repeat size	Not updated	Not updated
DMCRAL	Transfer count	Decremented by one	DMCRAH

Table 15.6 Register update operation in repeat transfer mode (2 of 2)

Register	Function	Update operation after completion of a transfer by one transfer request	
		When DMCRAL register is not 1	When DMCRAL register is 1 (Transfer of the last data in repeat size)
DMCRBH	Number of repeat transfer operations	Not updated	Not updated
DMCRBL	Count of repeat transfer operations	Not updated	Decremented by one

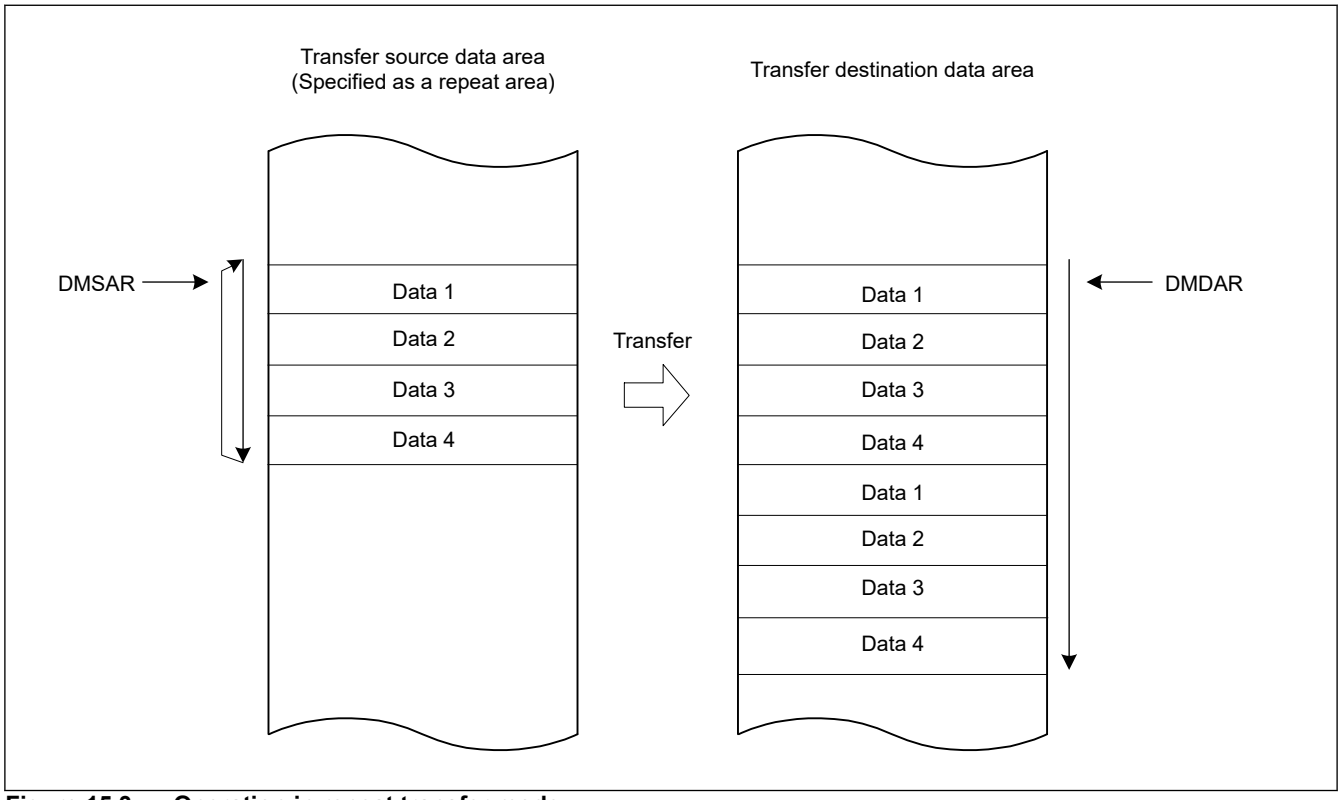


Figure 15.3 Operation in repeat transfer mode

15.3.1.3 Block Transfer Mode

In block transfer mode, a single block data is transferred by one transfer request.

A maximum of 1K data can be set as a total block transfer size using DMCRA register.

A maximum of 64K can be set as the number of block transfer operations using DMCRB register; therefore, a maximum of 64M data (1K data × 64K counts of block transfer operations) can be set as a total data transfer size.

Either the transfer source or transfer destination can be specified as a block area. When transfer of a single block data is completed, the address of the specified block area (DMSAR or DMDAR) returns to the transfer start address. When a single block data has all been transferred in block transfer mode, DMA transfer can be stopped, and the repeat size end interrupt can be requested. DMA transfer can be resumed by writing 1 to the DMCNT.DTE bit in the repeat size end interrupt handling.

Transfer end interrupt request can be generated after completion of the specified number of block transfer operations.

Table 15.7 summarizes the register update operation in block transfer mode, and Figure 15.4 shows the operation in block transfer mode.

Table 15.7 Register update operation in block transfer mode

Register	Function	Update operation after completion of single-block transfer by one transfer request
DMSAR	Transfer source address	• DMTMD.DTS[1:0] = 00b Increment/decrement/fixed/offset addition • DMTMD.DTS[1:0] = 01b Initial value of DMSAR • DMTMD.DTS[1:0] = 10b Increment/decrement/fixed/offset addition
DMDAR	Transfer destination address	• DMTMD.DTS[1:0] = 00b Initial value of DMDAR • DMTMD.DTS[1:0] = 01b Increment/decrement/fixed/offset addition • DMTMD.DTS[1:0] = 10b Increment/decrement/fixed/offset addition
DMCRAH	Block size	Not updated
DMCRAL	Transfer count	DMCRAH
DMCRBH	Number of block transfer operations	Not updated
DMCRBL	Count of block transfer operations	Decremented by one

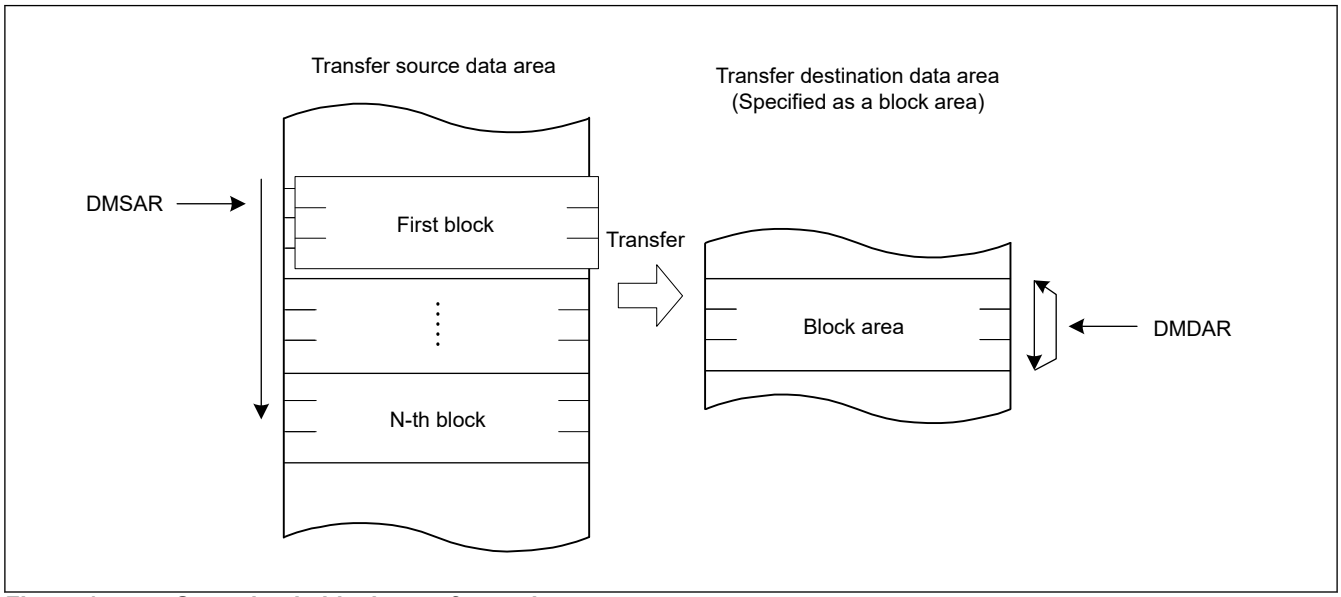


Figure 15.4 Operation in block transfer mode

15.3.1.4 Repeat-Block Transfer Mode

Repeat-block transfer is the operation mode with the following functions added to the block transfer function.

Repeat function: Added function (ring buffer) to repeat specified address area.

Offset function: Multiple areas with offset can be specified within one block transfer.

The repeat function and the offset function can be used for both the transfer source and the transfer destination of repeat-block transfer.

Figure 15.5 shows an example of adding a repeat function to the transfer destination.

Figure 15.6 shows repeat-block transfer with an offset to the transfer destination.

In repeat-block transfer mode, a single block data is transferred by one transfer request.

A maximum of 1K data can be set as a total block transfer size using DMCRA of the DMACn.

A maximum of 64K can be set as the number of block transfer operations using DMCRB of the DMACn; therefore, a maximum of 64M data (1K data × 64K counts of block transfer operations) can be set as a total data transfer size.

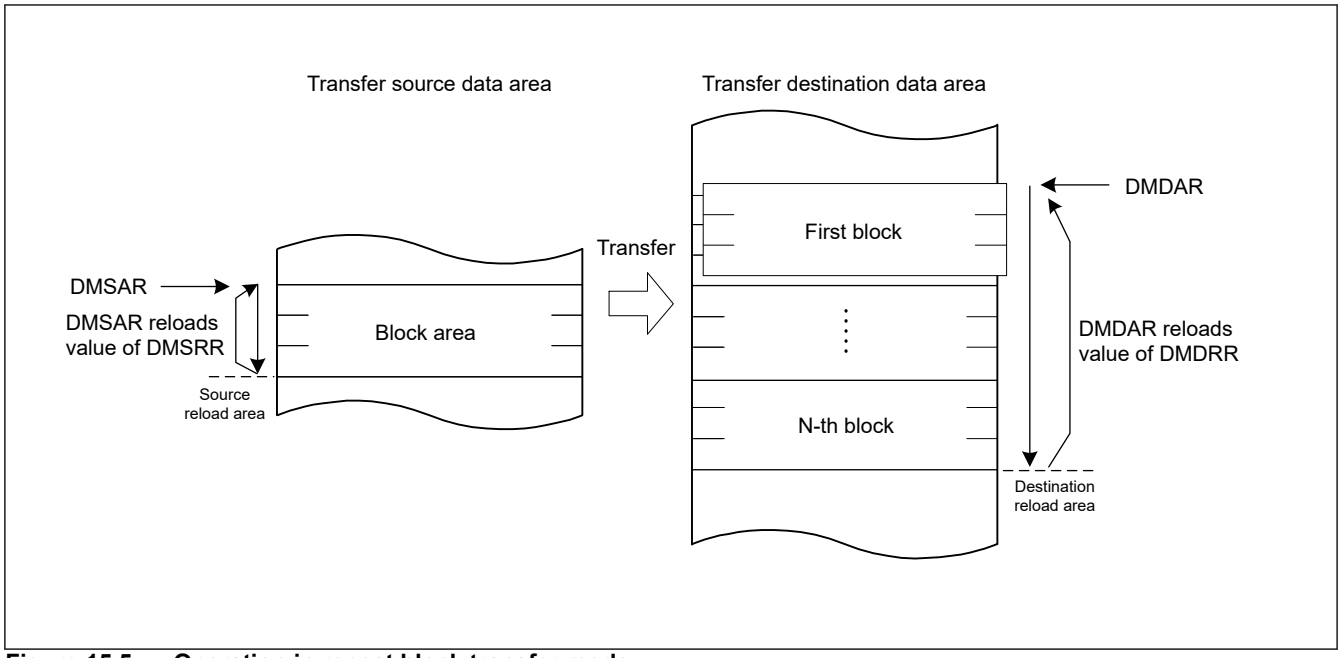


Figure 15.5 Operation in repeat block transfer mode

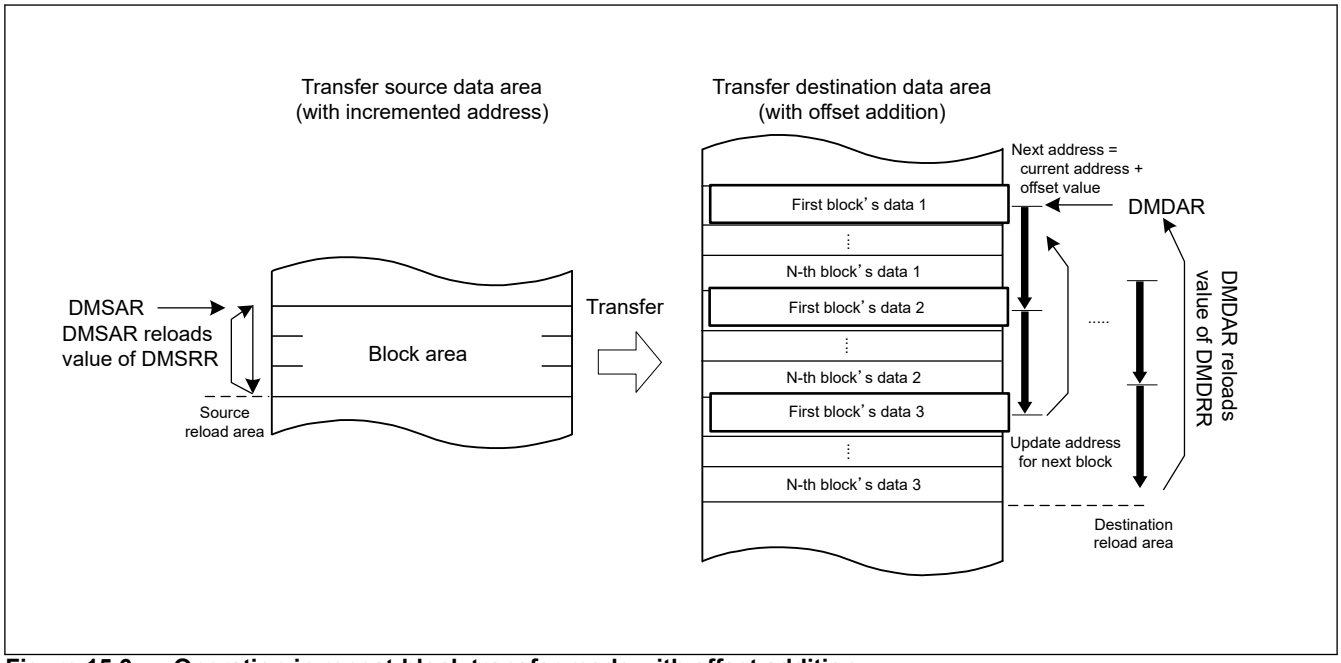


Figure 15.6 Operation in repeat-block transfer mode with offset addition

Table 15.8 to Table 15.13 summarize the register update operations in repeat-block transfer mode.

Table 15.8 Register update operation associated with source area in repeat-block transfer mode (fixed address DMAMD.SM[1:0] = 00b) (1 of 2)

Register	Function	Update operation after single data is transferred		
		DMCRAL[15:0] is not 1	DMCRAL[15:0] is 1 (single block is transferred)	
			DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1
DMSRR	Transfer source reload address	Not updated	Not updated	Not updated
DMSAR	Transfer source address	Not updated	Not updated	Not updated
DMCRAH[9:0]	Block size	Not updated	Not updated	Not updated

Table 15.8 Register update operation associated with source area in repeat-block transfer mode (fixed address DMAMD.SM[1:0] = 00b) (2 of 2)

Register	Function	Update operation after single data is transferred		
		DMCRAL[15:0] is not 1	DMCRAL[15:0] is 1 (single block is transferred)	
			DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1
DMCRAL[15:0]	Block size count	Decremented by 1	DMCRAH[9:0]	DMCRAH[9:0]
DMCRBH[15:0]	Number of block transfer operations	Not updated	Not updated	Not updated
DMCRBL[15:0]	Count of block transfer operations when DMTMD.TKP = 0	Not updated	Decremented by 1	0
	Count of block transfer operations when DMTMD.TKP = 1			DMCRBH[15:0]

Table 15.9 Register update operation associated with destination area in repeat-block transfer mode (fixed address DMAMD.DM[1:0] = 00b)

Register	Function	Update operation after single data is transferred		
		DMCRAL[15:0] is not 1	DMCRAL[15:0] is 1 (single block is transferred)	
			DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1
DMDRR	Transfer destination reload address	Not updated	Not updated	Not updated
DMDAR	Transfer destination address	Not updated	Not updated	Not updated
DMCRAH[9:0]	Block size	Not updated	Not updated	Not updated
DMCRAL[15:0]	Block size count	Decremented by 1	DMCRAH[9:0]	DMCRAH[9:0]
DMCRBH[15:0]	Number of block transfer operations	Not updated	Not updated	Not updated
DMCRBL[15:0]	Count of block transfer operations when DMTMD.TKP = 0	Not updated	Decremented by 1	0
	Count of block transfer operations when DMTMD.TKP = 1			DMCRBH[15:0]

Table 15.10 Register update operation associated with source area in repeat-block transfer mode (incremented or decremented address DMAMD.SM[1:0] = 10b or 11b) (1 of 2)

Register	Function	Update operation after single data is transferred					
		DMSBSL[15:0] is not 1			DMSBSL[15:0] is 1		
		DMCRAL[15:0] is not 1	DMCRAL[15:0] is 1 (single block is transferred)		DMCRAL[15:0] is not 1	DMCRAL[15:0] is 1 (single block is transferred)	
			DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1		DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1
DMSRR	Transfer source reload address	Not updated	Not updated	Not updated	Not updated	Not updated	Not updated

Table 15.10 Register update operation associated with source area in repeat-block transfer mode (incremented or decremented address DMAMD.SM[1:0] = 10b or 11b) (2 of 2)

Register	Function	Update operation after single data is transferred					
		DMSBSL[15:0] is not 1			DMSBSL[15:0] is 1		
		DMCRAL[15:0] is not 1	DMCRAL[15:0] is 1 (single block is transferred)		DMCRAL[15:0] is not 1	DMCRAL[15:0] is 1 (single block is transferred)	
			DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1		DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1
DMSAR	Transfer source address when DMTMD.SM[1:0] = 10b	Incremented by Data Size			DMSRR		
	Transfer source address when DMTMD.SM[1:0] = 11b	Decremented by Data Size			DMSRR		
DMCRAH[9:0]	Block size	Not updated	Not updated	Not updated	Not updated	Not updated	Not updated
DMCRAL[15:0]	Block size count	Decremented by 1	DMCRAH[9:0]	DMCRAH[9:0]	Decremented by 1	DMCRAH[9:0]	DMCRAH[9:0]
DMSBSH[15:0]	Source buffer size (Repeat-size)	Not updated	Not updated	Not updated	Not updated	Not updated	Not updated
DMSBSL[15:0]	Count of transfer data in source buffer	Decremented by 1	Decremented by 1	Decremented by 1	DMSBSH	DMSBSH	DMSBSH
DMCRBH[15:0]	Number of block transfer operations	Not updated	Not updated	Not updated	Not updated	Not updated	Not updated
DMCRBL[15:0]	Count of block transfer operations when DMTMD.TKP = 0	Not updated	Decremented by 1	0	Not updated	Decremented by 1	0
	Count of block transfer operations when DMTMD.TKP = 1			DMCRBH[15:0]			DMCRBH[15:0]

Table 15.11 Register update operation associated with destination area in repeat-block transfer mode (incremented or decremented address DMAMD.DM[1:0] = 10b or 11b) (1 of 2)

Register	Function	Update operation after single data is transferred					
		DMDBSL[15:0] is not 1			DMDBSL[15:0] is 1		
		DMCRAL[15:0] is not 1	DMCRAL[15:0] is 1 (single block is transferred)		DMCRAL[15:0] is not 1	DMCRAL[15:0] is 1 (single block is transferred)	
			DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1		DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1
DMDRR	Transfer destination reload address	Not updated	Not updated	Not updated	Not updated	Not updated	Not updated

Table 15.11 Register update operation associated with destination area in repeat-block transfer mode (incremented or decremented address DMAMD.DM[1:0] = 10b or 11b) (2 of 2)

Register	Function	Update operation after single data is transferred					
		DMDBSL[15:0] is not 1			DMDBSL[15:0] is 1		
		DMCRAL[15:0] is not 1	DMCRAL[15:0] is 1 (single block is transferred)		DMCRAL[15:0] is not 1	DMCRAL[15:0] is 1 (single block is transferred)	
			DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1		DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1
DMDAR	Transfer destination address when DMTMD.DM[1:0] = 10b	Incremented by Data Size			DMDRR		
	Transfer destination address when DMTMD.DM[1:0] = 11b	Decremented by Data Size			DMDRR		
DMCRAH[9:0]	Block size	Not updated	Not updated	Not updated	Not updated	Not updated	Not updated
DMCRAL[15:0]	Block size count	Decremented by 1	DMCRAH[9:0]	DMCRAH[9:0]	Decremented by 1	DMCRAH[9:0]	DMCRAH[9:0]
DMDBSH[15:0]	Destina-tion buffer size (Repeat-size)	Not updated	Not updated	Not updated	Not updated	Not updated	Not updated
DMDBSL[15:0]	Count of transfer data in destination buffer	Decremented by 1	Decremented by 1	Decremented by 1	DMDBSH	DMDBSH	DMDBSH
DMCRBH[15:0]	Number of block transfer operations	Not updated	Not updated	Not updated	Not updated	Not updated	Not updated
DMCRBL[15:0]	Count of block transfer operations when DMTMD.TKP = 0	Not updated	Decremented by 1	0	Not updated	Decremented by 1	0
	Count of block transfer operations when DMTMD.TKP = 1			DMCRBH[15:0]			DMCRBH[15:0]

Table 15.12 Register update operation associated with source area in repeat-block transfer mode (offset addition DMAMD.SM[1:0] = 01b) (1 of 2)

Register	Function	DMCRAL[15:0] is not 1	DMCRAL[15:0] is 1 (single block is transferred)			
			DMSBSL[15:0] is not 1		DMSBSL[15:0] is 1	
			DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1	DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1
DMSRR	Transfer source reload address	Not updated	Not updated	Not updated	Not updated	Not updated

Table 15.12 Register update operation associated with source area in repeat-block transfer mode (offset addition DMAMD.SM[1:0] = 01b) (2 of 2)

Register	Function	DMCRAL[15:0] is not 1	DMCRAL[15:0] is 1 (single block is transferred)			
			DMSBSL[15:0] is not 1		DMSBSL[15:0] is 1	
			DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1	DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1
DMSAR	Transfer source address when DMAMD.SADR = 0	Offset addition by DMSBSH	DMSRR		DMSRR	
	Transfer source address when DMAMD.SADR = 1		DMSRR + (DMS-BSH - DMSBSL) × DataSize			
DMCRAH[9:0]	Block size	Not updated	Not updated	Not updated	Not updated	Not updated
DMCRAL[15:0]	Block size count	Decremented by 1	DMCRAH[9:0]	DMCRAH[9:0]	DMCRAH[9:0]	DMCRAH[9:0]
DMSBSH[15:0]	Source buffer size (Repeat-size)	Not updated	Not updated	Not updated	Not updated	Not updated
DMSBSL[15:0]	Count of transfer data in source buffer	Not updated	Decremented by 1	Decremented by 1	DMSBSH	DMSBSH
DMCRBH[15:0]	Number of block transfer operations	Not updated	Not updated	Not updated	Not updated	Not updated
DMCRBL[15:0]	Count of block transfer operations when DMTMD.TKP = 0	Not updated	Decremented by 1	0	Decremented by 1	0
	Count of block transfer operations when DMTMD.TKP = 1			DMCRBH[15:0]		DMCRBH[15:0]

Table 15.13 Register update operation associated with destination area in repeat-block transfer mode (offset addition DMAMD.DM[1:0] = 01b) (1 of 2)

Register	Function	DMCRAL[15:0] is not 1	DMCRAL[15:0] is 1 (single block is transferred)			
			DMDBSL[15:0] is not 1		DMDBSL[15:0] is 1	
			DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1	DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1
DMDRR	Transfer destination reload address	Not updated	Not updated	Not updated	Not updated	Not updated
DMSAR	Transfer destination address when DMAMD.DADR = 0	Offset addition by DMDBSH	DMDRR		DMDRR	
	Transfer destination address when DMAMD.DADR = 1		DMDRR + (DMDBSH - DMDBSL) × DataSize			
DMCRAH[9:0]	Block size	Not updated	Not updated	Not updated	Not updated	Not updated
DMCRAL[15:0]	Block size count	Decremented by 1	DMCRAH[9:0]	DMCRAH[9:0]	DMCRAH[9:0]	DMCRAH[9:0]

Table 15.13 Register update operation associated with destination area in repeat-block transfer mode (offset addition DMAMD.DM[1:0] = 01b) (2 of 2)

Register	Function	DMCRAL[15:0] is not 1	DMCRAL[15:0] is 1 (single block is transferred)			
			DMDBSL[15:0] is not 1		DMDBSL[15:0] is 1	
			DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1	DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1
DMDBSH[15:0]	Destina-tion buffer size (Repeat-size)	Not updated	Not updated	Not updated	Not updated	Not updated
DMDBSL[15:0]	Count of transfer data in destination buffer	Not updated	Decremented by 1	Decremented by 1	DMDBSH	DMDBSH
DMCRBH[15:0]	Number of block transfer operations	Not updated	Not updated	Not updated	Not updated	Not updated
DMCRBL[15:0]	Count of block transfer operations when DMTMD.TKP = 0	Not updated	Decremented by 1	0	Decremented by 1	0
	Count of block transfer operations when DMTMD.TKP = 1			DMCRBH[15:0]		DMCRBH[15:0]

15.3.2 Extended Repeat Area Function

The DMAC supports a function to specify the extended repeat areas on the transfer source and destination addresses. With the extended repeat areas set, the address registers repeatedly indicate the addresses of the specified extended repeat areas.

The extended repeat areas can be specified separately to the transfer source address register (DMSAR) and transfer destination address register (DMDAR).

The extended repeat area on the source address is specified by the DMAMD.SARA[4:0] bits. The extended repeat area on the destination address is specified by the DMAMD.DARA[4:0] bits. The size can be specified separately for the source and destination sides.

However, the area (of transfer source or transfer destination) which is specified as the repeat area or block area should not be specified as the extended repeat area.

When the address register value reaches the end address of the extended repeat area and the extended repeat area overflows, DMA transfer is stopped and an interrupt by an extended repeat area overflow can be requested. When an overflow occurs in the extended repeat area on the transfer source while the DMINT.SARIE bit is set to 1, the DMSTS.ESIF flag is set to 1 and the DMCNT.DTE bit is cleared to 0 to stop DMA transfer. At this time, if the DMINT.ESIE bit is set to 1, an interrupt by an extended repeat area overflow is requested. When the DMINT.DARIE bit is set to 1, the destination address register becomes a target to apply the function. DMA transfer can be resumed by writing 1 to the DMCNT.DTE bit in the interrupt handling.

Figure 15.7 shows an example of the extended repeat area operation.

Eight bytes are specified as an extended repeat area by the lower three bits of DMSAR (DMAMD.SARA[4:0] bits = 00011b). The data size is eight bits (DMTMD.SZ[1:0] = 00b).

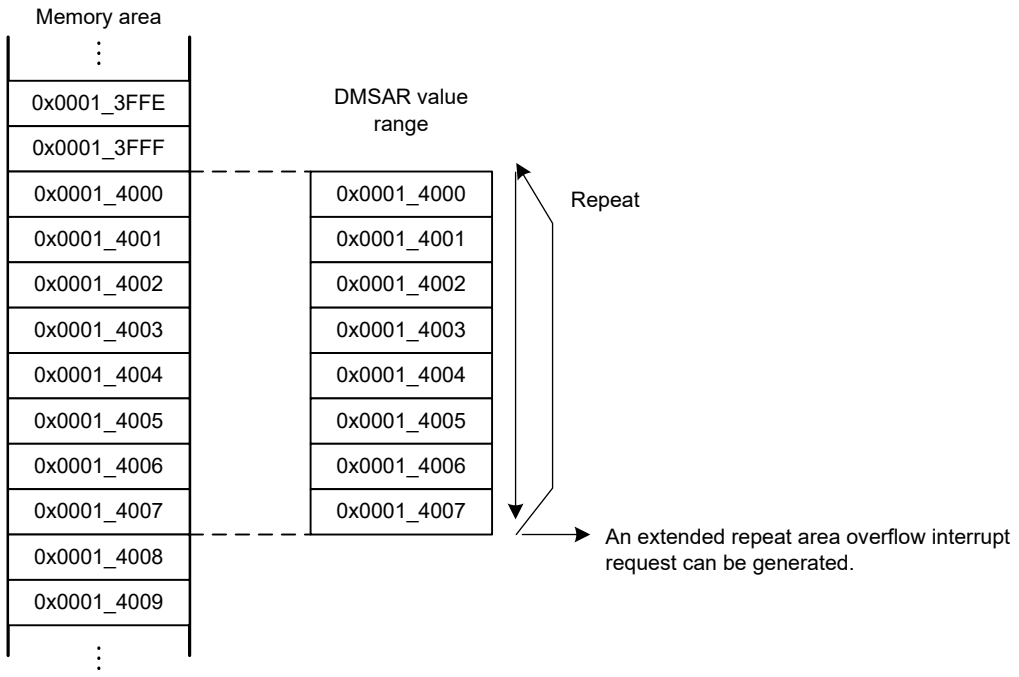


Figure 15.7 Example of extended repeat area operation

When an interrupt by an extended repeat area overflow is used in block transfer mode, the following should be taken into consideration.

When a transfer is stopped by an interrupt by an extended repeat area overflow, the address register must be set so that the block size is a power of 2 or the block size boundary is aligned with the extended repeat area boundary. When an overflow on the extended repeat area occurs during a transfer of one block, the interrupt by the overflow is suspended until transfer of the block is completed, and the transfer overruns.

Figure 15.8 shows an example when the extended repeat area function is used in block transfer mode.

Eight bytes are specified as an extended repeat area by the lower three bits of DMSAR (DMAMD.SARA[4:0] bits = 00011b), block transfer mode with block size 5 is set (DMCRA = 0x0005_0005), and transfer source address is not specified as a block area. Data size is eight bits (DMTMD.SZ[1:0] bits = 00b).

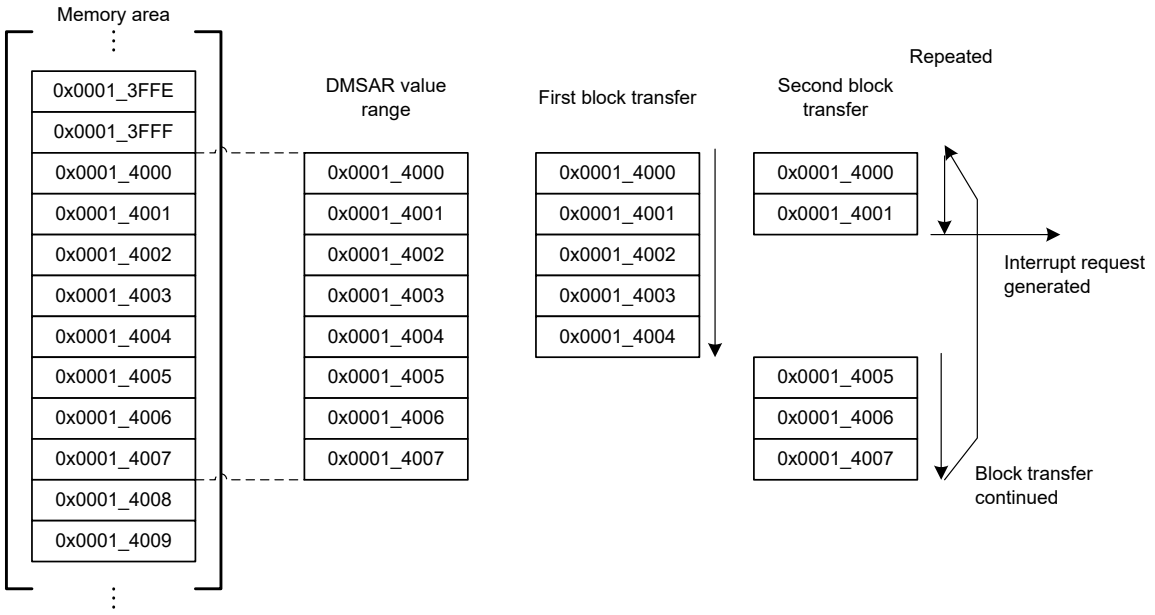


Figure 15.8 Example of extended repeat area function in block transfer mode

15.3.3 Free-running Function

The DMAC supports free-running function. This function allows to transfer repeatedly without reconfiguring in interrupt handler.

15.3.3.1 In Normal Transfer Mode

In normal transfer mode, when DMCRA.DMCRAL bits are set to 0x0000, no specific number of transfer operations is set; data transfer is performed with the transfer counter stopped.

For more information, see section 15.3.1.1. Normal Transfer Mode.

15.3.3.2 In Other Transfer Modes

In repeat, block and repeat-block transfer mode, the DMAC supports free-running function using the DMTMD.TKP bit. If the DMTMD.TKP bit is to be set to 1, the transfer is not stopped by completion of specified total number of transfer operations and reloads DMCRBH repeatedly.

Figure 15.9 show an example of block transfer operation without free-running function.

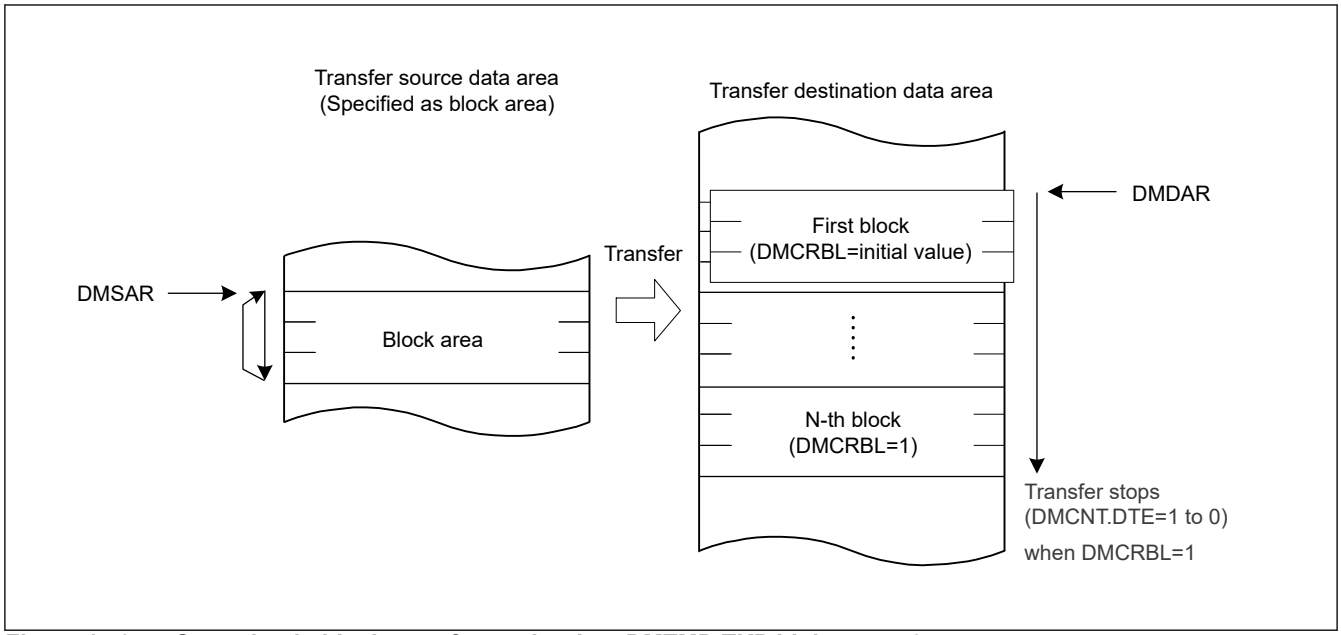


Figure 15.9 Operation in block transfer mode when DMTMD.TKP bit is set to 0

Figure 15.10 show an example of block transfer operation with free-running function.

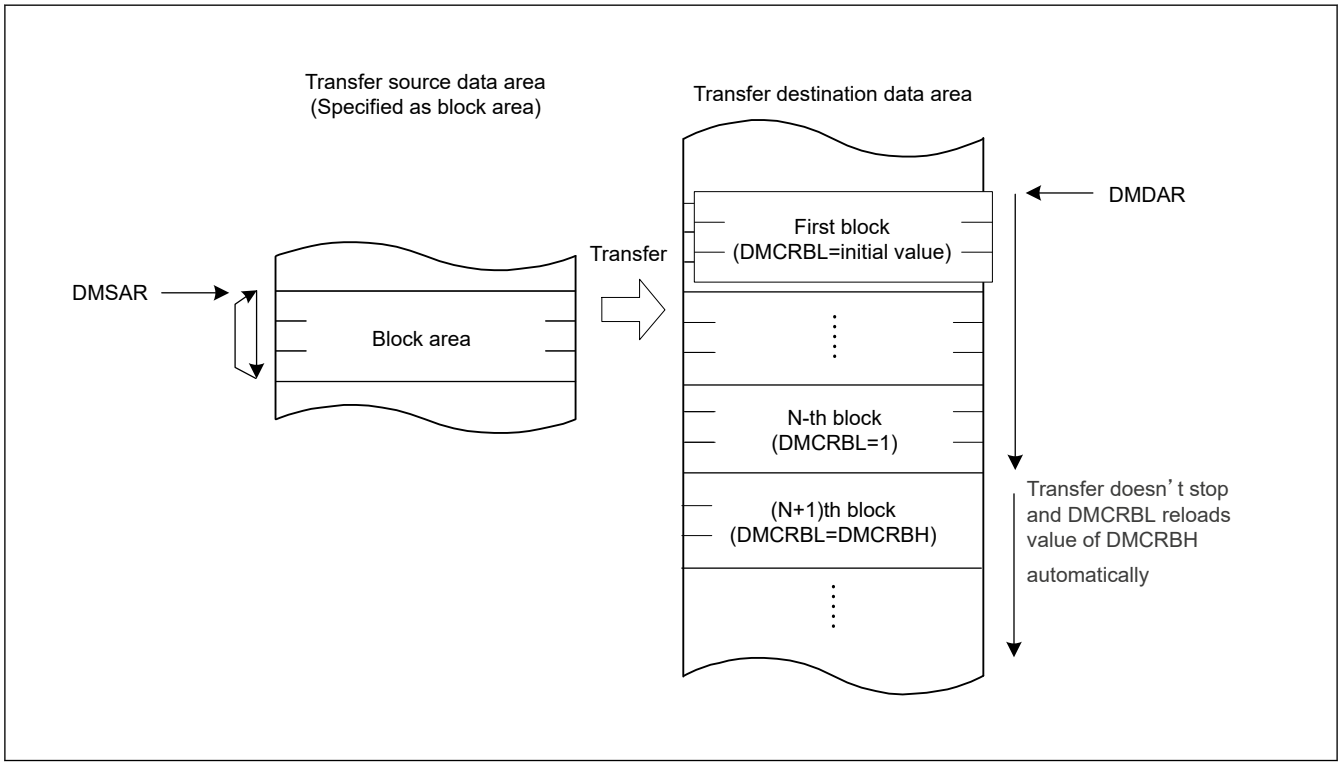


Figure 15.10 Operation in block transfer mode when DMTMD.TKP bit is set to 1

15.3.4 Address Update Function using Offset

The source and destination addresses can be updated by fixing, increment, decrement, or offset addition. In normal, repeat and block transfer mode, when the offset addition is selected, the offset specified by the DMA offset register (DMOFR) is added to the address every time the DMAC performs one data transfer. This function realizes a data transfer where addresses are allocated to separated areas.

Offset subtraction can also be realized by setting a negative value in DMOFR. In this case, the negative value must be 2's complement.

DMSBS or DMDBS are used instead of DMOFR in repeat-block transfer mode. For more information section 15.3.1.4. Repeat-Block Transfer Mode

Table 15.14 shows the address update method in each address update mode.

Table 15.14 Address update method in each address update mode

Address update mode	Settings of DMAMD.SM[1:0] and DMAMD.DM[1:0] for address update modes	Address update method (for different SZ[1:0] settings in DMTMD)		
		SZ[1:0] = 00b	SZ[1:0] = 01b	SZ[1:0] = 10b
Address fixed	00b	Fixed		
Offset addition	01b	+DMOFR*1		
Increment	10b	+1	+2	+4
Decrement	11b	–1	–2	–4

Note 1. When setting a negative value in the DMA Offset Register, the value must be in two's complement, obtained by the following formula:
two's complement of a negative offset value = ~ (offset) + 1 (~ = bit inversion)

15.3.4.1 Basic Transfer Using Offset Addition

Figure 15.11 shows an example of address updating using offset addition.

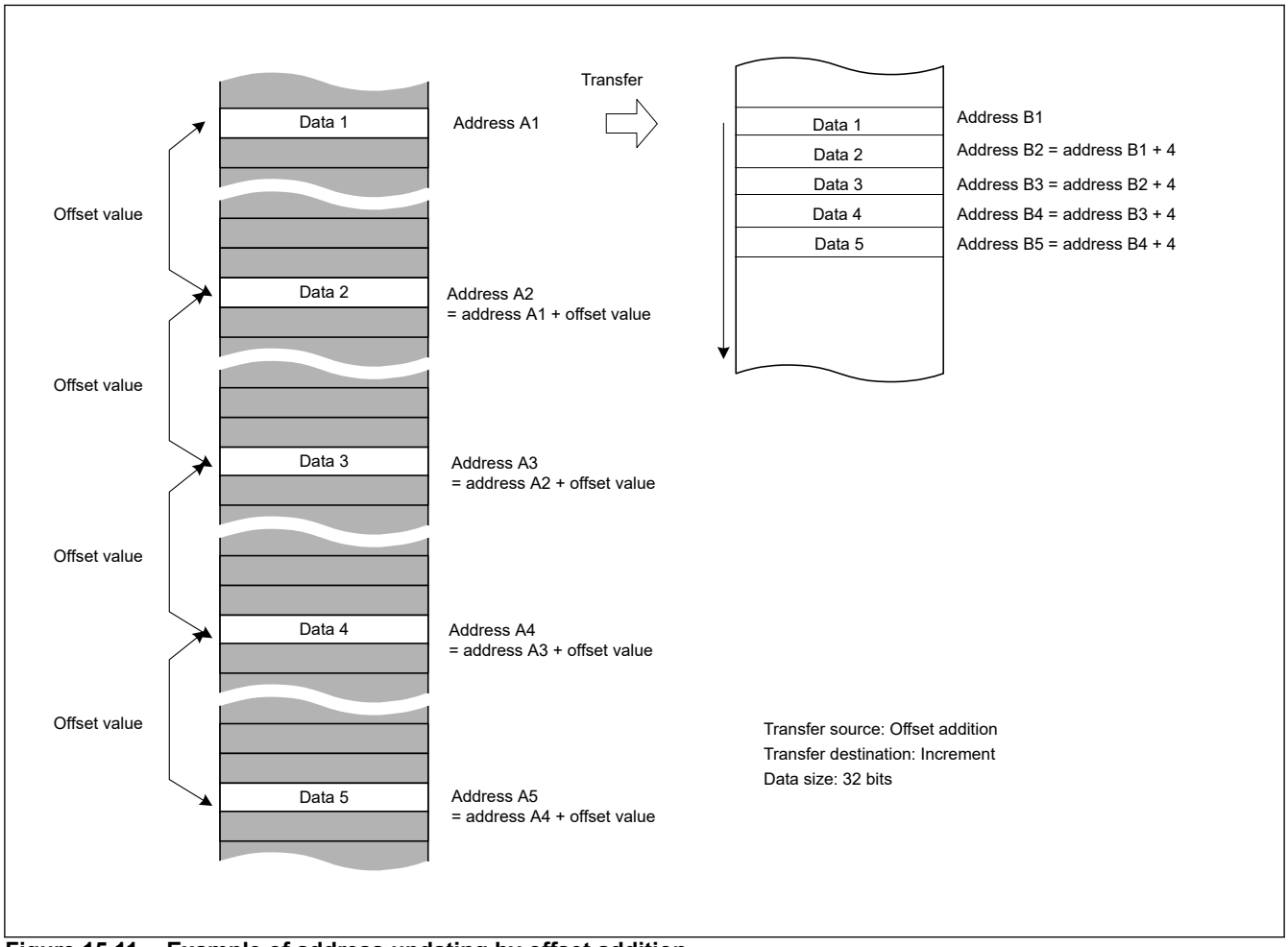


Figure 15.11 Example of address updating by offset addition

Figure 15.11 shows the setting of the following.

- The transfer data is 32 bits long.
- Offset addition is set as the transfer source address update mode.

- Increment is set as the transfer destination address update mode.

The second and subsequent data is each read from the transfer source address obtained by adding the offset value to the previous address. The data read from the addresses at the specified intervals is written to the continuous locations on the destination.

15.3.4.2 Example of XY Conversion Using Offset Addition

Figure 15.12 shows the XY conversion using offset addition in repeat transfer mode.

Settings are as follows:

- DMAMD.SM — Transfer source address update mode: Offset addition.
- DMAMD.DM — Transfer destination address update mode: Destination address is incremented.
- DMTMD.SZ — Transfer data size select: 32 bits.
- DMTMD.MD — Transfer mode select: Repeat transfer.
- DMTMD.DTS — Repeat area select: The source is specified as the repeat area.
- DMOFR — Offset address: 0x10.
- DMCRA — Repeat size: 0x4.
- DMINT.RPTIE — The repeat size end interrupt is enabled.

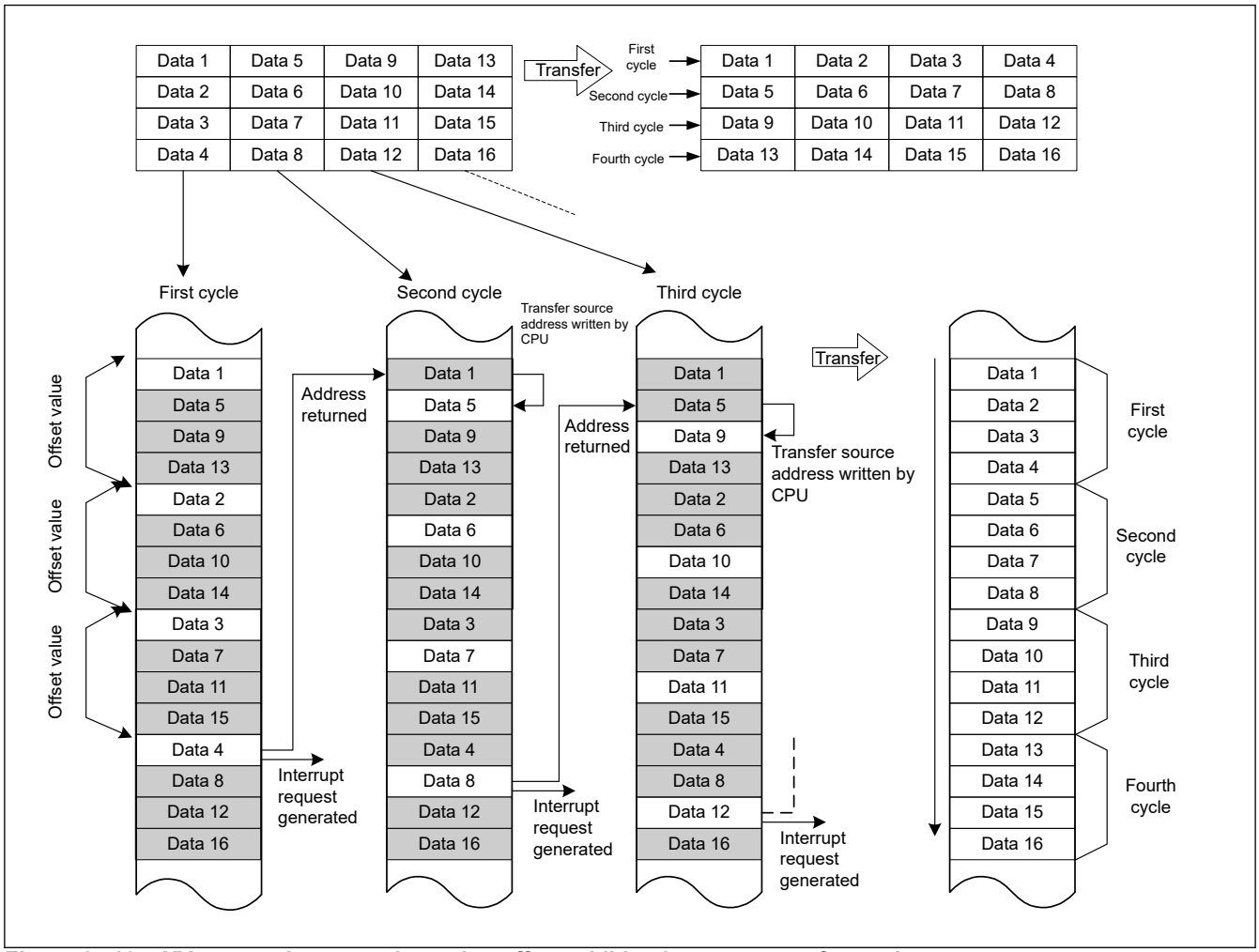


Figure 15.12 XY conversion operation using offset addition in repeat transfer mode

When a transfer starts, the offset value is added to the transfer source address every time data is transferred. The transfer data is written to continuous transfer destination addresses. When data 4 is transferred:

- The repeat size of data transfer is complete.

- The transfer source address returns to the transfer start address (the address of data 1 on the transfer source).
- A repeat size end interrupt is requested.

During the time this interrupt pauses the transfer, the following operations are performed.

- DMSAR — Rewrite the DMA transfer source address to the address of data 5
  (with the above example, the data 1 address + 4).
- DMCNT — Set the DTE bit to 1.

The DMA transfer is resumed from the state when the DMA transfer is stopped. After that, the operations described above are repeated until the transfer source data is transposed to the destination area (XY conversion).

Figure 15.13 shows a flowchart of the XY conversion.

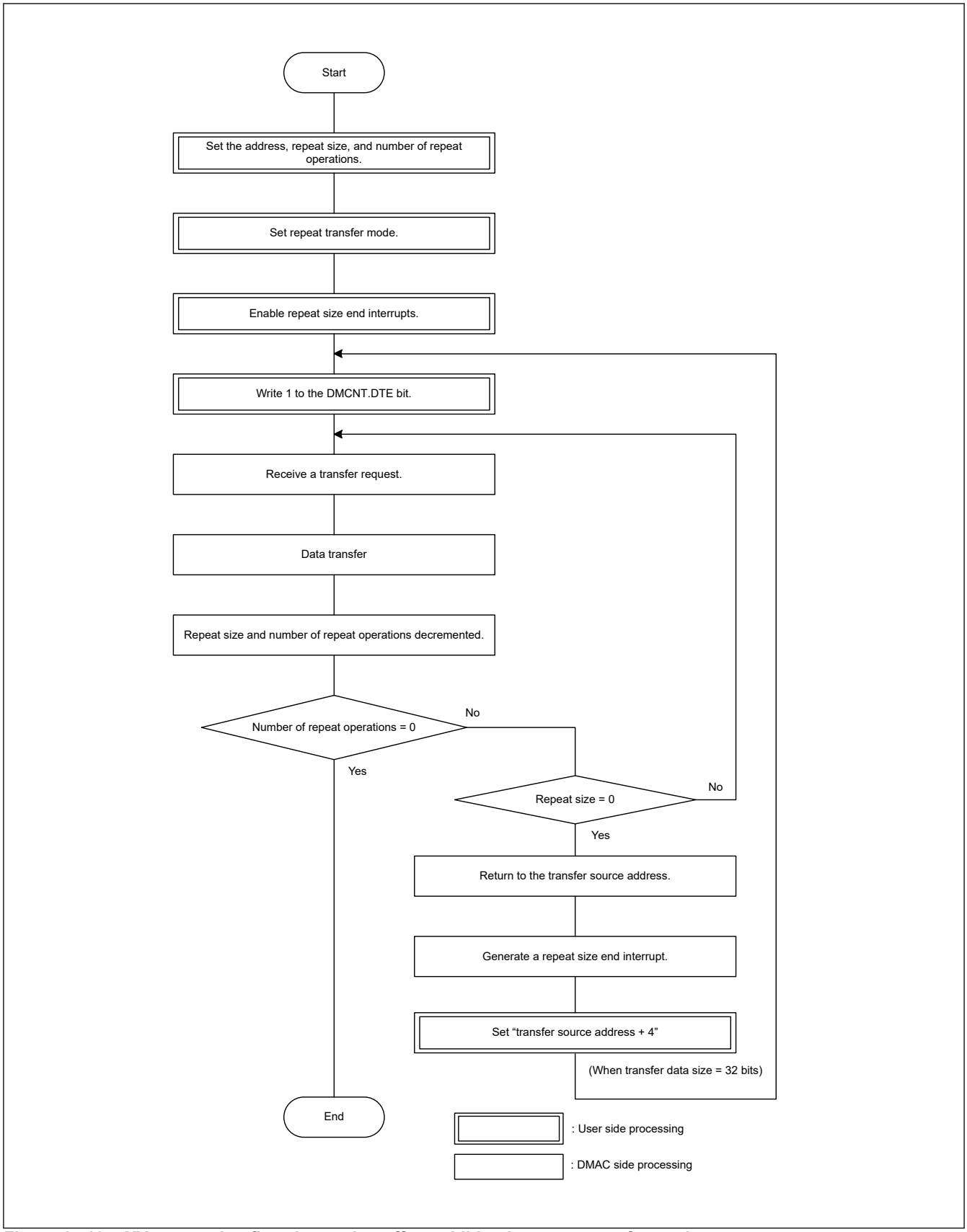


Figure 15.13 XY conversion flowchart using offset addition in repeat transfer mode

15.3.5 Address Update Function in Repeat-Block Transfer Mode

Repeat-block transfer mode is an extension of repeat transfer mode and block transfer mode. However, the detailed behavior of the address update is different from these two modes. Here are the details of the address update function in repeat-block transfer mode.

15.3.5.1 Fixed Address Mode

When DMAMD.SM[1:0] is set to 00b, the address update mode of the source is fixed address. And when DMAMD.DM[1:0] is set to 00b, the address update mode of the destination is fixed address.

In fixed address, the address is not updated from the initial value of DMSAR and DMDAR. If the block size (DMCRA) is larger than 1, the same data will be transferred multiple times for one request.

Figure 15.14 shows address update in fixed address.

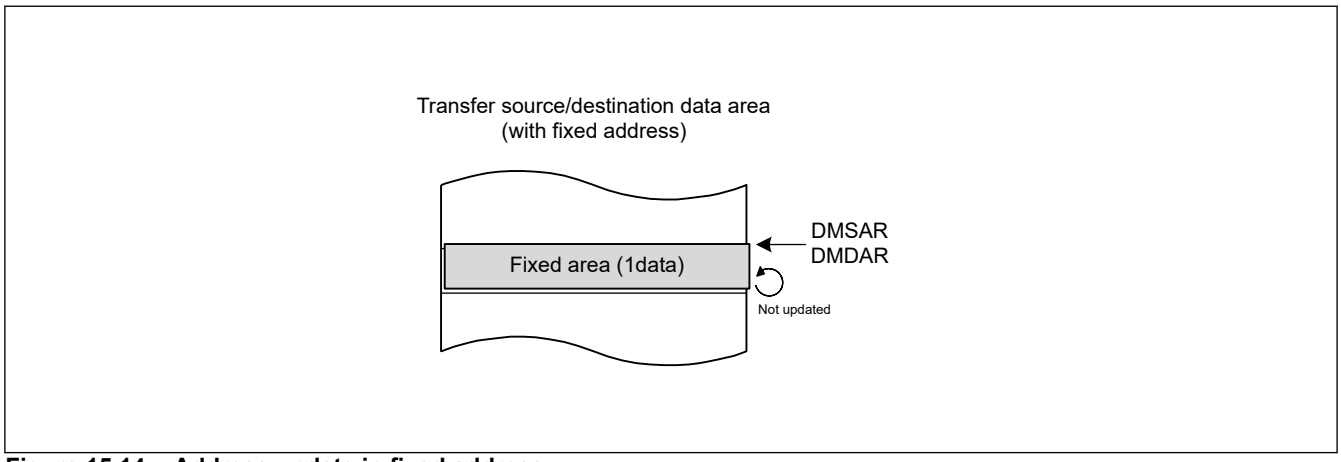


Figure 15.14 Address update in fixed address

15.3.5.2 Incremental and Decremental Address Mode

When DMAMD.SM[1:0] is set to 10b, the address update mode of the source is incremental address. And when DMAMD.DM[1:0] is set to 10b, the address update mode of the destination is incremental address. When DMAMD.SM[1:0] is set to 11b, the address update mode of the source is decremental address. And when DMAMD.DM[1:0] is set to 11b, the address update mode of the destination is decremental address.

In these update modes, the address is incremented or decremented according to the setting of DMTMD.SZ[1:0].

In these update modes DMSBS and DMDBS indicates a reload area. The unit of DMSBS and DMDBS is "number of data". At the start of transfer, DMSBSL and DMDBSL, which is the lower 16 bits of DMSBS and DMDBS, operates as a down counter and decrements each time one data transfer is performed. When the value becomes 1, DMSAR and DMDAR reloads the value of DMSRR and DMDRR.

Figure 15.15 shows address update in incremental address.

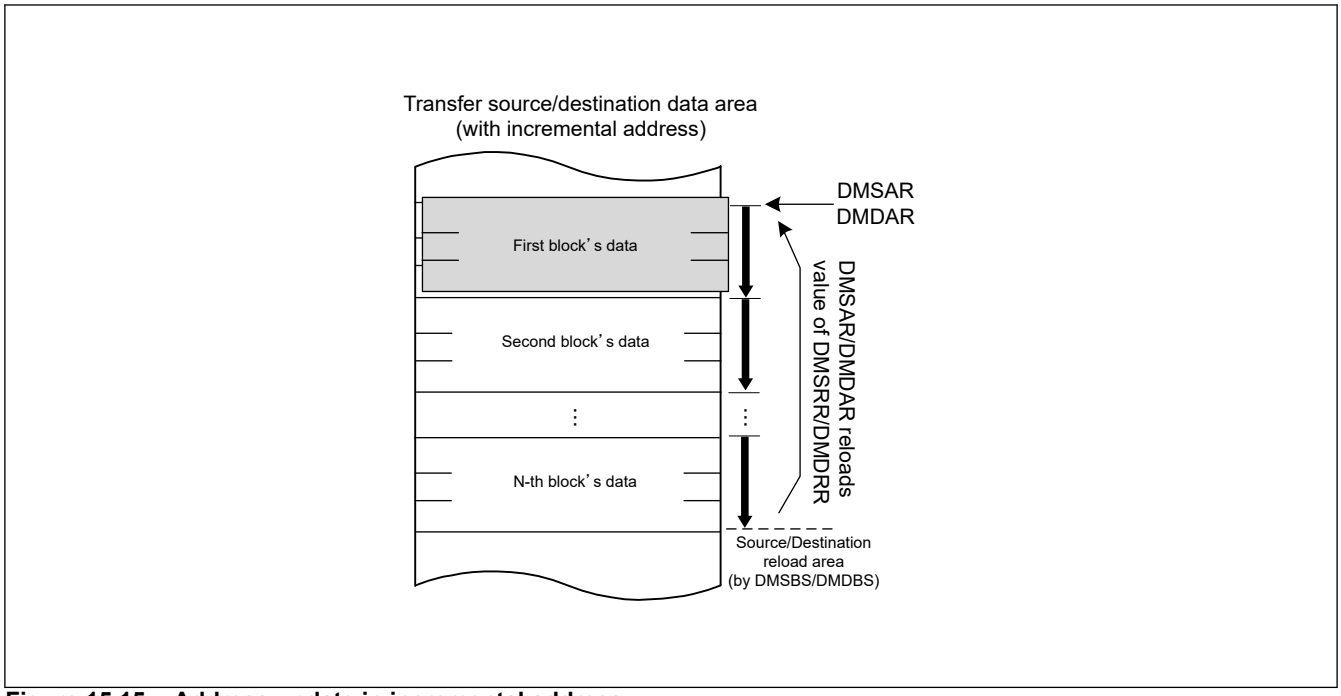


Figure 15.15 Address update in incremental address

15.3.5.3 Offset Addition Mode

When DMAMD.SM[1:0] is set to 01b, the address update mode of the source is offset addition. And when DMAMD.DM[1:0] is set to 01b, the address update mode of the destination is offset addition.

In offset addition, DMSBS and DMDBS indicates reload area and also works as an access offset value. Unlike other transfer modes, DMOFR register is not used in repeat-block transfer mode. In offset addition, the unit of DMSBS and DMDBS is the number of blocks. When the transfer starts, DMCRAL operates as a down counter, DMSAR and DMDAR reloads the value of DMSRR and DMDRR every time one block is transferred. In addition, DMSBSL and DMDBSL, which is the lower 16 bits of DMSBS and DMDBS, also operates as a down counter and decrements every time one block is transferred. When the DMSBS and DMDBS value becomes 1, DMSAR and DMDAR reloads the value of DMSRR and DMDRR.

When DMAMD.SADR and DMAMD.DADR is set to 0, offset addition operation of the same area is repeated. DMDAR only reloads DMDRR. Figure 15.16 shows address update in offset addition with DMAMD.SADR and DMAMD.DADR=0.

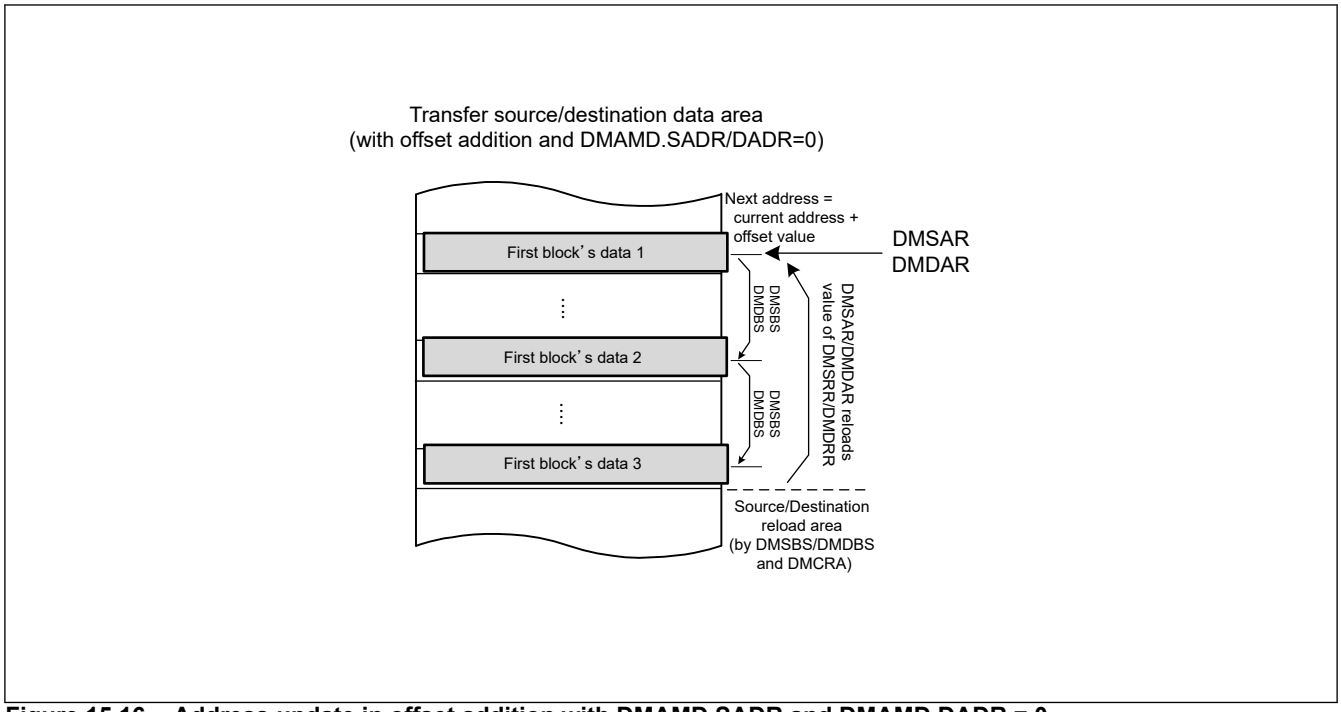


Figure 15.16 Address update in offset addition with DMAMD.SADR and DMAMD.DADR = 0

When DMAMD.SADR and DMAMD.DADR is set to 1, the address is incremented by one data unit after DMSRR and DMDRR is reloaded by DMCRAL=1. In other words, an index value ((DMDBSH-DMDBSL) × DataSize) is added to DMDAR after DMDRR is reloaded. This behavior is used to implement multiple ring buffers. Figure 15.17 shows address update in offset addition with DMAMD.SADR and DMAMD.DADR=1.

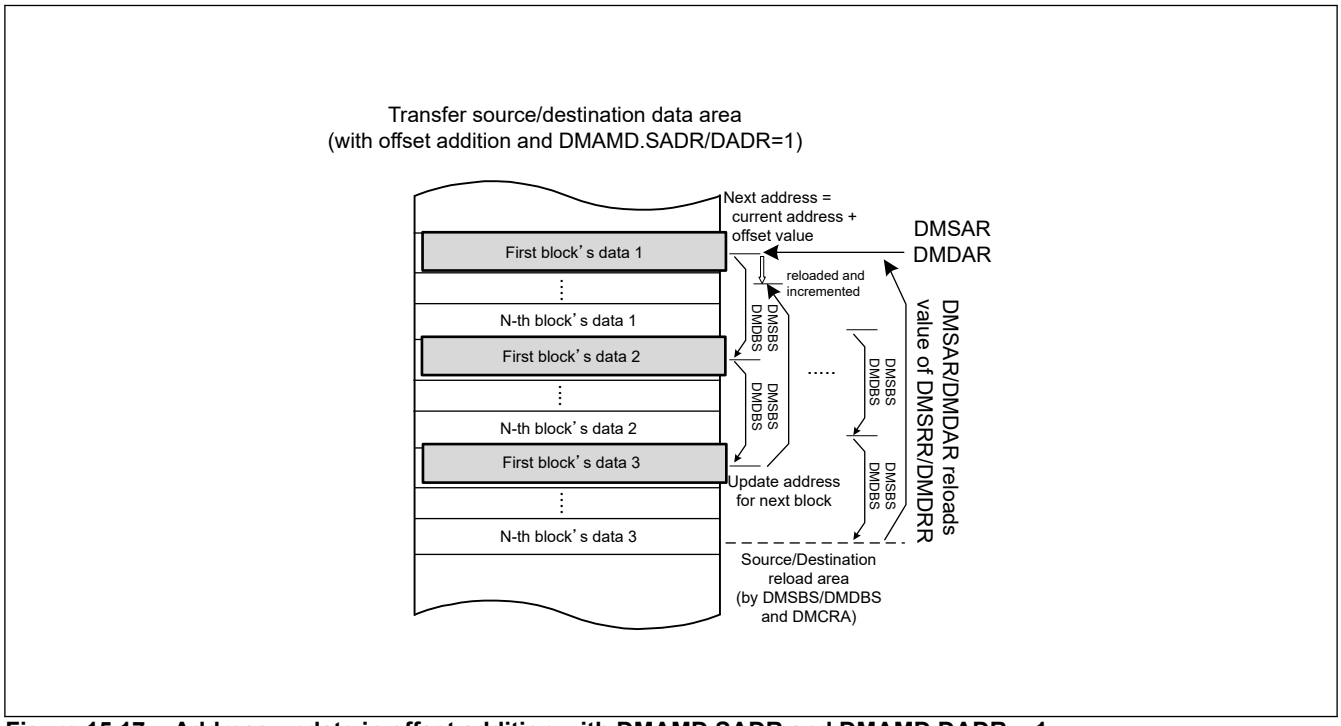


Figure 15.17 Address update in offset addition with DMAMD.SADR and DMAMD.DADR = 1

15.3.6 Example of Using Repeat-Block Transfer Mode

In repeat-block transfer mode, it is possible to realize repeated access to interval data and single or multiple ring buffers by combining the above address update modes. Following sections shows some usage examples.

15.3.6.1 Interval Address to Single Ring Buffer

Figure 15.18 shows an example of reading interval ADDRn registers (data register) of ADC module and storing it in single ring buffer. It transfers 2 data every 4 words per 1 request. DMSAR is incremented by one data every one request. This can be achieved by setting the transfer source to offset addition and DMAMD.SADR=1, the block size (DMCRA) to 2, and the transfer source offset (DMSBS) to 4. Table 15.15 shows setting of this example.

Table 15.15 Setting of use case: from interval address to single ring buffer

Register	Value	Description
DMSAR, DMSRR	0x4017_1000	Initial source address
DMDAR, DMDRR	0x2000_0000	Initial destination address
DMTMD.SZ[1:0]	10b	Data size is word
DMAMD.SADR	1	Incremental source address after reloading
DMAMD.SM[1:0]	01b	Source update mode is offset addition
DMAMD.DM[1:0]	10b	Destination update mode is incremental address
DMCRAH, DMCRAL	2	Transfer block size
DMSBSH, DMSBSL	4	Source whole buffer size (unit is ‘blocks’) and Source access offset (unit is ‘data’)
DMDBSH, DMDBSL	N × 2(DMCRA)	Destination buffer size (unit is ‘data’)

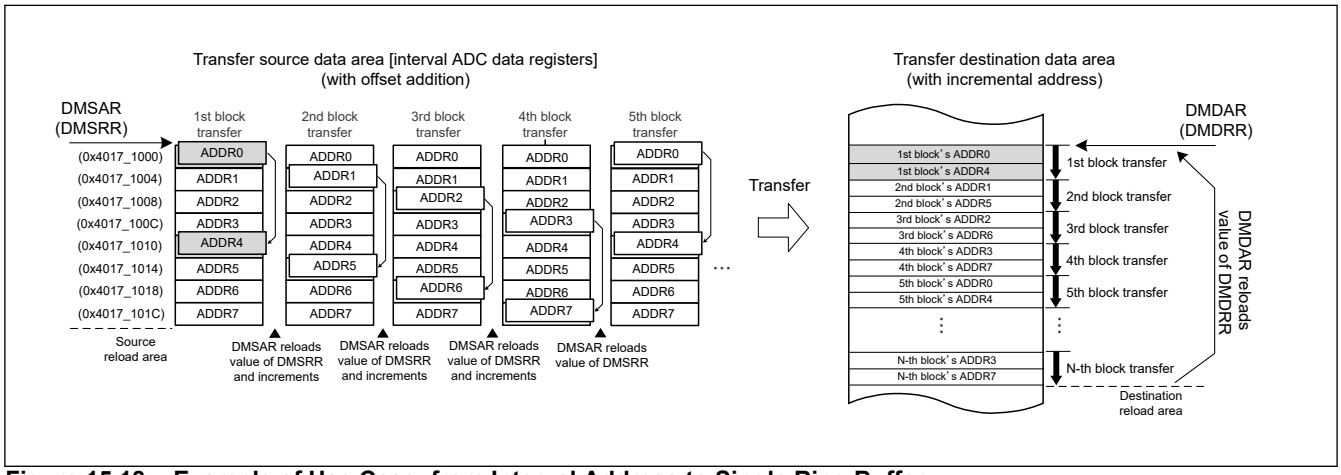


Figure 15.18 Example of Use Case: from Interval Address to Single Ring Buffer

15.3.6.2 Single Block to Multi Ring Buffer

Figure 15.19 shows an example of storing the continuous ADDRn registers (data register) of ADC module individually in multiple ring buffers. In this example, a ring buffer in which only the first element (ADDR0) in a single block is arranged in transfer order is created at the destination. Also, in the next area, create a ring buffer in which only the second element (ADDR1) is arranged in transfer order. In the following case, create a ring buffer of length N, which is defined by DMDBS. And the number of data elements in the block is 3, which is defined by DMCRA. Table 15.16 shows setting of this example.

Table 15.16 Setting of use case: from single block to multi ring buffer (1 of 2)

Register	Value	Description
DMSAR, DMSRR	0x4017_1000	Initial source address
DMDAR, DMDRR	0x2000_0000	Initial destination address
DMTMD.SZ[1:0]	10b	Data size is word
DMAMD.SADR	0	Only reloading
DMAMD.DADR	1	Incremental destination address after reloading

Table 15.16 Setting of use case: from single block to multi ring buffer (2 of 2)

Register	Value	Description
DMAMD.SM[1:0]	10b	Source update mode is incremental address
DMAMD.DM[1:0]	01b	Destination update mode is offset addition
DMCRAH, DMCRAL	3	Transfer block size
DMSBSH, DMSBSL	3	Source buffer size (unit is 'data')
DMDBSH, DMDBSL	N	Destination whole buffer size (unit is 'blocks') and Destination access offset (unit is 'data')

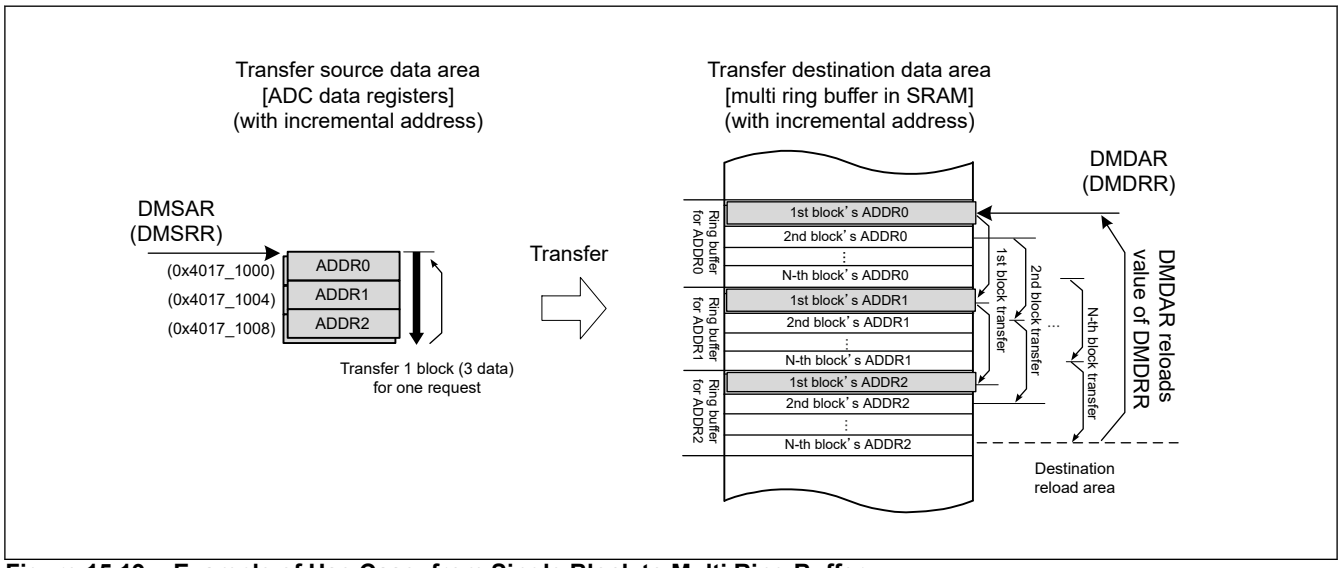


Figure 15.19 Example of Use Case: from Single Block to Multi Ring Buffer

15.3.7 Activation Sources

Software, interrupt requests from the peripheral modules, and external interrupt requests can all be specified as DMAC activation sources. Set the DMTMD.DCTG[1:0] bits to select the activation source.

15.3.7.1 DMAC Activation by Software

When DMA transfer is started by software, follow below procedure.

1. Set the DMTMD.DCTG[1:0] bits to 00b.
2. Set the DMCNT.DTE bit to 1 (DMA transfer is enabled).
3. Set the DMAST.DMST bit set to 1 (DMAC activation enabled).
4. Set the DMREQ.SWREQ bit to 1 (DMA requested).

When the DMAC is activated by software while the DMREQ.CLRS bit is 0, the DMREQ.SWREQ bit is cleared to 0 after data transfer is started in response to a DMA transfer request.

When the DMAC is activated by software while the CLRS bit is 1, the SWREQ bit is not cleared to 0 after data transfer is started. In this case, a DMA transfer request is issued again after completion of a transfer.

15.3.7.2 DMAC Activation through Interrupt Requests from On-Chip Peripheral Modules or External Interrupt Requests

You can specify interrupt requests from on-chip peripheral modules and external interrupt requests as DMAC activation sources. The activation sources can be selected individually for each channel in ICU.DELSRn.DELS[8:0] (n = 0 to 7).

To start DMA transfer through an interrupt request from an on-chip peripheral module or an external interrupt request, follow the procedures as indicated below.

1. Set ICU.DELSRn.DELS[8:0] (n = 0 to 7) to the event number (select the DMAC event link).

2. Set the DMTMD.DCTG[1:0] bits to 01b (interrupts from the peripheral modules and the external interrupt pins).
3. Set the DMCNT.DTE bit to 1 (enable DMA transfer).
4. Set the DMAST.DMST bit set to 1 (DMAC activation enabled).

For interrupt requests specified as DMAC activation sources, see Table 12.3, in section 12, Interrupt Controller Unit (ICU).

15.3.8 Operation Timing

The following timing charts show the minimum number of execution cycles.

Figure 15.20 and Figure 15.21 show DMAC operation timing examples.

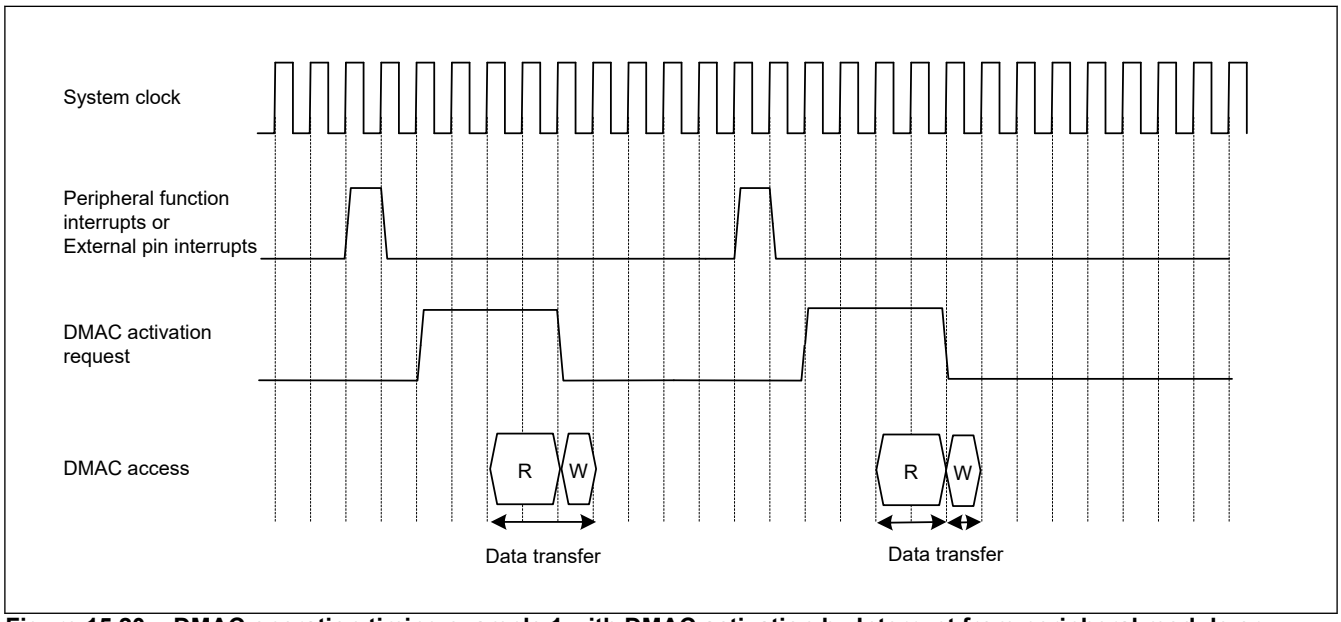


Figure 15.20 DMAC operation timing example 1 with DMAC activation by Interrupt from peripheral module or external interrupt input pin, in normal transfer mode or repeat transfer mode

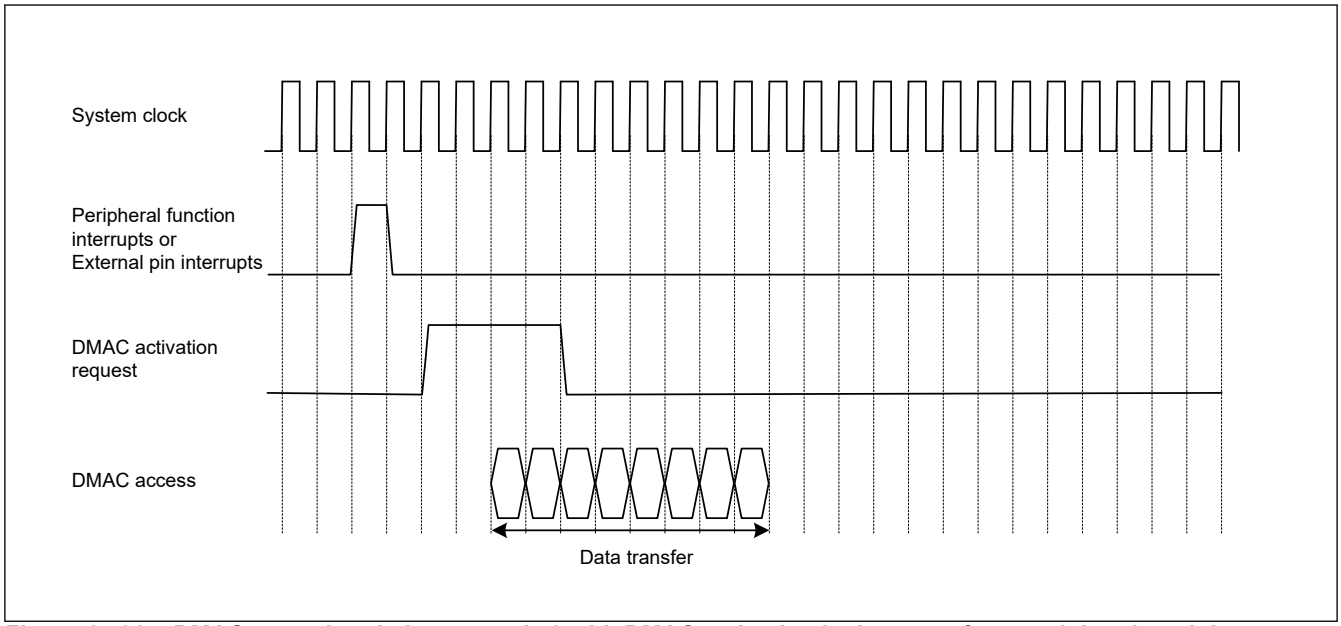


Figure 15.21 DMAC operation timing example 2 with DMAC activation by interrupt from peripheral module or external interrupt input pin, in block transfer mode with block size = 4

15.3.9 DMAC Execution Cycles

Table 15.17 lists execution cycles in one DMAC data transfer operation.

Table 15.17 DMAC execution cycles

Transfer mode	Data transfer (read)	Data transfer (write)
Normal	Cr+1	Cw
Repeat	Cr+1	Cw
Block*1	P × Cr	P × Cw

Note: P: Block size (DMCRAH register setting)
Cr: Data read destination access cycle
Cw: Data write destination access cycle
Note 1. This is the case when the block size is 2 or more. When the block size is 1, normal transfer cycle is applied.

Cr and Cw depend on the access destination. For the number of cycles for each access destination, see section 41, SRAM, section 43, Flash Memory, and section 13, Buses. The frequency ratio of the system clock and the peripheral clock is also taken into consideration.

The unit for +1 in “Data Transfer (Read)” column is one system clock cycle (ICLK). For the operation example, see section 15.3.8. Operation Timing.

15.3.10 Activating the DMAC

Table 15.18 shows the register setting procedure of normal, repeat and block transfer mode and Table 15.19 shows register setting procedure of repeat-block transfer mode.

Table 15.18 Register Setting Procedure of Normal Transfer Mode, Repeat Transfer Mode and Block Transfer Mode (1 of 2)

No.	Step Name	Description
1	Disable the peripheral function as the DMACn request source.	To use peripheral function interrupts as DMAC activation sources. Disable the control register for the peripheral function.
2	Disable the IRQn pin as the DMACn request source.	To use external pin interrupts as DMAC activation sources.
3	Set the DMACn Event Link select (ICU.DELSRn.DELS[8:0]) to 0x00	Disable the DMACn request.
4	Clear the DMCNT.DTE bit to 0	Disable DMA transfer.
5	Set the interrupt request as a DMACn request source in the DMAC Event Link Setting Register (ICU.DELSRn) by using the ICU.	To use internal peripheral interrupts or external pin interrupts as DMAC activation sources. Enable the interrupt bit for the activation source. Set the DMACn activation source.
6	Set the peripheral module as a DMACn request source	To use peripheral function interrupt as a DMAC activation source. Set the control register for the peripheral function without starting it.
7	Set the IRQn pin function by using the ICU.	To use external pin interrupt as a DMAC activation source. Set the IRQn pin function by using the Interrupt Controller Unit.
8	Set the DMAMD.DM[1:0] bits Set the DMAMD.SM[1:0] bits Set the DMAMD.DARA[4:0] bits Set the DMAMD.SARA[4:0] bits	Set the Transfer destination address update mode bits Set the Transfer source address update mode bits Set the Transfer destination address extended repeat area bits Set the Transfer source address extended repeat area bits
9	Set the DMTMD.DCTG[1:0] bits Set the DMTMD.SZ[1:0] bits Set the DMTMD.DTS[1:0] bits Set the DMTMD.MD[1:0] bits Set the DMTMD.TKP bit	Set the Transfer request select bits Set the Data transfer size bits Set the Repeat area select bits Set the Transfer mode select bits Set the transfer keeping select bit
10	Set the DMSAR register Set the DMDAR register Set the DMCRA register	Set the transfer source start address. Set the transfer destination start address. Set the number of transfer operations.
11	Set the DMCRB register	To use block transfer mode or repeat transfer mode. Set the number of block transfer operations.

Table 15.18 Register Setting Procedure of Normal Transfer Mode, Repeat Transfer Mode and Block Transfer Mode (2 of 2)

No.	Step Name	Description
12	Set the DMOFR register	To use the address update function with offset. Set the offset value.
13	Set the DMINT.DTIE bit to 1	To use the DMA transfer end interrupts. Enable DMACn transfer end interrupts.
14	Set the DMINT.RPTIE bit Set the DMINT.SARIE bit Set the DMINT.DARIE bit Set the DMINT.ESIE bit to 1	To use the DMA transfer escape end interrupts Set the repeat size end interrupt. Set the transfer source address extended repeat area overflow interrupt. Set the transfer destination address extended repeat area overflow interrupt. Enable the DMA transfer escape end interrupt.
15	Set the DMCNT.DTE bit to 1	Enable DMA transfer.
16	Set the DMAST.DMST bit to 1	Enable DMAC operation. *1 Common settings for DMAC
17	Start the peripheral function as a DMACn request source	To use peripheral function interrupt as a DMAC activation source
18	Enable the IRQn pin as a DMACn request source	To use external pin interrupt as a DMAC activation source
19	End of initial settings	For activation by software On completion of the initial settings, writing 1 to the DMA software start bit (DMREQ.SWREQ) starts DMA transfer.

Note: n: DMAC channel (n = 0 to 7)
Note 1. The DMAST.DMST bit setting does not necessarily have to follow the settings for the individual activation sources.

Table 15.19 Register Setting Procedure of Repeat-Block Transfer Mode (1 of 2)

No.	Step Name	Description
1	Disable the peripheral function as the DMACn request source.	To use peripheral function interrupts as DMA activation sources. Disable the control register for the peripheral function.
2	Disable the IRQ pin as the DMACn request source.	To use external pin interrupts as DMA activation sources.
3	Set the DMACn Event Link select (ICU.DELSRn.DELS[8:0]) to 00h	Disable the DMACn request.
4	Clear the DMCNT.DTE bit to 0	Disable DMACn transfer.
5	Set the interrupt request as a DMACn request source in the DMACn Event Link Setting Register (ICU.DELSRn) by using the ICU.	To use internal peripheral interrupts or external pin interrupts as DMA activation sources. Enable the interrupt bit for the activation source. Set the DMACn activation source.
6	Set the peripheral module as a DMACn request source	To use peripheral function interrupt as a DMA activation source. Set the control register for the peripheral function without starting it.
7	Set the IRQ pin function by using the Interrupt Controller Unit.	To use external pin interrupt as a DMA activation source. Set the IRQ pin function by using the Interrupt Controller Unit.
8	Set the DMAMD.DM[1:0] bits Set the DMAMD.SM[1:0] bits Set the DMAMD.DADR bit Set the DMAMD.SADR bit	Set the Transfer destination address update mode bits Set the Transfer source address update mode bits Set the Transfer destination address update select after reload Set the Transfer source address update select after reload
9	Set the DMTMD.DCTG[1:0] bits Set the DMTMD.SZ[1:0] bits Set the DMTMD.MD[1:0] bits Set the DMTMD.TKP bit	Set the Transfer request select bits Set the Data transfer size bits Set the Transfer mode to repeat-block transfer mode Set the transfer keeping select bit
10	Set the DMSAR register Set the DMDAR register Set the DMSRR register Set the DMDRR register Set the DMCRA register Set the DMCRB register	Set the transfer source start address Set the transfer destination start address Set the initial value of source start address Set the initial value of destination start address Set the number of transfer operations Set the number of block transfer operations

Table 15.19 Register Setting Procedure of Repeat-Block Transfer Mode (2 of 2)

No.	Step Name	Description
11	Set the DMSBS register Set the DMDBS register	To use the address update function with incremental, decremental or offset Set the source buffer size and access offset Set the destination buffer size and access offset
12	Set the DMINT.DTIE bit to 1	To use DMA transfer end interrupts. Enable DMACn transfer end interrupts.
13	Set the DMCNT.DTE bit to 1	Enable DMACn transfer
14	Set the DMAST.DMST bit to 1	Enable DMAC operation.*1
15	Start the peripheral function as a DMACn request source	To use peripheral function interrupt as a DMA activation source
16	Enable the IRQ pin as a DMACn request source	To use external pin interrupt as a DMA activation source
17	End of initial settings	For activation by software On completion of the initial settings, writing 1 to the DMA software start bit (DMREQ.SWREQ) starts DMA transfer.

Note: m: DELSRn.DELS bit number (m = 0 to 8)
n: DMAC channel (n = 0 to 7)
Note 1. The DMAST.DMST bit setting does not necessarily have to follow the settings for the individual activation sources.

15.3.11 Starting DMA Transfer

To enable the DMA transfer, set the DMCNT.DTE bit to 1 (enable the DMA transfer), and then set the DMAST.DMST bit to 1 (enable the DMAC activation).

New activation requests are not accepted during the transfer of another DMAC channel or DTC. When the preceding transfer is complete, channel arbitration selects the DMA transfer request of the highest priority channel, and the DMA transfer of that channel starts. When the DMA transfer starts, the DMSTS.ACT flag is set to 1 (the DMAC is in the active state).

15.3.12 Registers during DMA Transfer

The DMAC registers are updated by a DMA transfer. The value to be updated differs according to the other settings and the transfer state. The registers to be updated are DMSAR, DMDAR, DMCRA, DMCRB, DMSBS, DMDBS, DMCNT, and DMSTS.

DMA Source Address Register (DMSAR)

When data has been transferred in response to one transfer request, the contents of DMSAR are updated to the address to be accessed by the next transfer request.

For details on register update operation in each transfer mode, see Table 15.5 to Table 15.13.

DMA Destination Address Register (DMDAR)

When data has been transferred in response to one transfer request, the contents of DMDAR are updated to the address to be accessed by the next transfer request.

For details on register update operation in each transfer mode, see Table 15.5 to Table 15.13.

DMA Transfer Count Register (DMCRA)

When data has been transferred in response to one transfer request, the count value is updated. The update operation depends on the transfer mode selected.

For details on register update operation in each transfer mode, see Table 15.5 to Table 15.13.

DMA Block Transfer Count Register (DMCRB)

When data has been transferred in response to one transfer request, the count value is updated. The update operation depends on the transfer mode selected.

For details on register update operation in each transfer mode, see Table 15.5 to Table 15.13.

DMA Source Buffer Size Register (DMSBS)

When data has been transferred in response to one transfer request, the count value is updated. The update operation depends on the transfer mode selected.

For details on register update operation in each transfer mode, see Table 15.8 to Table 15.13.

DMA Destination Buffer Size Register (DMDBS)

When data has been transferred in response to one transfer request, the count value is updated. The update operation depends on the transfer mode selected.

For details on register update operation in each transfer mode, see Table 15.8 to Table 15.13.

DMA Transfer Enable Bit (DMCNT.DTE)

Although the DMCNT.DTE bit enables or disables data transfer by the register write access, it is automatically cleared to 0 by the DMAC according to the DMA transfer state.

The conditions for clearing this bit by the DMAC are as follows:

- When the specified total volume of data transfer is completed
- When DMA transfer is stopped by the repeat size end interrupt
- When DMA transfer is stopped by the extended repeat area overflow interrupt
- When DMA transfer error occurs

Writing to the registers for the channels when the corresponding DMCNT.DTE bit is set to 1 is prohibited (except for DMCNT). In this case, writing must be performed after the bit is cleared to 0.

DMAC Active Flag (DMSTS.ACT)

The DMSTS.ACT flag indicates whether the DMACn is in the idle or active state.

This flag is set to 1 when the DMAC starts data transfer, and is cleared to 0 when data transfer in response to one transfer request is completed.

Even when DMA transfer is stopped by writing 0 to the DMCNT.DTE bit during DMA transfer, this flag remains 1 until DMA transfer is completed.

Transfer End Interrupt Flag (DMSTS.DTIF)

The DMSTS.DTIF flag is set to 1 after DMA transfer of the total transfer size of data is completed.

When both this flag and the DMINT.DTIE bit are set to 1, a transfer end interrupt is requested.

This flag is set to 1 when the DMA transfer bus cycle is completed and the DMSTS.ACT flag is cleared to 0 indicating the DMA transfer end.

This flag is automatically cleared to 0 when the DMCNT.DTE bit is set to 1 during the interrupt handling.

Transfer Escape End Interrupt Flag (DMSTS.ESIF)

The DMSTS.ESIF flag is set to 1 when a repeat size end interrupt or extended repeat area overflow interrupt is requested. When this bit and the DMINT.ESIE bit are set to 1, a transfer escape end interrupt is requested.

This flag is set to 1 when the bus cycle of the DMA transfer having caused the interrupt request is completed and the DMSTS.ACT flag is cleared to 0 indicating the DMA transfer end.

This flag is automatically cleared to 0 when the DMCNT.DTE bit is set to 1 during an interrupt handling.

Before sending an interrupt request from the DMAC to the CPU or the DTC, the interrupt control register must be set.

For details, see section 12, Interrupt Controller Unit (ICU).

15.3.13 Channel Priority

When multiple DMA transfer requests are present, the DMAC determines the priority of channels that have DMA transfer requests.

- The channel priority is fixed as follows: Channel 0 > Channel 1 > Channel 2 > Channel 3… > Channel 7 (Channel 0: Highest).

When a DMA transfer request is generated during data transfer, channel arbitration is started after the final data has been transferred, and DMA transfer of the higher-priority channel starts.

15.3.14 Channel Security

The security attribute of transfer access of DMACn, security attribute of access to register of DMACn, security attribute of access to the ICU.DELSRn register are controlled by ICUSARC.SADMACn bit. For details on the ICUSARC register, see section 12, Interrupt Controller Unit (ICU).

When the ICUSARC.SADMACn bit is 0, transfer of DMACn is secure access for both read and write. At the same time, the registers of channel n and the DELSRn register are protected from a non-secure access.

When the ICUSARC.SADMACn bit is 1, transfer of DMACn is non-secure access for both read and write. At the same time, the registers of channel n and the DELSRn register are non-secure attributes.

Do not write to the ICUSARC.SADMACn bit while DMA transfer of same channel is enabled or a bus master is writing to the DMA registers of same channel.

Figure 15.22 shows security attribute about each DMAC channels.

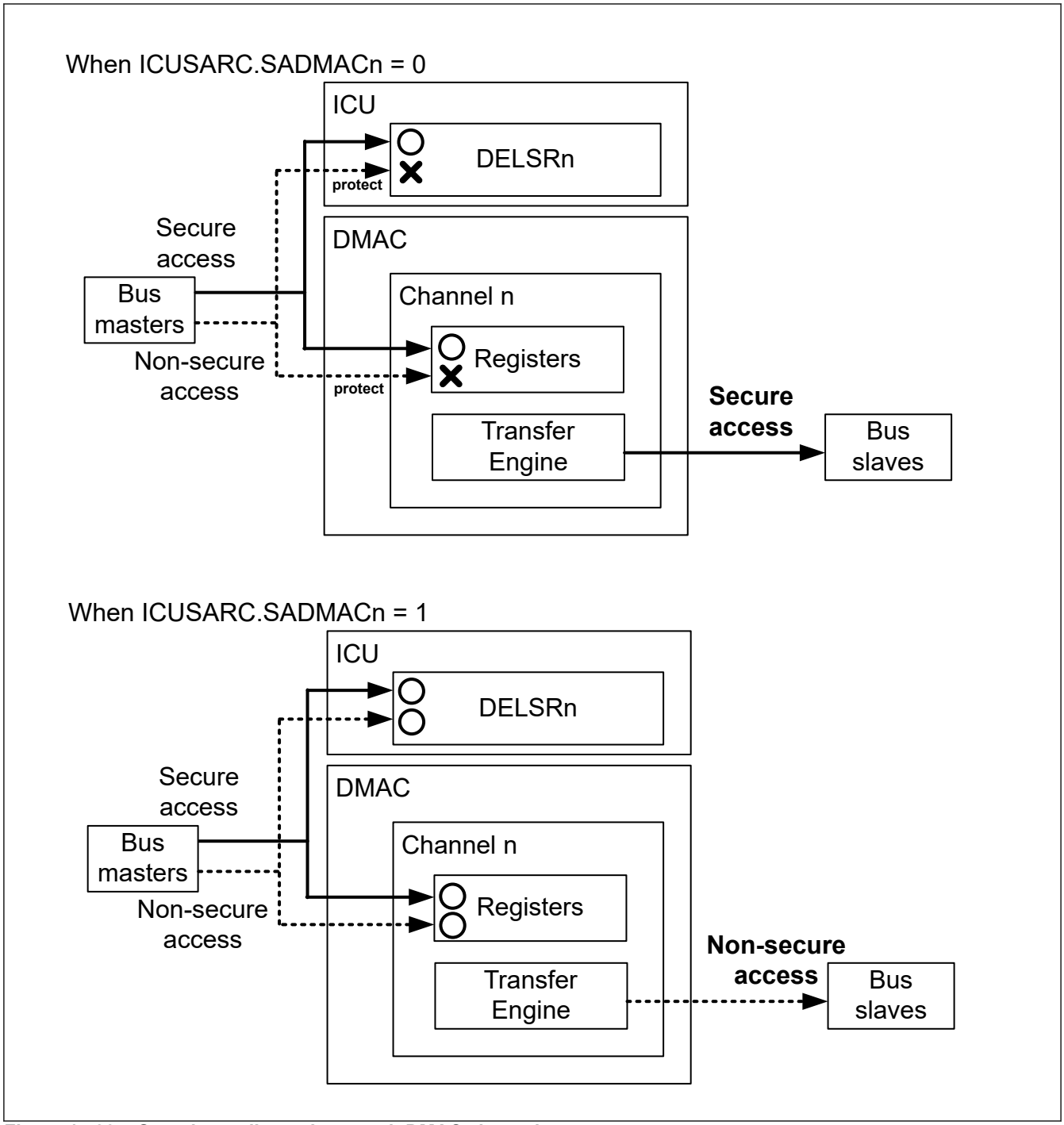


Figure 15.22 Security attribute about each DMAC channels

15.3.15 Master TrustZone Filter in DMAC

DMAC has the Master TrustZone Filter. The MasterTrustZone Filter in DMAC can detect the security areas of Flash area(code Flash and data Flash) and SRAM area defined by IDAU. When set No-secure channel accesses those addresses, it detects the security violation. Access of violation address is not performed. Detected the error is handled as the Master TrustZone Filter error.

15.4 Ending DMA Transfer

The operation for ending DMA transfer depends on the transfer end conditions. When DMA transfer ends, the DMCNT.DTE bit and the DMSTS.ACT flag are changed from 1 to 0, indicating that DMA transfer has ended.

15.4.1 Transfer End by Completion of Specified Total Number of Transfer Operations

(1) In Normal Transfer Mode (DMTMD.MD[1:0] = 00b)

When the value of DMCRAL changes from 1 to 0, DMA transfer ends on the corresponding channel, and the DMCNT.DTE bit is cleared to 0 and the DMSTS.DTIF flag is set to 1 at the same time. If the DMINT.DTIE bit is 1 at this time, a transfer end interrupt request is issued to the CPU or the DTC.

(2) In Repeat Transfer Mode (DMTMD.MD[1:0] = 01b)

When the value of DMCRBL changes from 1 to 0, DMA transfer ends on the corresponding channel, and the DMCNT.DTE bit is cleared to 0 and the DMSTS.DTIF flag is set to 1 at the same time. If the DMINT.DTIE bit is 1 at this time, an interrupt request is issued to the CPU or the DTC.

If the DMTMD.TKP bit is 1 (in free-running function), the DMSTS.DTIF bit is set to 1, but the DMCNT.DTE bit is not cleared to 0.

(3) In Block Transfer Mode (DMTMD.MD[1:0] = 10b)

When the value of DMCRBL changes from 1 to 0, DMA transfer ends on the corresponding channel, and the DMCNT.DTE bit is cleared to 0 and the DMSTS.DTIF flag is set to 1 at the same time. If the DMINT.DTIE bit is 1 at this time, an interrupt request is issued to the CPU or the DTC.

Before sending an interrupt request from the DMAC to the CPU or the DTC, the interrupt control register must be set.

For details, see section 12, Interrupt Controller Unit (ICU).

If the DMTMD.TKP bit is 1 (in free-running function),the DMSTS.DTIF bit is set to 1, but the DMCNT.DTE bit is not cleared to 0.

(4) In Repeat-Block Transfer Mode (DMTMD.MD[1:0] = 11b)

When the value of DMCRBL changes from 1 to 0, DMA transfer ends on the corresponding channel, and the DMCNT.DTE bit is cleared to 0 and the DMSTS.DTIF flag is set to 1 at the same time. If the DMINT.DTIE bit is 1 at this time, an interrupt request is issued to the CPU or the DTC.

Before sending an interrupt request from the DMAC to the CPU or the DTC, the interrupt control register must be set. For details, see section 12, Interrupt Controller Unit (ICU).

If the DMTMD.TKP bit is 1 (in free-running function),the DMSTS.DTIF bit is set to 1, but the DMCNT.DTE bit is not cleared to 0.

15.4.2 Transfer End by Repeat Size End Interrupt

In repeat transfer mode, a repeat size end interrupt is requested when transfer of a 1-repeat size of data is completed while the DMINT.RPTIE bit is set to 1. When the interrupt is requested to complete DMA transfer, the DMCNT.DTE bit is cleared to 0 and the DMSTS.ESIF flag is set to 1 even if the DMTMD.TKP bit is 1 (in free-running function). If the DMINT.ESIE bit is 1 at this time, an interrupt request is issued to the CPU or the DTC. Here, the transfer can be resumed by writing 1 to the DMCNT.DTE bit.

A repeat size end interrupt can be requested also in block transfer mode. In block transfer mode, the interrupt is requested in the same way as in repeat transfer mode when transfer of a 1-block size data is completed.

Repeat size end interrupt cannot be requested in repeat-block transfer mode.

Before sending an interrupt request from the DMAC to the CPU or the DTC, the interrupt control register must be set. For details, see section 12, Interrupt Controller Unit (ICU).

15.4.3 Transfer End by Interrupt on Extended Repeat Area Overflow

When an overflow on the extended repeat area occurs while the extended repeat area is specified and the DMINT.SARIE or DMINT.DARIE bit is set to 1 even if the DMTMD.TKP bit is 1 (in free-running function), an interrupt by an extended repeat area overflow is requested. When the interrupt is requested, the DMA transfer is terminated, the DMCNT.DTE bit is cleared to 0, and the ESIF flag in DMSTS is set to 1. If the DMINT.ESIE bit is 1 at this time, an interrupt request is issued to the CPU or the DTC.

Even if an interrupt by an extended repeat area overflow is requested during a read cycle, the following write cycle is performed.

In block transfer mode, even if an interrupt by an extended repeat area overflow is requested during a 1-block transfer, the remaining data in the block is transferred; transfer is terminated after a block transfer.

An interrupt by an extended repeat area overflow cannot be requested in repeat-block transfer mode.

Before sending an interrupt request from the DMAC to the CPU or the DTC, the interrupt control register must be set. For details, see section 12, Interrupt Controller Unit (ICU).

15.5 Processing on DMA Transfer Error

DMA transfer error occurs with the Master TrustZone Filter error in DMAC, the Slave TrustZone Filter error, the Master MPU error, the Slave Bus Error or the Illegal Access Error. If the access error occurs during the DMA transfer, the DMAC immediately stops the transfer of error occurred channel. At this time, the ICU setting of the corresponding channel is also cleared. If there is a request other than the channel which caused the error, it will be re-arbitration as it is.

When the transfer error occurs, DMCNT.DTE of the error causing channel is set to 0. Also, the error response is informed to the ICU.DELSRn of the corresponding channel is cleared. Write back to each register is not performed. Furthermore, it generates the error response detection interrupt request (DMA_TRANSERR) to notify that an error has occurred by DMAC/DTC transfer.

When the Master TrustZone Filter error occurs, the Slave TrustZone error occurs or the Master MPU error occurs, it is possible to confirm the error information of DMAC by selecting NMI. The DMAC error channel register is cleared by selecting reset. Under the conditions where NMI is generated due to transfer error in DMAC, two interrupts(NMI and DMA_TRANSERR) are generated. In this case, NMI always responds first.

The error response detection interrupt request (DMA_TRANSERR) occurs when the Slave Bus error or the Illegal Access error occurs. Furthermore, it occurs after NMI when the error response detection interrupt request (DMA_TRANSERR) is not cleared in NMI handler.

section 15.5.1. Processing on NMI handler describes how to confirm the error information of the DMAC in the NMI handler.

section 15.5.2. Processing on Error response detection interrupt request (DMA_TRANSERR) handler describes how to confirm the error information of the DMAC in the DMA_TRANSERR handler.

Interrupts and the error information generated due to transfer errors are shown in section 15.6.2. Transfer Error Interrupt.

15.5.1 Processing on NMI handler

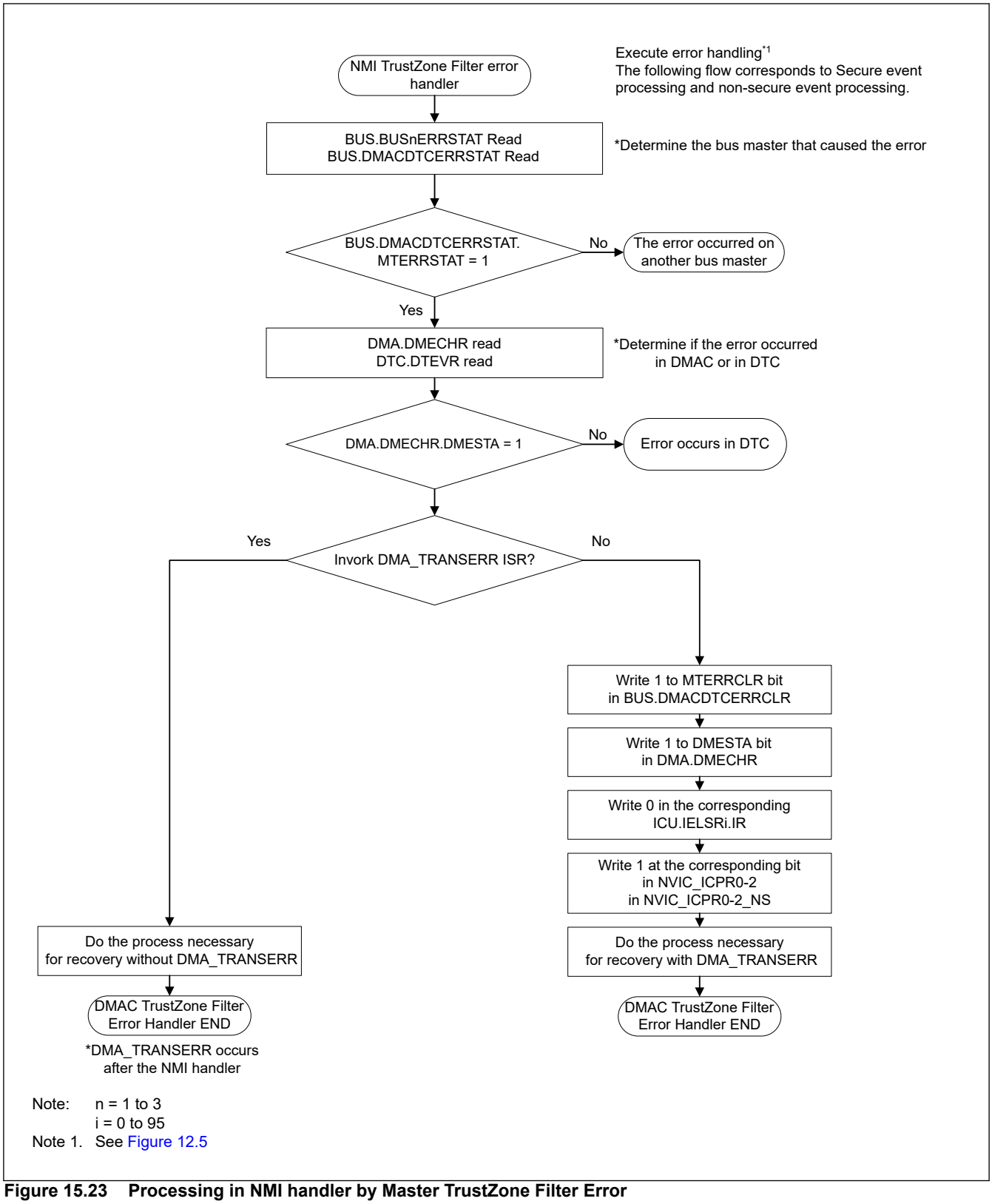
The cause of NMI due to the DMA transfer error is the Master TrustZone Filter error, the Slave TrustZone Filter error or the Master MPU error. When NMI occurs due to the DMAC transfer error, the error response detection interrupt request (DMA_TRANSERR) will occur after the end of NMI handler. It is possible to confirm the cause of the error and the DMAC channel in which the error occurred. When NMI occurs, perform the necessary processing according to the flow described in the ICU chapter.

Figure 15.23 shows the flow for confirm the channel that caused the Master TrustZone Filter Error in DMAC

Figure 15.24 shows the flow for confirm the channel that caused the Slave TrustZone Filter Error in DMAC

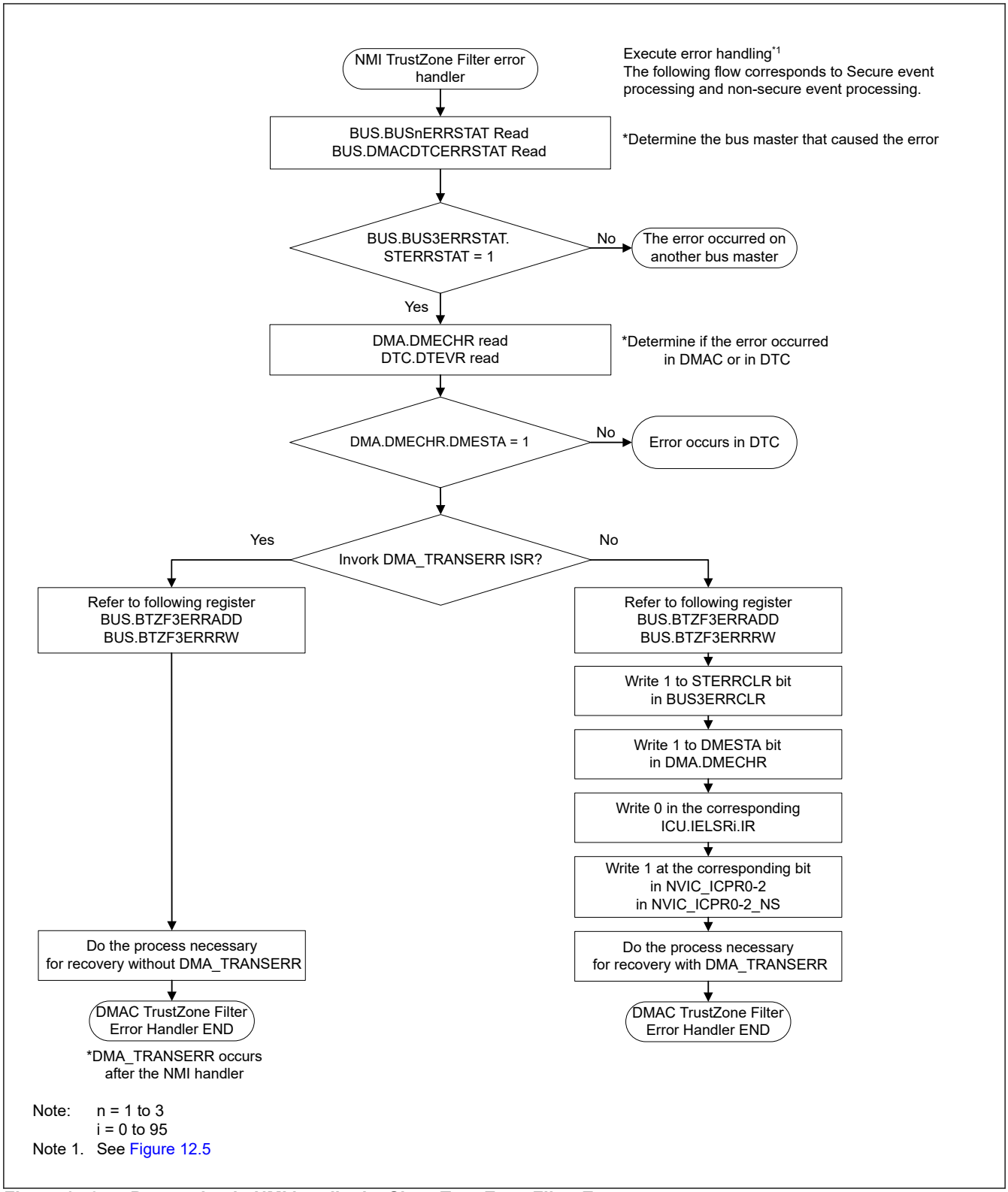
Figure 15.25 shows the flow for confirm the channel and Security Attribute that caused the Master MPU error in DMAC.

If completing all processing in NMI handler, it is possible to clear the error response detection interrupt request (DMA_TRANSERR) that occurs subsequently.



Note: n = 1 to 3
i = 0 to 95
Note 1. See Figure 12.5

Figure 15.23 Processing in NMI handler by Master TrustZone Filter Error



Note: n = 1 to 3
i = 0 to 95
Note 1. See Figure 12.5

Figure 15.24 Processing in NMI handler by Slave TrustZone Filter Error

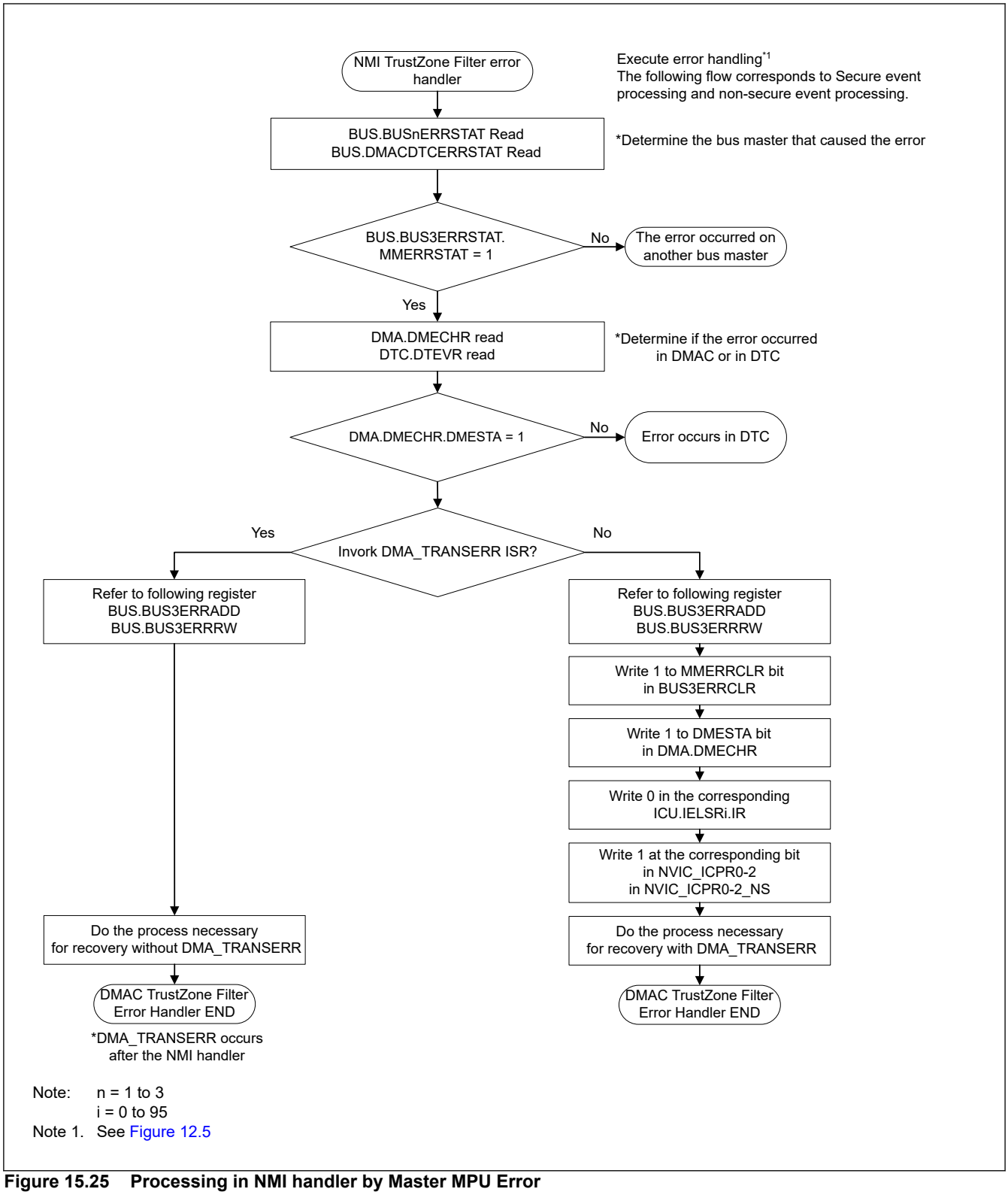


Figure 15.25 Processing in NMI handler by Master MPU Error

15.5.2 Processing on Error response detection interrupt request (DMA_TRANSERR) handler

The cause of error response detection interrupt request (DMA_TRANSERR) due to DMA transfer error is the Slave Bus Error or Illegal Access Error. Also, it occurs after the NMI handler error response detection interrupt request (DMA_TRANSERR) is not cleared by the NMI handler.

It is possible to confirm the cause of the error and the DMAC channel in which the error occurred.

Error cause confirmation procedure is shown Figure 15.26.

Figure 15.27 shows the flow for confirm the channel that caused the Master TrustZone Filter Error in DMAC

Figure 15.28 shows the flow for confirm the channel that caused the Slave TrustZone Filter Error in DMAC

Figure 15.29 shows the flow for confirm the channel and Security Attribute that caused the Master MPU Error in DMAC

Figure 15.30 shows the flow for confirm the channel and Security Attribute that caused the Slave Bus Error in DMAC

Figure 15.31 shows the flow for confirm the channel and Security Attribute that caused the Illegal Access Error in DMAC

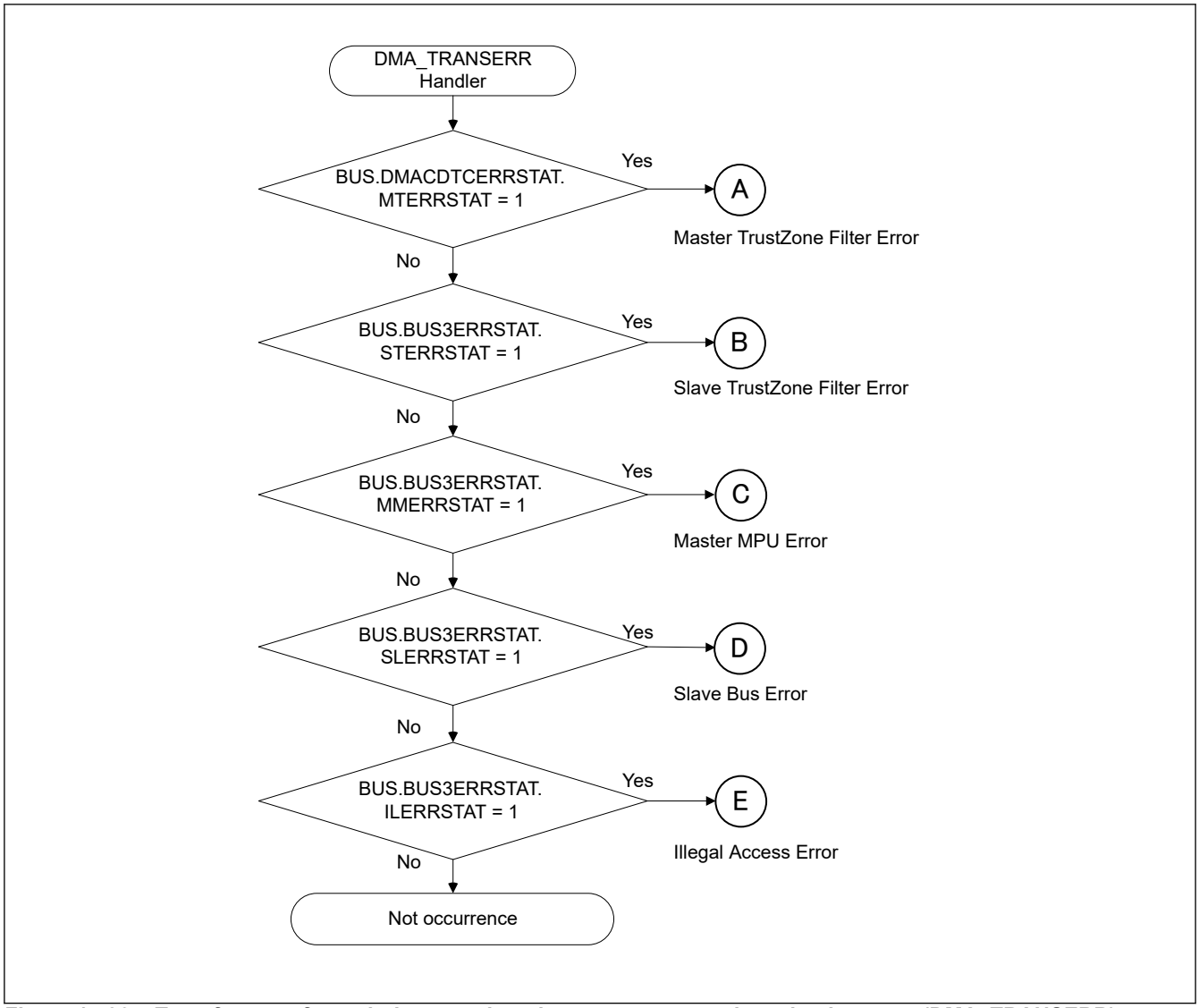


Figure 15.26 Transfer error factor judgment when the error response detection interrupt (DMA_TRANSERR) occurs

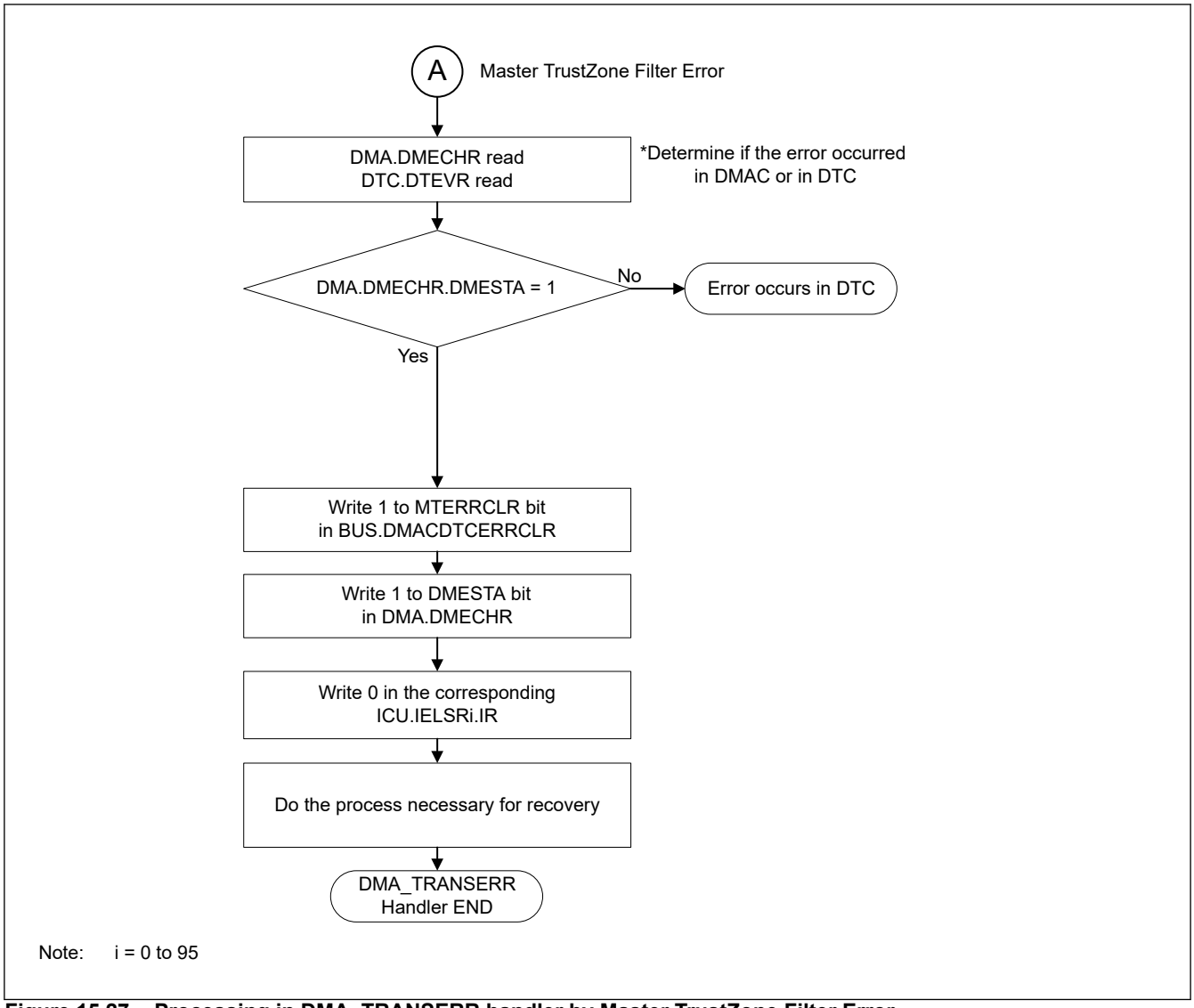


Figure 15.27 Processing in DMA_TRANSERR handler by Master TrustZone Filter Error

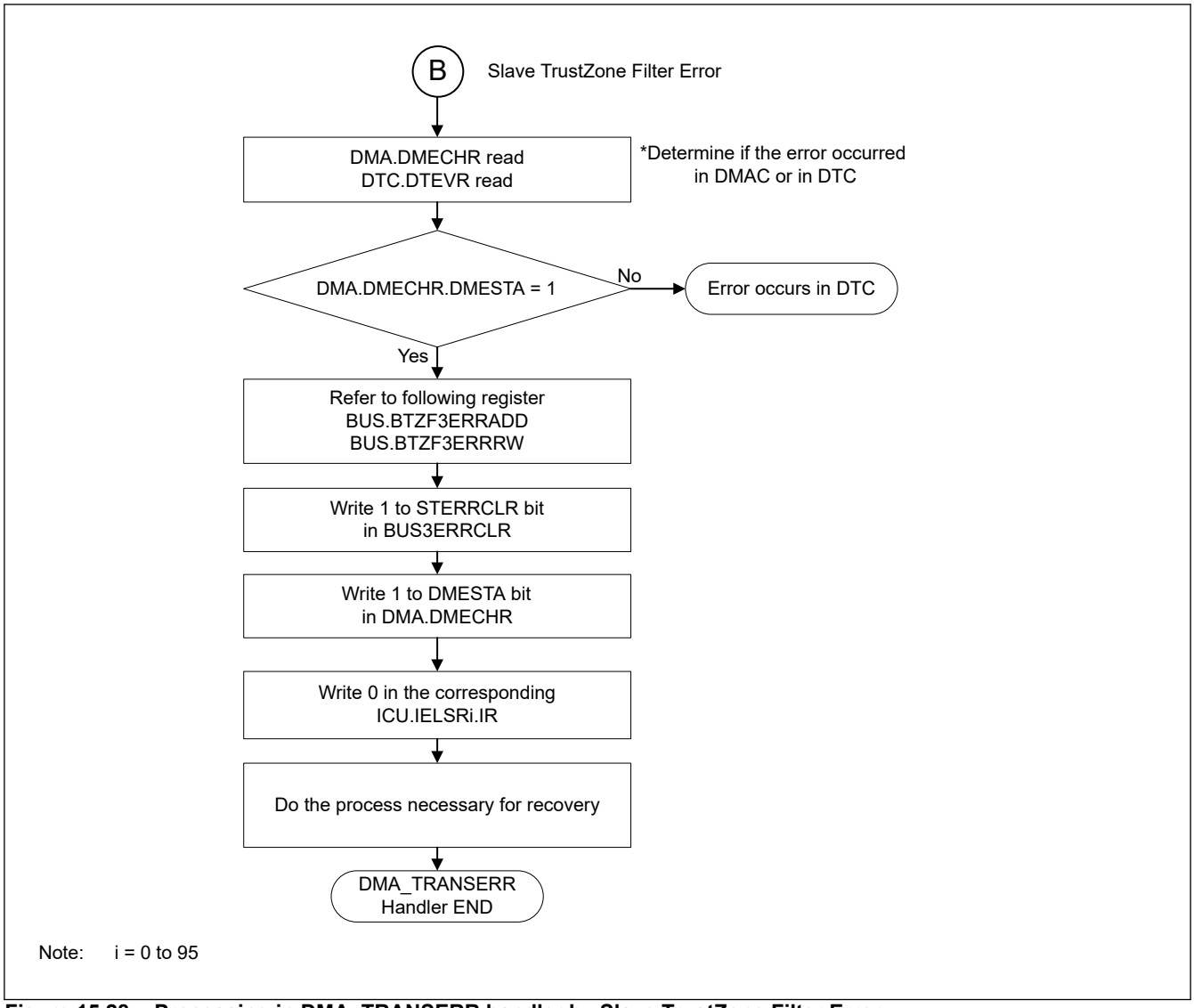
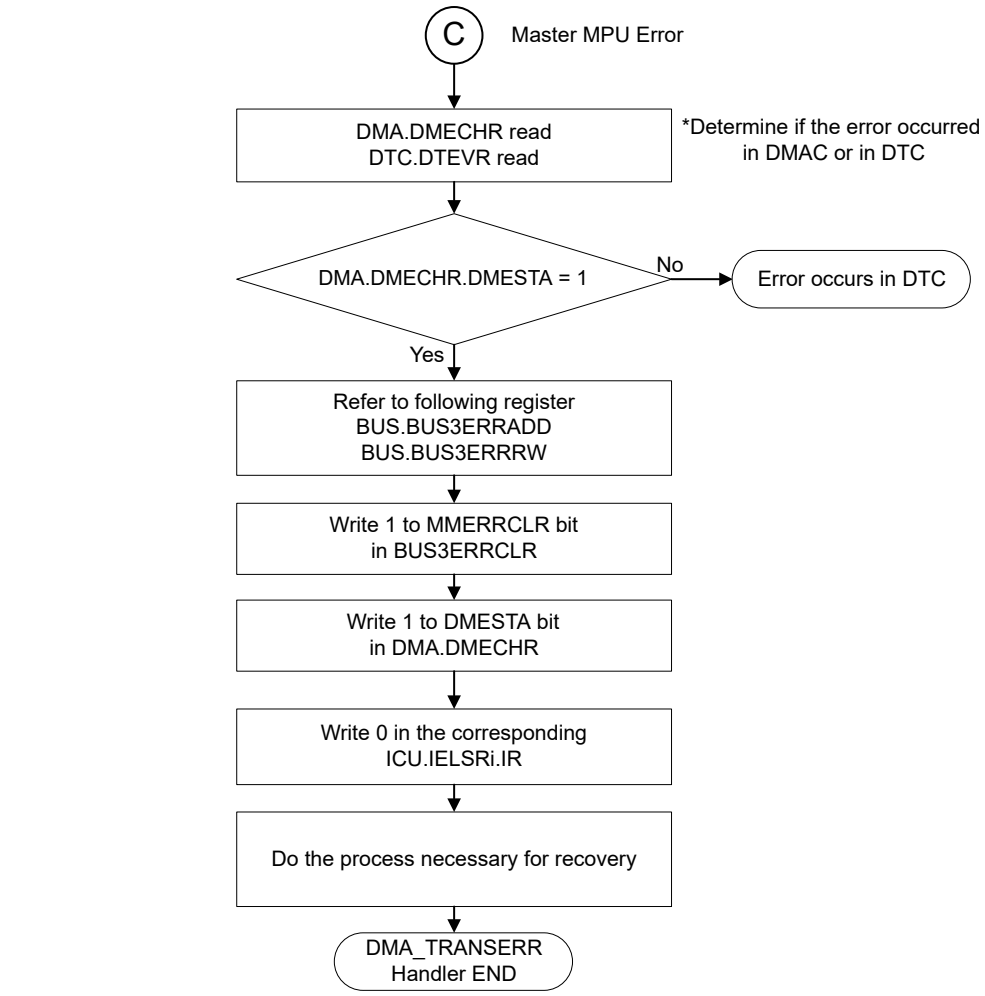


Figure 15.28 Processing in DMA_TRANSERR handler by Slave TrustZone Filter Error



Note: i = 0 to 95

Figure 15.29 Processing in DMA_TRANSERR handler by Master MPU Error

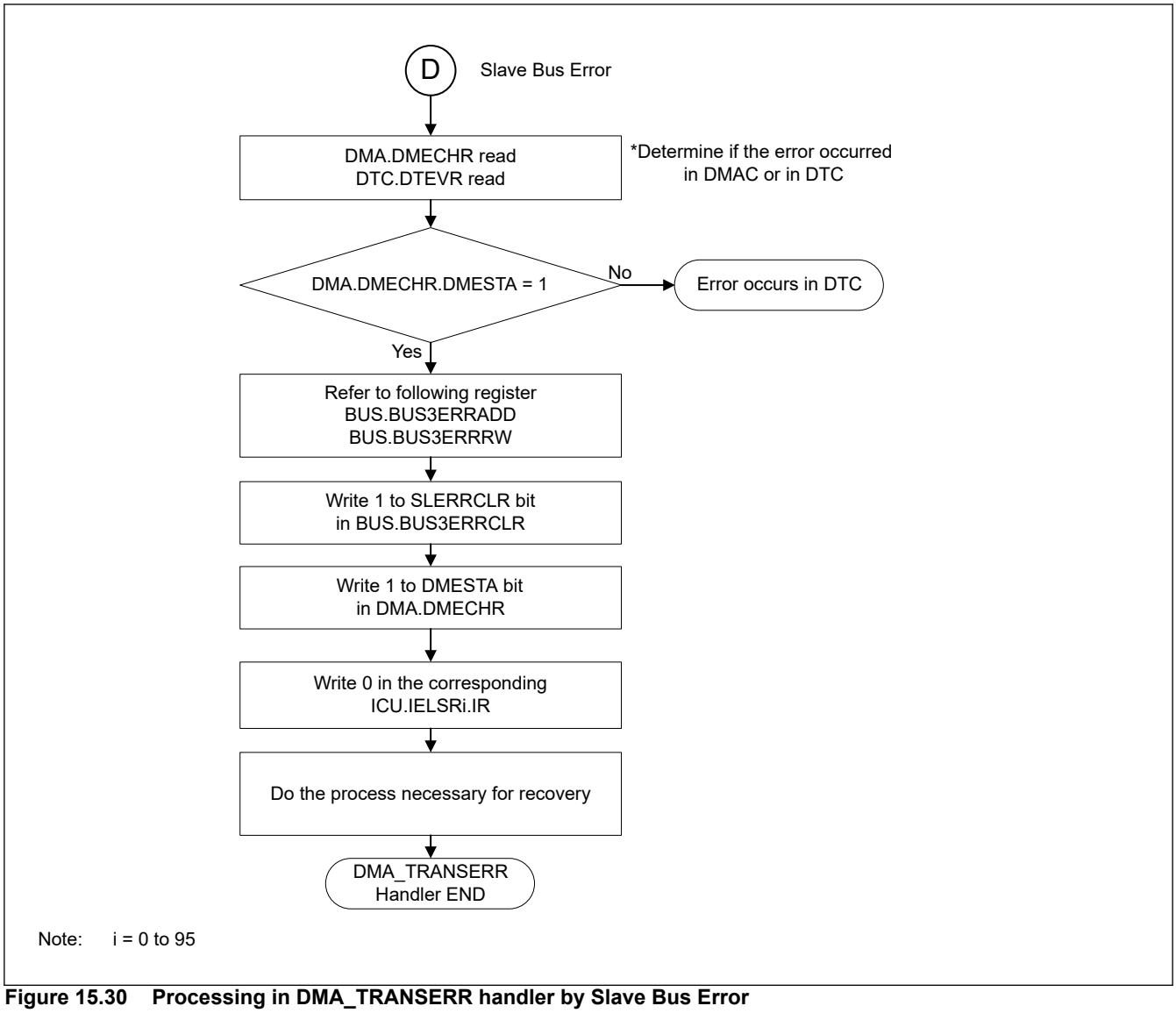


Figure 15.30 Processing in DMA_TRANSERR handler by Slave Bus Error

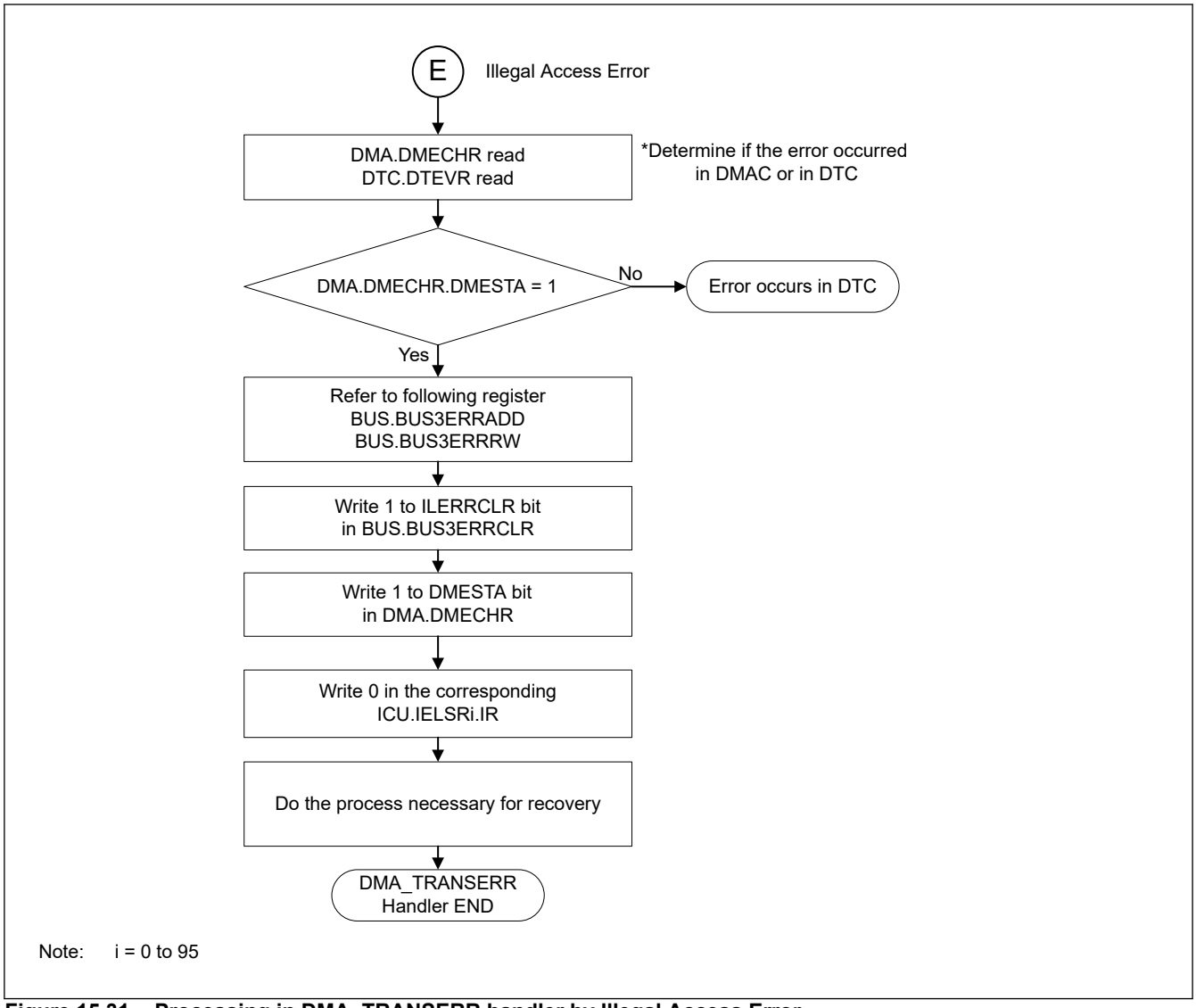


Figure 15.31 Processing in DMA_TRANSERR handler by Illegal Access Error

15.6 Interrupts

15.6.1 Transfer End Interrupt

Each DMAC channel can output an interrupt request (DMACn_INT) to the CPU or the DTC after transfer in response to one request is completed.

In repeat-block transfer mode, do not enable escape transfer end interrupt.

Table 15.20 lists the relation among the interrupt sources, the interrupt status flags, and the interrupt enable bits. Figure 15.32 shows the schematic logic diagram of interrupt outputs (DMACn (n = 0 to 7)). Figure 15.33 shows the DMAC interrupt handling routine to resume/terminate DMA transfer.

Table 15.20 Relation among interrupt sources, interrupt status flags, and interrupt enable bits (1 of 2)

Interrupt sources	Interrupt enable bits	Interrupt status flags	Request output enable bits
Transfer end	—	DMSTS.DTIF	DMINT.DTIE

Table 15.20 Relation among interrupt sources, interrupt status flags, and interrupt enable bits (2 of 2)

Interrupt sources		Interrupt enable bits	Interrupt status flags	Request output enable bits
Escape transfer end	Repeat size end	DMINT.RPTIE	DMSTS.ESIF	DMINT.ESIE
	Source address extended repeat area overflow	DMINT.SARIE		
	Destination address extended repeat area overflow	DMINT.DARIE		

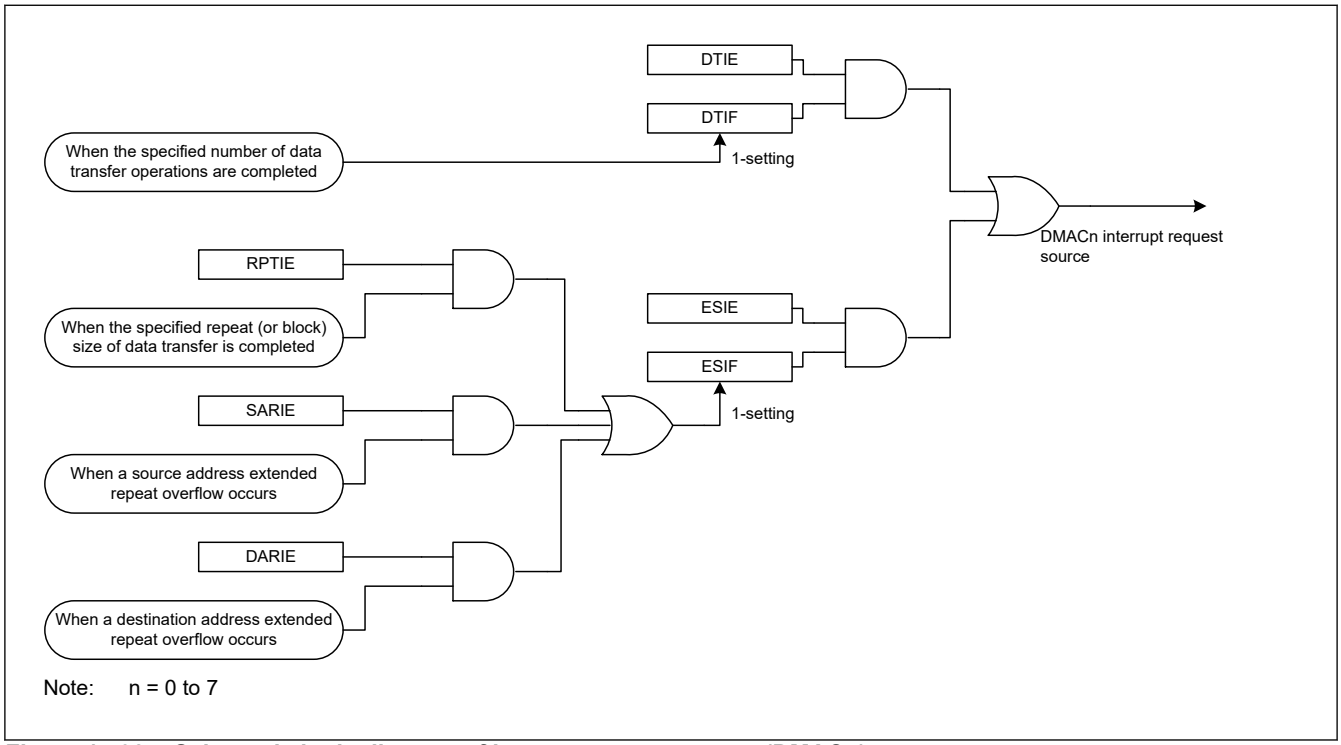


Figure 15.32 Schematic logic diagram of interrupt output source (DMACn)

Specifically, the different procedures are used for canceling an interrupt to restart DMA transfer in the following two cases:

- When terminating a DMA transfer
- When continuing a DMA transfer

15.6.1.1 When Terminating a DMA Transfer

Write 0 to the DMSTS.DTIF flag to clear a transfer end interrupt, and to the DMSTS.ESIF flag to clear a repeat size interrupt and an extended repeat area overflow interrupt. The DMACn remains in the stop state. When starting another DMA transfer after that, set the appropriate registers, and set the DMCNT.DTE bit to 1 (DMA transfer enabled).

15.6.1.2 When Continuing a DMA Transfer

Write 1 to the DMCNT.DTE bit. The DMSTS.ESIF flag is automatically cleared to 0 (interrupt source cleared), and DMA transfer is resumed.

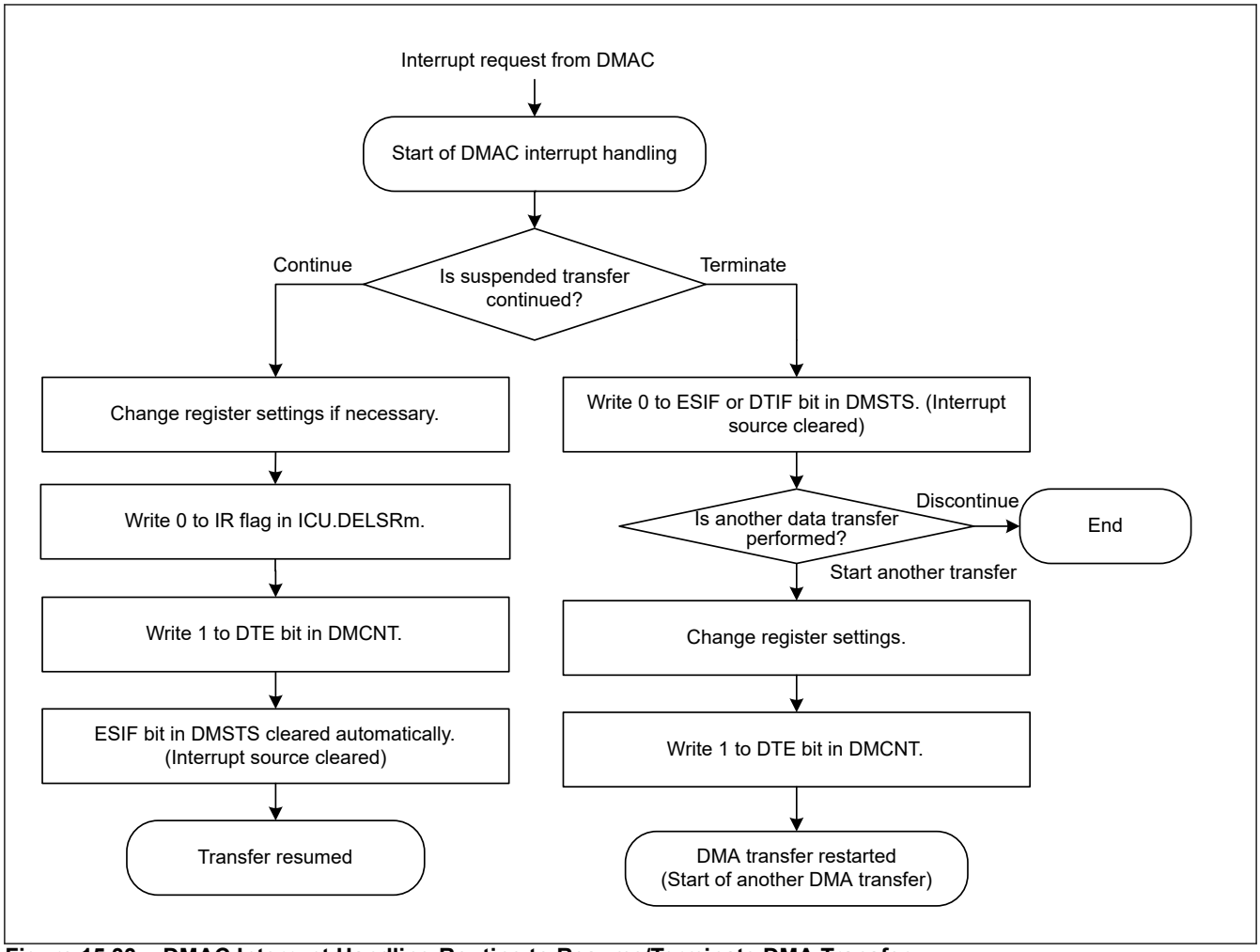


Figure 15.33 DMAC Interrupt Handling Routine to Resume/Terminate DMA Transfer

15.6.2 Transfer Error Interrupt

Error response detection interrupt request (DMA_TRANSERR) is generated from the DMAC/DTC when the transfer error is detected during DMAC transfer. The types of interrupts that occur when a DMAC transfer error occurs are listed in the Table 15.21. The Table 15.21 also shows error information stored when a transfer error occurs.

Table 15.21 Interrupt and error information due to DMAC transfer error cause

Transfer error factor	NMI/RESET*1 Request	Interrupt Request	Bus Error Status	Error Address Error R/W	Error Channel Information
Master TrustZone Filter (in DMAC/DTC)	ICU.NMISR.TZFST*1	DMA_TRANSERR	BUS.DMACDTCERRSTAT.MTERRSTAT*1	—	DMA.DMECHR
Slave TrustZone Filter	ICU.NMISR.TZFST*1	DMA_TRANSERR	BUS.BUS3ERRSTAT.STERRSTAT*1	BUS.BTZF3ERRADD BUS.BTZF3ERRRW	DMA.DMECHR
Master MPU	ICU.NMISR. BUSMST	DMA_TRANSERR	BUS.BUS3ERRSTAT.MMERRSTAT	BUS.BUS3ERRADD BUS.BUS3ERRRW	DMA.DMECHR
Slave Bus Error	—*2	DMA_TRANSERR	BUS.BUS3ERRSTAT.SLERRSTAT*2	BUS.BUS3ERRADD BUS.BUS3ERRRW	DMA.DMECHR
Illegal Access Error	—*2	DMA_TRANSERR	BUS.BUS3ERRSTAT.ILERRSTAT*2	BUS.BUS3ERRADD BUS.BUS3ERRRW	DMA.DMECHR

Note 1. Interrupt generated, when NMI request selected as the operation after detection of the Master MPU error and the TrustZone Filter error. By confirming BUS.BUS3ERRSTAT and BUS.DMACDTCERRSTAT, judge whether it is the Master or the Slave.

Note 2. If the error response detection interrupt (DMA_TRANSERR) occurs and NMI of the Master MPU or NMI of the TrustZone Filter has not occurred, treat it as the Illegal address access error or the Slave Bus Error. It can be judged also by BUS.BUS3ERRSTAT and BUS. DMACDTCERRSTAT.

Note that if the bus error occurs when writing the last data of transfer, the transfer end event and the error response detection interrupt (DMA_TRANSERR) occurs.

15.7 Event Link

Each DMAC channel outputs an event link request signal (DMACn_INT) every time it completes a data transfer, or a block transfer in block transfer mode.

For details, see section 17, Event Link Controller (ELC).

If a bus error occurs when writing the last data of transfer, a transfer end event and error response detection interrupt (DMA_TRANSERR) occurs.

15.8 Low-Power Consumption Function

Before entering the module-stop state or Software Standby mode, or Deep Software Standby mode, you must first set the DMAST.DMST bit to 0 (the DMAC module suspended) and use the settings in the sections that follow.

(1) Module-stop function

Writing 1 to the MSTPCRA.MSTPA22 bit enables the module-stop function of the DMAC. If a DMA transfer is in progress when 1 is written to the MSTPA22 bit, the transition to the module-stop state proceeds after the DMA transfer ends. Access to the DMAC registers is prohibited while the MSTPA22 bit is 1. Writing 0 to the MSTPA22 bit releases the DMAC from the module-stop state.

(2) Software Standby mode and Deep Software Standby mode

Use the settings described in section 10.7.1. Transitioning to Software Standby Mode, or in section 10.9.1. Transitioning to Deep Software Standby Mode .

If DMA transfer operations are in progress when the WFI instruction is executed, the DMA transfer completes before the transition to Software Standby mode or Deep Software Standby mode.

(3) Notes on low power consumption function

For information on the WFI instruction and register settings, see section 10.10.7. Timing of WFI Instruction.

To perform a DMA transfer after returning from a low power mode, set the DMAST.DMST bit to 1 again. To use a request that is generated in Software Standby mode as an interrupt request to the CPU but not as a DMAC startup request, specify the CPU as the interrupt request destination, as described in section 12.4.1. Detecting Interrupts, and then execute the WFI instruction.

15.9 Usage Notes

15.9.1 Access to the Registers during DMA Transfer

Do not write to the following registers while the DMSTS.ACT flag of the same channel is set to 1 (DMAC active state) or the DMCNT.DTE bit of the same channel is set to 1 (DMA transfer enabled):

- DMSAR
- DMDAR
- DMCRA
- DMCRB
- DMTMD
- DMINT
- DMAMD
- DMOFR
- DMSBS
- DMDBS
- DMSRR

- DMDRR
- ICUSARC
- DMACSAR

15.9.2 DMA Transfer to Reserved Areas

DMA transfer to the reserved areas is prohibited. If such an access is made, transfer results are not guaranteed. For details on the reserved areas, see section 4, Address Space.

15.9.3 Setting of DMAC Event Link Setting Register of the Interrupt Controller Unit (ICU.DELSRn n = 0 to 7)

The DMAC event link setting register (ICU.DELSRn) should be set while the DMA transfer enable bit (DMCNT.DTE) is cleared to 0 (DMA transfer is disabled). Moreover, the DTC activation enable register (ICU.IELSRn.DTCE (n = 0 to 95)) that corresponds to the same event number that has been set by the ICU.DELSRn register should not be set to 1. For details on the ICU.IELSRn.DTCE and ICU.DELSRn, see section 12, Interrupt Controller Unit (ICU).

15.9.4 Suspending or Restarting DMAC Activation

To suspend a DMAC activation request, write 0x00 to the DMAC Event Link select bits (ICU.DELSRn.DELS[8:0]). To restart the DMA transfer, write the event number to the ICU.DELSRn.DELS[8:0] bits following the settings shown in section 15.3.10. Activating the DMAC.

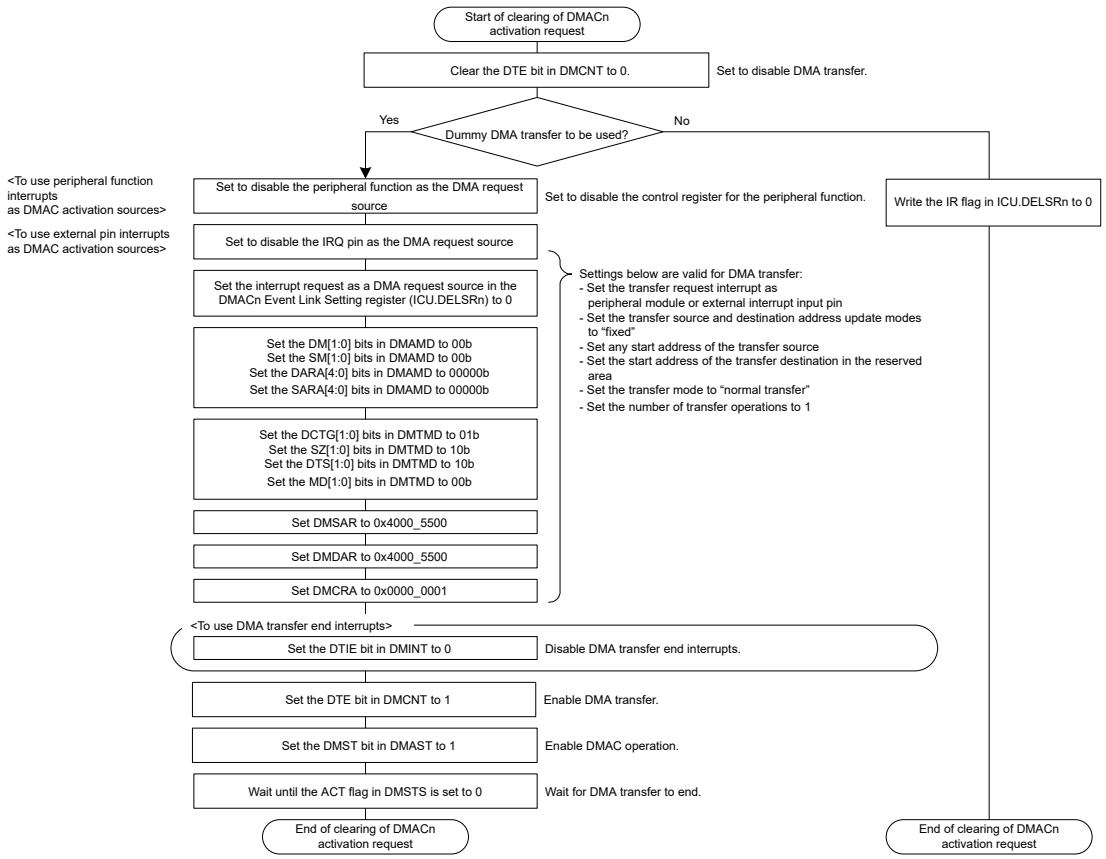
15.9.5 Precautions for Resuming DMA Transfer

A DMAC activation request might occur in the next request after a DMA transfer completes. If this happens, the DMA transfer starts and the DMAC activation request is held in the DMAC. To prevent this, stop the DMAC activation requests by setting the DELSRn.DELS[8:0] bits in the ICU to 0.

When a DMAC activation request occurs after the last round of the DMA transfer is generated, clear the DMAC activation request with either of the following approaches.

- Clear the DMAC activation request with a DMA dummy transfer.
- Set the DMCNT.DTE bit to 0 and then set the ICU.DELSRn.IR flag to 0.

See Figure 15.34.



Note: n: 0 to 7

Figure 15.34 Example of register setting procedure to clear the DMAC activation interrupt

16. Data Transfer Controller (DTC)

16.1 Overview

A Data Transfer Controller (DTC) module is provided for transferring data when activated by an interrupt request.

Table 16.1 lists the DTC specifications and Figure 16.1 shows DTC block diagram.

Table 16.1 DTC specifications

Parameter	Description
Transfer modes	• Normal transfer mode A single activation leads to a single data transfer. • Repeat transfer mode A single activation leads to a single data transfer. The transfer address returns to the start address after the number of data transfers reaches the specified repeat size. The maximum number of repeat transfers is 256 and the maximum data transfer size is 256 × 32 bits (1024 bytes) • Block transfer mode A single activation leads to a transfer of a single block. The maximum block size is 256 × 32 bits = 1024 bytes.
Transfer channel	• Channel transfer can be associated with the interrupt source (transferred by a DTC activation request from the ICU) • Multiple data units can be transferred on a single activation source (chain transfer) • Chain transfers are selectable to either execute when the counter is 0, or always execute.
Transfer space	• 4 GB area from 0x0000_0000 to 0xFFFF_FFFF, excluding reserved areas
Data transfer units	• Single data unit: 1 byte (8 bits), 1 halfword (16 bits), 1 word (32 bits) • Single block size: 1 to 256 data units.
CPU interrupt source	• An interrupt request can be generated to the CPU on a DTC activation interrupt • An interrupt request can be generated to the CPU after a single data transfer • An interrupt request can be generated to the CPU after a data transfer of a specified volume.
Processing on DTC transfer error	• When the DTC transfer error occurs, it stops the transfer that caused the error • Request to clear the register for activation request of DTC error number to ICU
Error response detection interrupt	Generated when the DTC transfer error occurs
Event link function	An event link request is generated after one data transfer (for block, after one block transfer)
Read skip	Read of transfer information can be skipped
Write-back skip	When the transfer source or destination address is specified as fixed, a write-back of transfer information can be skipped
TrustZone	TrustZone violation area of Flash and SRAM is detected in advance before access the bus.
Module-stop function	Module-stop state can be set to reduce power consumption
TrustZone Filter	Security attribution can be set for each activation source

Note: Security attribution Regiser of DTC is described in ICU.ICUSARG, ICU.ICUSARH and ICU.ICUSARI

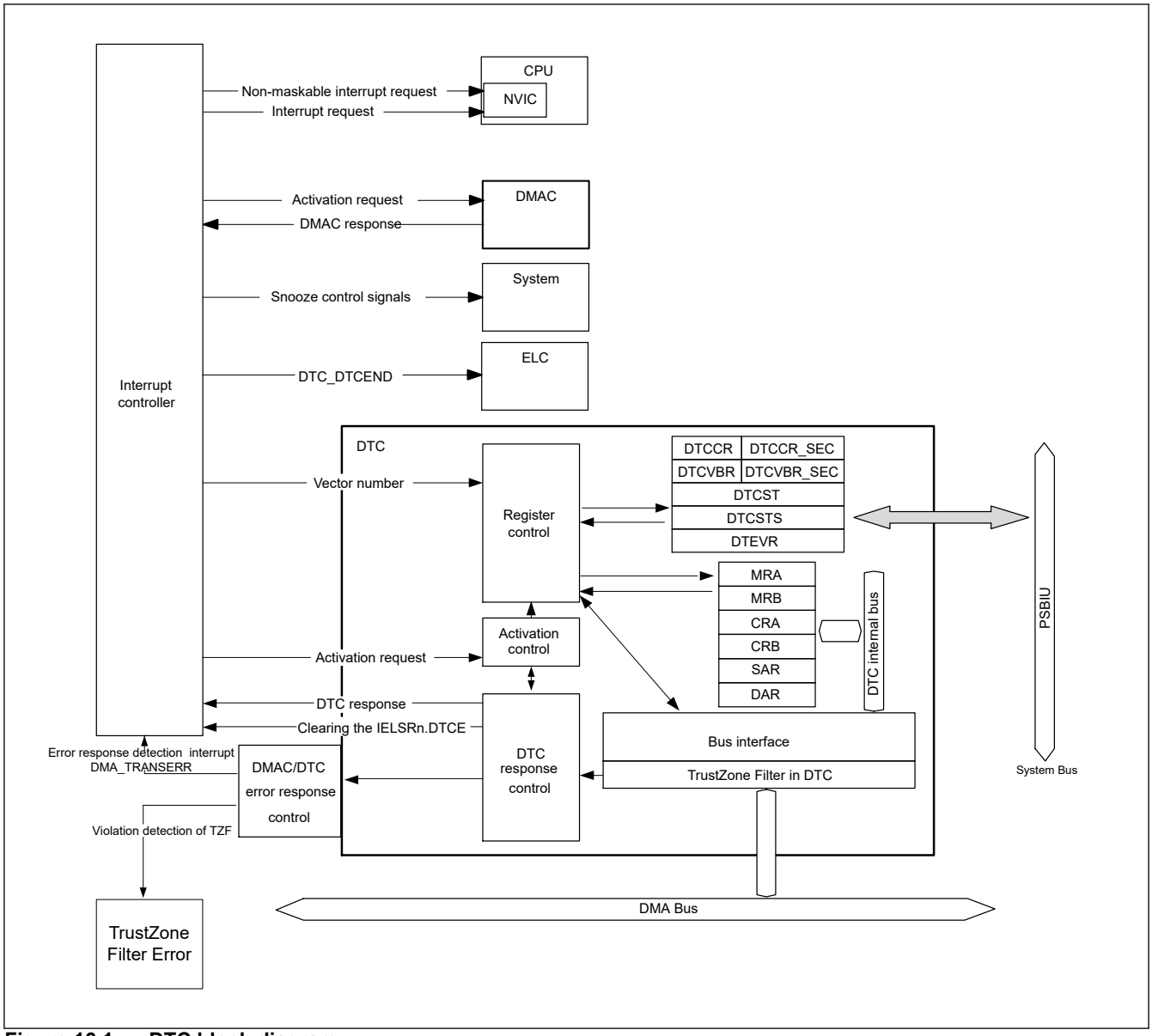


Figure 16.1 DTC block diagram

See section 12.1. Overview in section 12, Interrupt Controller Unit (ICU) for the connections between the DTC and NVIC in the CPU.

16.2 Register Descriptions

MRA, MRB, SAR, DAR, CRA, and CRB are all DTC internal registers that cannot be directly accessed from the CPU. Values to be set in these DTC internal registers are placed in the SRAM area as transfer information. When an activation request is generated, the DTC reads the transfer information from the SRAM area and sets it in its internal registers. After the data transfer ends, the internal register contents are written back to the SRAM area as transfer information.

16.2.1 DTCSAR : DTC Controller Security Attribution Register

Base address: CPSCU = 0x4000_8000

Offset address: 0x30

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

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

Bit	Symbol	Function	R/W
0	DTCSTSA	DTC Security Attribution 0: Secure. 1: Non-Secure.	R/W
31:1	—	This bit is read as 1.	R/W

Note: Only secure access can write to this register. Both secure access and Non-secure read access are allowed but Non-secure write access is not allowed and TrustZone access error is not generated.

Note: This register is write-protected by PRCR register.

This register only sets the DTCST security attribute.

DTCSTSA bit (DTC Security Attribution)

Security attributes of registers for DTCST.

Do not write to the DTCSTSA bit while DTC transfer is enabled or a bus master is writing to the DTC registers.

16.2.2 MRA : DTC Mode Register A

Base address: DTCVBR

Offset address: 0x03 + 0x4 × Vector number
(Inaccessible directly from the CPU. See section 16.3.1. Allocating Transfer Information and DTC Vector Table)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	MD[1:0]		SZ[1:0]		SM[1:0]		—	—
Value after reset:	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
1:0	—	The read values are undefined. The write value should be 0.	—
3:2	SM[1:0]	Transfer Source Address Addressing Mode 0 0: Address in the SAR register is fixed (write-back to SAR is skipped.) 0 1: Address in the SAR register is fixed (write-back to SAR is skipped.) 1 0: SAR value is incremented after data transfer: +1 when SZ[1:0] = 00b +2 when SZ[1:0] = 01b +4 when SZ[1:0] = 10b 1 1: SAR value is decremented after data transfer: -1 when SZ[1:0] = 00b -2 when SZ[1:0] = 01b -4 when SZ[1:0] = 10b	—

Bit	Symbol	Function	R/W
5:4	SZ[1:0]	DTC Data Transfer Size 0 0: Byte (8-bit) transfer 0 1: Halfword (16-bit) transfer 1 0: Word (32-bit) transfer 1 1: Setting prohibited	—
7:6	MD[1:0]	DTC Transfer Mode Select 0 0: Normal transfer mode 0 1: Repeat transfer mode 1 0: Block transfer mode 1 1: Setting prohibited	—

The MRA register cannot be accessed directly from the CPU, however the CPU can access the SRAM area (transfer information (n) start address + 0x03) and DTC transfers it automatically to and from the MRA register. See section 16.3.1. Allocating Transfer Information and DTC Vector Table.

16.2.3 MRB : DTC Mode Register B

Base address: DTCVBR

Offset address: 0x02 + 0x4 × Vector number
(Inaccessible directly from the CPU. See section 16.3.1. Allocating Transfer Information and DTC Vector Table)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	CHNE	CHNS	DISEL	DTS	DM[1:0]		—	—
Value after reset:	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
1:0	—	The read values are undefined. The write value should be 0.	—
3:2	DM[1:0]	Transfer Destination Address Addressing Mode 0 0: Address in the DAR register is fixed (write-back to DAR is skipped) 0 1: Address in the DAR register is fixed (write-back to DAR is skipped) 1 0: DAR value is incremented after data transfer: +1 when MRA.SZ[1:0] = 00b +2 when SZ[1:0] = 01b +4 when SZ[1:0] = 10b 1 1: DAR value is decremented after data transfer: -1 when MRA.SZ[1:0] = 00b -2 when SZ[1:0] = 01b -4 when SZ[1:0] = 10b	—
4	DTS	DTC Transfer Mode Select 0: Select transfer destination as repeat or block area. 1: Select transfer source as repeat or block area.	—
5	DISEL	DTC Interrupt Select 0: Generate an interrupt request to the CPU when specified data transfer is complete. 1: Generate an interrupt request to the CPU each time DTC data transfer is performed.	—
6	CHNS	DTC Chain Transfer Select 0: Chain transfer is continuous. 1: Chain transfer occurs only when the transfer counter changes from 1 to 0 or 1 to CRAH.	—
7	CHNE	DTC Chain Transfer Enable 0: Chain transfer is disabled. 1: Chain transfer is enabled.	—

The MRB register cannot be accessed directly from the CPU, however the CPU can access the SRAM area (transfer information (n) start address + 0x02) and DTC transfers it automatically to and from the MRB register. See section 16.3.1. Allocating Transfer Information and DTC Vector Table.

DM[1:0] bits (Transfer Destination Address Addressing Mode)

The DM[1:0] bits are to fix the address of the DAR register or specify increment / decrement of the DAR register after transfer.

DTS bit (DTC Transfer Mode Select)

The DTS bit specifies whether the transfer source or destination is the repeat or block area in repeat or block transfer mode.

DISEL bit (DTC Interrupt Select)

The DISEL bit specifies the condition for generating an interrupt request to the CPU.

CHNS bit (DTC Chain Transfer Select)

The CHNS bit selects the chain transfer condition. When CHNE is 0, the CHNS setting is ignored. For details on the conditions for chain transfer, see Table 16.3.

When the next transfer is chain transfer, completion of the specified number of transfers is not determined, the activation source flag is not cleared, and an interrupt request to the CPU is not generated.

CHNE bit (DTC Chain Transfer Enable)

The CHNE bit enables chain transfer. The chain transfer condition is selected by the CHNS bit. For details on chain transfer, see section 16.4.6. Chain Transfer.

16.2.4 SAR : DTC Transfer Source Register

Base address: DTCVBR

Offset address: 0x04 + 0x4 × Vector number
(Inaccessible directly from the CPU. See section 16.3.1. Allocating Transfer Information and DTC Vector Table)

Bit position:	31																												0
Bit field:																													
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

The SAR sets the transfer source start address and cannot be accessed directly from the CPU. However, the CPU can access the SRAM area (transfer information (n) start address + 0x04) and DTC transfers it automatically to and from the SAR register. See section 16.3.1. Allocating Transfer Information and DTC Vector Table.

Misalignment is prohibited for DTC transfers. Bit[0] must be 0 when MRA.SZ[1:0] = 01b, and bit[1] and bit[0] must be 0 when MRA.SZ[1:0] = 10b.

16.2.5 DAR : DTC Transfer Destination Register

Base address: DTCVBR

Offset address: 0x08 + 0x4 × Vector number
(Inaccessible directly from the CPU. See section 16.3.1. Allocating Transfer Information and DTC Vector Table)

Bit position:	31																												0
Bit field:																													
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

The DAR sets the transfer destination start address and cannot be accessed directly from the CPU. However, the CPU can access the SRAM area (transfer information (n) start address + 0x08) and DTC transfers it automatically to and from the DAR register. See section 16.3.1. Allocating Transfer Information and DTC Vector Table.

Misalignment is prohibited for DTC transfers. Bit[0] must be 0 when MRA.SZ[1:0] = 01b, and bit[1] and bit[0] must be 0 when MRA.SZ[1:0] = 10b.

16.2.6 CRA : DTC Transfer Count Register A

Base address: DTCVBR

Offset address: 0x0E + 0x4 × Vector number
(Inaccessible directly from the CPU. See section 16.3.1. Allocating Transfer Information and DTC Vector Table)

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

Bit	Symbol	Function	R/W
7:0	CRAL	Transfer Counter A Lower Register Specify the transfer count.	—
15:8	CRAH	Transfer Counter A Upper Register Specify the transfer count.	—

Note: The function depends on the transfer mode.
Note: Set CRAH and CRAL to the same value in repeat transfer mode and block transfer mode.

The CRA register consists of 16 bits. CRAL is the lower 8 bits and CRAH is the upper 8 bits. CRA is used in normal mode.

CRAL and CRAH are used in repeat transfer mode and block transfer mode.

The CRA register cannot be accessed directly from the CPU. However, the CPU can access the SRAM area (transfer information (n) start address + 0x0E) and DTC transfers it automatically to and from the CRA register. See section 16.3.1. Allocating Transfer Information and DTC Vector Table.

(1) Normal transfer mode (MRA.MD[1:0] = 00b)

In normal transfer mode, CRA functions as a 16-bit transfer counter. The transfer count is 1, 65535, and 65536 when the set value is 0x0001, 0xFFFF, and 0x0000, respectively. The CRA value is decremented (-1) on each data transfer.

(2) Repeat transfer mode (MRA.MD[1:0] = 01b)

In repeat transfer mode, the CRAH register holds the transfer count and the CRAL register functions as an 8-bit transfer counter. The transfer count is 1, 255, and 256 when the set value is 0x01, 0xFF, and 0x00, respectively. The CRAL value is decremented (-1) on each data transfer. When it reaches 0x00, the CRAH value is transferred to CRAL.

(3) Block transfer mode (MRA.MD[1:0] = 10b)

In block transfer mode, the CRAH register holds the block size and the CRAL register functions as an 8-bit block size counter. The transfer count is 1, 255, and 256 when the set value is 0x01, 0xFF, and 0x00, respectively. The CRAL value is decremented (-1) on each data transfer. When it reaches 0x00, the CRAH value is transferred to CRAL.

16.2.7 CRB : DTC Transfer Count Register B

Base address: DTCVBR

Offset address: 0x0C + 0x4 × Vector number
(Inaccessible directly from the CPU. See section 16.3.1. Allocating Transfer Information and DTC Vector Table)

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

The CRB sets the block transfer count for block transfer mode. The transfer count is 1, 65535, and 65536 when the set value is 0x0001, 0xFFFF, and 0x0000, respectively. The CRB value is decremented (-1) when the final data of a single block size is transferred. When normal transfer mode or repeat transfer mode is selected, this register is not used, and the set value is ignored.

The CRB cannot be accessed directly from the CPU. However, the CPU can access the SRAM area (transfer information (n) start address + 0x0C) and DTC transfers it automatically to and from the CRB register. See section 16.3.1. Allocating Transfer Information and DTC Vector Table.

16.2.8 DTCCR : DTC Control Register

Base address: DTC = 0x4000_5400

Offset address: 0x00

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

Bit	Symbol	Function	R/W
2:0	—	These bits are read as 0. The write value should be 0.	R/W
3	—	This bit is read as 1. The write value should be 1.	R/W
4	RRS	DTC Transfer Information Read Skip Enable 0: Transfer information read is not skipped 1: Transfer information read is skipped when vector numbers match	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

RRS bit (DTC Transfer Information Read Skip Enable)

The RRS bit enables skipping of transfer information reads when vector numbers match. The DTC vector number is compared with the vector number in the previous activation process. When these vector numbers match and the RRS bit is set to 1, DTC data transfer is performed without reading the transfer information. However, when the previous transfer is a chain transfer, the transfer information is read regardless of the RRS bit.

When the transfer counter (CRA register) becomes 0 during the previous normal transfer and when the transfer counter (CRB register) becomes 0 during the previous block transfer, the transfer information is read regardless of the RRS bit value.

16.2.9 DTCCR_SEC : DTC Control Register for secure Region

Base address: DTC = 0x4000_5400

Offset address: 0x10

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

Bit	Symbol	Function	R/W
2:0	—	These bits are read as 0. The write value should be 0.	R/W
3	—	This bit is read as 1. The write value should be 1.	R/W
4	RRS	DTC Transfer Information Read Skip Enable for Secure 0: Transfer information read is not skipped. 1: Transfer information read is skipped when vector numbers match.	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: Secure access is allowed. Non-secure access is read-only.

RRS bit (DTC Transfer Information Read Skip Enable for Secure)

The RRS bit enables skipping of transfer information reads when vector numbers match. The DTC vector number is compared with the vector number in the previous activation process. When these vector numbers match and the RRS bit is

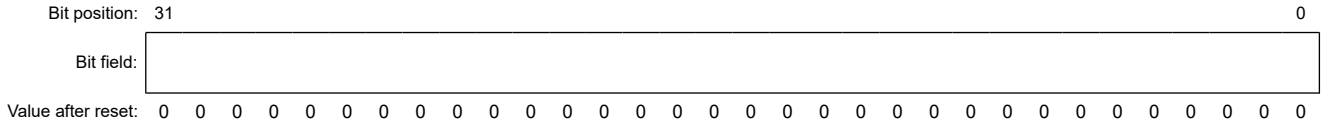
set to 1, DTC data transfer is performed without reading the transfer information. However, when the previous transfer is a chain transfer, the transfer information is read regardless of the RRS bit.

When the transfer counter (CRA register) becomes 0 during the previous normal transfer and when the transfer counter (CRB register) becomes 0 during the previous block transfer, the transfer information is read regardless of the RRS bit value.

16.2.10 DTCVBR : DTC Vector Base Register

Base address: DTC = 0x4000_5400

Offset address: 0x04



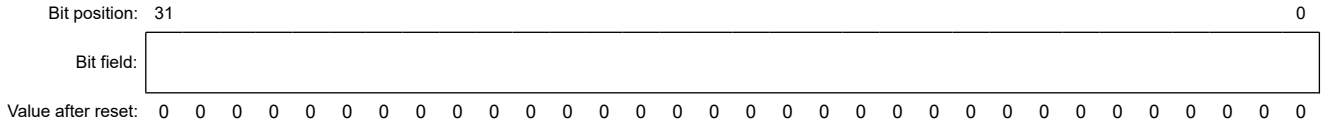
Bit	Symbol	Function	R/W
31:0	n/a	DTC Vector Base Address Set the DTC vector base address. The lower 10 bits should be 0.	R/W

The DTCVBR sets the base address for calculating the DTC vector table address, which can be set in the range of 0x0000_0000 to 0xFFFF_FFFF (4 GB) in 1-KB units.

16.2.11 DTCVBR_SEC : DTC Vector Base Register for secure Region

Base address: DTC = 0x4000_5400

Offset address: 0x14



Bit	Symbol	Function	R/W
31:0	n/a	DTC Vector Base Address for secure region Set DTC Vector Base Address for secure region. The lower 10 bits should be 0.	R/W

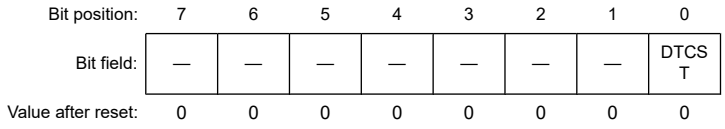
Note: Secure access is allowed. Non-secure access is read-only.

The DTCVBR_SEC sets the base address for calculating the DTC vector table address, which can be set in the range of 0x0000_0000 to 0xFFFF_FFFF (4 GB) in 1-KB units.

16.2.12 DTCST : DTC Module Start Register

Base address: DTC = 0x4000_5400

Offset address: 0x0C



Bit	Symbol	Function	R/W
0	DTCST	DTC Module Start 0: DTC module stopped. 1: DTC module started.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

DTCST bit (DTC Module Start)

Set the DTCST bit to 1 to enable the DTC to accept transfer requests. When this bit is set to 0, transfer requests are no longer accepted. If this bit is set to 0 during a data transfer, the accepted transfer request is active until processing completes.

DTCST must be set to 0 before transitioning to one of the following state or mode:

- Module-stop state
- Software Standby mode without Snooze mode transition
- Deep Software standby mode

For details on these transitions, see section 16.10. Low Power Consumption Function and section 10, Low Power Modes.

16.2.13 DTCSTS : DTC Status Register

Base address: DTC = 0x4000_5400

Offset address: 0x0E

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

Bit	Symbol	Function	R/W
7:0	VECN[7:0]	DTC-Activating Vector Number Monitoring These bits indicate the vector number for the activation source when a DTC transfer is in progress. The value is only valid if a DTC transfer is in progress (ACT flag is 1).	R
14:8	—	These bits are read as 0.	R
15	ACT	DTC Active Flag 0: DTC transfer operation is not in progress. 1: DTC transfer operation is in progress.	R

VECN[7:0] bits (DTC-Activating Vector Number Monitoring)

While transfer by the DTC is in progress, the VECN[7:0] bits indicate the vector number associated with the activation source for the transfer. The value read from the VECN[7:0] bits is valid if the ACT flag is 1, indicating a DTC transfer in progress, and invalid if the ACT flag is 0, indicating no DTC transfer is in progress.

ACT flag (DTC Active Flag)

The ACT flag indicates the state of the DTC transfer operation.

[Setting condition]

- When the DTC is activated by a transfer request.

[Clearing condition]

- When transfer by the DTC, in response to a transfer request, is complete.

16.2.14 DTEVR : DTC Error Vector Register

Base address: DTC = 0x4000_5400

Offset address: 0x20

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

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

Bit	Symbol	Function	R/W
7:0	DTEV[7:0]	DTC Error Vector Number These bits represent error vector of the DTC.	R
8	DTEVSAM	DTC Error Vector Number SA Monitor Indicates the SA of vector number causing the error. 0: Secure vector number 1: Non-Secure vector number	R
15:9	—	These bits are read as 0. The write value should be 0.	R
16	DTESTA	DTC Error Status Flag 0: No DTC transfer error occurred 1: DTC transfer error occurred	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R

Note: Writing to DTESTA depends on the value of DTEVSAM

DTEV[7:0] bits (DTC Error Vector Number)

When a transfer error due to DTC transfer occurs, the DTEV[7:0] bits store the violating DTC channel.

When reset is selected in MPU.MMPUOAD.OAD and TZF.TZFOAD.OAD, this register is also reset. Select NMI when you want to debug the program.

[Set condition]

- When the DTC transfer error occurs and DTESTA = 0.

[Clearing condition]

- When 1 is written to DTEVR.DTESTA.

DTEVSAM bit (DTC Error Vector Number SA Monitor)

When a transfer error due to DTC transfer occurs, the DTEVSAM bit indicates the SA of the violating DTC vector number.

When reset is selected in MPU.MMPUOAD.OAD and TZF.TZFOAD.OAD, this register is also reset. Select NMI when you want to debug the program.

[Set condition]

- When the DTC transfer error occurs and DTESTA = 0.

[Clearing condition].

- When 1 is written to DTEVR.DTESTA.

DTESTA bit (DTC Error Status Flag)

The DTESTA bit indicates whether or not a DTC transfer error occurred.

DTEV, DTEVSAM, DTESTA are cleared by writing 1 to DTESTA.

Writing 0 to DTESTA is ignored.

When reset is selected in MPU.MMPUOAD.OAD and TZF.TZFOAD.OAD, this register is also reset. Select NMI when you want to debug the program.

[Set condition]

- When the DMAC transfer error occurs.

[Clearing condition]

- When 1 is written to DTEVR.DTESTA.

Note: When DTEVSAM = 1, it can be cleared in the secure state and non-secure state. DTEVSAM = 0, it cannot be cleared in the non-secure state.

16.3 Activation Sources

The DTC is activated by an interrupt request. Setting the ICU.IELSRn.DTCE bit to 1 enables activation of the DTC by the associated interrupt. The selector output *n* number set in ICU.IELSRn is defined as the interrupt vector number, where n = 0 to 95. For an enabled interrupt, the specific DTC interrupt source associated with each interrupt vector number *n* is selected in ICU.IELSRn.IELS[8:0] where n = 0 to 95, as listed in section 12.3.2. Event Number in section 12, Interrupt Controller Unit (ICU). For activation by software, see section 17.2.2. ELSEGRn : Event Link Software Event Generation Register n (n = 0, 1).

The interrupt vector number is equivalent to the DTC vector table number. After the DTC accepted an activation request, it does not accept another activation request until the transfer for that single request is complete, regardless of the priority of the requests. When multiple activation requests are generated during a DTC transfer, the highest priority request is accepted on completion of the transfer. When multiple activation requests are generated while the DTC Module Start bit (DTCST.DTCST) is 0, the DTC accepts the highest priority request when DTCST.DTCST is subsequently set to 1. The smaller interrupt vector number has higher priority.

The DTC performs the following operations at the start of a single data transfer or for a chain transfer, after the last of the consecutive transfers:

- On completion of a specified round of data transfer, the ICU.IELSRn.DTCE bit is set to 0, and an interrupt request is sent to the CPU.
- If the MRB.DISEL bit is 1, an interrupt request is sent to the CPU on completion of a data transfer.
- For other transfers, the ICU.IELSRn.IR flag of the activation source is set to 0 at the start of the data transfer.

16.3.1 Allocating Transfer Information and DTC Vector Table

The DTC reads the start address of the transfer information associated with each activation source from the vector table and reads the transfer information starting at that address.

DTC has two vector tables, non-secure side or secure side. Because the interrupt vector number that serves as a trigger for DTC is divided into non-secure or secure. Place the vector table of the interrupt vector number of SA = 1 in DTCVBR which is the non-secure side. Place the vector table of interrupt number SA = 0 in DTCVBR_SEC which is the secure side.

The vector table must be located so that the lower 10 bits of the base address (start address) are 0. Use the DTC Vector Base Register (DTCVBR) to set the base address of the DTC vector table. Transfer information is allocated in the SRAM area. In the SRAM area, the start address of the transfer information *n* with vector number *n* must be 4n added to the base address in the vector table.

Figure 16.2 shows the relationship between the DTC vector table and transfer information. Figure 16.3 shows the allocation of transfer information in the SRAM area.

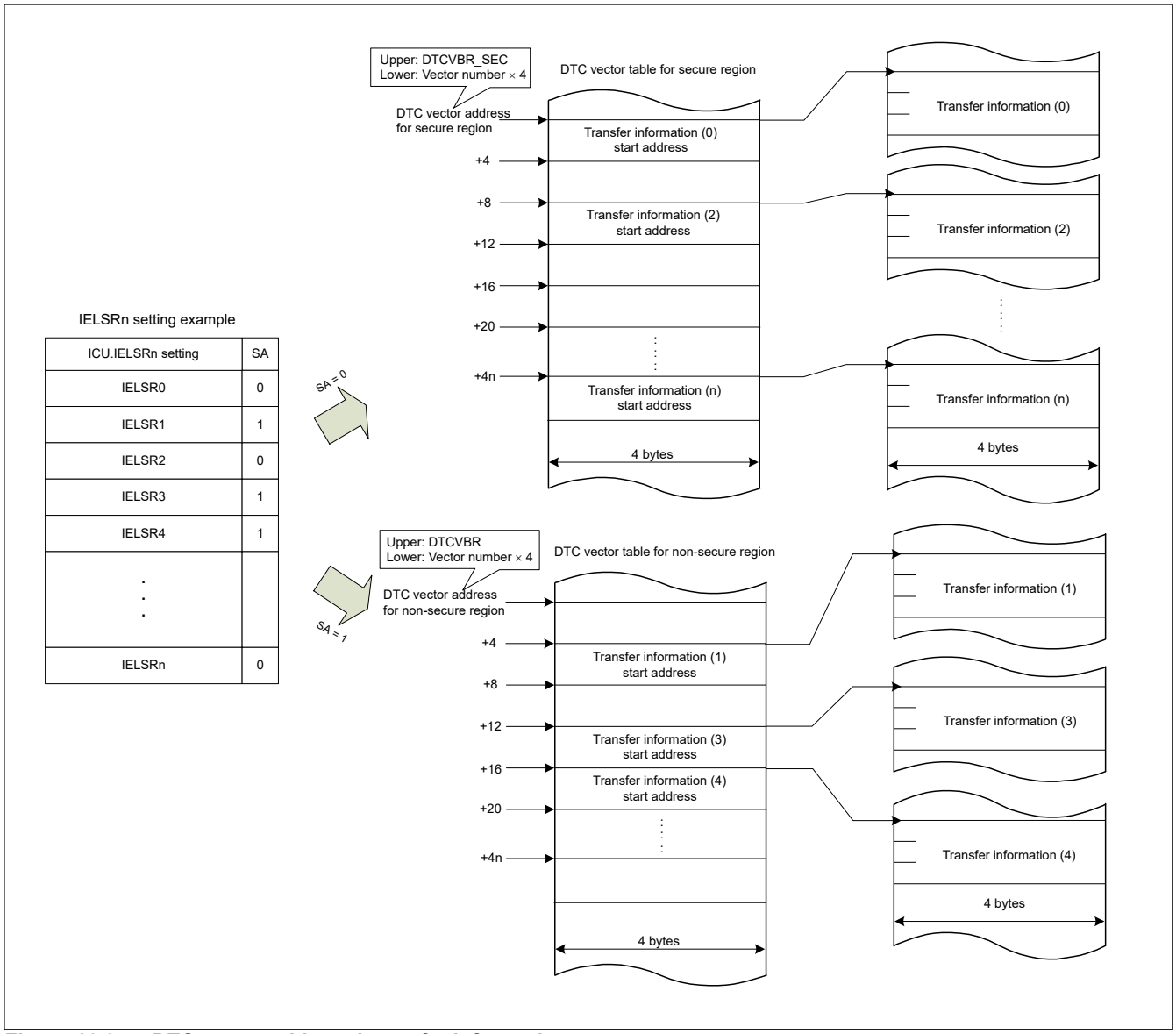


Figure 16.2 DTC vector table and transfer information

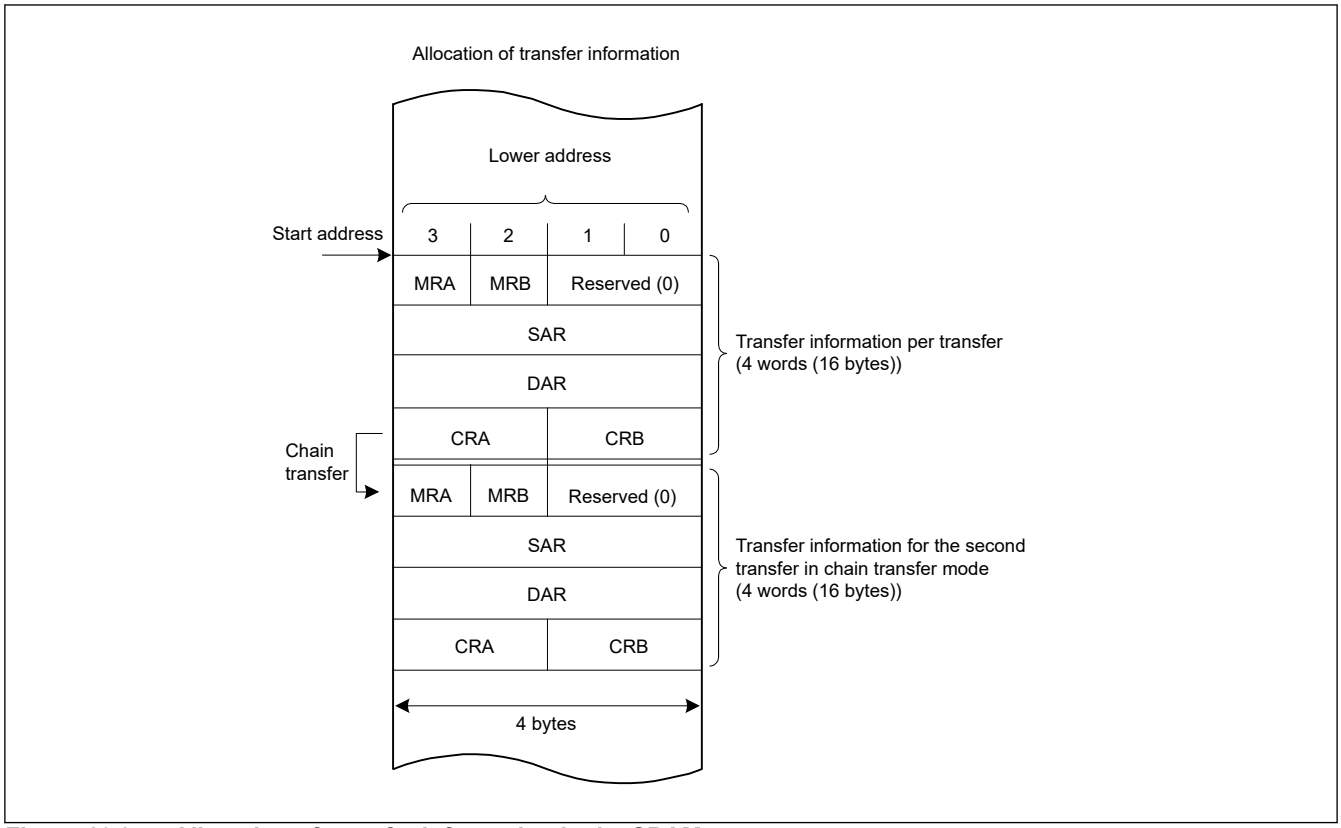


Figure 16.3 Allocation of transfer information in the SRAM area

16.4 Operation

The DTC transfers data according to the transfer information. Storage of the transfer information in the SRAM area is required before a DTC operation. When the DTC is activated, it reads the DTC vector associated with the vector number. The DTC reads the transfer information from the transfer information store address referenced by the DTC vector and transfers the data. After the data transfer, the DTC writes back the transfer information. Storing the transfer information in the SRAM area allows data transfer of any number of channels.

The transfer modes include:

- Normal transfer mode
- Repeat transfer mode
- Block transfer mode.

The DTC specifies a transfer source address in the SAR register and a transfer destination address in the DAR register. The values of these registers are incremented, decremented, or address-fixed independently after the data transfer.

Table 16.2 describes the DTC transfer modes.

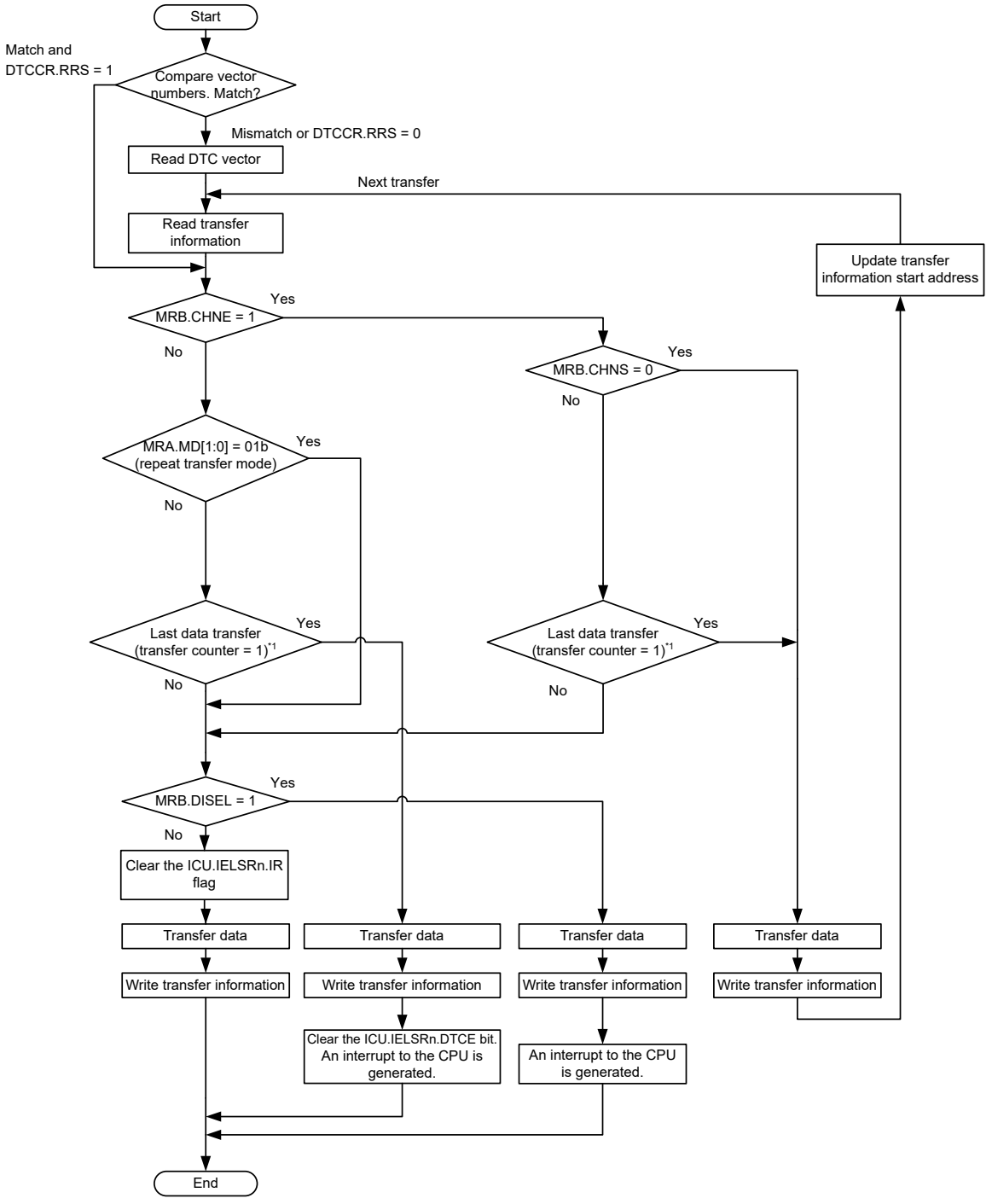
Table 16.2 DTC transfer modes

Transfer mode	Data size transferred on single transfer request	Increment or decrement of memory address	Settable transfer count
Normal transfer mode	1 byte (8 bit), 1 halfword (16 bit), 1 word (32 bit)	Incremented or decremented by 1, 2, or 4 or address-fixed	1 to 65536
Repeat transfer mode*1	1 byte (8 bit), 1 halfword (16 bit), 1 word (32 bit)	Incremented or decremented by 1, 2, or 4 or address-fixed	1 to 256*3
Block transfer mode*2	Block size specified in CRAH (1 to 256 bytes, 1 to 256 halfwords (2 to 512 bytes), or 1 to 256 words (4 to 1024 bytes))	Incremented or decremented by 1, 2, or 4 or address-fixed	1 to 65536

Note 1. Set the transfer source or transfer destination as the repeat area.
Note 2. Set the transfer source or transfer destination as the block area.
Note 3. After a data transfer of the specified count, the initial state is restored and operation restarts.

Setting the MRB.CHNE bit to 1 allows multiple transfers or chain transfer on a single activation source. It also enables a chain transfer when the specified data transfer is complete.

Figure 16.4 shows the operation flow of the DTC. Table 16.3 lists the chain transfer conditions. The combination of control information for the second and subsequent transfers are omitted in this table.



Note 1. Counter value before starting data transfer.

Figure 16.4 DTC operation flow

Table 16.3 Chain transfer conditions

First transfer				Second transfer*3				
CHNE bit	CHNS bit	DISEL bit	Transfer counter*1 *2	CHNE bit	CHNS bit	DISEL bit	Transfer counter*1 *2	DTC transfer
0	—	0	Other than (1 → 0)	—	—	—	—	Ends after the first transfer
0	—	0	(1 → 0)	—	—	—	—	Ends after the first transfer with an interrupt request to the CPU
0	—	1	—	—	—	—	—	
1	0	—	—	0	—	0	Other than (1 → 0)	Ends after the second transfer
				0	—	0	(1 → 0)	Ends after the second transfer with an interrupt request to the CPU
				0	—	1	—	
1	1	0	Other than (1 → *)	—	—	—	—	Ends after the first transfer
1	1	—	(1 → *)	0	—	0	Other than (1 → 0)	Ends after the second transfer
				0	—	0	(1 → 0)	Ends after the second transfer with an interrupt request to the CPU
				0	—	1	—	
1	1	1	Other than (1 → *)	—	—	—	—	Ends after the first transfer with an interrupt request to the CPU

Note 1. The transfer counter used depends on the transfer modes as follows:
Normal transfer mode — CRA register
Repeat transfer mode — CRAL register
Block transfer mode — CRB register

Note 2. On completion of a data transfer, the counters operate as follows:
1 → 0 in normal and block transfer modes
1 → CRAH in repeat transfer mode
(1 → *) in the table indicates both of these two operations, depending on the mode.

Note 3. Chain transfer can be selected for the second or subsequent transfers. The conditions for the combination of the second transfer and CHNE = 1 is omitted.

16.4.1 Transfer Information Read Skip Function

Reading of vector addresses and transfer information can be skipped by setting the DTCCR.RRS bit. When a DTC activation request is generated, the current DTC vector number is compared with the DTC vector number in the previous activation process. When these vector numbers match and the RRS bit is set to 1, the DTC data transfer is performed without reading the vector address and transfer information. However, when the previous transfer is a chain transfer, the vector address and transfer information are read. Additionally, when the transfer counter (CRA register) becomes 0 during the previous normal transfer, and when the transfer counter (CRB register) becomes 0 during the previous block transfer, transfer information is read regardless of the RRS bit. Figure 16.12 shows an example when reading the transfer information is skipped.

To update the vector table and transfer information, set the RRS bit to 0, update the vector table and transfer information, then set the RRS bit to 1. The stored vector number is discarded by setting the RRS bit to 0. The updated DTC vector table and transfer information are read in the next activation process.

16.4.2 Transfer Information Write-Back Skip Function

When the MRA.SM[1:0] bits or the MRB.DM[1:0] bits are set to address fixed, a part of the transfer information is not written back. Table 16.4 lists the transfer information write-back skip conditions and the associated registers. The CRA and CRB registers are written back, and the write-back of the MRA and MRB registers is skipped.

Table 16.4 Transfer information write-back skip conditions and applicable registers

MRA.SM[1:0] bits		MRB.DM[1:0] bits			
b3	b2	b3	b2	SAR register	DAR register
0	0	0	0	Skip	Skip
0	0	0	1		
0	1	0	0		
0	1	0	1		
0	0	1	0	Skip	Write-back
0	0	1	1		
0	1	1	0		
0	1	1	1		
1	0	0	0	Write-back	Skip
1	0	0	1		
1	1	0	0		
1	1	0	1		
1	0	1	0	Write-back	Write-back
1	0	1	1		
1	1	1	0		
1	1	1	1		

16.4.3 Normal Transfer Mode

The normal transfer mode allows a 1-byte (8 bit), 1-halfword (16 bit), 1-word (32 bit) data transfer on a single activation source. The transfer count can be set from 1 to 65536. Transfer source and destination addresses can be independently set to increment, decrement, or fixed. This mode enables an interrupt request to the CPU to be generated at the end of a specified-count transfer.

Table 16.5 lists register functions in normal transfer mode, and Figure 16.5 shows the memory map of normal transfer mode.

Table 16.5 Register functions in normal transfer mode

Register	Description	Value written back by writing transfer information
SAR	Transfer source address	Increment, decrement, or fixed*1
DAR	Transfer destination address	Increment, decrement, fixed*1
CRA	Transfer counter A	CRA - 1
CRB	Transfer counter B	Not updated

Note 1. Write-back operation is skipped in address-fixed mode.

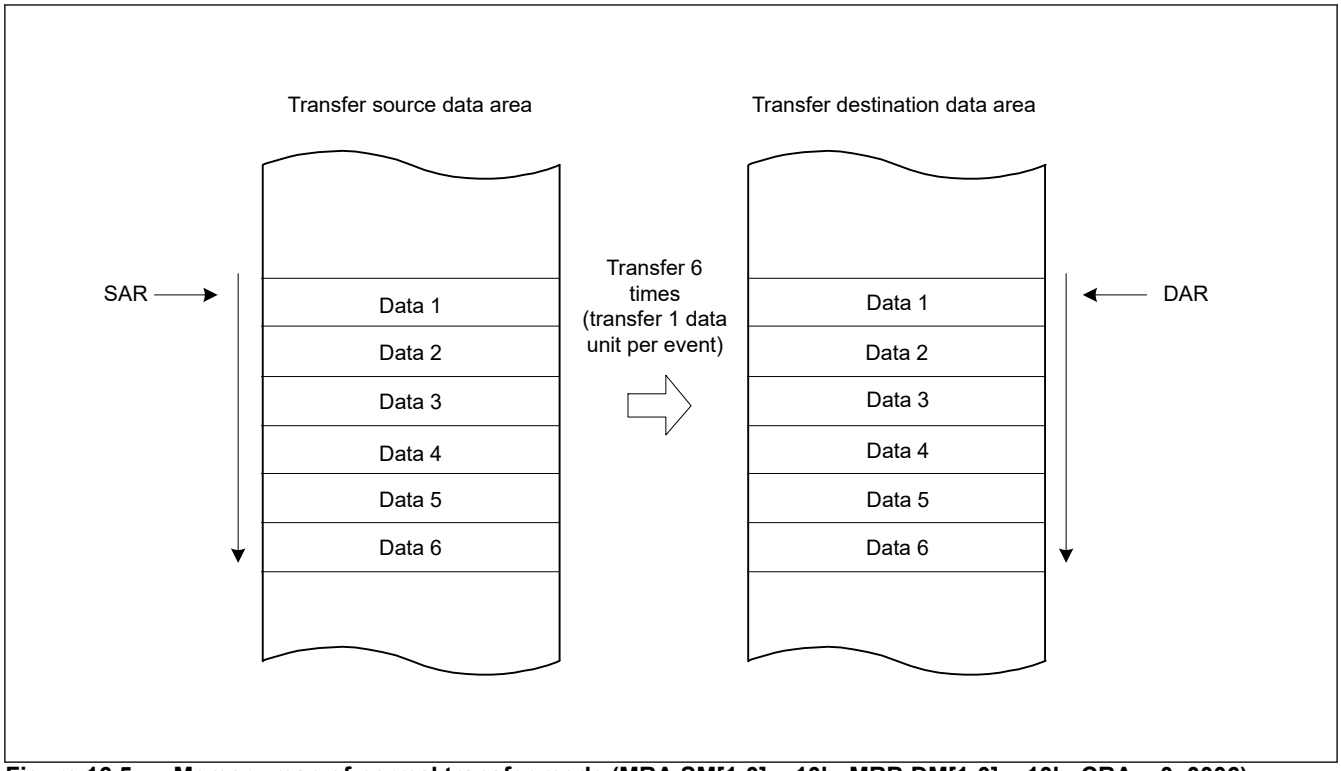


Figure 16.5 Memory map of normal transfer mode (MRA.SM[1:0] = 10b, MRB.DM[1:0] = 10b, CRA = 0x0006)

16.4.4 Repeat Transfer Mode

The repeat transfer mode allows a 1-byte (8-bit), 1-halfword (16-bit), or 1-word (32-bit) data transfer on a single activation source. Transfer source or transfer destination for the repeat area must be specified in the MRB.DTS bit. The transfer count can be set from 1 to 256. When the specified transfer count is complete, the initial value of the address register specified in the repeat area is restored, the initial value of the transfer counter is restored, and transfer is repeated. The other address register is incremented or decremented continuously or remains unchanged.

When the transfer counter CRAL decrements to 0x00 in repeat transfer mode, the CRAL value is updated to the value set in the CRAH register. As a result, the transfer counter does not clear to 0x00, which disables interrupt requests to the CPU when the MRB.DISEL bit is set to 0. An interrupt request to the CPU is generated when the specified data transfer completes.

Table 16.6 lists the register functions in repeat transfer mode, and Figure 16.6 shows the memory map of repeat transfer mode.

Table 16.6 Register functions in repeat transfer mode

Register	Description	Value written back by writing transfer information	
		When CRAL is not 1	When CRAL is 1
SAR	Transfer source address	Increment, decrement, fixed*1	• When the MRB.DTS bit is 0 Increment, decrement, or fixed*1 • When the MRB.DTS bit is 1 SAR register initial value
DAR	Transfer destination address	Increment, decrement, or fixed*1	• When the MRB.DTS bit is 0 DAR register initial value • When the MRB.DTS bit is 1 Increment, decrement, or fixed*1
CRAH	Retains transfer counter	CRAH	CRAH
CRAL	Transfer counter A	CRAL - 1	CRAH
CRB	Transfer counter B	Not updated	Not updated

Note 1. Write-back is skipped in address-fixed mode.

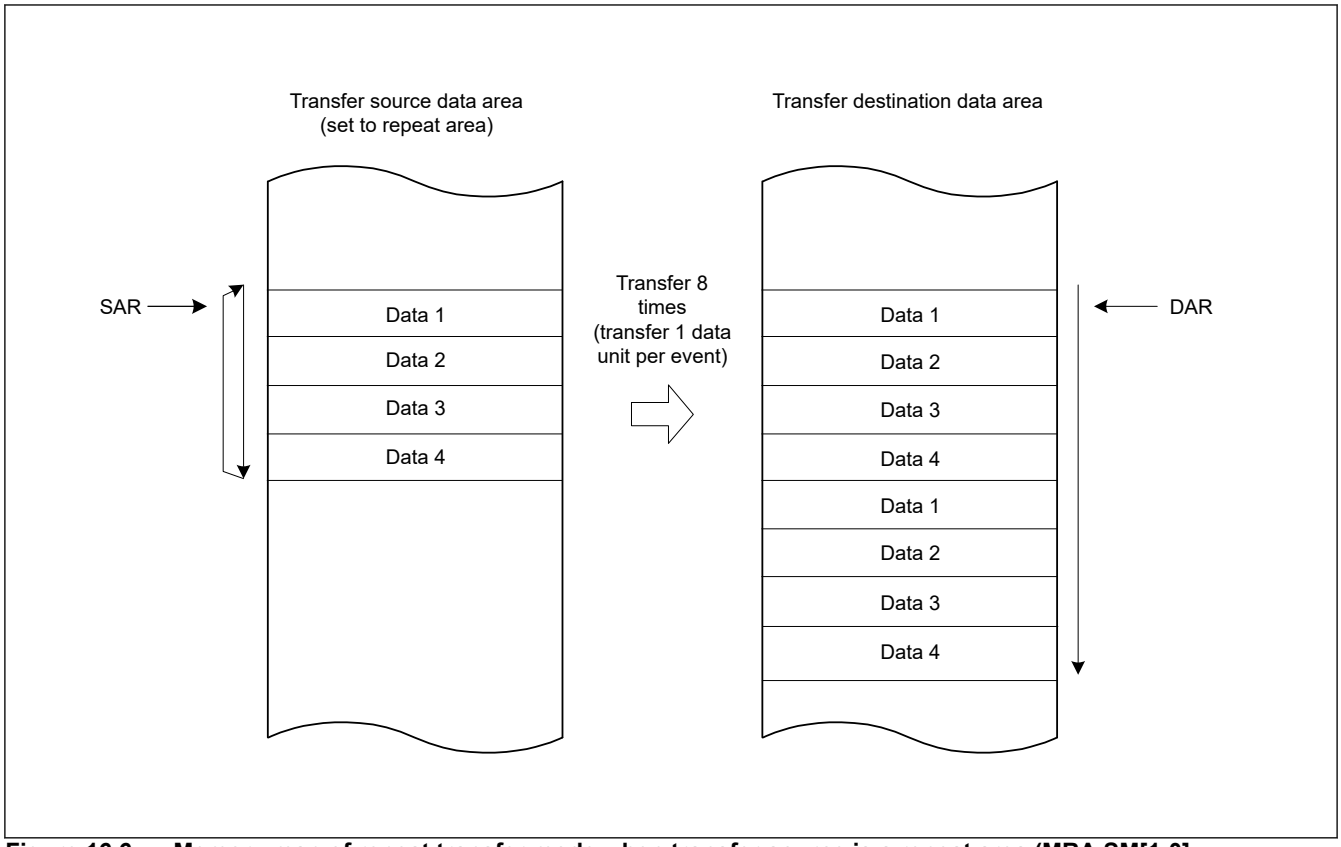


Figure 16.6 Memory map of repeat transfer mode when transfer source is a repeat area (MRA.SM[1:0] = 10b, MRB.DM[1:0] = 10b, CRAH = 0x04)

16.4.5 Block Transfer Mode

The block transfer mode allows single-block data transfer on a single activation source. Transfer source or transfer destination for the block area must be specified in the MRB.DTS bit. The block size can be set from 1 to 256 bytes, 1 to 256 halfwords (2 to 512 bytes), or 1 to 256 words (4 to 1024 bytes). When transfer of the specified block completes, the initial values of the block size counter CRAL and the address register (the SAR register when the MRB.DTS = 1 or the DAR register when the DTS = 0) specified in the block area are restored. The other address register is incremented or decremented continuously or remains unchanged.

The transfer count (block count) can be set from 1 to 65536. This mode enables an interrupt request to the CPU to be generated at the end of the specified-count block transfer.

Table 16.7 lists the register functions in block transfer mode, and Figure 16.7 shows the memory map for block transfer mode.

Table 16.7 Register functions in block transfer mode

Register	Description	Value written back by writing transfer information
SAR	Transfer source address	• When MRB.DTS bit is 0 Increment, decrement, or fixed*1 • When MRB.DTS bit is 1 SAR register initial value.
DAR	Transfer destination address	• When MRB.DTS bit is 0 DAR register initial value • When MRB.DTS bit is 1 Increment, decrement, or fixed*1.
CRAH	Holds block size	CRAH
CRAL	Block size counter	CRAH
CRB	Block transfer counter	CRB - 1

Note 1. Write-back is skipped in address-fixed mode.

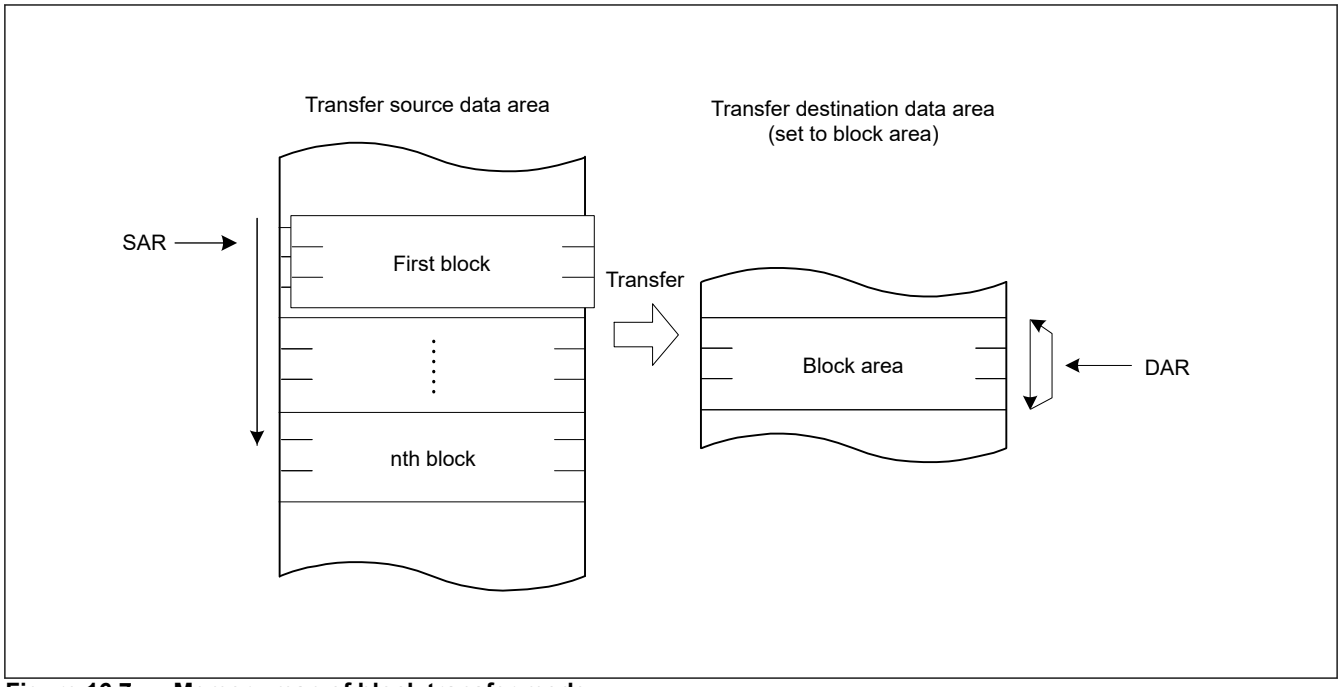


Figure 16.7 Memory map of block transfer mode

16.4.6 Chain Transfer

Setting the MRB.CHNE bit to 1 allows chain transfer to be performed continuously on a single activation source. If the MRB.CHNE is set to 1 and CHNS to 0, an interrupt request to the CPU is not generated on completion of the specified number of rounds of transfer or by setting the MRB.DISEL bit to 1. An interrupt request is sent to the CPU each time DTC data transfer is performed. Data transfer has no effect on the ICU.IELSRn.IR flag of the activation source.

The SAR, DAR, CRA, CRB, MRA, and MRB registers can be set independently of each other to define the data transfer. Figure 16.8 shows a chain transfer operation.

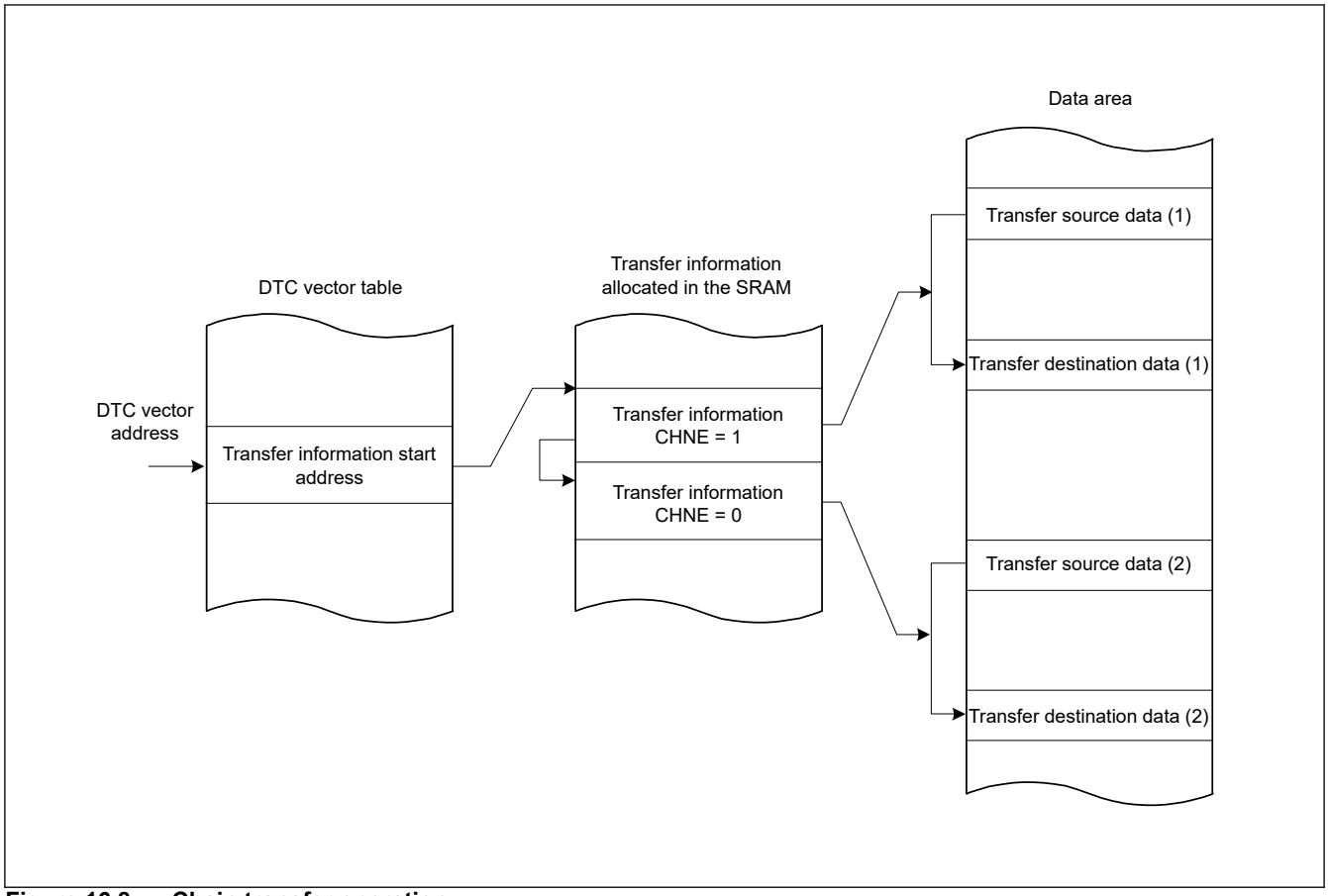


Figure 16.8 Chain transfer operation

Writing 1 to the MRB.CHNE and CHNS bits enables chain transfer to be performed only after completion of the specified data transfer. In repeat transfer mode, chain transfer is performed after completion of the specified data transfer. For details on chain transfer conditions, see Table 16.3.

16.4.7 Operation Timing

Figure 16.9 to Figure 16.12 are timing diagrams that show the minimum number of execution cycles.

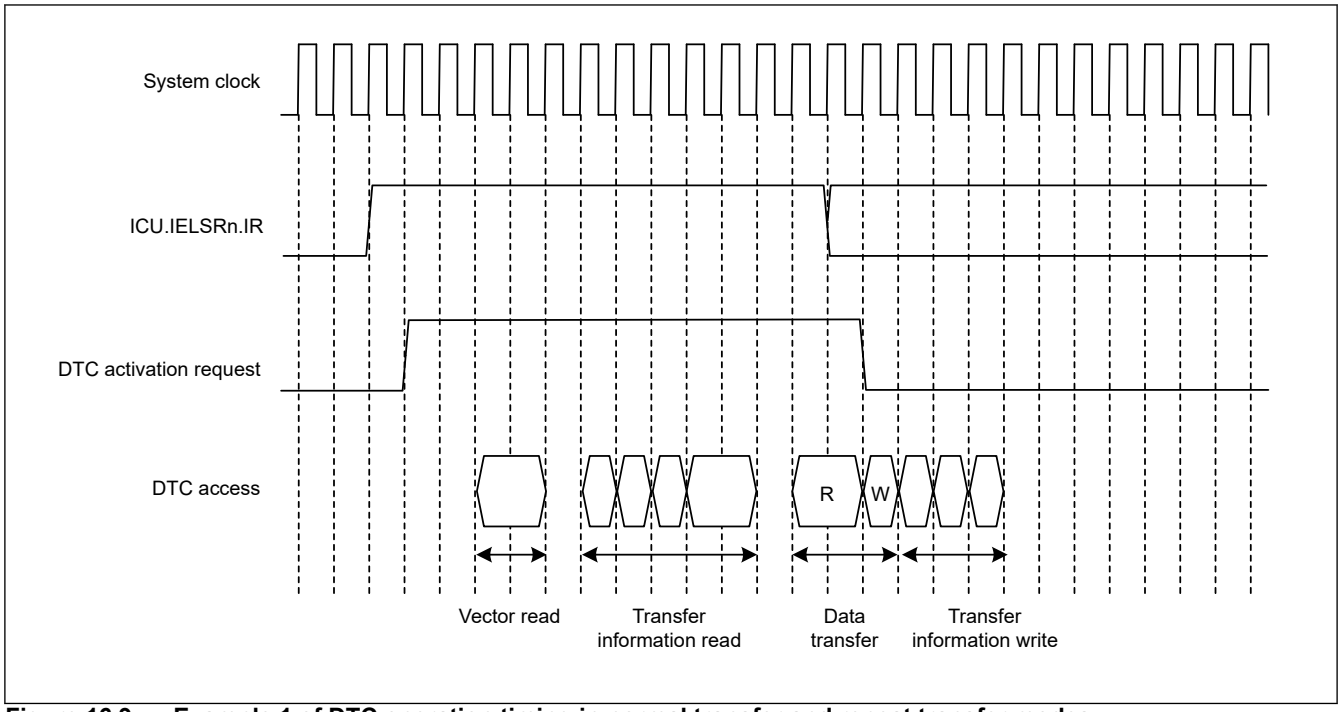


Figure 16.9 Example 1 of DTC operation timing in normal transfer and repeat transfer modes

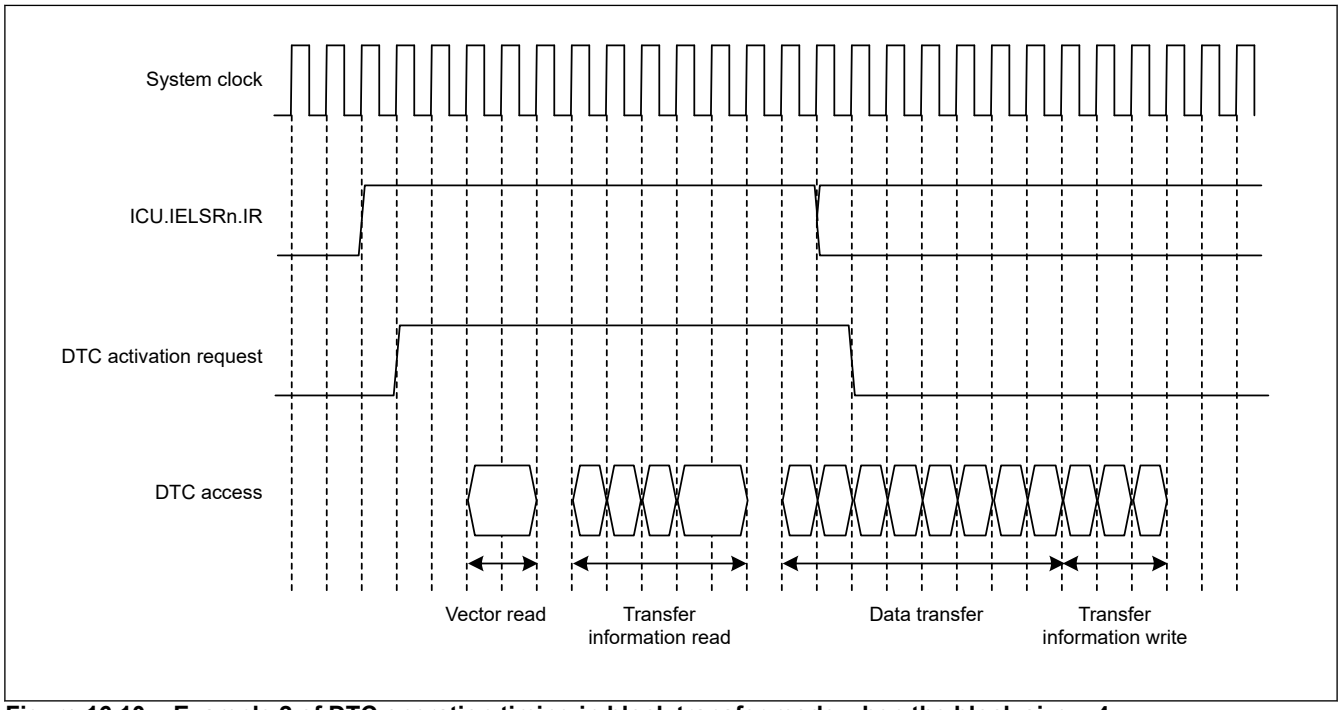


Figure 16.10 Example 2 of DTC operation timing in block transfer mode when the block size = 4

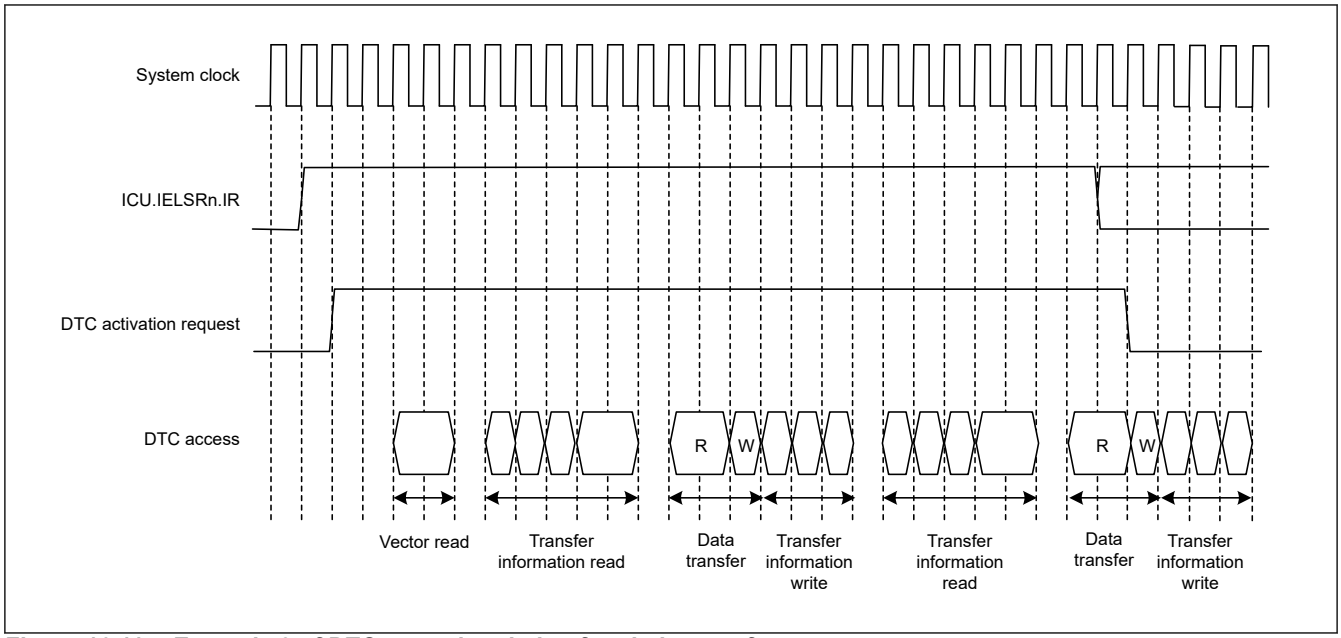
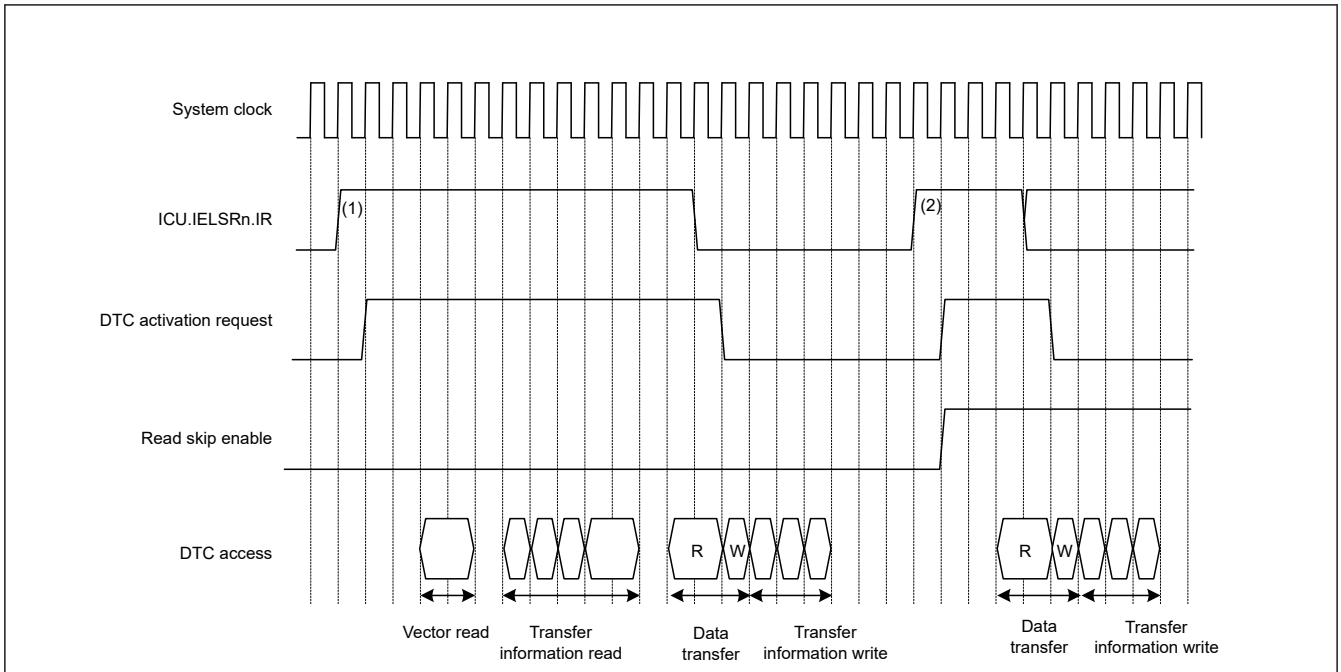


Figure 16.11 Example 3 of DTC operation timing for chain transfer



Note: When activation sources (vector numbers) of (1) and (2) are the same and the RRS = 1, the transfer information read for request (2) is skipped.

Figure 16.12 Example of operation when a transfer information read is skipped with the vector, transfer information, and transfer destination data on the SRAM, and the transfer source data on the peripheral module

16.4.8 Execution Cycles of DTC

Table 16.8 lists the execution cycles of single data transfer of the DTC. For the order of the execution states, see section 16.4.7. Operation Timing.

Table 16.8 Execution cycles of DTC

P: Block size (initial settings of CRAH and CRAL)
Cv: Cycles for access to vector transfer information storage destination
Ci: Cycles for access to transfer information storage destination address
Cr: Cycles for access to data read destination
Cw: Cycles for access to data write destination
The unit is for system clocks (ICLK) + 1 in the Vector read, Transfer information read, and Data transfer read columns and 2 in the Internal operation column.
Cv, Ci, Cr, and Cw vary depending on the corresponding access destination. For the number of cycles for respective access destinations, see section 41, SRAM, section 43, Flash Memory, and section 13, Buses.
The frequency ratio of the system clock and peripheral clock is also taken into consideration.
The DTC response time is the time from when the DTC activation source is detected until DTC transfer starts.
Table 16.8 does not include the time until DTC data transfer starts after the DTC activation source becomes active.

Transfer mode	Vector read		Transfer information read		Transfer information write			Data transfer		Internal operation	
								Read	Write		
Normal	Cv + 1	0*1	4 × Ci + 1	0*1	3 × Ci + 1*2	2 × Ci + 1*3	Ci*4	Cr + 1	Cw + 1	2	0*1
Repeat								Cr + 1	Cw + 1		
Block*5								P × Cr	P × Cw		

Note 1. When transfer information read is skipped.
Note 2. When neither SAR nor DAR is set to address-fixed mode.
Note 3. When SAR or DAR is set to address-fixed mode.
Note 4. When SAR and DAR are set to address-fixed mode.
Note 5. When the block size is 2 or more. If the block size is 1, the cycle number for normal transfer applies.

16.4.9 DTC Bus Mastership Release Timing

The DTC does not release the bus mastership during transfer information reads. Before the transfer information is read or written, the bus is arbitrated according to the priority determined by the bus master arbitrator. For bus arbitration, see section 13, Buses.

16.4.10 Vector Security

The security attribute of transfer access of DTC vector n and security attribute of access to the IELSRn(n = 0 to 95) register of ICU are controlled by SAIELSRn bit of ICUSARx(x = G,H or I) registers in CPSCU. For details on the CPSCU.ICUSARx registers, see section 12, Interrupt Controller Unit (ICU).

When the CPSCU.ICUSARx.SAIELSRn bit is 0, transfer of DTC vector n is secure access for both read and write. At the same time, the IELSRn register are protected from a non-secure access.

When the CPSCU.ICUSARx.SAIELSRn bit is 1, transfer of DTC vector n is non-secure access for both read and write. At the same time, the IELSRn register are non-secure attributes.

Do not write to the CPSCU.ICUSARx.SAIELSRn bit while DTC transfer is enabled or a bus master is writing to the DTC registers of same channel.

section 16.3.1. Allocating Transfer Information and DTC Vector Table shows security attribute about each DTC vectors.

16.4.11 Master TrustZone Filter in DTC

DTC has the Master TrustZone Filter. The Master TrustZone Filter in DTC can detect the security areas of Flash area(code Flash and data Flash) and SRAM area defined by IDAU. When no-secure accesses those addresses, it detects the security violation. Access of violation address is not performed. Detected the error is handled as the Master TrustZone Filter error.

16.5 DTC Setting Procedure

Before using the DTC, set the DTC Vector Base Register (DTCVBR). Set the ICU.IELSRn.IELS[8:0] bits to 0 to disable the interrupt in the NVIC and follow the procedure in Table 16.9 to set the DTC.

Table 16.9 DTC setting procedure

No.	Step Name	Description
1	Set the DTCCR*1.RRS bit to 0	Set the DTCCR*1.RRS bit to 0 to reset the transfer information read skip flag. After that, the transfer information read is not skipped while the DTC is activated. Be sure to specify this setting when the transfer information is updated.
2	Set transfer information (MRA, MRB, SAR, DAR, CRA, and CRB)	Allocate transfer information (MRA, MRB, SAR, DAR, CRA, and CRB) in the data area. To set transfer information, see section 16.2. Register Descriptions. To allocate transfer information, see section 16.3.1. Allocating Transfer Information and DTC Vector Table.
3	Set transfer information start addresses in the DTC vector table	Set the transfer information start addresses in the DTC vector table. To set the DTC vector table, see section 16.3.1. Allocating Transfer Information and DTC Vector Table.
4	Set the DTCCR*1.RRS bit to 1	Set the DTCCR*1.RRS bit to 1 to enable skipping of the second and subsequent transfer information read cycles for continuous DTC activation from the same interrupt source. The RRS bit can be set to 1, but if this is set during DTC transfer, it becomes valid from the next transfer.
5	Set the ICU.IELSRn.DTCE bit to 1. Set the ICU.IELSRn.IELS[8:0] as interrupt source. The interrupt should be enabled in the NVIC.	Set the ICU.IELSRn.DTCE bit to 1. Set ICU.IELSRn.IELS[8:0] as interrupt sources that trigger DTC. The interrupt must be enabled in the NVIC. See section 12.3.2. Event Number in section 12, Interrupt Controller Unit (ICU).
6	Set the enable bit for an activation source interrupt	Set the enable bit for the activation source interrupts to 1. When a source interrupt is generated, the DTC is activated. To set the interrupt source enable bit, see the settings for the modules that are to be the activation sources.
7	Set the DTCST.DTCST bit to 1	Set the DTC Module Start bit (DTCST.DTCST) to 1.

Note: The DTCST.DTCST bit can be set even if the setting for each activation source is not completed.
Note: When used in non-secure state, DTCSAR.DTCSTSA = 1 or DTCST.DTCST = 1 must be set.
Note 1. When used in secure state, access DTCCR_SEC instead of DTCCR.

16.6 Examples of DTC Usage

16.6.1 Normal Transfer

This section provides an example of DTC usage and its application when receiving 128 bytes of data from an SCI.

(1) Transfer information settings

In the MRA register, select a fixed source address (MRA.SM[1:0] = 00b), normal transfer mode (MRA.MD[1:0] = 00b), and byte-sized transfer (MRA.SZ[1:0] = 00b). In the MRB register, specify incrementation of the destination address (MRB.DM[1:0] = 10b) and single data transfer by a single interrupt (MRB.CHNE = 0 and MRB.DISEL = 0). The MRB.DTS bit can be set to any value. Set the RDR register address of the SCI in the SAR register, the start address of the SRAM area for data storage in the DAR register, and 128 (0x0080) in the CRA register. The CRB register can be set to any value.

(2) DTC vector table settings

The start address of the transfer information for the RXI interrupt is set in the vector table for the DTC.

(3) ICU settings and DTC module activation

Set the ICU.IELSRn.DTCE bit to 1 and set ICU.IELSRn.IELS[8:0] as the SCI interrupt. The interrupt must be enabled in the NVIC. Set the DTCST.DTCST bit to 1.

(4) SCI settings

Enable the SCIn_RXI (n = 0 to 4, 9) interrupt by setting the CCR0.RIE bit in the SCI to 1. If a reception error occurs during the SCI receive operation, reception stops. To manage this, use settings that allow the CPU to accept receive error interrupts.

(5) DTC transfer

Each time a reception of 1 byte by the SCI is complete, an SCIn_RXI interrupt is generated to activate the DTC. The DTC transfers the received byte from the RDR of the SCI to the SRAM, after which the DAR register is incremented and the CRA register is decremented.

(6) Interrupt handling

After 128 rounds of data transfer are complete and the value in the CRA register becomes 0, an SCIn_RXI interrupt request is generated for the CPU. Complete the process in the handling routine for this interrupt.

16.6.2 Chain transfer

This section provides an example of chain transfer by the DTC and describes its use in the output of pulses by the General PWM Timer (GPT). You can use chain transfer to transfer PWM timer compare data and change the period of the PWM timer for the GPT.

For the first of the chain transfers, normal transfer mode is specified for transfer to the GPTm.GTCCRC register (m = 320 to 329). For the second transfer, normal transfer mode is specified for transfer to the GPTm.GTCCRE register (m = 320 to 329). For the third transfer of the chained transfer, normal transfer mode for transfer to the GPTm.GTPBR register (m = 320 to 329) is specified. This is because clearing of the activation source and generation of an interrupt on completion of the specified number of transfers are restricted to the third of the chain transfers, that is, transfer while MRB.CHNE = 0.

The following example shows how to use the counter overflow interrupt with the GPT320.GTPR register as an activating source for the DTC.

(1) First transfer information setting

Set up transfer to the GPT320.GTCCRC register.

1. In the MRA register, select incrementation of the source address (MRA.SM[1:0] = 10b).
2. Set the transfer to normal transfer mode (MRA.MD[1:0] = 00b) and word-sized transfer (MRA.SZ[1:0] = 10b).
3. In the MRB register, select the destination address as fixed (MRB.DM[1:0] = 00b) and set up chain transfer (MRB.CHNE = 1 and MRB.CHNS = 0).
4. Set the SAR register to the first address of the data table.
5. Set the DAR register to the address of the GPT320.GTCCRC register.
6. Set the CRAH and CRAL registers to the size of the data table. The CRB register can be set to any value.

(2) Second transfer information setting

Set up for transfer to the GPT320.GTCCRE register.

1. In the MRA register, select incrementation of the source address (MRA.SM[1:0] = 10b).
2. Set the transfer to normal transfer mode (MRA.MD[1:0] = 00b) and word-sized transfer (MRA.SZ[1:0] = 10b).
3. In the MRB register, select the destination address as fixed (MRB.DM[1:0] = 00b) and set up chain transfer (MRB.CHNE = 1, MRB.CHNS = 0).
4. Set the SAR register to the first address of the data table.
5. Set the DAR register to the address of the GPT320.GTCCRE register.
6. Set the CRAH and CRAL registers to the size of the data table. The CRB register can be set to any value.

(3) Third transfer information set

Set up transfer to the GPT320.GTPBR register.

1. In the MRA register, select incrementation of the source address (MRA.SM[1:0] = 10b).
2. Set the transfer to normal transfer mode (MRA.MD[1:0] = 00b) and word-sized transfer (MRA.SZ[1:0] = 10b).
3. In the MRB register, select the destination address as fixed (MRB.DM[1:0] = 00b) and set up single data transfer per interrupt (MRB.CHNE = 0, MRB.DISEL = 0). The MRB.DTS bit can be set to any value.
4. Set the SAR register to the first address of the data table.
5. Set the DAR register to the address of the GPT320.GTPBR register.
6. Set the CRA register to the size of the data table. The CRB register can be set to any value.

(4) Transfer information assignment

Place the transfer information for use in the transfer to the GPT320.GTPBR immediately after the transfer control information for use in the GPT320.GTCCRC and GPT320.GTCCRE registers.

(5) DTC vector table

In the DTC vector table, set the address where the transfer control information for use in transfer to the GPT320.GTCCRC and GPT320.GTCCRE registers starts.

(6) ICU setting and DTC module activation

1. Set the ICU.IELSRn.DTCE bit associated with the GPT320 counter overflow interrupt.
2. Set the ICU.IELSRn.IELS[8:0] bits and specify the GPT320 counter overflow.
3. Set the DTCST.DTCST bit to 1.

(7) GPT settings

1. Set the GPT320.GTIOR register so that the GTCCRA and GTCCRB registers operate as output compare registers.
2. Set the default PWM timer compare values in the GPT320.GTCCRA and GPT320.GTCCRB registers and the next PWM timer compare values in the GPT320.GTCCRC and GPT320.GTCCRE registers.
3. Set the default PWM timer period values in the GPT320.GTPR register and the next PWM timer period values in the GPT320.GTPBR register.
4. Set 1 to the output bit in PmnPFS.PDR, and set 00011b to the Peripheral Select bits in PmnPFS.PSEL[4:0].

(8) GPT activation

Set the GPT320.GTSTR.CSTRT bits to 1 to start the GPT320.GTCNT counter.

(9) DTC transfer

Each time a GPT320 counter overflow is generated with the GPT320.GTPR register, the next PWM timer compare values are transferred to the GPT320.GTCCRC and GPT320.GTCCRE registers. The setting for the next PWM timer period is transferred to the GPT320.GTPBR register.

(10) Interrupt handling

After the specified rounds of data transfer are complete, for example when the value in the CRA register for GPT transfer becomes 0, a GPT320 counter overflow interrupt request is issued for the CPU. Complete the process for this interrupt in the handling routine.

16.6.3 Chain Transfer when Counter = 0

The second data transfer is performed only when the transfer counter is set to 0 in the first data transfer, and the first data transfer information is repeatedly changed in the second transfer. Chain transfer enables transfers to be repeated 256 times or more.

The following procedure shows an example of configuring a 1-KB input buffer, where the input buffer is set so that its lower address starts with 0x00. Figure 16.13 shows a chain transfer when the counter = 0.

1. Set the normal transfer mode to input data for the first data transfer. Set the following:
   (a) Transfer source address = fixed.
   (b) CRA register = 0x0200 (512) times.
   (c) MRB.CHNE bit = 1 (chain transfer is enabled).
   (d) MRB.CHNS bit = 1 (chain transfer is performed only when the transfer counter is 0).
   (e) MRB.DISEL bit = 0 (an interrupt request to the CPU is generated when the specified data transfer completes).
2. Prepare the upper 8-bit address of the start address at every 512 times of the transfer destination address for the first data transfer in different area such as the flash. For example, when setting the input buffer to 0x8000 to 0x83FF, prepare 0x82 and 0x80.

3. For the second data transfer:
   (a) Set the repeat transfer mode (with transfer source and destination address = fixed.) to reset the transfer counter of the first data transfer.
   (b) Specify the CRA register in the first transfer information area for the transfer destination.
   (c) Set the MRB.CHNE bit = 1 (chain transfer is enabled).
   (d) Set the MRB.CHNS bit = 0 (select continuous chain transfer).
   (e) Set the MRB.DISEL bit = 0 (an interrupt request to the CPU is generated when the specified data transfer completes).
   (f) CRA register = 0x0101 (The transfer count is 1).
4. For the third data transfer:
   (a) Set the repeat transfer mode (with the source as the repeat area) to reset the transfer destination address of the first data transfer.
   (b) Specify the upper 8 bits of the DAR register in the first transfer information area for the transfer destination.
   (c) Set the MRB.CHNE bit = 0 (chain transfer is disabled).
   (d) Set the MRB.DISEL bit = 0 (an interrupt request to the CPU is generated when the specified data transfer completes).
   (e) When setting the input buffer to 0x8000 to 0x83FF, also set the transfer counter to 2.
5. The first data transfer is performed by an interrupt 512 times. When the transfer counter of the first data transfer becomes 0, the second data transfer starts. Set the transfer counter of the first data transfer to 0x0200. The lower 8 bits of the transfer destination address and the transfer counter of the first data transfer becomes 0x0200.
6. The second data transfer is performed by an interrupt 1 times. When the transfer counter of the first data transfer becomes 0, the third data transfer starts. Set the upper 8 bits of the transfer destination address of the first data transfer to 0x82. The lower 8 bits of the transfer destination address becomes 0x00 and the transfer counter of the first data transfer becomes 0x0200.
7. In succession, the first data transfer is performed by an interrupt 512 times as specified for the first data transfer. When the transfer counter of the first data transfer becomes 0, the second data transfer starts. Set the transfer counter of the first data transfer to 0x0200. The lower 8 bits of the transfer destination address and the transfer counter of the first data transfer becomes 0x0200.
8. The second data transfer is performed by an interrupt 1 times. When the transfer counter of the first data transfer becomes 0, the third data transfer starts. Set the upper 8 bits of the transfer destination address of the first data transfer to 0x80. The lower 8 bits of the transfer destination address becomes 0x00 and the transfer counter of the first data transfer becomes 0x0200.
9. Steps 5 to 8 are repeated indefinitely. Because the second data transfer is in repeat transfer mode, no interrupt request to the CPU is generated.

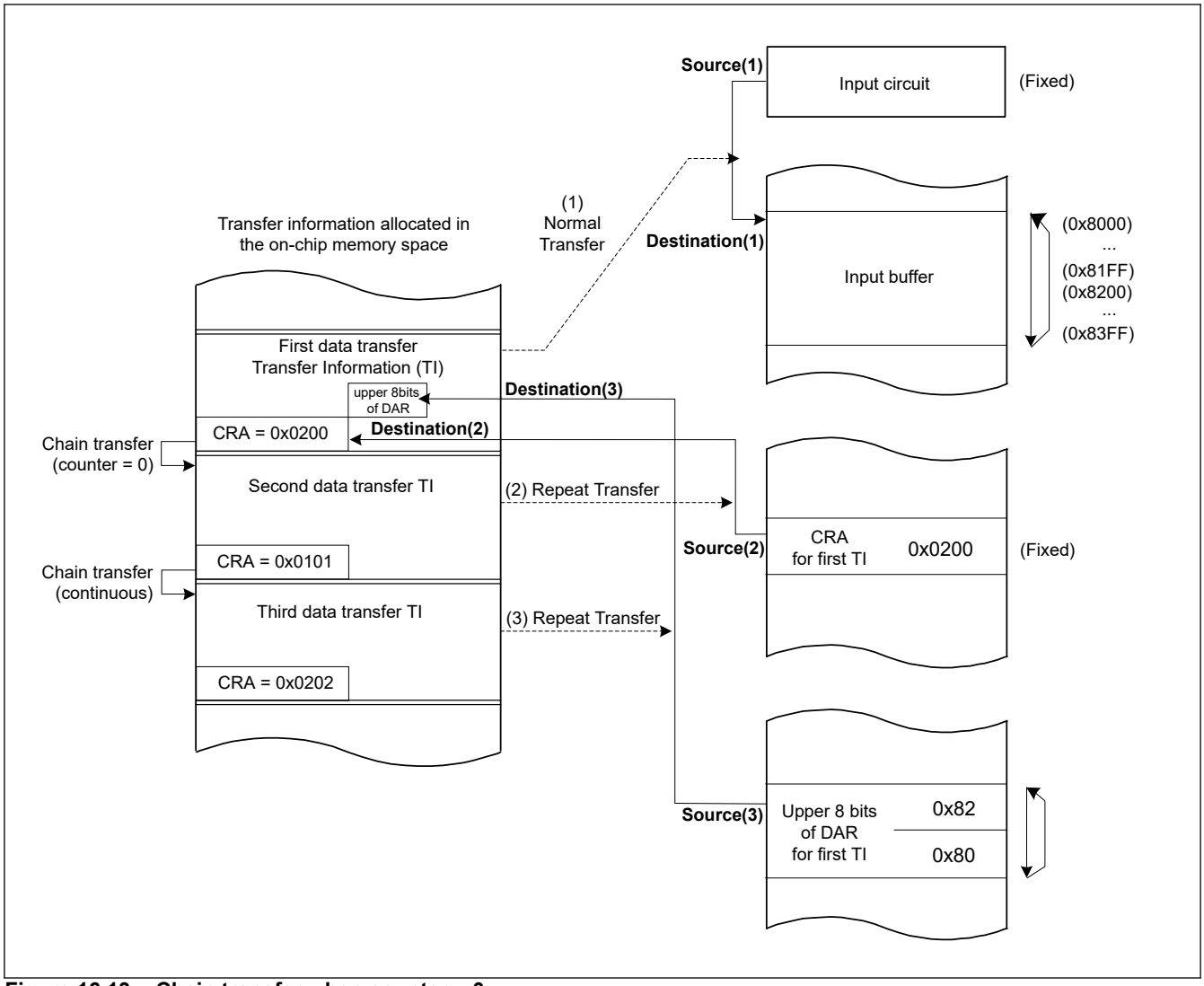


Figure 16.13 Chain transfer when counter = 0

16.7 Processing on DTC Transfer Error

If the access error occurs during DTC transfer, the DTC immediately stops access during transfer. To stop only the vector number that caused the error, inform the vector number that caused the error to the ICU and clear the corresponding ICU setting. After that, if there is a request other than the vector number which caused the error, it will be re-arbitration as it is. The condition under which the transfer error occurs is indicated when TrustZone Filter in DTC detects a violation.

The error response is informed to ICU when the transfer error occurs. ICU clears the ICU.IELSRn of the corresponding vector number which caused the transfer error. Furthermore, it generates an error response detection interrupt to notify that an error has occurred by DMAC/DTC transfer. (section 16.8.2. Interrupt Request of Transfer Error). Write back to SRAM is not performed.

When the Master TrustZone Filter error occurs, the Slave TrustZone error occurs or the Master MPU error occurs, it is possible to confirm the error information of DTC by selecting NMI. The DTC error vector register is cleared by selecting reset. Under the conditions where NMI is generated due to transfer error in DTC, two interrupts(NMI and DMA_TRANSERR) are generated. In this case, NMI always responds first.

The error response detection interrupt request (DMA_TRANSERR) occurs when the Slave Bus error or the Illegal Access error occurs. Furthermore, it occurs after NMI when the error response detection interrupt request (DMA_TRANSERR) is not cleared in NMI handler.

section 16.7.1. Processing on NMI handler describes how to confirm the error information of the DTC in the NMI handler. section 16.7.2. Processing on Error response detection interrupt request (DMA_TRANSERR) handler describes how to confirm the error information of the DTC in the DMA_TRANSERR handler.

Interrupts and the error information generated due to transfer errors are shown in section 16.8.2. Interrupt Request of Transfer Error.

16.7.1 Processing on NMI handler

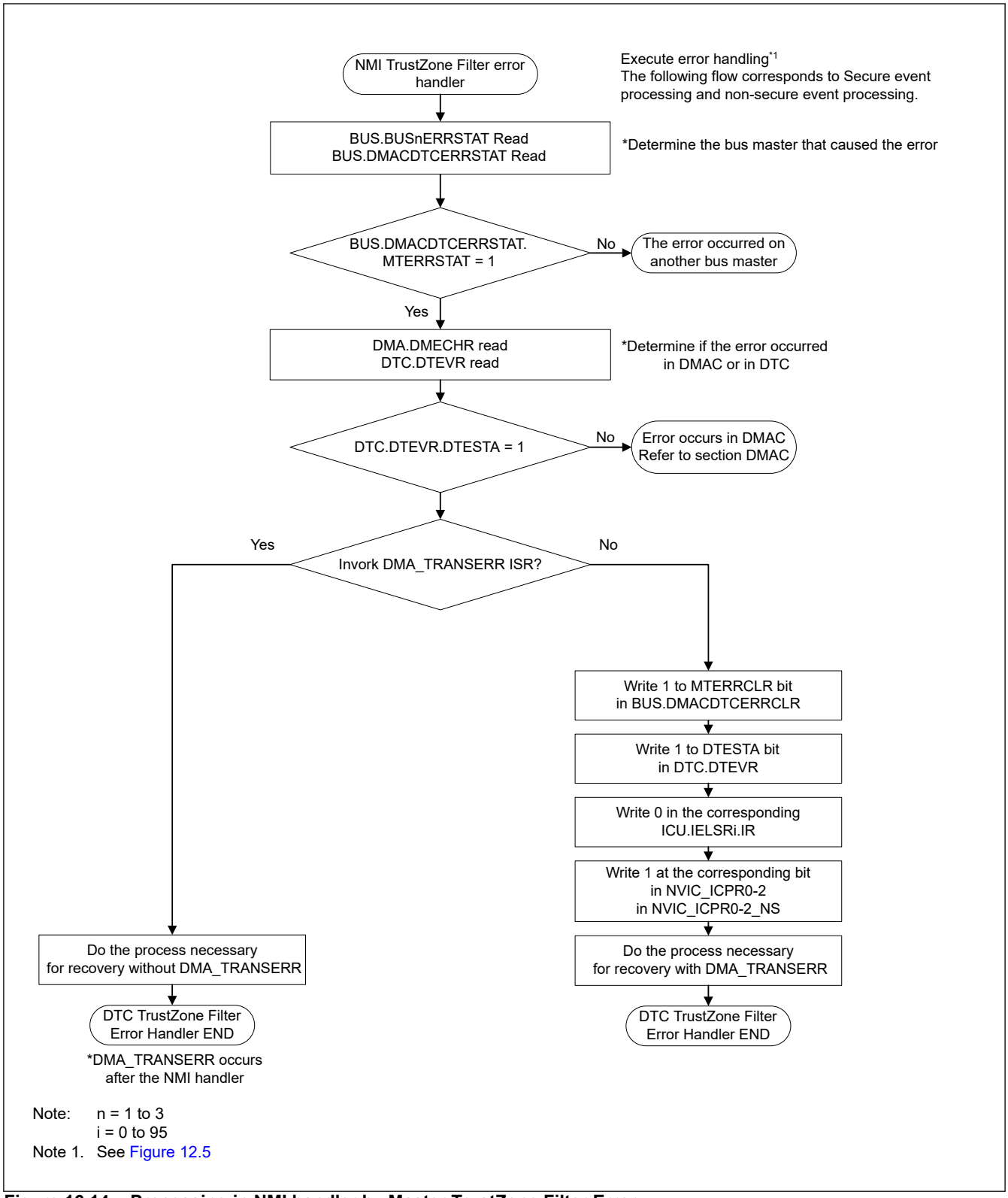
The cause of NMI due to the DMA transfer error is the Master TrustZone Filter error, the Slave TrustZone Filter error or the Master MPU error. When NMI occurs due to the DTC transfer error, the error response detection interrupt request (DMA_TRANSERR) will occur after the end of NMI handler. It is possible to confirm the cause of the error and the DTC vector number in which the error occurred. When NMI occurs, perform the necessary processing according to the flow described in the ICU chapter.

Figure 16.14 shows the flow for confirm the vector number that caused the Master TrustZone Filter Error in DTC

Figure 16.15 shows the flow for confirm the vector number that caused the Slave TrustZone Filter Error in DTC

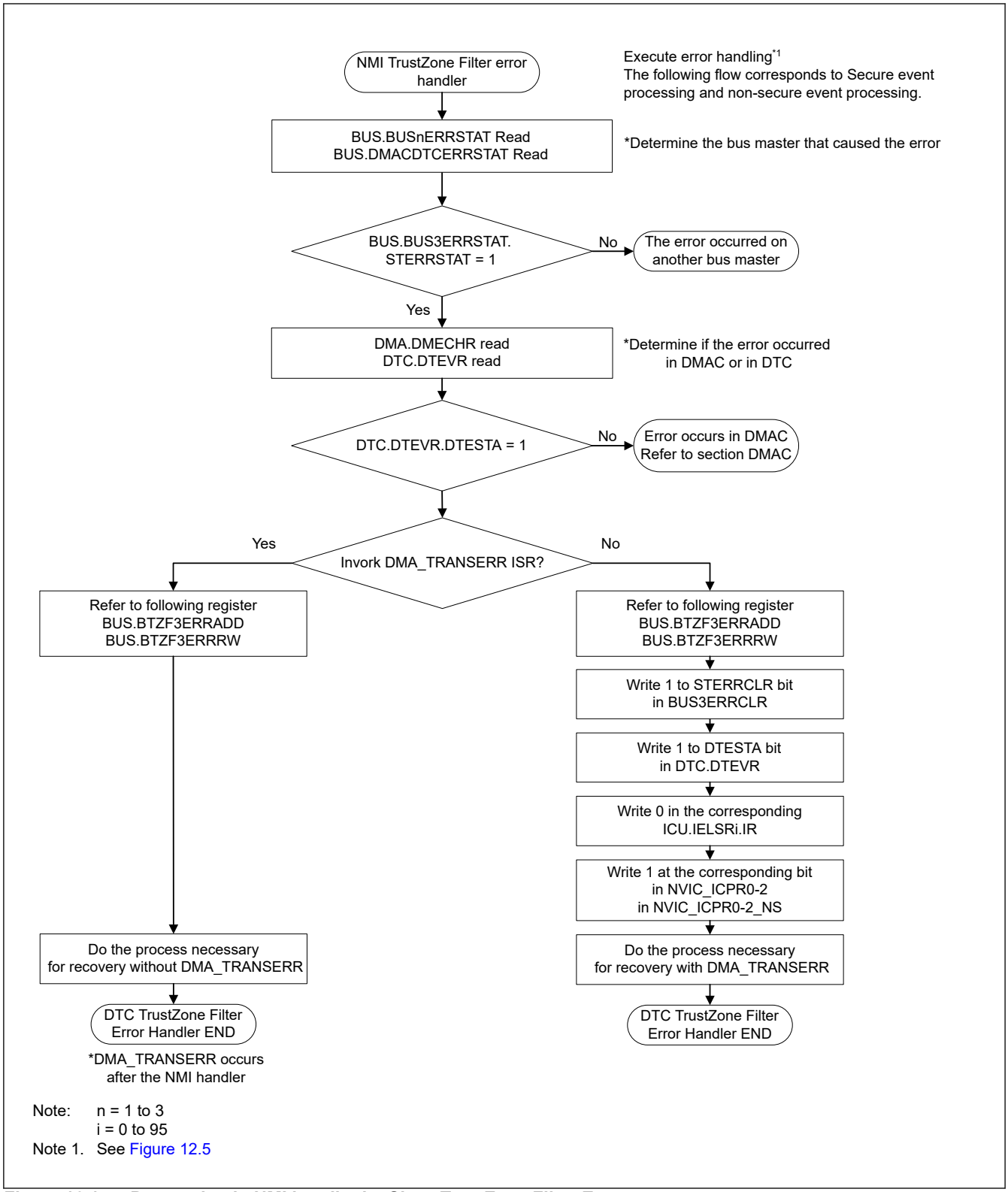
Figure 16.16 shows the flow for confirm the vector number and Security Attribute that caused the Master MPU error in DTC

If completing all processing in NMI handler, it is possible to clear the error response detection interrupt request (DMA_TRANSERR) that occurs subsequently.



Note: n = 1 to 3
i = 0 to 95
Note 1. See Figure 12.5

Figure 16.14 Processing in NMI handler by Master TrustZone Filter Error



Note: n = 1 to 3
i = 0 to 95
Note 1. See Figure 12.5

Figure 16.15 Processing in NMI handler by Slave TrustZone Filter Error

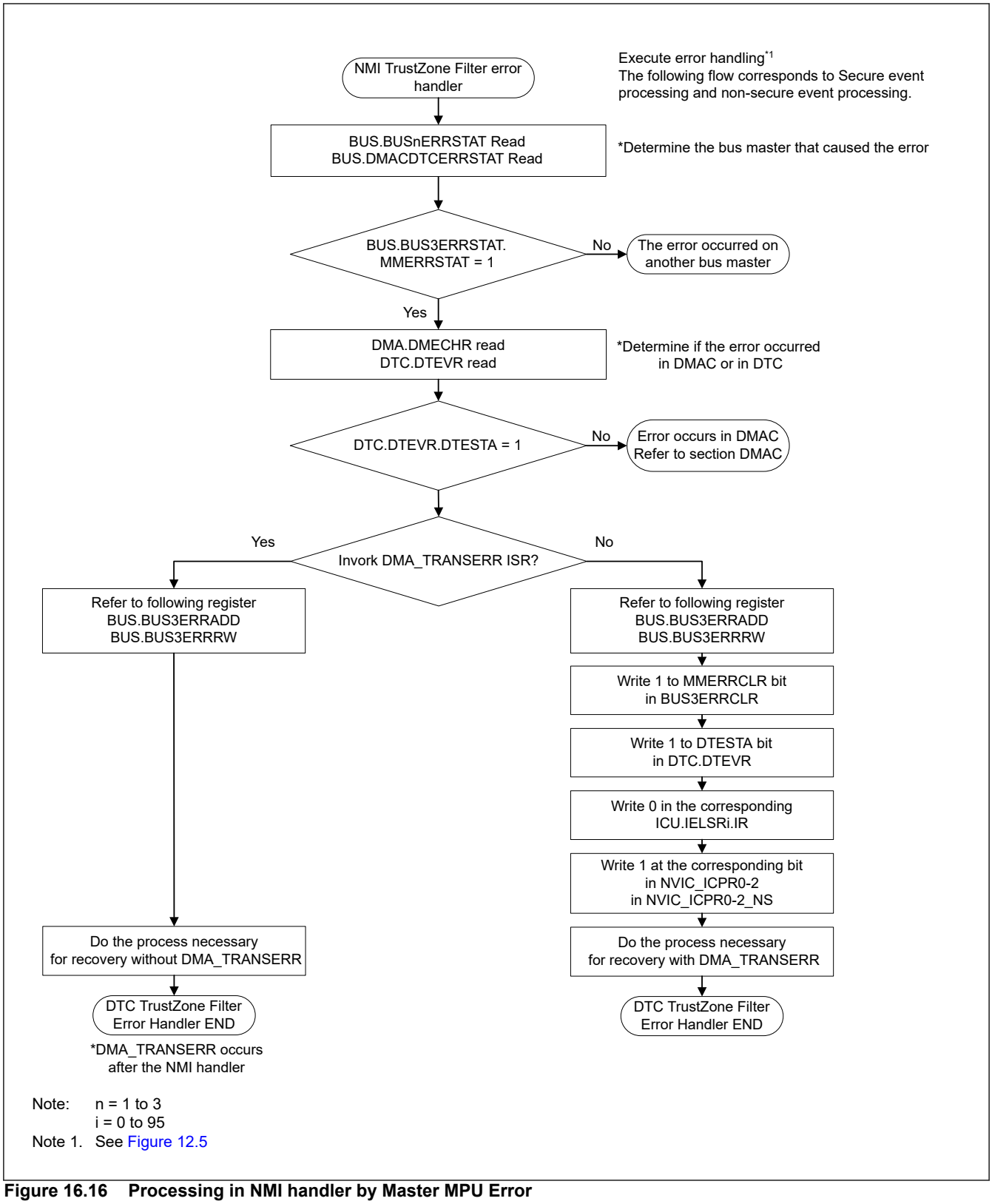


Figure 16.16 Processing in NMI handler by Master MPU Error

16.7.2 Processing on Error response detection interrupt request (DMA_TRANSERR) handler

The cause of error response detection interrupt request (DMA_TRANSERR) due to DMA transfer error is the Slave Bus Error or Illegal Access Error. Also, it occurs after the NMI handler error response detection interrupt request (DMA_TRANSERR) is not cleared by the NMI handler.

It is possible to confirm the cause of the error and the vector number of DTC in which the error occurred.

Error cause confirmation procedure is shown Figure 16.17.

Figure 16.18 shows the flow for confirm the vector number that caused the Master TrustZone Filter Error in DTC

Figure 16.19 shows the flow for confirm the vector number that caused the Slave TrustZone Filter Error in DTC

Figure 16.20 shows the flow for confirm the vector number and Security Attribute that caused the Master MPU Error in DTC

Figure 16.21 shows the flow for confirm the vector number and Security Attribute that caused the Slave Bus Error in DTC

Figure 16.22 shows the flow for confirm the vector number and Security Attribute that caused the Illegal Access Error in DTC

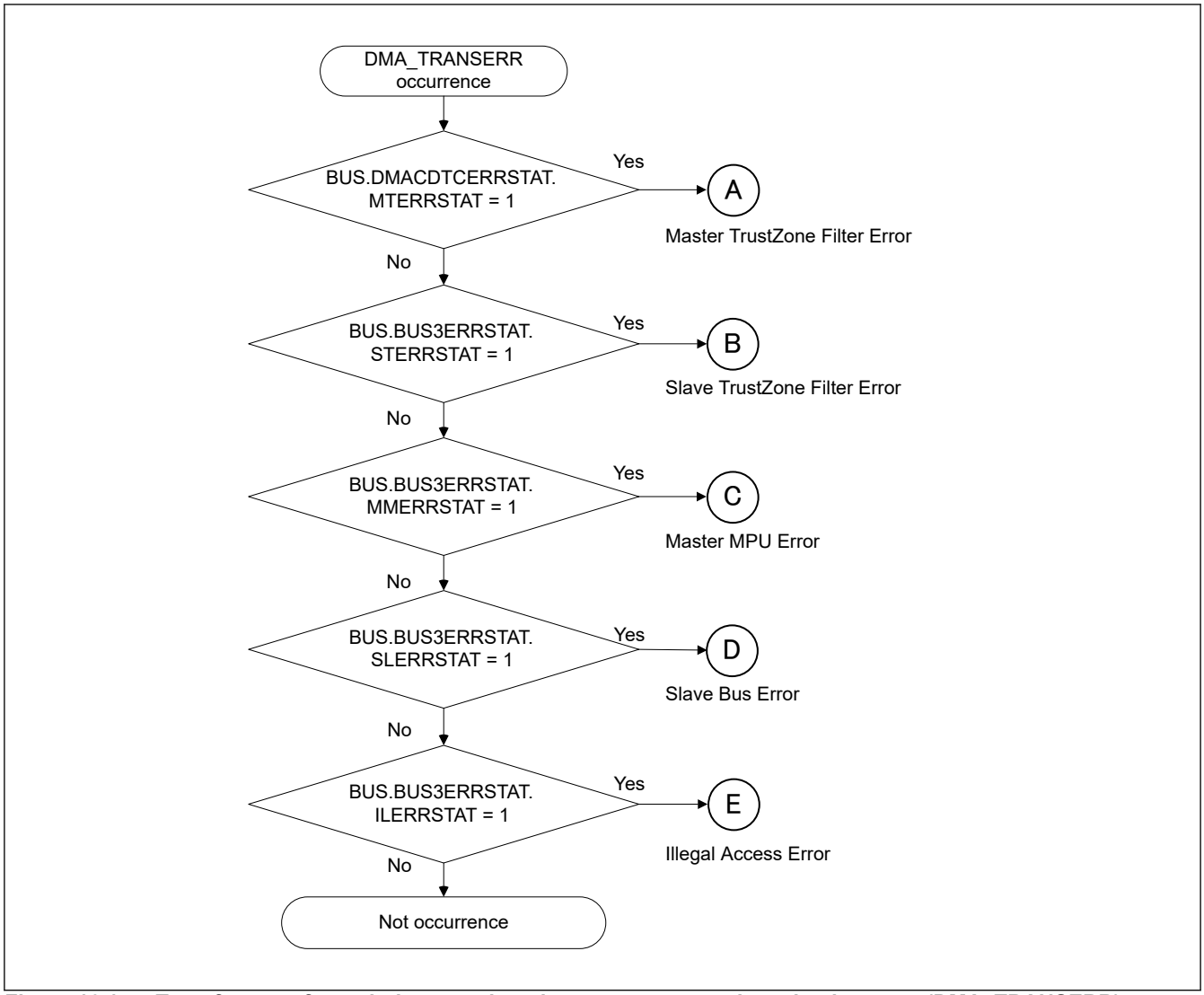
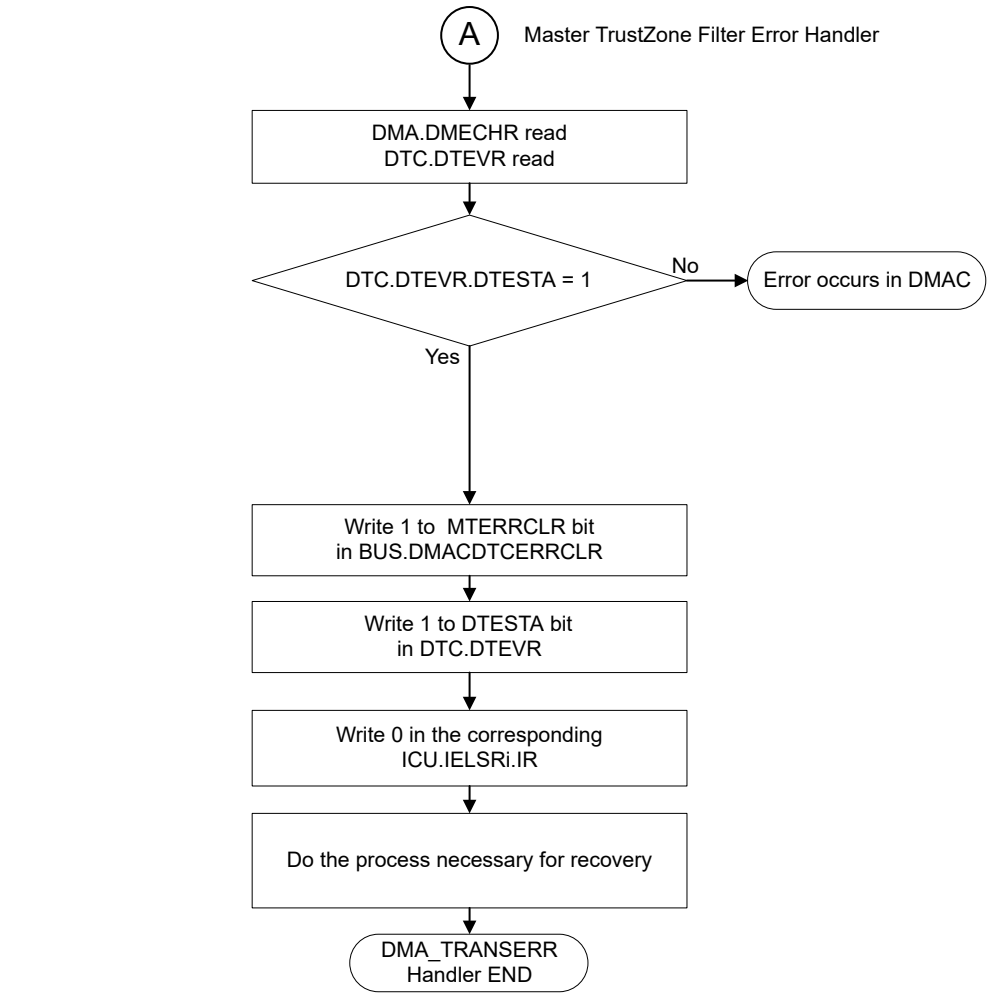


Figure 16.17 Transfer error factor judgment when the error response detection interrupt (DMA_TRANSERR) occurs



Note: i = 0 to 95

Figure 16.18 Processing in DMA_TRANSERR handler by Master TrustZone Filter Error

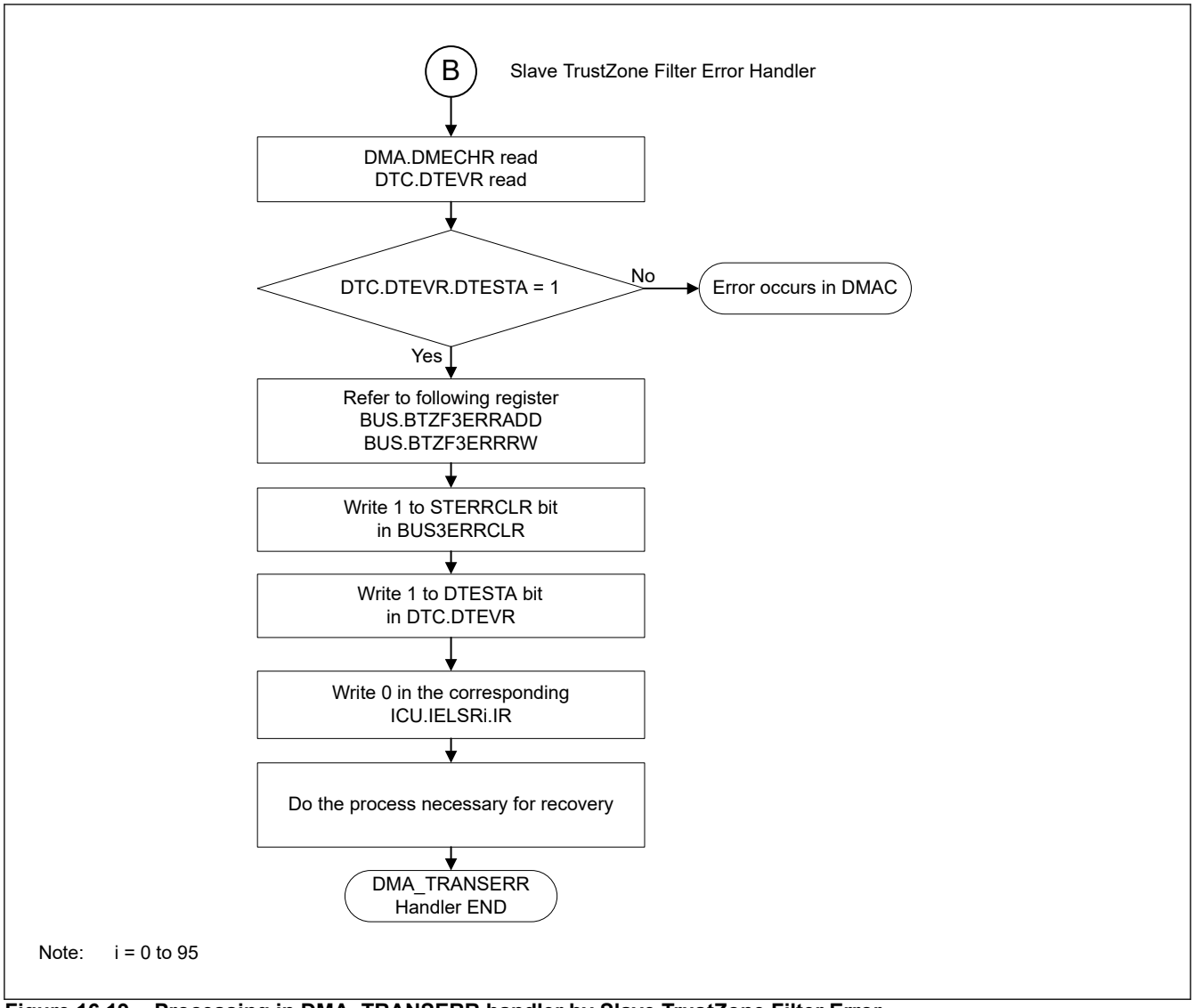
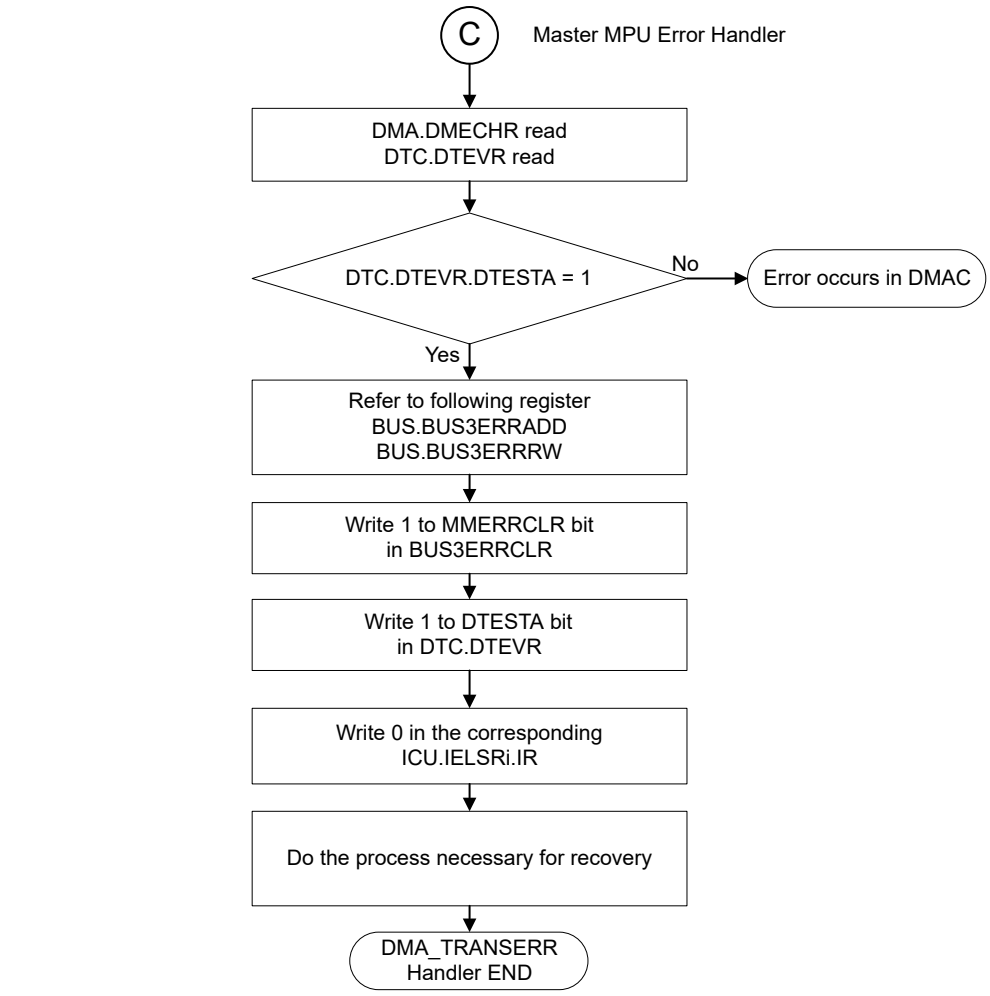
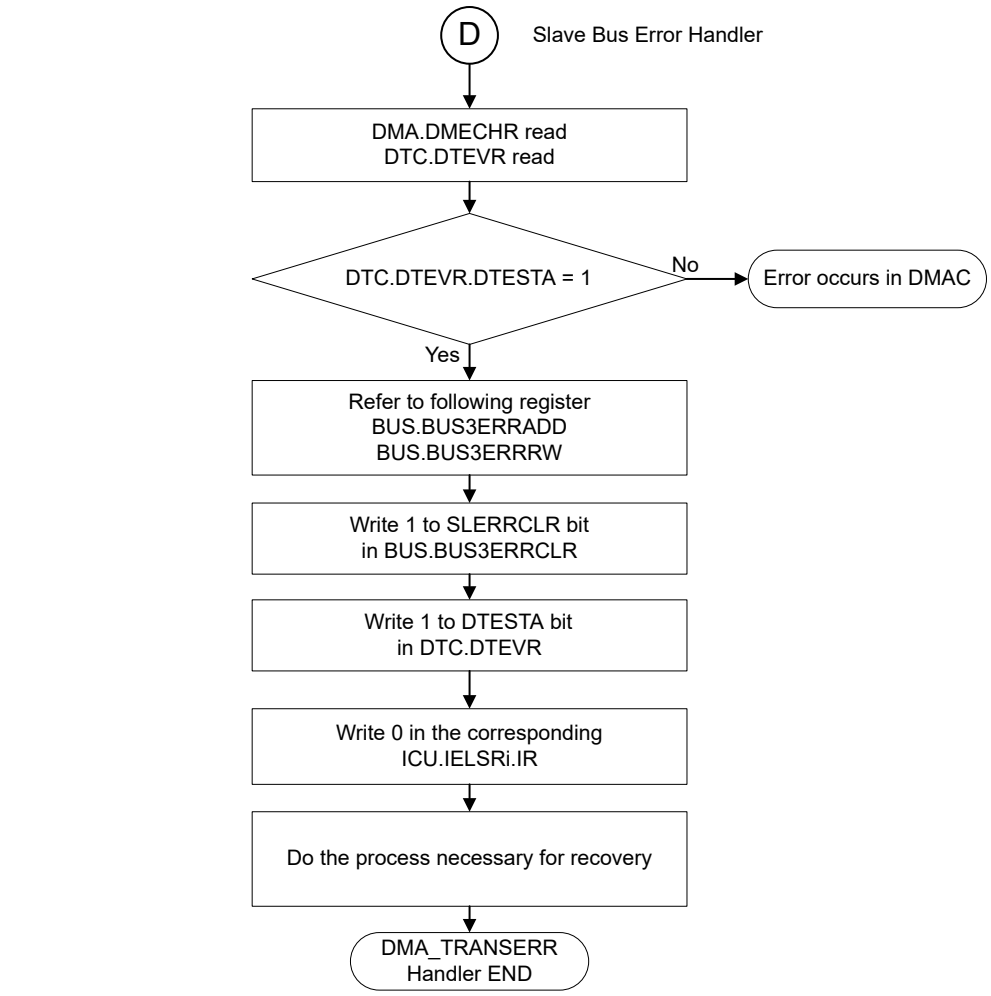


Figure 16.19 Processing in DMA_TRANSERR handler by Slave TrustZone Filter Error



Note: i = 0 to 95

Figure 16.20 Processing in DMA_TRANSERR handler by Master MPU Error



Note: i = 0 to 95

Figure 16.21 Processing in DMA_TRANSERR handler by Slave Bus Error

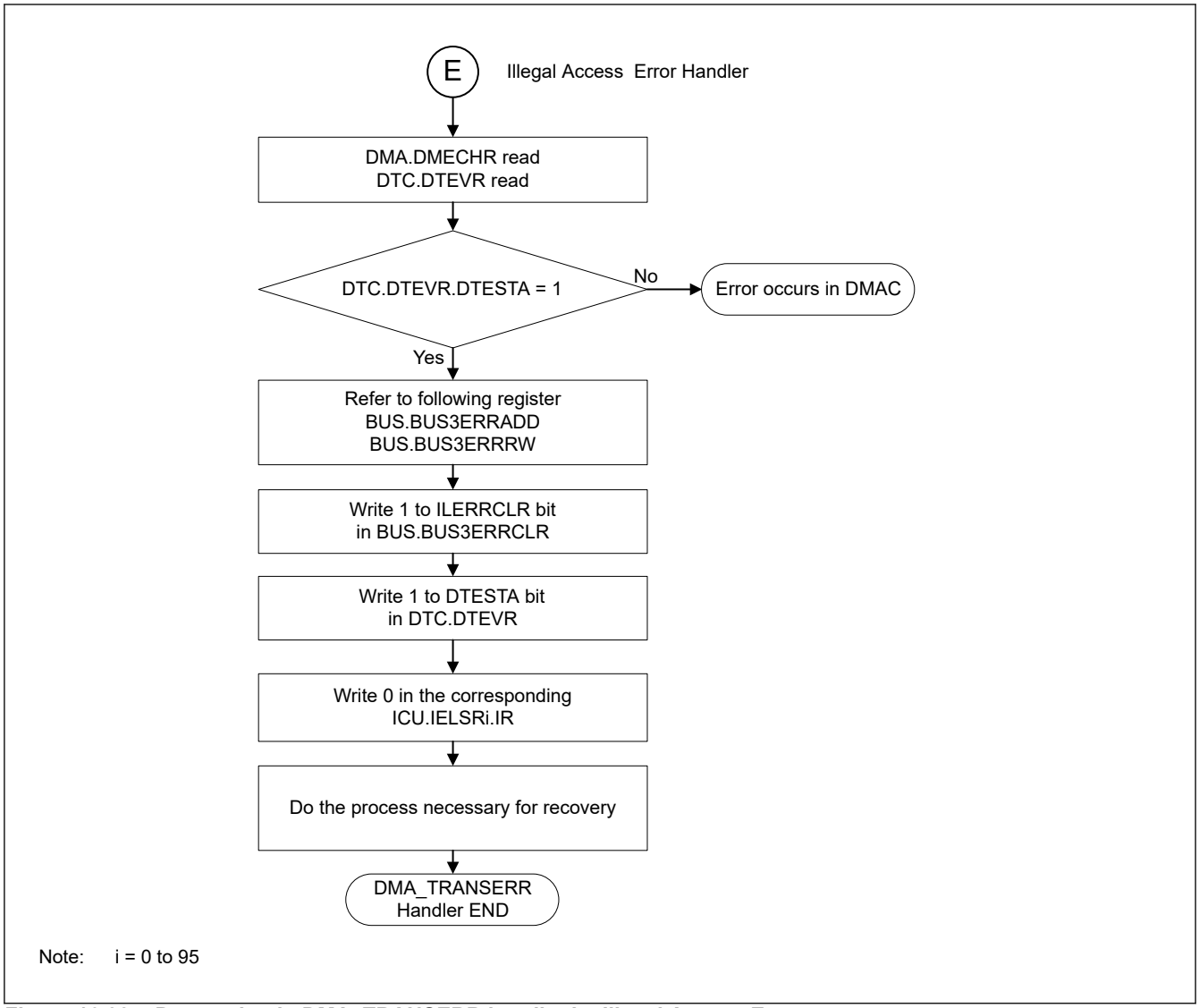


Figure 16.22 Processing in DMA_TRANSERR handler by Illegal Access Error

16.8 Interrupt

16.8.1 Interrupt Request of Transfer End

When the DTC completes data transfer of the specified count or when data transfer with MRB.DISEL set to 1 is complete, a DTC activation source generates an interrupt to the CPU. Two types of interrupt are available: interrupts triggered by a DTC activation (per channel) and an interrupt triggered by the event signal DTC_COMPLETE (common to all channels). Interrupts to the CPU are controlled according to the settings in the NVIC and the ICU.IELSRn.IELS[8:0] bits. See section 12, Interrupt Controller Unit (ICU). The DTC prioritizes activation sources by granting the smaller interrupt vector numbers higher priority. The priority of interrupts to the CPU is determined by the NVIC priority.

16.8.2 Interrupt Request of Transfer Error

The error response detection interrupt request (DMA_TRANSERR) is generated from the DMAC/DTC when the transfer error is detected during DTC transfer. The types of interrupts that occur when the DTC transfer error occurs are listed in the Table 16.10. The Table 16.10 also shows error information stored when a transfer error occurs.

Table 16.10 Interrupt and error information due to DMAC transfer error cause

Transfer error factor	NMI/RESET*1 Request	Interrupt Request	Bus Error Status	Error Address Error R/W	Error Channel Information
Master TrustZone Filter (in DMAC/DTC)	ICU.NMISR.TZFST*1	DMA_TRANSERR	BUS.DMACDTCERRSTAT.MTERRSTAT*1	—	DTC.DTEVR
Slave TrustZone Filter	ICU.NMISR.TZFST*1	DMA_TRANSERR	BUS.BUS3ERRSTAT.STERRSTAT*1	BUS.BTZF3ERRADD BUS.BTZF3ERRRW	DTC.DTEVR
Master MPU	ICU.NMISR. BUSMST	DMA_TRANSERR	BUS.BUS3ERRSTAT.MMERRSTAT	BUS.BUS3ERRADD BUS.BUS3ERRRW	DTC.DTEVR
Slave Bus Error	—*2	DMA_TRANSERR	BUS.BUS3ERRSTAT.SLERRSTAT*2	BUS.BUS3ERRADD BUS.BUS3ERRRW	DTC.DTEVR
Illegal Access Error	—*2	DMA_TRANSERR	BUS.BUS3ERRSTAT.ILERRSTAT*2	BUS.BUS3ERRADD BUS.BUS3ERRRW	DTC.DTEVR

Note 1. Interrupt generated, when NMI request selected as the operation after detection of the Master MPU error and The TrustZone Filter error. By confirming BUS.BUS3ERRSTAT and BUS.DMACDTCERRSTAT, judge whether it is the Master or the Slave.
Note 2. If the error response detection interrupt (DMA_TRANSERR) occurs and NMI of the Master MPU or NMI of the TrustZone Filter has not occurred, treat it as the Illegal address access error or the Slave Bus Error. It can be judged also by BUS.BUS3ERRSTAT and BUS. DMACDTCERRSTAT.

Note that if the bus error occurs when writing the last data of transfer, the transfer end event and the error response detection interrupt (DMA_TRANSERR) occurs.

16.9 Event Link

The DTC can produce an event link request on completion of one transfer request.

16.10 Low Power Consumption Function

Before transitioning to the module-stop state, Software Standby mode without Snooze mode transition, Deep Software Standby mode, set the DTCST.DTCST bit to 0, and then perform the operations described in the following sections. The DTC is available in Snooze mode by setting the SYSTEM.SNZCR.SNZDTCEN bit to 1. See section 10, Low Power Modes.

(1) Module-Stop Function

Writing 1 to the MSTPCRA.MSTPA22 bit enables the module-stop function of the DTC. If a DTC transfer is in progress when 1 is written to the MSTPCRA.MSTPA22 bit, the transition to the module-stop state proceeds after the DTC transfer ends. While the MSTPCRA.MSTPA22 bit is 1, accessing the DTC registers is prohibited. Writing 0 to the MSTPCRA.MSTPA22 bit releases the DTC from the module-stop state.

(2) Software Standby Mode and Deep Software Standby Mode

Use the settings described in section 10.7.1. Transitioning to Software Standby Mode or section 10.9.1. Transitioning to Deep Software Standby Mode.

If DTC transfer operations are in progress when the WFI instruction is executed, the transition to Software Standby mode or Deep Software Standby mode is executed after the completion of the DTC transfer.

(3) Snooze Mode

When the snooze control circuit receives a snooze request in Software Standby mode, the MCU transitions to Snooze mode. See section 10.8.1. Transition to Snooze Mode. DTC operation in Snooze mode can be selected in the SYSTEM.SNZCR.SNZDTCEN bit. If DTC operation is enabled in Snooze mode, before transitioning to Software Standby mode, set the DTCST.DTCST bit to 1. To return to Software Standby mode through DTC, set SYSTEM.SNZEDCR0.DTCZRED or SYSTEM.SNZEDCR0.DTCNZRED to 1. See section 10.8.3. Returning from Snooze Mode to Software Standby Mode. SYSTEM.SNZEDCR0.DTCZRED enables or disables a snooze end request on completion of the last DTC transmission, detected on DTC transmission completion when CRA and CRB are 0. SYSTEM.SNZEDCR0.DTCNZRED enables or disables a snooze end request on a not last DTC transmission completion (CRA and CRB are not 0), detected on DTC transmission completion when CRA and CRB are not 0. The DTC activation request from the ICU is stopped during Software Standby mode but not stopped during Snooze mode.

(4) Notes on Low Power Consumption Function

For the WFI instruction and the register setting procedure, see section 10, Low Power Modes.

To perform a DTC transfer after returning from a low power mode without a Snooze mode transition, set the DTCST.DTCST bit to 1 again.

To use a request that is generated in Software Standby mode as an interrupt request to the CPU but not as a DTC activation request, specify the CPU as the interrupt request destination as described in section 12.4.1. Detecting Interrupts, then execute the WFI instruction. If DTC operation is enabled in Snooze mode, do not use the module-stop function of the DTC.

16.11 Usage Notes

16.11.1 Transfer Information Start Address

You must set multiples of 4 for the transfer information start addresses in the vector table. Otherwise, such addresses are accessed with their lowest 2 bits regarded as 00b.

17. Event Link Controller (ELC)

This is the ELC_B version of the ELC peripheral module.

ELC_B is referred to as ELC in this chapter.

17.1 Overview

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

Table 17.1 lists the ELC specifications, and Figure 17.1 shows a block diagram.

Table 17.1 ELC Specifications

Item	Description
Event link function	215 types of event signals can be directly connected to modules. The ELC generates the ELC event signal, and events that activate the DTC.
Module-stop function	Module-stop state can be set.
TrustZone Filter	Security attribution can be set for each registers

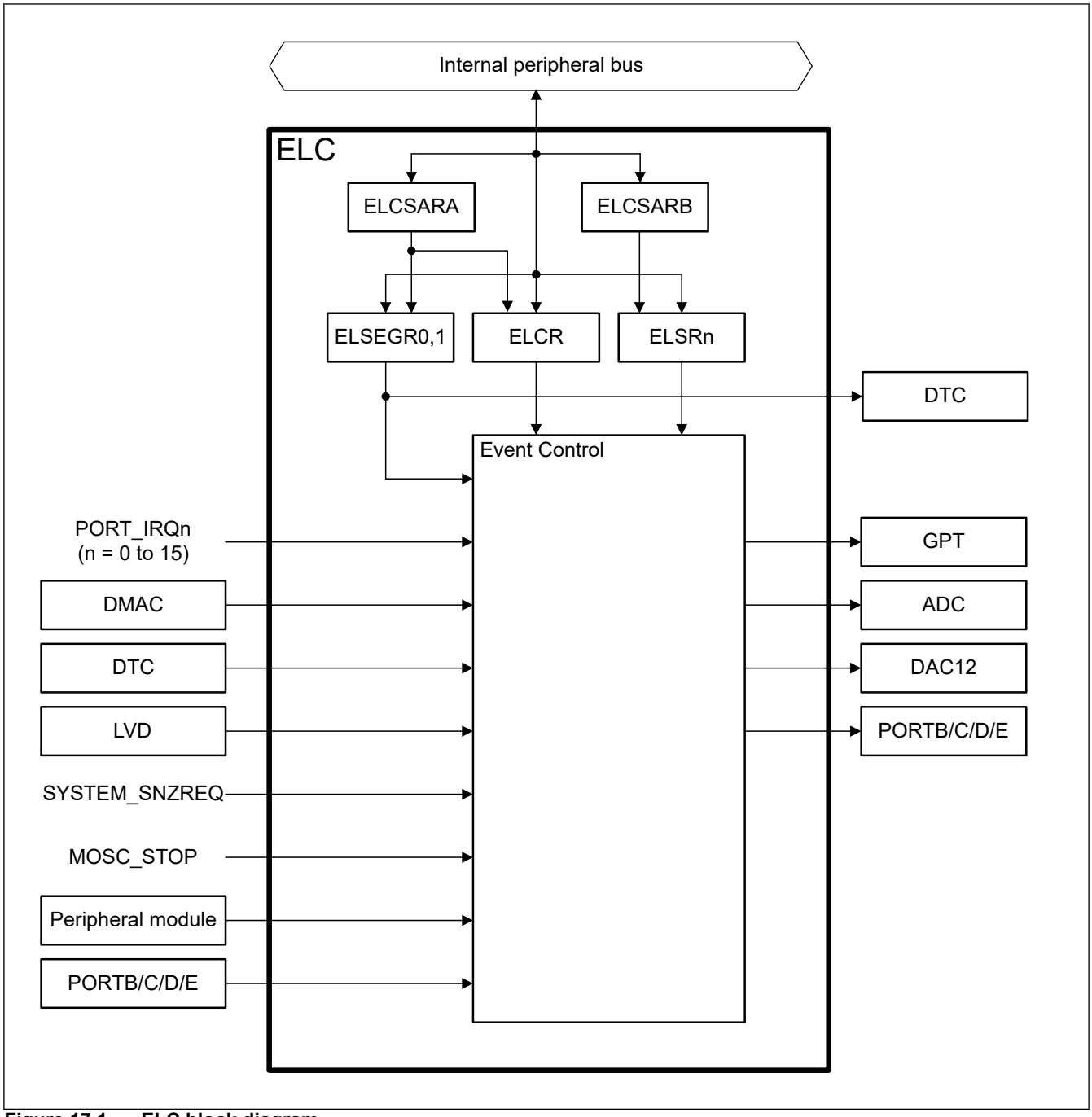


Figure 17.1 ELC block diagram

17.2 Register Descriptions

17.2.1 ELCR : Event Link Controller Register

Base address: ELC_B = 0x4008_2000

Offset address: 0x00

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

Bit	Symbol	Function	R/W
6:0	—	These bits are read as 0. The write value should be 0.	R/W
7	ELCON	All Event Link Enable 0: ELC function is disabled. 1: ELC function is enabled.	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

The ELCR register controls the ELC operation.

17.2.2 ELSEGRn : Event Link Software Event Generation Register n (n = 0, 1)

Base address: ELC_B = 0x4008_2000

Offset address: 0x04 + 0x04 × n

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

Bit	Symbol	Function	R/W
0	SEG	Software Event Generation 0: Normal operation 1: Software event is generated.	W
5:1	—	These bits are read as 0. The write value should be 0.	R/W
6	WE	SEG Bit Write Enable 0: Write to SEG bit disabled. 1: Write to SEG bit enabled.	R/W
7	WI	ELSEGR Register Write Disable 0: Write to ELSEGR register enabled. 1: Write to ELSEGR register disabled.	W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

SEG bit (Software Event Generation)

When 1 is written to the SEG bit while the WE bit is 1, a software event is generated. This bit is read as 0. Even when 1 is written to this bit, data is not stored. The WE bit must be set to 1 before writing to this bit.

A software event can trigger a linked DTC event.

WE bit (SEG Bit Write Enable)

The SEG bit can only be written to when the WE bit is 1. Clear the WI bit to 0 before writing to this bit.

[Setting condition]

- If 1 is written to this bit while the WI bit is 0, this bit becomes 1.

[Clearing condition]

- If 0 is written to this bit while the WI bit is 0, this bit becomes 0.

WI bit (ELSEGR Register Write Disable)

The ELSEGR register can only be written to when the write value to the WI bit is 0. This bit is read as 1. Before setting the WE or SEG bit, the WI bit must be set to 0.

17.2.3 ELSRn : Event Link Setting Register n (n = 0 to 7, 12 to 17, 19 to 24, 28, 29)

Base address: ELC_B = 0x4008_2000

Offset address: 0x20 + 0x04 × n

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

Bit	Symbol	Function	R/W
8:0	ELS[8:0]	Event Link Select 0x000: Event output disabled for the associated peripheral module 0x001: Number setting for the event signal to be linked ⋮ 0x1DB: Number setting for the event signal to be linked Others: Settings prohibited	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

The ELSRn register specifies an event signal to be linked to each peripheral module. Table 17.2 shows the association between the ELSRn register and the peripheral modules. Table 17.3 shows the association between the event signal names set in the ELSRn register and the signal numbers.

Table 17.2 Association between the ELSRn registers and peripheral functions (1 of 2)

Register name	Peripheral function (module)	Event name
ELSR0	GPT (A)	ELC_GPTA
ELSR1	GPT (B)	ELC_GPTB
ELSR2	GPT (C)	ELC_GPTC
ELSR3	GPT (D)	ELC_GPTD
ELSR4	GPT (E)	ELC_GPTE
ELSR5	GPT (F)	ELC_GPTF
ELSR6	GPT (G)	ELC_GPTG
ELSR7	GPT (H)	ELC_GPTH
ELSR12	DAC12 channel 0	ELC_DA0
ELSR13	DAC12 channel 1	ELC_DA1
ELSR14	PORTB	ELC_PORTB
ELSR15	PORTC	ELC_PORTC
ELSR16	PORTD	ELC_PORTD
ELSR17	PORTE	ELC_PORTE
ELSR19	ADCA0	ELC_AD00
ELSR20	ADCB0	ELC_AD01
ELSR21	ADCC0	ELC_AD02
ELSR22	ADCA1	ELC_AD10
ELSR23	ADCB1	ELC_AD11
ELSR24	ADCC1	ELC_AD12
ELSR28	DAC12 channel 2	ELC_DA2

Table 17.2 Association between the ELSRn registers and peripheral functions (2 of 2)

Register name	Peripheral function (module)	Event name
ELSR29	DAC12 channel 3	ELC_DA3

Table 17.3 Association between event signal names set in ELSRn.ELS[8:0] bits and signal numbers (1 of 6)

Event number	Interrupt request source	Name	Description
0x001	Port	PORT_IRQ0*1	External pin interrupt 0
0x002		PORT_IRQ1*1	External pin interrupt 1
0x003		PORT_IRQ2*1	External pin interrupt 2
0x004		PORT_IRQ3*1	External pin interrupt 3
0x005		PORT_IRQ4*1	External pin interrupt 4
0x006		PORT_IRQ5*1	External pin interrupt 5
0x007		PORT_IRQ6*1	External pin interrupt 6
0x008		PORT_IRQ7*1	External pin interrupt 7
0x009		PORT_IRQ8*1	External pin interrupt 8
0x00A		PORT_IRQ9*1	External pin interrupt 9
0x00B		PORT_IRQ10*1	External pin interrupt 10
0x00C		PORT_IRQ11*1	External pin interrupt 11
0x00D		PORT_IRQ12*1	External pin interrupt 12
0x00E		PORT_IRQ13*1	External pin interrupt 13
0x00F		PORT_IRQ14*1	External pin interrupt 14
0x010		PORT_IRQ15*1	External pin interrupt 15
0x020	DMAC0	DMAC0_INT	DMAC transfer end 0
0x021	DMAC1	DMAC1_INT	DMAC transfer end 1
0x022	DMAC2	DMAC2_INT	DMAC transfer end 2
0x023	DMAC3	DMAC3_INT	DMAC transfer end 3
0x024	DMAC4	DMAC4_INT	DMAC transfer end 4
0x025	DMAC5	DMAC5_INT	DMAC transfer end 5
0x026	DMAC6	DMAC6_INT	DMAC transfer end 6
0x027	DMAC7	DMAC7_INT	DMAC transfer end 7
0x02A	DTC	DTC_DTCEND*2	DTC transfer end
0x038	LVD	LVD_LVD1	Voltage monitor 1 interrupt
0x039		LVD_LVD2	Voltage monitor 2 interrupt
0x03B	MOSC	MOSC_STOP	Mail Clock oscillation stop
0x03C	LPW	SYSTEM_SNZREQ*2 *3	Snooze enrty
0x040	AGT0	AGT0_AGTI	AGT interrupt
0x041		AGT0_AGTCMAI	Compare match A
0x042		AGT0_AGTCMBI	Compare match B
0x043	AGT1	AGT1_AGTI	AGT interrupt
0x044		AGT1_AGTCMAI	Compare match A
0x045		AGT1_AGTCMBI	Compare match B
0x052	IWDT	IWDT_NMIUNDF	IWDT underflow
0x053	WDT	WDT_NMIUNDF	WDT underflow

Table 17.3 Association between event signal names set in ELSRn.ELS[8:0] bits and signal numbers (2 of 6)

Event number	Interrupt request source	Name	Description
0x08F	ACMPHS	ACMP_HS0	High-Speed Analog Comparator interruput 0
0x090		ACMP_HS1	High-Speed Analog Comparator interruput 1
0x091		ACMP_HS2	High-Speed Analog Comparator interruput 2
0x092		ACMP_HS3	High-Speed Analog Comparator interruput 3
0x0B1	I/O Port	IOPORT_GROUPB	Port B event
0x0B2		IOPORT_GROUPC	Port C event
0x0B3		IOPORT_GROUPD	Port D event
0x0B4		IOPORT_GROUPE	Port E event
0x0B5	ELC	ELC_SWEVT0	Software event 0
0x0B6		ELC_SWEVT1	Software event 1
0x0C0	GPT0	GPT0_CCMPA	Compare match A
0x0C1		GPT0_CCMPB	Compare match B
0x0C2		GPT0_CMPC	Compare match C
0x0C3		GPT0_CMPD	Compare match D
0x0C4		GPT0_CMPE	Compare match E
0x0C5		GPT0_CMPF	Compare match F
0x0C6		GPT0_OVF	Overflow
0x0C7		GPT0_UDF	Underflow
0x0C8		GPT0_PC	Cycle count function end
0x0CA		GPT0_ADTRGA	A/D converter start request A
0x0CB		GPT0_ADTRGB	A/D converter start request B
0x0CC	GPT1	GPT1_CCMPA	Compare match A
0x0CD		GPT1_CCMPB	Compare match B
0x0CE		GPT1_CMPC	Compare match C
0x0CF		GPT1_CMPD	Compare match D
0x0D0		GPT1_CMPE	Compare match E
0x0D1		GPT1_CMPF	Compare match F
0x0D2		GPT1_OVF	Overflow
0x0D3		GPT1_UDF	Underflow
0x0D4		GPT1_PC	Cycle count function end
0x0D6		GPT1_ADTRGA	A/D converter start request A
0x0D7		GPT1_ADTRGB	A/D converter start request B

Table 17.3 Association between event signal names set in ELSRn.ELS[8:0] bits and signal numbers (3 of 6)

Event number	Interrupt request source	Name	Description
0x0D8	GPT2	GPT2_CCMPA	Compare match A
0x0D9		GPT2_CCMPB	Compare match B
0x0DA		GPT2_CMPC	Compare match C
0x0DB		GPT2_CMPD	Compare match D
0x0DC		GPT2_CMPE	Compare match E
0x0DD		GPT2_CMPF	Compare match F
0x0DE		GPT2_OVF	Overflow
0x0DF		GPT2_UDF	Underflow
0x0E0		GPT2_PC	Cycle count function end
0x0E2		GPT2_ADTRGA	A/D converter start request A
0x0E3		GPT2_ADTRGB	A/D converter start request B
0x0E4	GPT3	GPT3_CCMPA	Compare match A
0x0E5		GPT3_CCMPB	Compare match B
0x0E6		GPT3_CMPC	Compare match C
0x0E7		GPT3_CMPD	Compare match D
0x0E8		GPT3_CMPE	Compare match E
0x0E9		GPT3_CMPF	Compare match F
0x0EA		GPT3_OVF	Overflow
0x0EB		GPT3_UDF	Underflow
0x0EC		GPT3_PC	Cycle count function end
0x0EE		GPT3_ADTRGA	A/D converter start request A
0x0EF		GPT3_ADTRGB	A/D converter start request B
0x0F0	GPT4	GPT4_CCMPA	Compare match A
0x0F1		GPT4_CCMPB	Compare match B
0x0F2		GPT4_CMPC	Compare match C
0x0F3		GPT4_CMPD	Compare match D
0x0F4		GPT4_CMPE	Compare match E
0x0F5		GPT4_CMPF	Compare match F
0x0F6		GPT4_OVF	Overflow
0x0F7		GPT4_UDF	Underflow
0x0FA		GPT4_ADTRGA	A/D converter start request A
0x0FB		GPT4_ADTRGB	A/D converter start request B
0x0FC	GPT5	GPT5_CCMPA	Compare match A
0x0FD		GPT5_CCMPB	Compare match B
0x0FE		GPT5_CMPC	Compare match C
0x0FF		GPT5_CMPD	Compare match D
0x100		GPT5_CMPE	Compare match E
0x101		GPT5_CMPF	Compare match F
0x102		GPT5_OVF	Overflow
0x103		GPT5_UDF	Underflow
0x106		GPT5_ADTRGA	A/D converter start request A
0x107		GPT5_ADTRGB	A/D converter start request B

Table 17.3 Association between event signal names set in ELSRn.ELS[8:0] bits and signal numbers (4 of 6)

Event number	Interrupt request source	Name	Description
0x108	GPT6	GPT6_CCMPA	Compare match A
0x109		GPT6_CCMPB	Compare match B
0x10A		GPT6_CMPC	Compare match C
0x10B		GPT6_CMPD	Compare match D
0x10C		GPT6_CMPE	Compare match E
0x10D		GPT6_CMPF	Compare match F
0x10E		GPT6_OVF	Overflow
0x10F		GPT6_UDF	Underflow
0x112		GPT6_ADTRGA	A/D converter start request A
0x113		GPT6_ADTRGB	A/D converter start request B
0x114	GPT7	GPT7_CCMPA	Compare match A
0x115		GPT7_CCMPB	Compare match B
0x116		GPT7_CMPC	Compare match C
0x117		GPT7_CMPD	Compare match D
0x118		GPT7_CMPE	Compare match E
0x119		GPT7_CMPF	Compare match F
0x11A		GPT7_OVF	Overflow
0x11B		GPT7_UDF	Underflow
0x11E		GPT7_ADTRGA	A/D converter start request A
0x11F		GPT7_ADTRGB	A/D converter start request B
0x120	GPT8	GPT8_CCMPA	Compare match A
0x121		GPT8_CCMPB	Compare match B
0x122		GPT8_CMPC	Compare match C
0x123		GPT8_CMPD	Compare match D
0x124		GPT8_CMPE	Compare match E
0x125		GPT8_CMPF	Compare match F
0x126		GPT8_OVF	Overflow
0x127		GPT8_UDF	Underflow
0x12A		GPT8_ADTRGA	A/D converter start request A
0x12B		GPT8_ADTRGB	A/D converter start request B
0x12C	GPT9	GPT9_CCMPA	Compare match A
0x12D		GPT9_CCMPB	Compare match B
0x12E		GPT9_CMPC	Compare match C
0x12F		GPT9_CMPD	Compare match D
0x130		GPT9_CMPE	Compare match E
0x131		GPT9_CMPF	Compare match F
0x132		GPT9_OVF	Overflow
0x133		GPT9_UDF	Underflow
0x136		GPT9_ADTRGA	A/D converter start request A
0x137		GPT9_ADTRGB	A/D converter start request B
0x138	GPT	GPT_UVWEDGE	UVW edge event

Table 17.3 Association between event signal names set in ELSRn.ELS[8:0] bits and signal numbers (5 of 6)

Event number	Interrupt request source	Name	Description
0x140	IIC0	IIC0_RX	Rx Data buffer full
0x141		IIC0_TX	Tx Data buffer empty
0x142		IIC0_TEND	Transmit end
0x145		IIC0_COM	Communication event
0x146	IIC1	IIC1_RX	Rx Data buffer full
0x147		IIC1_TX	Tx Data buffer empty
0x148		IIC1_TEND	Transmit end
0x14A		IIC1_COM	Communication event
0x159	ADC	ADC_ADI0	A/D scan end for scan group 0
0x15A		ADC_ADI1	A/D scan end for scan group 1
0x15B		ADC_ADI2	A/D scan end for scan group 2
0x15E		ADC_CCMPM0	Composite compare match 0
0x167		ADC_ADI3	A/D scan end for scan group 3
0x168		ADC_ADI4	A/D scan end for scan group 4
0x169		ADC_ADI5678	A/D scan end for scan group 5 to 8
0x16C		ADC_CCMPM1	Composite compare match 1
0x18D	SCI0	SCI0_RXI*4	Receive data full
0x18E		SCI0_TXI*4	Transmit data empty
0x18F		SCI0_TEI*4	Transmit end
0x190		SCI0_ERI	Receive error
0x191		SCI0_AED	Effective edge detection
0x193		SCI0_AM	Address match event
0x195	SCI1	SCI1_RXI*4	Received data full
0x196		SCI1_TXI*4	Transmit data empty
0x197		SCI1_TEI*4	Transmit end
0x198		SCI1_ERI	Receive error
0x199		SCI1_AED	Effective edge detection
0x19B		SCI1_AM	Address match event
0x19C	SCI2	SCI2_RXI*4	Received data full
0x19D		SCI2_TXI*4	Transmit data empty
0x19E		SCI2_TEI*4	Transmit end
0x19F		SCI2_ERI	Receive error
0x1A0		SCI2_AED	Effective edge detection
0x1A2		SCI2_AM	Address match event
0x1A3	SCI3	SCI3_RXI*4	Received data full
0x1A4		SCI3_TXI*4	Transmit data empty
0x1A5		SCI3_TEI*4	Transmit end
0x1A6		SCI3_ERI	Receive error
0x1A7		SCI3_AED	Effective edge detection
0x1A9		SCI3_AM	Address match event

Table 17.3 Association between event signal names set in ELSRn.ELS[8:0] bits and signal numbers (6 of 6)

Event number	Interrupt request source	Name	Description
0x1AA	SCI4	SCI4_RXI*4	Received data full
0x1AB		SCI4_TXI*4	Transmit data empty
0x1AC		SCI4_TEI*4	Transmit end
0x1AD		SCI4_ERI	Receive error
0x1AE		SCI4_AED	Effective edge detection
0x1B0		SCI4_AM	Address match event
0x1B1	SCI9	SCI9_RXI*4	Received data full
0x1B2		SCI9_TXI*4	Transmit data empty
0x1B3		SCI9_TEI*4	Transmit end
0x1B4		SCI9_ERI	Receive error
0x1B5		SCI9_AED	Effective edge detection
0x1B7		SCI9_AM	Address match event
0x1C4	SPI0	SPI0_SPRI	Receive buffer full
0x1C5		SPI0_SPTI	Transmit buffer empty
0x1C6		SPI0_SPII	Idle
0x1C7		SPI0_SPEI	Error
0x1C8		SPI0_SPCEND	Communication complete event
0x1C9	SPI1	SPI1_SPRI	Receive buffer full
0x1CA		SPI1_SPTI	Transmit buffer empty
0x1CB		SPI1_SPII	Idle
0x1CC		SPI1_SPEI	Error
0x1CD		SPI1_SPCEND	Transmission complete event
0x1DB	DOC	DOC_DOPCI*2	Data operation circuit interrupt

Note 1. Only pulse (edge detection) is supported.
Note 2. This event can occur in Snooze mode.
Note 3. ELSR14 to ELSR17 and ELSR19 to ELSR24 can select this event.
Note 4. This event is not supported in FIFO mode.

17.2.4 ELCSARA : Event Link Controller Security Attribution Register A

Base address: ELC_B = 0x4008_2000

Offset address: 0xE0

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	ELSEGR1	ELSEGR0	ELCR
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	ELCR	Event Link Controller Register Security Attribution Target register: ELCR 0: Secure 1: Non-secure	R/W
1	ELSEGR0	Event Link Software Event Generation Register 0 Security Attribution 0: Secure 1: Non-secure	R/W
2	ELSEGR1	Event Link Software Event Generation Register 1 Security Attribution 0: Secure 1: Non-secure	R/W
31:3	—	These bits are read as 1. The write value should be 1.	R/W

Note: Only secure access can write to this register. Both secure access and Non-secure read access are allowed but Non-secure write access is not allowed and TrustZone access error is not generated.
Note: This register is write-protected by PRCR register.

The ELCR register controls operation of the ELC.

17.2.5 ELCSARB : Event Link Controller Security Attribution Register B

Base address: ELC_B = 0x4008_2000

Offset address: 0xE4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	ELSR29	ELSR28	—	—	—	ELSR24	ELSR23	ELSR22	ELSR21	ELSR20	ELSR19	—	ELSR17	ELSR16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ELSR15	ELSR14	ELSR13	ELSR12	—	—	—	—	ELSR7	ELSR6	ELSR5	ELSR4	ELSR3	ELSR2	ELSR1	ELSR0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
7:0	ELSR7 to ELSR0	Event Link Setting Register n Security Attribution Target register: ELSRn (n = 0 to 7) 0: Secure 1: Non-secure	R/W
11:8	—	These bits are read as 1. The write value should be 1.	R/W
17:12	ELSR17 to ELSR12	Event Link Setting Register n Security Attribution Target register: ELSRn (n = 12 to 17) 0: Secure 1: Non-secure	R/W
18	—	This bit is read as 1. The write value should be 1.	R/W
24:19	ELSR24 to ELSR19	Event Link Setting Register n Security Attribution Target register: ELSRn (n = 19 to 24) 0: Secure 1: Non-secure	R/W
27:25	—	These bits are read as 1. The write value should be 1.	R/W
29:28	ELSR29 to ELSR28	Event Link Setting Register n Security Attribution Target register: ELSRn (n = 28, 29) 0: Secure 1: Non-secure	R/W
31:30	—	These bits are read as 1. The write value should be 1.	R/W

Note: Only secure access can write to this register. Both secure access and Non-secure read access are allowed but Non-secure write access is not allowed and TrustZone access error is not generated.
Note: This register is write-protected by PRCR register.

This register specifies the security attribution for the Register ELSRn (n = 0 to 7, 12 to 17, 19 to 24, 28, 29).

17.3 Operation

17.3.1 Relation between Interrupt Handling and Event Linking

Event number for an event link is the same as that for the associated interrupt source. For information on generating event signals, see the explanation in the chapter for each event source module.

17.3.2 Linking Events

When an event occurs and that event is already set as a trigger in the Event Link Setting Register (ELSRn), the associated module is activated. The operation of the module must be set up in advance. Table 17.4 lists the operations of modules when an event occurs.

Table 17.4 Module operations when event occurs

Module	Operations When Event is Input
GPT	• Start counting • Stop counting • Clear counting • Up counting • Down counting • Input capture
DAC12	Start D/A conversion
I/O Ports	• Change pin output based on the EORR (reset) or EOSR (set) • Latch pin state to EIDR • The following ports can be used for the ELC: Port B Port C Port D Port E
ADC	Start A/D conversion
DTC	Start DTC data transfer

17.3.3 Example of Procedure for Linking Events

To link events:

1. Set the operation of the module for which an event is to be linked.
2. Set the appropriate ELSRn.ELS[8:0] bits for the module to be linked.
3. Set the ELCR.ELCON bit to 1 to enable linkage of all events.
4. Configure the module from which an event is output and activate the module. The link between the two modules is now active.
5. To stop event linkage of modules individually, set 0 to the ELSRn.ELS[8:0] bit associated with the modules. To stop linkage of all the events, set the ELCR.ELCON bit to 0.

If event link output from the LVD is to be used, set the ELC after setting the LVD. To disable the LVD, do so after setting 0x00 to the associated ELSRn register.

17.4 Usage Notes

17.4.1 Linking DMAC/DTC Transfer End Signals as Events

When linking the DMAC/DTC transfer end signals as events, do not set the same peripheral module as the DMAC/DTC transfer destination and event link destination. If set, the peripheral module might be started before DMAC/DTC transfer to the peripheral module is complete.

17.4.2 Setting Clocks

To link events, you must enable the ELC and the related modules. The modules cannot operate if the related modules are in the module-stop state or in low power mode in which the module is stopped (Software Standby mode or Deep Software Standby mode).

Some modules can perform in Snooze mode. For more information, see Table 17.3 and section 10, Low Power Modes.

17.4.3 Module-Stop Function Setting

The Module Stop Control Register C (MSTPCRC) can enable or disable ELC operation. The ELC is initially stopped after reset. Releasing the module-stop state enables access to the registers. The ELCON bit must be set to 0 before disabling ELC operation using the Module Stop Control Register. For more information, see Table 17.3 and section 10, Low Power Modes.

17.4.4 ELC Delay Time

In Figure 17.2, module A accesses module B through the ELC. There is a delay time in the ELC between module A and module B. Table 17.5 shows the ELC delay time.

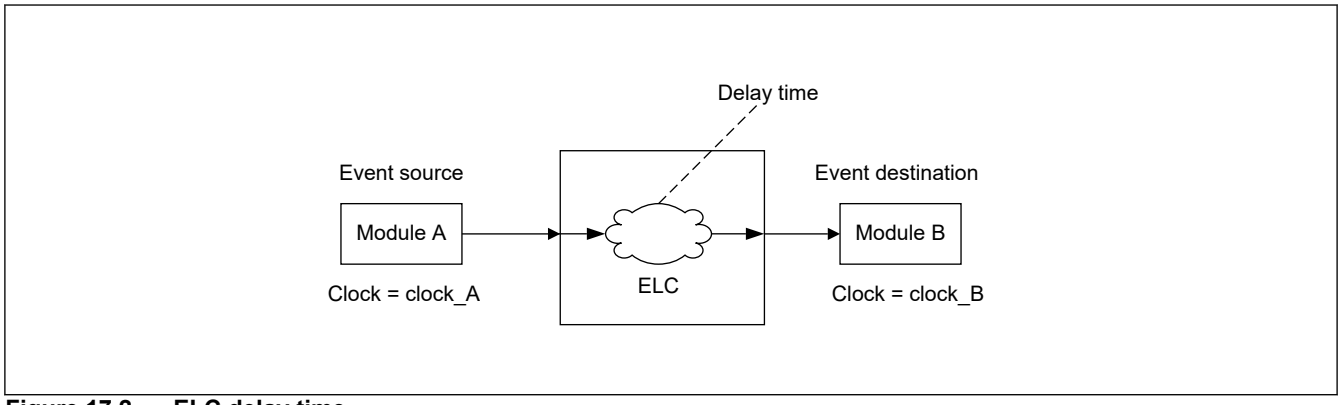


Figure 17.2 ELC delay time

Table 17.5 ELC delay time

Clock domain	Clock frequency	ELC delay time
clock_A = clock_B	clock_A = clock_B	0 cycle
clock_A ≠ clock_B	clock_A = clock_B	1 cycle to 2 cycles
	clock_A > clock_B	1 cycle to 2 cycles of clock_B
	clock_A < clock_B	1 cycle to 2 cycles of clock_A

17.4.5 Interval of Event Request

If the clocks of Event Source and Event Destination are combined as shown in Table 17.6, the Event request may be lost if the interval between one Event request and the next is less than the following value (Event_Interval) for the same Event request signal.

However, this restriction does not apply if the Event Destination is GPT or ADC and uses a different ELSR register.

Table 17.6 shows a combination of clocks with the restricted event interval, and Figure 17.3 shows an example of the event interval for GPT0_ADTRGA.

The event interval is expressed by the following formula.

Event_Interval [ns] = Period_of_Source_clock [ns] × 6 + Period_of_Destination_clock [ns] × 4

Table 17.6 Combination of clocks with the restricted Event Interval

Event Source	Source clock	Event Destination	Destination clock
Other than GPT	PCLKA or PCLKB	GPT	GPTCLK
		ADC	GPTCLK or PCLKC
GPT	PCLKD	ADC	GPTCLK or PCLKC
	GPTCLK	ADC	PCLKA or PCLKC
		DAC12	PCLKA
		I/O Port	PCLKB

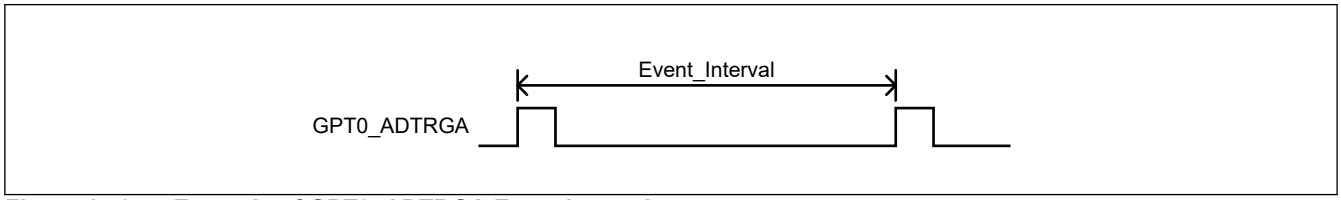


Figure 17.3 Example of GPT0_ADTRGA Event Interval

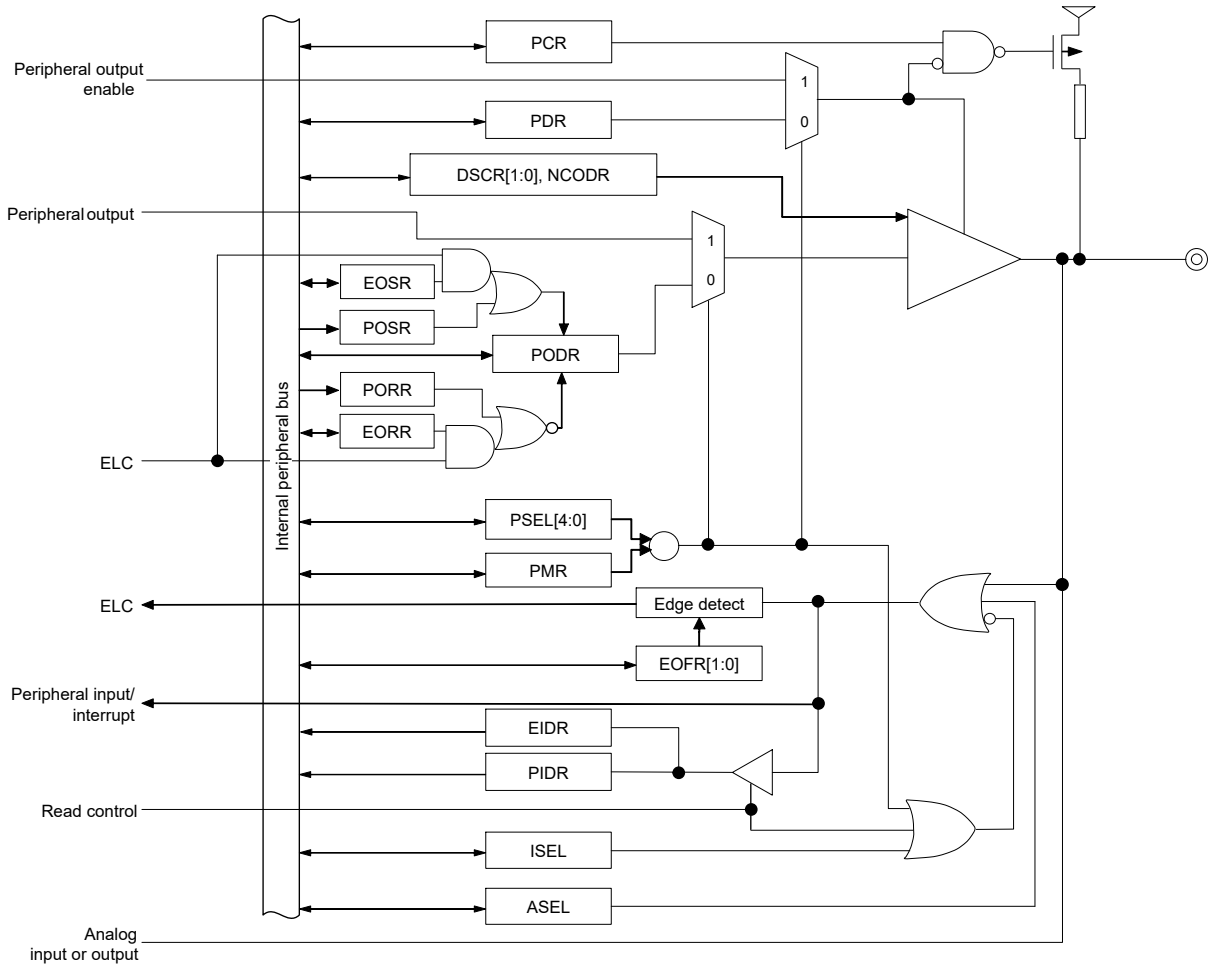
18. I/O Ports

18.1 Overview

The I/O port pins operate as general I/O port pins, I/O pins for peripheral modules, interrupt input pins, analog I/O, or port group function for the ELC.

All pins except PB03 (as TDO of JTAG ports) operate as input pins immediately after a reset, and pin functions are switched by register settings. The I/O ports and peripheral modules for each pin are specified in the associated registers.

Figure 18.1 shows a connection diagram for the I/O port registers. The configuration of the I/O ports differs for different packages. Table 18.1 lists the I/O port specifications by package, and Table 18.2 lists the port functions.



Note: This figure shows a basic port configuration. The configuration differs depending on the ports.

Figure 18.1 Connection diagram for I/O port registers

Table 18.1 I/O port specifications (1 of 2)

Port	Package		Package		Package	
	100 pins	Number of pins	64 pins	Number of pins	48 pins	Number of pins
PORT0	P000 to P002	3	P002	1	—	0
PORT2	P201, P212, P213	3	P201, P212, P213	3	P201, P212, P213	3
PORTA	PA00 to PA15	16	PA00 to PA15	16	PA00 to PA15	16

Table 18.1 I/O port specifications (2 of 2)

Port	Package 100 pins	Number of pins	Package 64 pins	Number of pins	Package 48 pins	Number of pins
PORTB	PB00 to PB10, PB12 to PB15	15	PB00 to PB09, PB12 to PB15	14	PB00, PB01, PB03 to PB09, PB12 to PB15	13
PORTC	PC00 to PC15	16	PC00 to PC15	16	PC13 to PC15	3
PORTD	PD00 to PD15	16	PD02	1	—	0
PORTE	PE00 to PE06, PE08 to PE15	15	—	0	—	0

Table 18.2 I/O port functions

Port	Port name	Input pull-up	Open-drain output	Drive capacity switching	5V tolerant	I/O
PORT0	P000, P001	✓	—	—	—	Input
	P002	—	—	—	—	Input
PORT2	P201	✓	✓	Low	—	Input / Output
	P212, P213	✓	✓	Low, middle, high	—	Input / Output
PORTA	PA00 to PA05	—	—	—	—	Input
	PA06, PA07	✓	—	—	—	Input
	PA08 to PA11	✓	✓	Low, middle, high, High current drive	—	Input / Output
	PA12 to PA15	✓	✓	Low, middle, high	✓	Input / Output
PORTB	PB00, PB01	✓	—	—	—	Input
	PB02	—	—	—	—	Input
	PB03, PB05 to PB09	✓	✓	Low, middle, high	✓	Input / Output
	PB04, PB10	✓	✓	Low, middle, high	—	Input / Output
	PB12 to PB15	✓	✓	Low, middle, high, High current drive	—	Input / Output
PORTC	PC00 to PC05	✓	—	—	—	Input
	PC06 to PC09	✓	✓	Low, middle, high, High current drive	—	Input / Output
	PC10 to PC12	✓	✓	Low, middle, high	✓	Input / Output
	PC13	✓	—	—	—	Input
	PC14, PC15	✓	✓	Low	✓	Input / Output
PORTD	PD00 to PD07	✓	✓	Low, middle, high	✓	Input / Output
	PD08 to PD15	✓	✓	Low, middle, high, High current drive	—	Input / Output
PORTE	PE00, PE01	✓	✓	Low, middle, high	✓	Input / Output
	PE02 to PE06	✓	✓	Low, middle, high, high-speed high-drive	—	Input / Output
	PE08, PE09	✓	✓	Low, middle, high	—	Input / Output
	PE10 to PE15	✓	✓	Low, middle, high, High current drive	—	Input / Output

Note: ✓: Available
—: Setting prohibited

18.2 Register Descriptions

18.2.1 PCNTR1/PODR/PDR : Port Control Register 1

Base address: PORTm = 0x4001_F000 + 0x0020 × m (m = 0, 2, A to E)

Offset address: 0x000 (PCNTR1/PDR)
0x002 (PODR)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	PODR15	PODR14	PODR13	PODR12	PODR11	PODR10	PODR09	PODR08	PODR07	PODR06	PODR05	PODR04	PODR03	PODR02	PODR01	PODR00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PDR15	PDR14	PDR13	PDR12	PDR11	PDR10	PDR09	PDR08	PDR07	PDR06	PDR05	PDR04	PDR03	PDR02	PDR01	PDR00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	PDR15 to PDR00	Pmn Direction 0: Input (functions as an input pin) 1: Output (functions as an output pin)	R/W*1
31:16	PODR15 to PODR00	Pmn Output Data 0: Low output 1: High output	R/W*2

Note: m = 0, 2, A to E, n = 00 to 15

Note 1. If the security attribution is configured as Secure,
- Secure access and Non-secure read access are allowed,
- Non-secure write access is ignored, but TrustZone access error is not generated.

If the security attribution is configured as Non-secure,
- Secure and Non-secure access are allowed.

Note 2. If the security attribution is configured as Secure,
- Secure access is allowed,
- Non-secure read value is 0, but TrustZone access error is not generated,
- Non-secure write access is ignored, but TrustZone access error is not generated.

If the security attribution is configured as Non-secure,
- Secure and Non-secure access are allowed.

The Port Control Register 1 (PCNTR1/PODR/PDR) is a 32-bit or 16-bit read/write register that controls port direction and port output data. The PCNTR1 specifies the port direction and output data, and is accessed in 32-bit units. The PDRn (bits [15:0] in PCNTR1) and PODRn (bits [31:16] in PCNTR1) respectively, are accessed in 16-bit units.

PDRn bits (Pmn Direction)

The PDRn bits select the input or output direction for individual pins on the associated port when the pins are configured as general I/O pins. Each pin on port m is associated with a PORTm.PCNTR1.PDRn bit. The I/O direction can be specified in 1-bit unit. Bits associated with non-existent pins are reserved. Reserved bits are read as 0. The write value should be 0. In the case of input only ports, PDRn bits are reserved. See section 18.1. Overview. The PDRn bit in the PORTm.PCNTR1 register serves the same function as the PDR bit in the PFS.PmnPFS register.

PODRn bits (Pmn Output Data)

The PODRn bits hold data to be output from the general I/O pins. Bits of non-existent port m are reserved. Reserved bits are read as 0. The write value should be 0. In the case of input only ports, PODRn bits are reserved. See section 18.1. Overview. The PODRn bit in the PORTm.PCNTR1 register serves the same function as the PODR bit in the PFS.PmnPFS register.

18.2.2 PCNTR2/EIDR/PIDR : Port Control Register 2

Base address: PORTm = 0x4001_F000 + 0x0020 × m (m = 0, 2, A to E)

Offset address: 0x004 (PCNTR2/PIDR)
0x006 (EIDR)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	EIDR15	EIDR14	EIDR13	EIDR12	EIDR11	EIDR10	EIDR09	EIDR08	EIDR07	EIDR06	EIDR05	EIDR04	EIDR03	EIDR02	EIDR01	EIDR00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PIDR15	PIDR14	PIDR13	PIDR12	PIDR11	PIDR10	PIDR09	PIDR08	PIDR07	PIDR06	PIDR05	PIDR04	PIDR03	PIDR02	PIDR01	PIDR00
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
15:0	PIDR15 to PIDR00	Pmn State 0: Low level 1: High level	R
31:16	EIDR15 to EIDR00 *2	Port Event Input Data*1 When an ELC_PORTx signal occurs 0: Low input 1: High input	R

Note: If the security attribution is configured as Secure,
- Secure read access is allowed,
- Non-secure read value is 0, but TrustZone access error is not generated.

If the security attribution is configured as Non-secure,
- Secure and Non-secure read access are allowed.

Note: m = 0, 2, A to E, n = 00 to 15
Note 1. x = B, C, D or E for EIDR only
Note 2. Supported by ports B, C, D or E.

The Port Control Register 2 (PCNTR2/EIDR/PIDR) allows read access to the Pmn state and the port event input data using 32-bit or 16-bit access.

The PCNTR2 represents the Pmn state and the port event input data, and is accessed in 32-bit units.

The PIDRn (bits [15:0] in PCNTR2) and EIDRn (bits [31:16] in PCNTR2) respectively, are accessed in 16-bit units. Bits associated with non-existent pins are reserved. Reserved bits are read as undefined.

PIDRn bits (Pmn State)

The PIDRn bits reflect the individual pin states of the port, regardless of the values set in PmnPFS.PMR and PORTm.PCNTR1.PDRn. The PIDRn bit in the PORTm.PCNTR2 register serves the same function as the PIDR bit in the PFS.PmnPFS register.

A pin state cannot be reflected in PIDRn when the following functions is enabled:

- Analog function (ASEL = 1)

EIDRn bits (Port Event Input Data)

The EIDRn bits latch a pin state when an ELC_PORTx signal occurs. Pin states can only be input to EIDRn when PmnPFS.PMR and PORTm.PCNTR1.PDRn are 0. When the PmnPFS.ASEL bit is set to 1, the associated pin state is not reflected in EIDRn.

18.2.3 PCNTR3/PORR/POSR : Port Control Register 3

Base address: PORTm = 0x4001_F000 + 0x0020 × m (m = 0, 2, A to E)

Offset address: 0x008 (PCNTR3/POSR)
0x00A (PORR)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	PORR 15	PORR 14	PORR 13	PORR 12	PORR 11	PORR 10	PORR 09	PORR 08	PORR 07	PORR 06	PORR 05	PORR 04	PORR 03	PORR 02	PORR 01	PORR 00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	POSR 15	POSR 14	POSR 13	POSR 12	POSR 11	POSR 10	POSR 09	POSR 08	POSR 07	POSR 06	POSR 05	POSR 04	POSR 03	POSR 02	POSR 01	POSR 00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	POSR15 to POSR00	Pmn Output Set 0: No effect on output 1: High output	W
31:16	PORR15 to PORR00	Pmn Output Reset 0: No effect on output 1: Low output	W

Note: If the security attribution is configured as Secure,
- Secure write access is allowed,
- Non-secure write access is ignored, but TrustZone access error is not generated.

If the security attribution is configured as Non-secure,
- Secure and Non-secure write access are allowed.

Note: m = 0, 2, A to E, n = 00 to 15

The Port Control Register 3 (PCNTR3/PORR/POSR) is a 32-bit or 16-bit write register that controls the setting or resetting of the port output data.

The PCNTR3 controls the setting or resetting of the port output data, and is accessed in 32-bit units.

The POSRn (bits [15:0] in PCNTR3) and the PORRn (bits [31:16] in PCNTR3) respectively, are accessed in 16-bit units.

POSRn bits (Pmn Output Set)

POSR changes PODR when set by a software write. For example, for PD00, when PORTD.PCNTR3.POSR00 = 1, PORTD.PCNTR1.PODR00 outputs 1. Bits associated with non-existent pins are reserved. The write value should always be 0. In the case of input only ports, POSRn bits are reserved. See section 18.1. Overview.

PORRn bits (Pmn Output Reset)

PORR changes PODR when reset by a software write. For example, for PD00, when PORTD.PCNTR3.PORR00 = 1, PORTD.PCNTR1.PODR00 outputs 0. Bits associated with non-existent pins are reserved. The write value should always be 0. In the case of input only ports, PORRn bits are reserved. See section 18.1. Overview.

Note: When EORRn or EOSRn is set, writing is prohibited to PODRn, PORRn, and POSRn.
Note: PORRn and POSRn should not be set at the same time.

18.2.4 PCNTR4/EORR/EOSR : Port Control Register 4

Base address: PORTm = 0x4001_F000 + 0x0020 × m (m = B to E)

Offset address: 0x00C (PCNTR4/EOSR)
0x00E (EORR)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	EORR 15	EORR 14	EORR 13	EORR 12	EORR 11	EORR 10	EORR 09	EORR 08	EORR 07	EORR 06	EORR 05	EORR 04	EORR 03	EORR 02	EORR 01	EORR 00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	EOSR 15	EOSR 14	EOSR 13	EOSR 12	EOSR 11	EOSR 10	EOSR 09	EOSR 08	EOSR 07	EOSR 06	EOSR 05	EOSR 04	EOSR 03	EOSR 02	EOSR 01	EOSR 00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	EOSR15 to EOSR00	Pmn Event Output Set When an ELC_PORTx signal occurs 0: No effect on output 1: High output	R/W
31:16	EORR15 to EORR0	Pmn Event Output Reset When an ELC_PORTx signal occurs 0: No effect on output 1: Low output	R/W

Note: If the security attribution is configured as Secure,
- Secure access is allowed,
- Non-secure read value is 0, but TrustZone access error is not generated,
- Non-secure write access is ignored, but TrustZone access error is not generated.

If the security attribution is configured as Non-secure,
- Secure and Non-secure access are allowed.

Note: m = B to E, n = 00 to 15, x = B to E

The Port Control Register 4 (PCNTR4/EORR/EOSR) is a 32-bit or 16-bit read/write register that controls the setting or resetting of the port output data by an event input from the ELC.

The PCNTR4 controls the setting or resetting of the port output data by an event input from the ELC, and is accessed in 32-bit units.

The EOSRn (bits [15:0] in PCNTR4) and EORRn (bits [31:16] in PCNTR4) respectively, are accessed in 16-bit units.

EOSRn bits (Pmn Event Output Set)

EOSR changes PODR when set because an ELC_PORTx signal occurs. For example, for PD00 if PORTD.PCNTR4.EOSR00 is set to 1 when the ELC_PORTx occurs, PORTD.PCNTR1.PODR00 outputs 1. Bits associated with non-existent pins are reserved. The write value should always be 0. In the case of input only ports, EOSRn bits are reserved. See section 18.1. Overview.

EORRn bits (Pmn Event Output Reset)

EORR changes PODR when reset because an ELC_PORTx signal occurs. For example, for PD00 if PORTD.PCNTR4.EORR00 = 1 when the ELC_PORTx occurs, PORTD.PCNTR1.PODR00 outputs 0. Bits associated with non-existent pins are reserved. The write value should always be 0. In the case of input only ports, EORRn bits are reserved. See section 18.1. Overview.

Note: When EORRn or EOSRn is set, writing is prohibited to PODRn, PORRn, and POSRn.

Note: EORRn and EOSRn should not be set at the same time.

18.2.5 PmnPFS/PmnPFS_HA/PmnPFS_BY : Port mn Pin Function Select Register (m = 0, 2, A to E, n = 00 to 15)

Base address: PFS_B = 0x4001_F800

Offset address: 0x000 + 0x040 × m + 0x004 × n (m = 0, 2, A to E)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	PSEL[4:0]					—	—	—	—	—	—	—	PMR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0*1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ASEL	ISEL	EOFR[1:0]		DSCR[1:0]		—	—	—	NCODR	—	PCR	—	PDR	PIDR	PODR
Value after reset:	0*1	0	0	0	0	0*1	0	0	0	0	0	0*1	0	0	x	0

Bit	Symbol	Function	R/W
0	PODR	Port Output Data 0: Low output 1: High output	R/W*5
1	PIDR	Pmn State 0: Low level 1: High level	R*6
2	PDR	Port Direction 0: Input (functions as an input pin) 1: Output (functions as an output pin)	R/W*7
3	—	This bit is read as 0. The write value should be 0.	R/W
4	PCR	Pull-up Control 0: Disable input pull-up 1: Enable input pull-up	R/W*7
5	—	This bit is read as 0. The write value should be 0.	R/W
6	NCODR	N-Channel Open-Drain Control 0: CMOS output 1: NMOS open-drain output	R/W*7
9:7	—	These bits are read as 0. The write value should be 0.	R/W
11:10	DSCR[1:0]	Port Drive Capability 0 0: Low drive 0 1: Middle drive 1 0: High-speed high-drive*2 / High current drive*3 1 1: High drive	R/W*7
13:12	EOFR[1:0]	Event on Falling/Event on Rising*4 0 0: Don't care 0 1: Detect rising edge 1 0: Detect falling edge 1 1: Detect both edges	R/W*7
14	ISEL	IRQ Input Enable 0: Not used as an IRQn input pin 1: Used as an IRQn input pin	R/W*7
15	ASEL	Analog Input Enable 0: Not used as an analog pin 1: Used as an analog pin	R/W*7
16	PMR	Port Mode Control 0: Used as a general I/O pin 1: Used as an I/O port for peripheral functions	R/W*7
23:17	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
28:24	PSEL[4:0]	Peripheral Select These bits select the peripheral function. For individual pin functions, see the associated tables in this chapter.	R/W*7
31:29	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. The initial value of P002, P201, PA00 to PA05, PA13 to PA15, PB02 and PB03 is not 0x0000_0000. P002, PA00 to PA05 and PB02 is 0x0000_8000, P201 is 0x0000_0010, PA13 is 0x0001_0410, PA14 and PA15 is 0x0001_0010, and PB03 is 0x0001_0400.
Note 2. Only pins that support high-speed and high-drive can be set. Setting for other pins are prohibited.
Note 3. Only pins that support high current drive can be set. Setting for other pins are prohibited.
Note 4. Supported by PORTn (n = B to E).
Note 5. If the security attribution is configured as Secure,
- Secure access is allowed,
- Non-secure read value is 0, but TrustZone access error is not generated,
- Non-secure write access is ignored, but TrustZone access error is not generated.

If the security attribution is configured as Non-secure,
- Secure and Non-secure access are allowed.

Note 6. If the security attribution is configured as Secure,
- Secure read access is allowed,
- Non-secure read value is 0, but TrustZone access error is not generated.

If the security attribution is configured as Non-secure,
- Secure and Non-secure read access are allowed.

Note 7. If the security attribution is configured as Secure,
- Secure access and Non-secure read access are allowed,
- Non-secure write access is ignored, but TrustZone access error is not generated.

If the security attribution is configured as Non-secure,
- Secure and Non-secure access are allowed.

Port mn Pin Function Select Register (PmnPFS/PmnPFS_HA/PmnPFS_BY) is a 32-bit, 16-bit, or 8-bit read/write control register that selects the port mn pin function, and is accessed in 32-bit units. PmnPFS_HA (PmnPFS [15:0] bits) is accessed in 16-bit units. PmnPFS_BY (PmnPFS[7:0] bits) is accessed in 8-bit units.

The available Port mn pin depends on the product. For details, see Table 18.1

PODR bit (Port Output Data), PIDR bit (Port State), PDR bit (Port Direction)

The PDR, PIDR, and PODR bits serve the same function as the PCNTR. When these bits are read, the PCNTR value is read.

PCR bit (Pull-up Control)

The PCR bit enables or disables an input pull-up resistor on the individual port pins. When a pin is in the input state with the associated bit in PmnPFS.PCR set to 1, the pull-up resistor connected to the pin is enabled. When a pin is set as a general port output pin, or a peripheral function output pin, the pull-up resistor for the pin is disabled regardless of the PCR setting. The pull-up resistor is also disabled in the reset state. Bits associated with non-existent pins are reserved. Reserved bits are read as 0. The write value should be 0.

NCODR bit (N-Channel Open-Drain Control)

The NCODR bit specifies the output type for the port pins. Bits associated with non-existent pins are reserved. Reserved bits are read as 0. The write value should be 0.

DSCR[1:0] bits (Port Drive Capability)

The DSCR[1:0] bits switches the drive capacity of the port. If the drive capacity of a pin is fixed, the associated bit is a read/write bit, but the drive capacity cannot be changed. Bits associated with non-existent pins are reserved. Reserved bits are read as 0. The write value should be 0.

EOFR[1:0] bits (Event on Falling/Event on Rising)

The EOFR[1:0] bits select the edge detection method for the port group input signal. These bits support rising, falling, or both edge detections. When the EOFR[1:0] bits are set to 01b, 10b, or 11b, the input enable of the I/O cell is asserted. Following that, the event pulse is input from the external pin, and the GPIO outputs the event pulse to the ELC. Bits associated with non-existent pins are reserved. Reserved bits are read as 0. The write value should be 0.

ISEL bit (IRQ Input Enable)

The ISEL bit specifies IRQ input pins. An IRQn (external pin interrupt) of the same number must only be enabled for one pin. The ISEL bit for an unspecified IRQn is reserved.

ASEL bit (Analog Input Enable)

The ASEL bit specifies analog pins. When a pin is set as an analog pin by this bit:

1. Specify it as a general I/O port in the Port Mode Control bit (PmnPFS.PMR).
2. Disable the pull-up resistor in the Pull-up Control bit (PmnPFS.PCR).
3. Specify the input in the Port Direction bit (PmnPFS.PDR). The pin state cannot be read at this point. The PmnPFS register is protected by the Write-Protect Register (PWPR). Release write-protect before modifying the register.

The ASEL bit for an unspecified analog I/O pin is reserved.

PMR bit (Port Mode Control)

The PMR bit specifies the port pin function. Bits associated with non-existent pins are reserved. The write value should be 0.

PSEL[4:0] bits (Peripheral Select)

The PSEL[4:0] bits assign the peripheral function.

18.2.6 PWPR : Write-Protect Register

Base address: PFS_B = 0x4001_F800

Offset address: 0x50C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	B0WI	PFSWE	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	—	These bits are read as 0. The write value should be 0.	R/W
6	PFSWE	PmnPFS Register Write Enable 0: Writing to the PmnPFS register is disabled 1: Writing to the PmnPFS register is enabled	R/W
7	B0WI	PFSWE Bit Write Disable 0: Writing to the PFSWE bit is enabled 1: Writing to the PFSWE bit is disabled	R/W

PFSWE bit (PmnPFS Register Write Enable)

Writing to the PmnPFS register is enabled only when the PFSWE bit is set to 1. You must first write 0 to the B0WI bit before setting PFSWE to 1.

B0WI bit (PFSWE Bit Write Disable)

Writing to the PFSWE bit is enabled only when the B0WI bit is set to 0.

18.2.7 PWPRS : Write-Protect Register for Secure

Base address: PFS_B = 0x4001_F800

Offset address: 0x514

Bit position:	7	6	5	4	3	2	1	0
Bit field:	B0WI	PFSWE	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
6	PFSWE	PmnPFS Register Write Enable 0: Disable writes to the PmnPFS register 1: Enable writes to the PmnPFS register	R/W
7	B0WI	PFSWE Bit Write Disable 0: Enable writes the PFSWE bit 1: Disable writes to the PFSWE bit	R/W

Note: Only secure access can write to this register. Both secure access and Non-secure read access are allowed but Non-secure write access is not allowed and TrustZone access error is not generated.
Note: This register is write-protected by PRCR register.

PFSWE bit (PmnPFS Register Write Enable)

Writing to the PmnPFS register of the IO port pin set as secure by the PmSAR register is enabled only when the PFSWE bit is set to 1. You must first write 0 to the B0WI bit before setting PFSWE to 1.

B0WI bit (PFSWE Bit Write Disable)

Writing to the PFSWE bit is enabled only when the B0WI bit is set to 0.

18.2.8 PmSAR : Port m Security Attribution register (m = 0, 2, A to E)

Base address: PFS_B = 0x4001_F800

Offset address: 0x530 + 0x004 × m

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

Bit	Symbol	Function	R/W
15:0	PMNSA[15:0]	Pmn Security Attribution Target I/O port pin : Pmn 0: Secure 1: Non Secure	R/W

Note: Only secure access can write to this register. Both secure access and Non-secure read access are allowed but Non-secure write access is not allowed and TrustZone access error is not generated.
Note: This register is write-protected by PRCR register.
Note: m = 0, 2, A to E, n = 00 to 15

Port Security Attribution Register is a 16-bit register that setting the Security Attribution of the each port, the registers are accessed only in 16-bit units.

PMNSA[15:0] bits (Pmn Security Attribution)

The PMNSA bit specifies the Security Attribution of Pmn.

18.3 Operation

18.3.1 General I/O Ports

All pins except P002, PA00 to PA05, and PB02 operate as general I/O ports after reset. General I/O ports are organized as 16 bits per port and can be accessed by port with the Port Control Registers (PCNTRn, where n = 1 to 4), or by individual pins with the Port mn Pin Function Select register. For details on these registers, see section 18.2. Register Descriptions.

Each port has the following bits:

- Port Security Attribution register (PmSAR)(m = 0, 2, A to E), which indicates the security attribution.
- Port Direction bit (PDRn), which selects input or output direction
- Port Output Data bit (PODRn), which holds data for output
- Port Input Data bit (PIDRn), which indicates the pin states

- Event Input Data bit (EIDRn), which indicates the pin state when an ELC_PORTn (n = B, C, D or E) signal occurs
- Port Output Set bit (POSRn), which indicates the output value when a software write occurs
- Port Output Reset bit (PORRn), which indicates the output value when a software write occurs
- Event Output Set bit (EOSRn), which indicates the output value when an ELC_PORTn (n = B, C, D or E) signal occurs
- Event Output Reset bit (EORRn), which indicates the output value when an ELC_PORTn (n = B, C, D or E) signal occurs.

18.3.2 Port Function Select

The following port functions are available for configuring each pin:

- Security function: Security attribution for each pins
- I/O configuration: CMOS output or NMOS open-drain output, pull-up control, and drive capacity
- General I/O port: Port direction, output data setting, and read input data
- Alternate function: Configured function mapping to the pin.

Each pin is associated with a Port mn Pin Function Select register (PmnPFS), which includes the associated PODR, PIDR, and PDR bits. In addition, the PmnPFS register includes the following:

- PCR: Pull-up resistor control bit that turns the input pull-up MOS on or off
- NCODR: N-channel open-drain control bit that selects the output type for each pin
- DSCR[1:0]: Drive capacity control bit that selects the drive capacity
- EOFR[1:0]: For selecting the edge of the event that input from the port group
- ISEL: IRQ input enable bit to specify an IRQ input pin
- ASEL: Analog input enable bit to specify an analog pin
- PMR: Port mode bit to specify the pin function of each port
- PSEL[4:0]: Port function select bits to select the associated peripheral function.

These configurations can be made by a single-register access to the Port mn Pin Function Select register. For details, see section 18.2.5. PmnPFS/PmnPFS_HA/PmnPFS_BY : Port mn Pin Function Select Register (m = 0, 2, A to E, n = 00 to 15).

18.3.3 Port Group Function for ELC

In the MCU, Port B to Port E are assigned for the ELC port group function.

18.3.3.1 Behavior When ELC_PORTn (n = B, C, D or E) is Input from ELC

The MCU supports the two functions described in this section when an ELC_PORTn (n = B, C, D or E) signal comes from the ELC.

(1) Input to EIDR

For the GPI function (PDR = 0 and PMR = 0 in the PmnPFS register), when an ELC_PORTn (n = B, C, D or E) signal comes from the ELC, the input enable of the I/O cell is asserted, and data from the external pins is read into the EIDR bit. See Figure 18.2

For the GPO function (PDR = 1) or the peripheral mode (PMR = 1), 0 is input into the EIDR bit from the external pins.

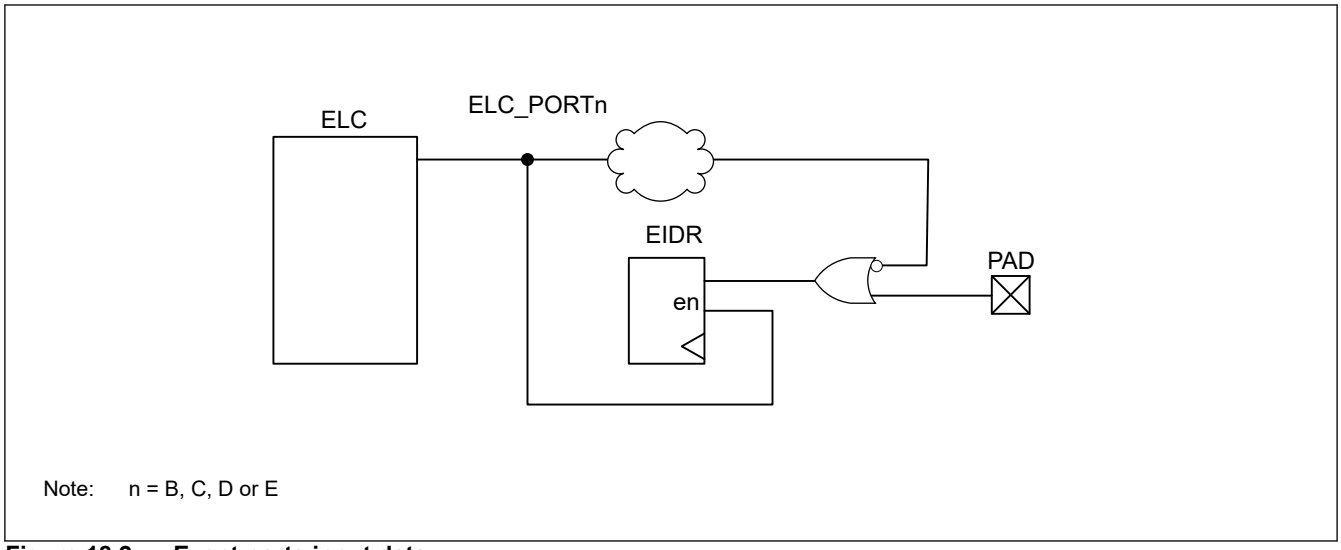


Figure 18.2 Event ports input data

(2) Output from PODR by EOSR and EORR

When an ELC_PORTn (n = B, C, D or E) signal occurs, the data is output from the PODR to the external pin based on the settings in the EOSR and EORR registers.

- If EOSR is set to 1, when an ELC_PORTn (n = B, C, D or E) signal occurs, the PODR register outputs 1 to the external pin. Otherwise, when EOSR = 0, the PODR value is retained.
- If EORR is set to 1, when ELC_PORTn (n = B, C, D or E) signal occurs, the PODR register outputs 0 to the external pin. Otherwise, when EORR = 0, the PODR value is retained.

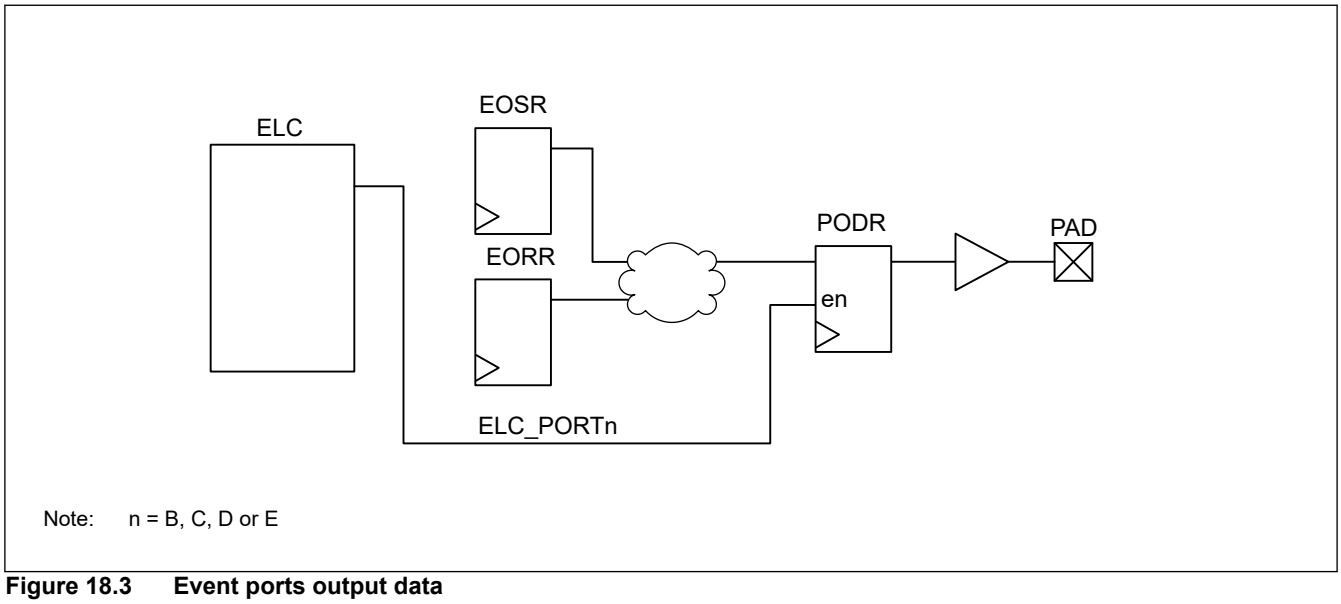


Figure 18.3 Event ports output data

18.3.3.2 Behavior When an Event Pulse is Output to ELC

To output the event pulse from the external pins to the ELC, set the EOFR[1:0] bits in the PmnPFS register. For details, see section 18.2.5. PmnPFS/PmnPFS_HA/PmnPFS_BY : Port mn Pin Function Select Register (m = 0, 2, A to E, n = 00 to 15). When the EOFR[1:0] bits are set, the input enable of the I/O cell is asserted.

Data from the external pin is the input. For example, for Port E, when the data is input from PE00 to PE15, the data of those 16 pins is organized by OR logic. This data is formed into a one-shot pulse that goes to the ELC. The operation of Port n (n = B to D) is also the same as Port E. See Figure 18.4.

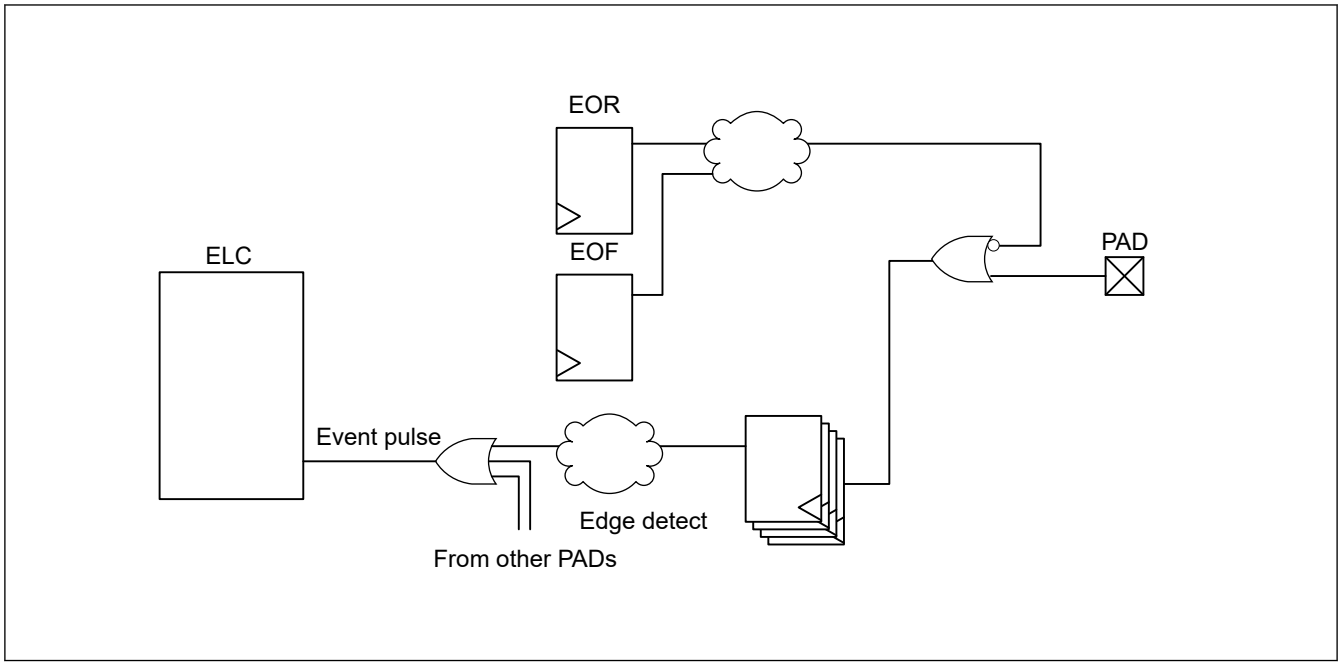


Figure 18.4 Generation of event pulse

18.4 Handling of Unused Pins

Table 18.3 shows how to handle unused pins.

Table 18.3 Handling of unused pins

Pin name	Description
MD	Use as a mode selection pin
RES	Connect to VCC through a resistor (pulling up)
PC13/NM1	Connect to VCC through a resistor (pulling up)
EXTAL	When the main clock oscillator is not used, set the MOSCCR.MOSTP bit to 1 (general port P212). When this pin is not used as port P212, configure it in the same way as ports A to E.
XTAL	When the main clock oscillator is not used, set the MOSCCR.MOSTP bit to 1 (general port P213). When the external clock is input to the EXTAL pin, the XTAL pin functions as P213. When this pin is not used as port P213, configure it in the same way as ports A to E.
P000, P001 PA00, PA02, PA04, PA06, PA07 PB00 to PB02 PC00 to PC05	Connect to AVCC0 (pulled up) through a resistor or to AVSS0 (pulled down) through a resistor*1
P002 PA01, PA03, PA05	Connect to AVCC0 (pulling up) through a resistor
Other ports	• If the direction is set to input (PCNTR1.PDRn = 0), connect the associated pin to VCC (pulled up) through a resistor or to VSS (pulled down) through a resistor.*1 *2 • If the direction is set to output (PCNTR1.PDRn = 1), keep pin open.*1 *3
VREFH0	Connect to AVCC0
VREFL0	Connect to AVSS0

Note 1. Clear the PmnPFS.PMR, PmnPFS.ISEL, PmnPFS.PCR, and PmnPFS.ASEL bits to 0.
Note 2. PA13 to PA15 should be enabled for input pull-up from the initial value (PmnPFS.PCR = 1).
Note 3. PB03 is recommended for setting the direction to output (PCNTR1.PDRn = 1), because this pin is output from the initial value.

18.5 Usage Notes

18.5.1 Procedure for Specifying the Pin Functions

To specify the I/O pin functions:

1. Clear the B0WI bit in the PWPR register. This enables writing to the PFSWE bit in the PWPR register.*1
2. Set 1 to the PFSWE bit in the PWPR register. This enables writing to the PmnPFS register.*1
3. Clear the Port Mode Control bit in the PMR to 0 for the target pin to select the general I/O port.
4. Specify the I/O function for the pin through the PSEL[4:0] bits settings in the PmnPFS register.
5. Set the PMR bit to 1 as required to switch to the selected I/O function for the pin.
6. Clear the PFSWE bit in the PWPR register. This disables writing to the PmnPFS register.*1
7. Set 1 to the B0WI bit in the PWPR register. This disables writing to the PFSWE bit in the PWPR register.*1

Note 1. When the security attribution of Pmn is set to 0, set the PWPRS register to write to the PmnPFS register.

18.5.2 Procedure for Using Port Group Input

To use the port group input (port n (n = B to E)):

1. Set the ELSRx.ELS[8:0] bits to all 0 to ignore unexpected pulses. For more information, see section 17, Event Link Controller (ELC).
2. Set the EOFR[1:0] bits of the PmnPFS register to specify the rising, falling, or both edge detections.
3. Execute a dummy read or wait for a short time, for example 100 ns. Ignoring of unexpected pulses depends on the initial value of the external pin.
4. Set the ELSRx.ELS[8:0] bits to enable the event signals.

18.5.3 Port Output Data Register (PODR) Summary

This register outputs data as follows:

1. Outputs 0 if PCNTR4.EORR is set to 1 when ELC_PORTn (n = B, C, D or E) signal occurs.
2. Outputs 1 if PCNTR4.EOSR is set to 1 when ELC_PORTn (n = B, C, D or E) signal occurs.
3. Outputs 0 if PCNTR3.PORR is set to 1.
4. Outputs 1 if PCNTR3.POSR is set to 1.
5. Outputs 0 or 1 because PCNTR1.PODRn is set.
6. Outputs 0 or 1 because PmnPFS.PODRn is set.

Numbers in this list correspond to the priority for writing to the PODRn. For example, if 1. and 3. from the list occur at the same time, the higher priority event 1. is executed.

18.5.4 Notes on Using Analog Functions

To use an analog function, set the Port Mode Control bit (PMR) and the Port Direction bit (PDRn) to 0 so that the pin acts as a general input port. Next, set the Analog Input Enable bit (ASEL) in the Port mn Pin Function Select Register (PmnPFS.ASEL) to 1.

The pin to which the PGA function (PGAINn, PGAVSSn (n = 0 to 3)) is assigned cannot be used as general ports when the PGA is set to pseudo-differential input mode.

When using the corresponding pin as general ports, set the corresponding PGA to single-ended input mode (see section 36.3.15.2. PGA Operation Setting).

Then, set the corresponding pin to function as general ports.

18.6 Peripheral Select Settings for Each Product

This section describes the pin function select configuration using the PmnPFS register. Some pin names have added _A, _B, _C, _D, _E or _F suffixes. When assigning IIC and SPI functionality, select the functional pins having the same suffix. The other pins can be selected regardless of the suffix. Assigning the same function to two or more pins simultaneously is prohibited.

1. In Pmn pin function select register(PmnPFS), the PSEL bits have to be set when the PMR bit of the target pin is 0. If the PSEL bits are set when the PMR bit is 1, the unexpected edges may be input at the input function or the unexpected pulses may be output to the external pin at the output function.
2. Only the allowed values (functions) should be specified in the PSEL bits of PmnPFS register. If a value which is not allowed for the register is specified, the correct operation is not guaranteed.
3. The single function should not be assigned to the multiple pins by PmnPFS register.
4. The PORT0 and A, B, C, E have the analog functions such as A/D converter. When these pins are used as an analog function, for avoiding the loss of resolution, the PMR bit should be set to 0 and PDR bit should be set to 0. After that, ASEL bit should be set to 1

Table 18.4 Register settings for input/output pin function (PORT0)

PSEL[4:0] settings	Function	pin		
		P000	P001	P002
00000b (value after reset)	Hi-Z/JTAG/SWD	Hi-Z		
ASEL bit		AN016/IVREF0	AN017/IVREF1	AN019/PGAVSS3
ISEL bit		IRQ0	IRQ2	—
DSCR[1:0] bits	Drive capacity control	—	—	—
NCODR bit	N-ch open-drain	—	—	—
PCR bit	Pull-up	✓	✓	—
100 pins product		✓	✓	✓
64 pins product		—	—	✓
48 pins product		—	—	—

✓: Available
—: Setting prohibited

Table 18.5 Register settings for input/output pin function (PORT2)

PSEL[4:0] settings	Function	Pin		
		P201	P212	P213
00000b (value after reset)	Hi-Z/JTAG/SWD	Hi-Z		
ASEL bit		—	—	—
ISEL bit		—	—	IRQ0
DSCR[1:0] bits	Drive capacity control	L*1	L/M/H	L/M/H
NCODR bit	N-ch open-drain	✓	✓	✓
PCR bit	Pull-up	✓	✓	✓
100 pins product		✓	✓	✓
64 pins product		✓	✓	✓
48 pins product		✓	✓	✓

✓: Available
—: Setting prohibited

Note 1. The drive capacity of this port cannot be controlled by PmnPFS.DSCR[1:0] bits.

Table 18.6 Register settings for input/output pin function (PORTA) (1 of 2)

PSEL[4:0] settings	Function	Pin															
		PA00	PA01	PA02	PA03	PA04	PA05	PA06	PA07	PA08	PA09	PA10	PA11	PA12	PA13	PA14	PA15
00000b (value after reset)	Hi-Z/ JTAG/SW D	Hi-Z													TMS/ SWDIO	TCK/ SWCLK	TDI
00001b	AGT	—	—	—	—	—	—	—	—	AGTIO 0	—	—	—	—	AGTO0	AGTO1	—
00010b	GPT	—	—	—	—	—	—	—	—	GTOUU P	GTOVU P	GTOW UP	GTETR GD	GTETR GB	—	—	GTETR GB
00011b	GPT*1	—	—	—	—	—	—	—	—	GTIOC 8A	GTIOC 8B	GTIOC 9A	GTIOC 9B	GTCPP O0	—	—	—
00100b	SCI*3	—	—	—	—	—	—	—	—	SCK0_ A	TXD0_ A/ MOSI0 _A/ SDA0	RXD0_ A/ MISO0 _A/ SCL0	CTS0_ A	CTS0_ RTS0/ SS0_A	SCK0_ C	TXD0_ C/ MOSI0 _C/ SDA0	RXD0_ C/ MISO0 _C/ SCL0

Table 18.6 Register settings for input/output pin function (PORTA) (2 of 2)

PSEL[4:0] settings	Function	Pin															
		PA00	PA01	PA02	PA03	PA04	PA05	PA06	PA07	PA08	PA09	PA10	PA11	PA12	PA13	PA14	PA15
00101b	SCI*3	—	—	—	—	—	—	—	—	SCK1_C	—	—	RXD1_C/MISO1_C/SCL1	TXD1_C/MOSI1_C/SDA1	CTS1_RTS1/SS1_C	SCK9_B	RXD9_B/MISO9_B/SCL9
00110b	SPI*2	—	—	—	—	—	—	—	—	SSLA1_B	SSLA0_B	RSPCKA_B	MOSIA_B	MISOA_B	—	—	SSLA0_A
00111b	IIC*2	—	—	—	—	—	—	—	—	SCL0_D	SCL1_C	SDA1_C	—	—	—	—	—
01000b	KINT	—	—	—	—	—	—	—	—	KR00	KR01	KR02	KR03	KR04	—	—	KR02
01001b	CLKOUT	—	—	—	—	—	—	—	—	CLKOUT	—	—	—	—	—	—	—
01010b	ADC	—	—	—	—	—	—	—	—	—	—	—	—	ADTRG1	—	—	ADTRG0
01100b	ACMPHS	—	—	—	—	—	—	—	—	CMPOUT2	CMPOUT3	CMPOUT0	CMPOUT1	—	—	—	CMPOUT012
01101b	SCI	—	—	—	—	—	—	—	—	DE1	—	—	—	—	DE1	DE9	—
01110b	SCI	—	—	—	—	—	—	—	—	DE0	—	—	—	DE0	DE0	—	—
10000b	CANFD	—	—	—	—	—	—	—	—	—	—	—	CTX0	CRX0	—	—	—
10100b	GPT*1	—	—	—	—	—	—	—	—	GTIOC7B	GTIOC8B	GTIOC9B	GTETRGC	GTCPPO2	—	—	—
10101b	GPT*1	—	—	—	—	—	—	—	—	GTIOC2A	GTIOC2B	GTIOC3A	GTIOC3B	GTADSM0	—	—	GTADSM1
10110b	GPT*1	—	—	—	—	—	—	—	—	GTIOC9A	GTIOC7B	GTIOC8B	—	GTCPPO7	—	—	GTCPPO4
11001b	CAC	—	—	—	—	—	—	—	—	—	—	—	—	CACREF	—	—	—
ASEL bit		AN000/PGAIN0/IVCMP02/IVCMP03	AN001/PGAVSS0	AN002/PGAIN1/IVCMP12/IVCMP13	AN003/PGAVSS1	AN004/PGAIN2/IVCMP22/IVCMP23	AN005/PGAVSS2	AN006/DA0	AN007/DA1	—	—	—	—	—	—	—	—
ISEL bit		IRQ0-DS	IRQ1	IRQ2	IRQ3	IRQ4	IRQ5	IRQ6	IRQ7	IRQ8	IRQ9	IRQ10	IRQ11	IRQ12	—	—	IRQ1
DSCR[1:0] bits	Drive capacity control	—	—	—	—	—	—	—	—	L/M/H/HC	L/M/H/HC	L/M/H/HC	L/M/H/HC	L/M/H	L/M/H	L/M/H	L/M/H
NCODR	N-ch open-drain	—	—	—	—	—	—	—	—	✓	✓	✓	✓	✓	✓	✓	✓
PCR bit	Pull-up	—	—	—	—	—	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
100 pins product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
64 pins product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
48 pins product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

✓: Available
—: Setting prohibited

Note 1. There are 2 types output buffer which are middle drive and high drive. Recommend using same drive buffer for output skew spec (tGTISK).

Note 2. Recommend using pins that have a letter appended to their names, for instance _A or _B or _C or _D, to indicate group membership. For the interface, the AC portion of the electrical characteristics is measured for each group.

Note 3. Recommend using pins that have a letter appended to their names, for instance _A or _B or _C, to indicate group membership. For details, see section 46, Electrical Characteristics.

Table 18.7 Register settings for input/output pin function (PORTB) (1 of 2)

PSEL[4:0] settings	Function	pin														
		PB00	PB01	PB02	PB03	PB04	PB05	PB06	PB07	PB08	PB09	PB10	PB12	PB13	PB14	PB15
00000b (value after reset)	Hi-Z/JTAG/SWD	Hi-Z			TDO/SWO	Hi-Z										
00001b	AGT	—	—	—	AGTO1	AGTOA0	AGTOB0	AGTOA1	AGTOB1	AGTIO0	AGTIO1	—	—	—	—	—
00010b	GPT	—	—	—	—	—	GTIU	GTIV	GTIW	—	—	GTIU	GTETRGA	GTOULO	GTOVLO	GTOWLO

Table 18.7 Register settings for input/output pin function (PORTB) (2 of 2)

PSEL[4:0] settings	Function	pin														
		PB00	PB01	PB02	PB03	PB04	PB05	PB06	PB07	PB08	PB09	PB10	PB12	PB13	PB14	PB15
00011b	GPT*1	—	—	—	—	GTIOC4A	GTIOC4B	GTIOC5A	GTIOC5B	GTIOC6A	GTIOC6B	GTETRGA	GTIOC0A	GTIOC0B	GTIOC1A	GTIOC1B
00100b	SCI*3	—	—	—	TXD2_A/ MOSI2_A/ SDA2	RXD2_A/ MISO2_A/ SCL2	SCK2_A	TXD0_D/ MOSI0_D/ SDA0	RXD0_D/ MISO0_D/ SCL0	RXD4/ MISO4_C/ SCL4	TXD4/ MOSI4_C/ SDA4	TXD4_A/ MOSI4_A/ SDA4	SCK4_A	CTS4_A	CTS4_RTS4/ SS4_A	RXD4_A/ MISO4_A/ SCL4
00101b	SCI*3	—	—	—	TXD9_B/ MOSI9_B/ SDA9	RXD3_D/ MISO3_D/ SCL3	TXD3_D/ MOSI3_D/ SDA3	CTS3_RTS3/ SS3_D	CTS1_RTS1/ SS1_D	RXD1_D/ MISO1_D/ SCL1	TXD1_D/ MOSI1_D/ SDA1	CTS3_B	RXD3_B/ MISO3_B/ SCL3	TXD3_B/ MOSI3_B/ SDA3	SCK3_B	CTS3_RTS3/ SS3_B
00110b	SPI*2	—	—	—	RSPCKA_A	MISOA_A	MOSIA_A	—	—	—	—	—	SSLB0_A	RSPCKB_A	MISOB_A	MOSIB_A
00111b	IIC*2	—	—	—	—	—	—	SCL0_A	SDA0_A	SCL1_A	SDA1_A	—	—	—	SDA0_C	SCL0_C
01000b	KINT	—	—	—	KR03	KR04	KR05	KR06	KR07	KR00	KR01	—	—	—	—	—
01001b	CLKOUT	—	—	—	—	VCOUT	—	—	—	—	—	VCOUT	—	—	—	—
01010b	ADC	—	—	—	ADTRG1	—	—	—	—	—	—	—	ADTRG0	—	—	—
01100b	ACMPHS	—	—	—	CMPOUT3	—	—	—	—	—	—	—	—	—	—	—
01101b	SCI	—	—	—	—	—	—	DE3	DE1	—	—	—	—	—	DE3	DE3
01110b	SCI	—	—	—	—	—	DE2	—	—	—	—	—	DE4	—	DE4	—
10000b	CANFD	—	—	—	CRX0	CTX0	CRX0	CTX0	—	CRX0	CTX0	—	CRX0	CTX0	—	—
10100b	GPT*1	—	—	—	GTIOC4A	GTIOC5A	GTIOC6A	GTIOC4B	GTETRGC	GTIOC5B	—	GTETRGB	—	GTIOC7A	GTIOC8A	GTIOC9A
10101b	GPT*1	—	—	—	GTCPPO1	GTIOC0A	GTIOC0B	GTIOC1A	GTIOC1B	GTIOC2A	GTIOC2B	GTCPPO4	GTIOC4A	GTIOC5A	GTIOC6A	GTIOC4B
10110b	GPT	—	—	—	GTCPPO3	—	—	—	—	—	—	GTCPPO7	—	—	—	—
11001b	CAC	—	—	—	—	CACREF	—	—	—	—	—	CACREF	—	—	—	—
11101b + ASEL bit	PGAOUT*4	PGAOUT0	PGAOUT1	—	—	—	—	—	—	—	—	—	—	—	—	—
11110b + ASEL bit	PGAOUT*4	PGAOUT2	PGAOUT3	—	—	—	—	—	—	—	—	—	—	—	—	—
ASEL bit		AN008	AN009	AN018/ PGAIN3/ IVCMP32/ IVCMP33	—	—	—	—	—	—	—	AN028	—	—	—	—
ISEL bit		IRQ0	IRQ1	IRQ15-DS	IRQ0	IRQ13	IRQ3-DS	IRQ4-DS	IRQ5-DS	IRQ1-DS	IRQ2-DS	IRQ10-DS	IRQ2	IRQ3	IRQ4	IRQ5
DSCR[1:0] bits	Drive capacity control	—	—	—	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H/HC	L/M/H/HC	L/M/H/HC	L/M/H/HC
NCODR	N-ch open-drain	—	—	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
PCR bit	Pull-up	✓	✓	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
100 pins product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
64 pins product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	—	✓	✓	✓	✓
48 pins product		✓	✓	—	✓	✓	✓	✓	✓	✓	✓	—	✓	✓	✓	✓

✓: Available
—: Setting prohibited

Note 1. There are 2 types output buffer which are middle drive and high drive. Recommend using same drive buffer for output skew spec (tGTISK).

Note 2. Recommend using pins that have a letter appended to their names, for instance _A or _C, to indicate group membership. For the interface, the AC portion of the electrical characteristics is measured for each group.

Note 3. Recommend using pins that have a letter appended to their names, for instance _A or _B or _C or _D, to indicate group membership. For details, see section 46, Electrical Characteristics.

Note 4. The PGAOUTn (n = 0 to 3) function is selected when the ASEL bit of the corresponding pin = 1 and the PSEL [4:0] bits are the specified combination.When using the corresponding pin as PGAOUTn, do not use it with other analog pin functions.

Table 18.8 Register settings for input/output pin function (PORTC)

PSEL[4:0] settings	Function	pin															
		PC00	PC01	PC02	PC03	PC04	PC05	PC06	PC07	PC08	PC09	PC10	PC11	PC12	PC13	PC14	PC15
00000b (value after reset)	Hi-Z/ JTAG/SW D	Hi-Z															
00001b	AGT	—	—	—	—	—	—	AGTO0	AGTEE 0	AGTOA 0	AGTOB 0	AGTIO 1	AGTOA 1	AGTOB 1	—	AGTIO 0	AGTIO 1
00010b	GPT	—	—	—	—	—	—	GTETR GD	GTETR GA	GTIV	GTIW	—	—	—	GTETR GD	GTETR GA	GTETR GB
00011b	GPT*1	—	—	—	—	—	—	GTIOC 6A	GTIOC 6B	GTIOC 7A	GTIOC 7B	—	—	—	—	GTIOC 3A	GTIOC 3B
00100b	SCI*3	—	—	—	—	—	—	TXD2_ B/ MOSI2 _B/ SDA2	RXD2_ B/ MISO2 _B/ SCL2	SCK2_ B	CTS2_ RTS2/ SS2_B	—	—	TXD4_ B/ MOSI4 _B/ SDA4	—	—	—
00101b	SCI*3	—	—	—	—	—	—	CTS9_ RTS9/ SS9_C	CTS9_ C	CTS3_ RTS3/ SS3_C	CTS3_ C	TXD1_ B/ MOSI1 _B/ SDA1	RXD1_ B/ MISO1 _B/ SCL1	SCK1_ B	—	—	—
00110b	SPI*2	—	—	—	—	—	—	—	—	SSLA3 _B	SSLA2 _B	RSPCK B_B	MISOB _B	MOSIB _B	—	—	—
00111b	IIC*2	—	—	—	—	—	—	SCL1_ E	SDA1_ E	SCL0_ E	SDA0_ D/ SDA0_ E	SCL0_ B	SDA0_ B	—	—	—	—
01000b	KINT	—	—	—	—	—	—	—	—	—	—	KR05	KR06	KR07	—	—	—
01001b	CLKOUT	—	—	—	—	—	—	—	—	—	CLKOU T	—	—	—	—	—	—
01010b	ADC	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADTRG 0	ADTRG 1
01100b	ACMPHS	—	—	—	—	—	—	—	—	—	—	CMPO UT0	CMPO UT1	CMPO UT2	—	CMPO UT012	CMPO UT3
01101b	SCI	—	—	—	—	—	—	DE9	—	DE3	—	—	—	DE1	—	—	—
01110b	SCI	—	—	—	—	—	—	—	—	DE2	DE2	—	—	—	—	—	—
10100b	GPT	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GTCPP O0	GTCPP O1
10101b	GPT*1	—	—	—	—	—	—	GTIOC 5B	—	—	—	—	—	—	—	GTADS M0	GTADS M1
10110b	GPT*1	—	—	—	—	—	—	—	—	—	GTIOC 8A	—	—	—	—	GTCPP O4	GTCPP O7
11001b	CAC	—	—	—	—	—	—	—	—	CACRE F	—	—	—	—	—	—	—
11101b + ASEL bit	PGAOUT*4	PGAOU T0	PGAOU T1	PGAOU T2	PGAOU T3	—	—	—	—	—	—	—	—	—	—	—	—
ASEL bit		AN012/ IVCMP 00	AN013/ IVCMP 10	AN014/ IVCMP 20	AN015/ IVCMP 30	AN010/ DA2	AN011/ DA3	—	—	—	—	—	—	—	—	—	—
ISEL bit		IRQ11-DS	IRQ12-DS	IRQ13-DS	IRQ14-DS	IRQ10	IRQ11	IRQ6	IRQ7	IRQ8	IRQ9	IRQ6-DS	IRQ7-DS	IRQ8-DS	NMI	IRQ14	IRQ15
DSCR[1:0] bits	Drive capacity control	—	—	—	—	—	—	L/M/H/ HC	L/M/H/ HC	L/M/H/ HC	L/M/H/ HC	L/M/H	L/M/H	L/M/H	—	L*5	L*5
NCODR	N-ch open-drain	—	—	—	—	—	—	✓	✓	✓	✓	✓	✓	✓	—	✓	✓
PCR bit	Pull-up	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
100 pins product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
64 pins product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
48 pins product		—	—	—	—	—	—	—	—	—	—	—	—	—	✓	✓	✓

✓: Available
—: Setting prohibited

Note 1. There are 2 types output buffer which are middle drive and high drive. Recommend using same drive buffer for output skew spec (tGTISK).

Note 2. Recommend using pins that have a letter appended to their names, for instance _B or _D or _E, to indicate group membership. For the interface, the AC portion of the electrical characteristics is measured for each group.

Note 3. Recommend using pins that have a letter appended to their names, for instance _B or _C, to indicate group membership. For details, see section 46, Electrical Characteristics.

Note 4. The PGAOUTn (n = 0 to 3) function is selected when the ASEL bit of the corresponding pin = 1 and the PSEL [4:0] bits are the specified combination.When using the corresponding pin as PGAOUTn, do not use it with other analog pin functions.
Note 5. The drive capacity of this port cannot be controlled by PmnPFS.DSCR[1:0] bits.

Table 18.9 Register settings for input/output pin function (PORTD)

PSEL[4:0] settings	Function	Pin															
		PD00	PD01	PD02	PD03	PD04	PD05	PD06	PD07	PD08	PD09	PD10	PD11	PD12	PD13	PD14	PD15
00000b (value after reset)	Hi-Z/ JTAG/SWD	Hi-Z															
00001b	AGT	—	—	AGTEE1	—	—	—	—	—	—	—	—	—	—	—	—	—
00010b	GPT	—	—	—	—	—	—	—	—	—	—	GTETRGC	—	—	—	—	—
00011b	GPT*1	—	—	—	—	—	—	—	—	GTIOC2A	GTIOC2B	GTIOC3A	GTIOC3B	GTIOC4A	GTIOC4B	GTIOC5A	GTIOC5B
00100b	SCI*3	CTS2_A	CTS2_RTS2/SS2_A	RXD4_B/MISO4_B/SCL4	SCK4_B	CTS4_RTS4/SS4_B	—	—	—	CTS2_B	CTS2_RTS2/SS2_B	SCK2_C	RXD2_C/MISO2_C/SCL2	TXD2_C/MOSI2_C/SDA2	SCK4_C	RXD4_C/MISO4_C/SCL4	TXD4_C/MOSI4_C/SDA4
00101b	SCI*3	RXD3_C/MISO3_C/SCL3	TXD3_C/MOSI3_C/SDA3	SCK3_C	CTS9_A	CTS9_RTS9/SS9_A	TXD9/MOSI9_A/SDA9	RXD9_A/MISO9_A/SCL9	SCK9_A	TXD1_A/MOSI1_A/SDA1	RXD1_A/MISO1_A/SCL1	SCK1_A	CTS1_A	CTS1_RTS1/SS1_A	SCK9_C	RXD9_C/MISO9_C/SCL9	TXD9_C/MOSI9_C/SDA9
00110b	SPI*2	SSLB0_B	SSLB1_B	—	SSLB2_B	SSLB3_B	SSLA3_A	SSLA2_A	SSLA1_A	SSLB1_A	SSLB2_A	SSLB3_A	—	—	—	—	—
00111b	IIC*2	—	—	—	—	—	SDA1_B	SCL1_B	—	—	—	—	—	SCL1_D	SDA1_D	SCL0_F	SDA0_F
01000b	KINT	KR00	KR01	KR02	KR03	KR04	KR05	KR06	KR07	KR00	KR01	KR02	KR03	KR04	KR05	KR06	KR07
01001b	CLKOUT	—	—	CLKOUT	—	—	—	—	—	—	—	—	—	—	—	—	—
01100b	ACMPHS	—	—	CMPOUT3	CMPOUT0	CMPOUT1	—	—	—	—	—	—	—	—	—	—	—
01101b	SCI	—	—	DE3	—	DE9	—	—	DE9	—	—	DE1	—	DE1	DE9	—	DE9
01110b	SCI	—	DE2	—	DE4	DE4	—	—	—	—	DE2	DE2	—	—	DE4	—	—
10000b	CANFD	CRX0	CTX0	—	—	—	—	—	—	—	—	—	—	—	—	—	—
10101b	GPT	GTADSM0	GTADSM1	GTCPPO0	—	—	GTADSM0	—	GTADSM1	—	—	—	—	—	—	—	—
10110b	GPT	GTCPPO4	GTCPPO7	GTCPPO2	GTCPPO0	GTCPPO1	GTCPPO3	GTCPPO4	GTCPPO7	—	—	—	—	—	—	—	—
ISEL bit		—	—	IRQ9-DS	—	—	—	—	—	—	—	—	—	IRQ12	IRQ13	IRQ14	IRQ15
DSCR[1:0] bits	Drive capacity control	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H/HC	L/M/H/HC	L/M/H/HC	L/M/H/HC	L/M/H/HC	L/M/H/HC	L/M/H/HC	L/M/H/HC
NCODR	N-ch open-drain	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
PCR bit	Pull-up	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
100 pins product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
64 pins product		—	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—
48 pins product		—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

✓: Available
—: Setting prohibited

Note 1. There are 2 types output buffer which are middle drive and high drive. Recommend using same drive buffer for output skew spec (tGTISK).
Note 2. Recommend using pins that have a letter appended to their names, for instance _A or _B or _D or _F, to indicate group membership. For the interface, the AC portion of the electrical characteristics is measured for each group.
Note 3. Recommend using pins that have a letter appended to their names, for instance _A or _B or _C, to indicate group membership. For details, see section 46, Electrical Characteristics.

Table 18.10 Register settings for input/output pin function (PORTE) (1 of 2)

PSEL[4:0] settings	Function	Pin													
		PE00	PE01	PE02	PE03	PE04	PE05	PE06	PE08	PE09	PE10	PE11	PE12	PE13	PE14	PE15
00000b (value after reset)	Hi-Z/ JTAG/SWD	Hi-Z														

Table 18.10 Register settings for input/output pin function (PORTE) (2 of 2)

PSEL[4:0] settings	Function	Pin														
		PE00	PE01	PE02	PE03	PE04	PE05	PE06	PE08	PE09	PE10	PE11	PE12	PE13	PE14	PE15
00001b	AGT	AGTEE0	AGTEE1	—	—	—	—	—	—	—	—	—	—	—	—	—
00010b	GPT	—	GTOULO	GTOVLO	GTOWLO	GTOUUP	GTOVUP	GTOWUP	GTIV	GTIW	GTOULO	GTOUUP	GTOVLO	GTOVUP	GTOWLO	GTOWUP
00011b	GPT*1	GTETRGA	GTIOC7A	GTIOC7B	GTIOC8A	GTIOC8B	GTIOC9A	GTIOC9B	GTIOC3A	GTIOC3B	GTIOC2A	GTIOC2B	GTIOC1A	GTIOC1B	GTIOC0A	GTIOC0B
00100b	SCI*3	TXD0_E/ MOSI0_E/ SDA0	RXD0_E/ MISO0_E/ SCL0	SCK0_B	RXD0_B/ MISO0_B/ SCL0	TXD0_B/ MOSI0_B/ SDA0	CTS0_RTS0/ SS0_B	CTS0_B	—	—	—	—	—	—	—	RXD4_A/ MISO4_A/ SCL4
00101b	SCI*3	TXD9_D/ MOSI9_D/ SDA9	RXD9_D/ MISO9_D/ SCL9	SCK3_A	CTS3_A	CTS3_RTS3/ SS3_A	RXD3_A/ MISO3_A/ SCL3	TXD3_A/ MOSI3_A/ SDA3	—	—	—	—	—	—	—	—
00110b	SPI*2	SSLB3_C	SSLB2_C	RSPCKB_C	SSLB0_C	SSLB1_C	MISOB_C	MOSIB_C	SSLA3_C	SSLA2_C	SSLA1_C	SSLA0_C	RSPCKA_C	MISOA_C	MOSIA_C	—
01000b	KINT	—	—	—	—	—	—	—	KR00	KR01	KR02	KR03	KR04	KR05	KR06	KR07
01001b	CLKOUT	—	—	CLKOUT	—	—	—	—	—	—	—	—	—	—	—	—
01010b	ADC	ADTRG0	ADTRG1	—	—	—	—	—	ADTRG0	ADTRG1	—	—	—	—	—	—
01100b	ACMPHS	—	—	CMPOUT0	CMPOUT1	CMPOUT2	CMPOUT3	—	CMPOUT012	CMPOUT3	—	—	—	—	—	—
01101b	SCI	—	—	DE3	—	DE3	—	—	—	—	—	—	—	—	—	—
01110b	SCI	—	—	DE0	—	—	DE0	—	—	—	—	—	—	—	—	—
10100b	GPT*1	GTIOC4A	GTIOC4B	GTIOC8A	GTIOC9A	GTIOC7B	GTIOC8B	—	GTETRGC	GTETRGD	GTIOC4A	GTIOC5A	GTIOC6A	GTIOC4B	GTIOC5B	GTIOC6B
10101b	GPT*1	GTADSM0	GTADSM1	—	—	—	—	—	GTADSM0	GTADSM1	GTIOC7A	GTIOC8A	GTIOC9A	GTIOC7B	GTIOC8B	GTIOC9B
10110b	GPT	—	—	—	—	—	GTCPPO2	GTCPPO3	—	—	—	—	—	—	—	—
11001b	CAC	CACREF	—	—	—	—	—	—	—	CACREF	—	—	—	—	—	—
11010b	Trace (Debug)	—	—	TRCLK	TRDATA0	TRDATA1	TRDATA2	TRDATA3	—	—	—	—	—	—	—	—
ASEL bit		—	—	—	—	—	—	—	AN020	AN021	AN022	AN023	AN024	AN025	AN026	AN027
ISEL bit		—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
DSCR[1:0] bits	Drive capacity control	L/M/H	L/M/H	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H	L/M/H	L/M/H/HC	L/M/H/HC	L/M/H/HC	L/M/H/HC	L/M/H/HC	L/M/H/HC
NCODR	N-ch open-drain	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
PCR bit	Pull-up	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
100 pins product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
64 pins product		—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
48 pins product		—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

✓: Available
—: Setting prohibited

Note 1. There are 2 types output buffer which are middle drive and high drive. Recommend using same drive buffer for output skew spec (tGTISK).

Note 2. Recommend using pins that have a letter appended to their names, for instance _C, to indicate group membership. For the interface, the AC portion of the electrical characteristics is measured for each group.

Note 3. Recommend using pins that have a letter appended to their names, for instance _A or _B or _D or _E, to indicate group membership. For details, see section 46, Electrical Characteristics.

19. Key Interrupt Function (KINT)

19.1 Overview

The key interrupt function (KINT) generates the key interrupt by detecting rising or falling edge on the key interrupt input pins. Figure 19.1 shows a block diagram and Table 19.1 lists the input pins.

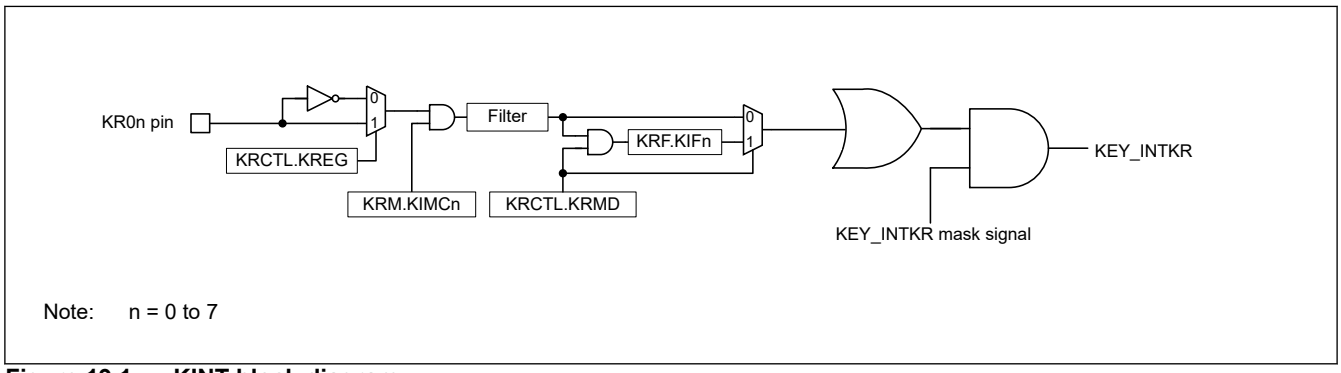


Figure 19.1 KINT block diagram

All key return factors are merged by an OR gate, and the key interrupt signal, KEY_INTKR, is the output of the AND gate to mask the merged key return factor by the KEY_INTKR mask signal. When using KRF.KIFn flag (KRCTL.KRMD = 1), the KEY_INTKR mask signal is used as the output mask that is asserted by clearing KRF.KIFn flag.

Table 19.1 KINT I/O pins

Pin name	I/O	Function
KR00 to KR07	Input	Key interrupt input pins

19.2 Register Descriptions

19.2.1 KRCTL : Key Return Control Register

Base address: KINT = 0x4008_5000

Offset address: 0x00

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

Bit	Symbol	Function	R/W
0	KREG	Detection Edge Selection (KR00 to KR07 pins) 0: Falling edge 1: Rising edge	R/W
6:1	—	These bits are read as 0. The write value should be 0.	R/W
7	KRMD	Usage of Key Interrupt Flags (KRF.KIF0 to KRF.KIF7) 0: Do not use key interrupt flags 1: Use key interrupt flags	R/W

The KRCTL register controls the usage of the key interrupt flags, KRF.KIFn (n = 0 to 7), and sets the detection edge.

19.2.2 KRF : Key Return Flag Register

Base address: KINT = 0x4008_5000

Offset address: 0x04

Bit position:	7	6	5	4	3	2	1	0
Bit field:	KIF7	KIF6	KIF5	KIF4	KIF3	KIF2	KIF1	KIF0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	KIF0 to KIF7	Key Interrupt Flag n 0: No interrupt detected 1: Interrupt detected	R/W

The KRF register controls the key interrupt flags, KIFn.

When KRCTL.KRMD = 0, setting the KIFn flag to 1 is prohibited. When setting the KIFn flag to 1, the KIFn value does not change.

To clear the KIFn flag, confirm the target flag is 1 before writing 0 to the bit, then write 1 to the other flags.

19.2.3 KRM : Key Return Mode Register

Base address: KINT = 0x4008_5000

Offset address: 0x08

Bit position:	7	6	5	4	3	2	1	0
Bit field:	KIMC7	KIMC6	KIMC5	KIMC4	KIMC3	KIMC2	KIMC1	KIMC0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	KIMC0 to KIMC7	Key Interrupt Mode Control n 0: Do not detect key interrupt signals 1: Detect key interrupt signals	R/W

The KRM register sets the key interrupt mode.

An interrupt is generated when the target bit in the KRM register is set while a low level (KRCTL.KREG = 0) or a high level (KRCTL.KREG = 1) is being input to the KR0n pin. To ignore this interrupt, set the KRM register after disabling the interrupt handling.

KINT can be assigned in the PmnPFS.PSEL[4:0] bits. The on-chip pull-up resistors can also be applied by setting the associated key interrupt input pin in the pull-up resistor. For details, see section 18, I/O Ports.

19.3 Operation

19.3.1 Operation When Not Using the Key Interrupt Flags (KRCTL.KRMD = 0)

A key interrupt signal, KEY_INTKR, is generated when the valid edge specified in the KRCTL.KREG bit is input to a KR0n pin. To identify the channel to which the valid edge is input, read the port register and check the port level after the KEY_INTKR signal is generated.

The KEY_INTKR signal changes based on the input level of the KR0n pin.

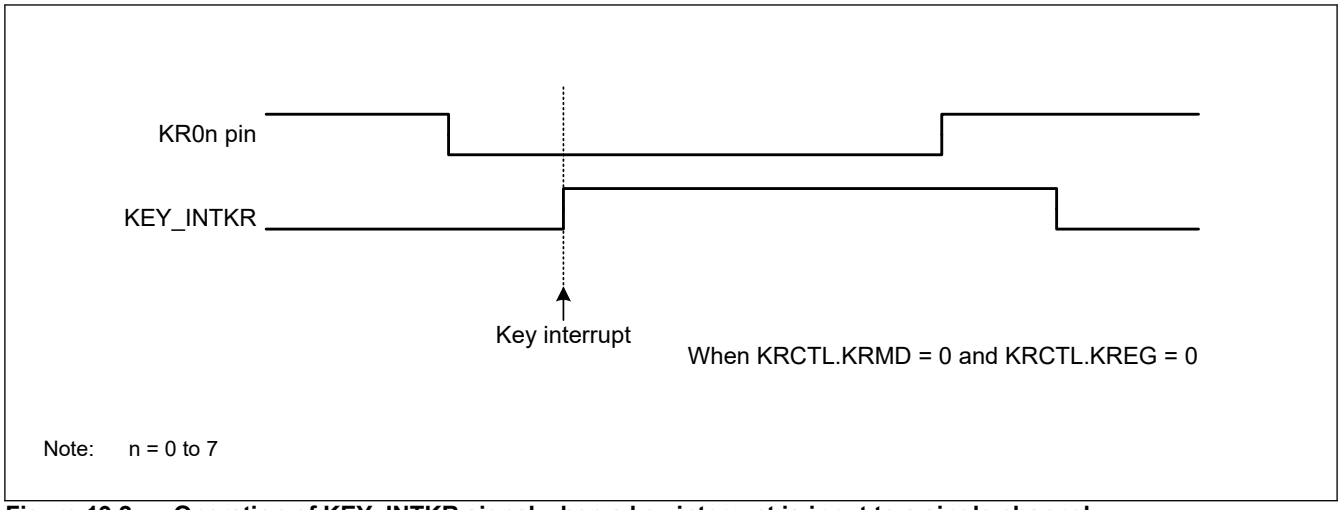


Figure 19.2 Operation of KEY_INTKR signal when a key interrupt is input to a single channel

Figure 19.3 shows the operation when a valid edge is input to multiple KR0n pins. The KEY_INTKR signal is set while a low level is being input to one pin (when KRCTL.KREG = 0). Therefore, even if a falling edge is input to another pin in this period, the KEY_INTKR signal is not generated again. See [1] in Figure 19.3.

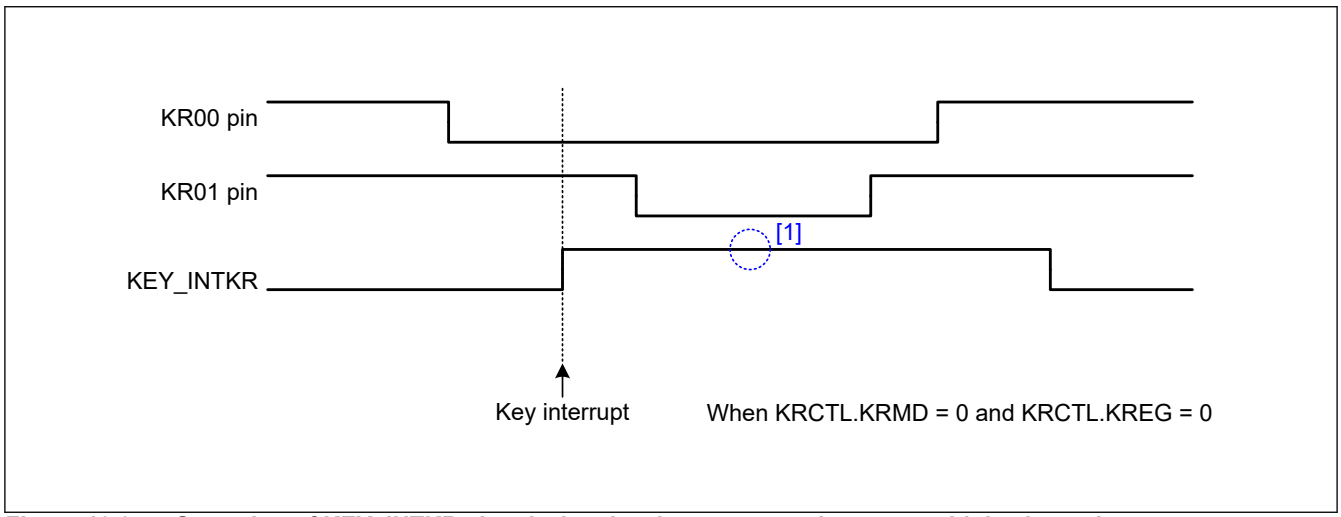


Figure 19.3 Operation of KEY_INTKR signal when key interrupts are input to multiple channels

19.3.2 Operation When Using the Key Interrupt Flags (KRCTL.KRMD = 1)

The KEY_INTKR signal is generated when the valid edge specified in the KRCTL.KREG bit is input to KR0n pins. To identify the channels to which the valid edge is input, read the KRF register after the KEY_INTKR signal is generated. If the KRCTL.KRMD bit is set to 1, clear the KEY_INTKR signal by clearing the associated bit in the KRF register.

As Figure 19.4 shows, only one interrupt is generated each time a falling edge is input to one channel, (when KRCTL.KREG = 0), regardless of whether the KRF.KIFn flag is cleared before or after a rising edge is input.

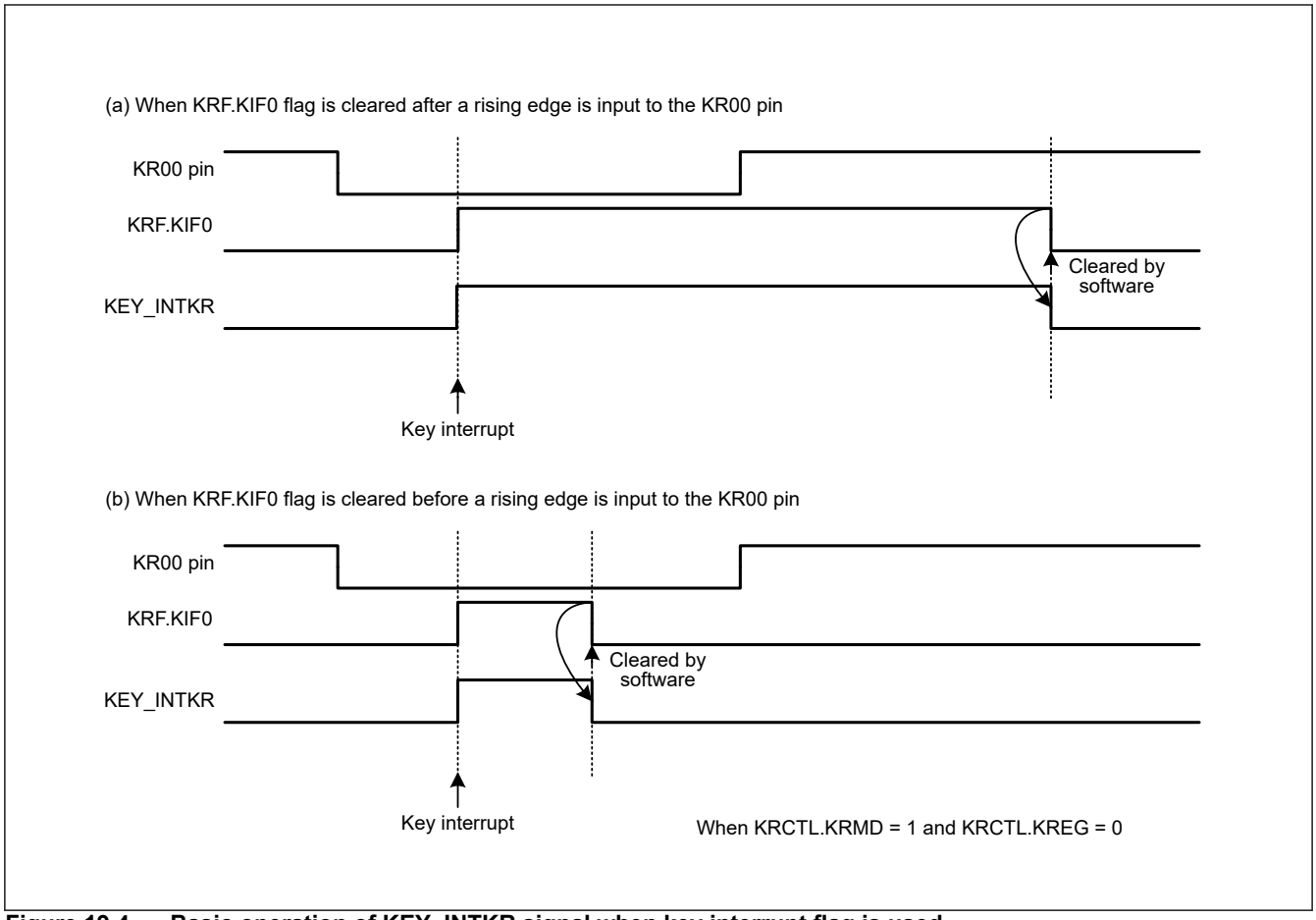


Figure 19.4 Basic operation of KEY_INTKR signal when key interrupt flag is used

Figure 19.5 shows the operation when a valid edge is input to multiple KR0n pins. A falling edge is also input to the KR01 and KR05 pins after a falling edge is input to the KR00 pin (when KRCTL.KREG = 0). The KRF.KIF1 flag is set when the KRF.KIF0 flag is cleared. The KEY_INTKR signal is negated 1 PCLKB clock cycle, after the KRF.KIF0 flag is cleared. See [1] in Figure 19.5.

Also, after a falling edge is input to the KR05 pin, the KRF.KIF5 flag is set. The KRF.KIF1 flag is cleared at time [2] in the figure. The KEY_INTKR signal is negated 1 PCLKB clock cycle, after the KRF.KIF1 flag is cleared. See [3] in the figure. It is therefore possible to generate each key interrupt when a valid edge is input to multiple channels.

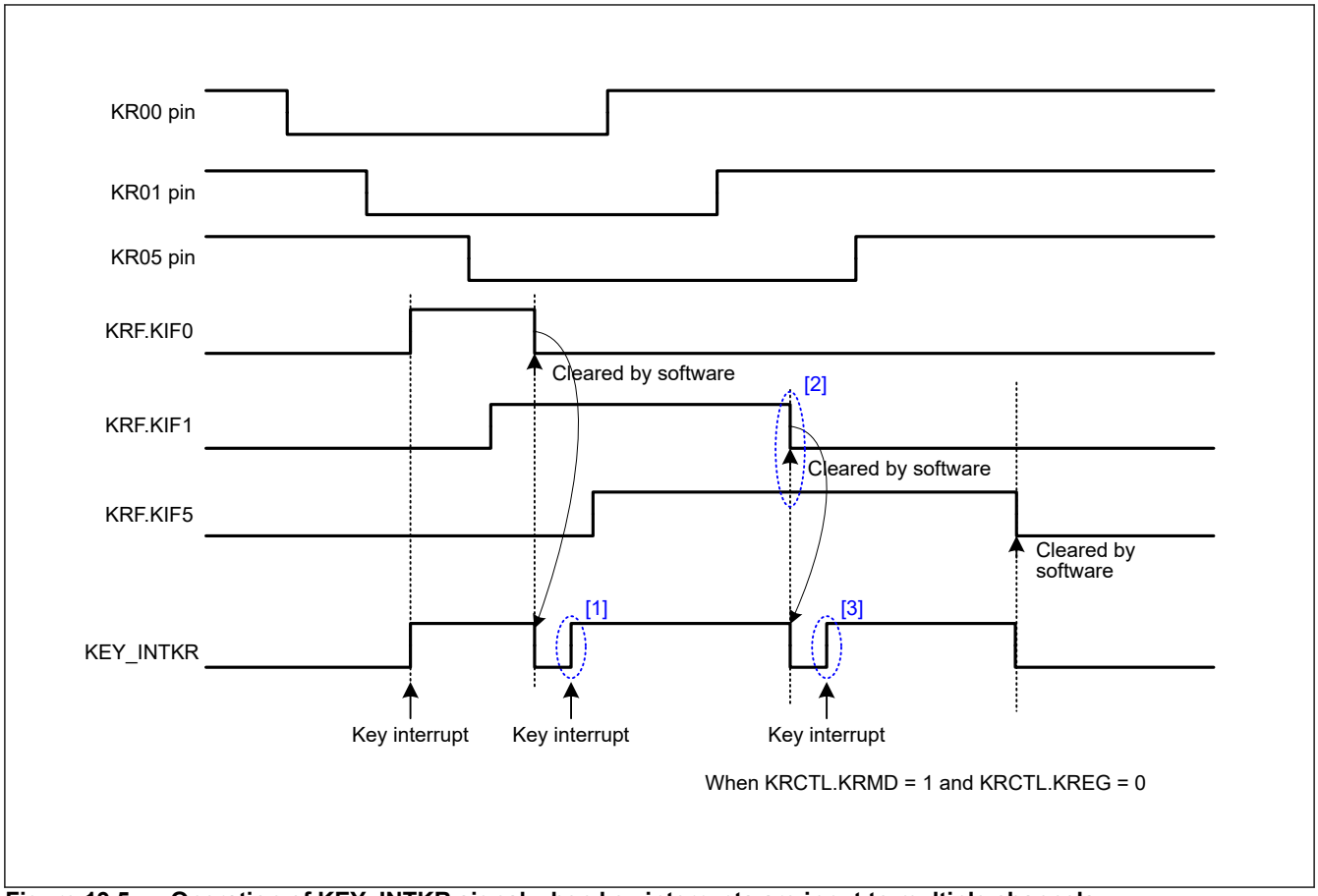


Figure 19.5 Operation of KEY_INTKR signal when key interrupts are input to multiple channels

19.4 Usage Notes

- If the KEY_INTKR signal is used as the snooze request, the KRCTL.KRMD bit should be set to 0.
- If the KEY_INTKR signal is used as the interrupt source for returning to Normal mode from Snooze and Software Standby modes, the KRCTL.KRMD bit should be set to 1.
- When KINT is assigned to a pin, this pin input is always enabled in the Software Standby mode, and if the pin level changes, the associated KRF.KIFn flag can be set. Therefore, a KEY_INTKR signal might be generated on canceling Software Standby mode. To ignore changes to the KR0n pin during a Software Standby, clear the associated KRM.KIMCn bit before entering Software Standby. After canceling Software Standby mode, the KRF.KIFn flag should be cleared before the associated KRM.KIMCn bit can be set.

20. Port Output Enable for GPT (POEG)

20.1 Overview

The POEG issues requests to stop output from output pins of the general PWM timer (GPT). Select the method of detection for stopping the output from the list below.

- Input level detection of the GTETRGn pin (n = A to D)
- Detection from the GPT to stop output
- Detection by comparator (edge detection or level detection)
- Detection of stopping of oscillation by the oscillation stop detection circuit for the main clock
- Register setting

The GTETRGn pin can be used for output to the external trigger input pins of the GPT.

Table 20.1 shows specifications, Figure 20.1 shows a block diagram, and Table 20.2 shows input pins.

Table 20.1 POEG specifications

Item	Description
Request of output stopping in response to input level detection	• The request to stop output is issued to the GPT when a POEGGn.PIDF flag is set in response to the detection of input of the selected level on the corresponding GTETRGn pin (n = A to D). • The request to stop output is issued to the GPT immediately upon detection of input of the selected level on the corresponding GTETRGn pin.
Requests to stop output in the form of an output stopping signal from the GPT	• The request to stop output is issued to the GPT when the GPT detects the active level (high or low) on the GTIOCA and GTIOCB pins at the same time while the corresponding POEGGn.IOCF flag is set. • The request to stop output is issued to the GPT when the GPT detects a deadtime error while the corresponding POEGGn.IOCF flag is set.
Requests to stop output in response to detection by a comparator	• The request to stop output is issued to the GPT when a POEGGn.IOCF flag is set in response to edge-detection by a comparator. • The request to stop output is issued directly to the GPT upon detecting level on a comparator.
Requests to stop output in response to detecting the stopping of oscillation	A request to stop output is issued to the GPT when the oscillation stop detection circuit for the main clock detects the stopping of oscillation while the corresponding POEGGn.OSTPF flag is set.
Requests by software to stop output	The request to stop output is issued to the GPT when the software sets the POEGGn.SSF flag.
Interrupt	• An interrupt is generated in response to the request to stop output in the form of the POEGGn.PIDF flag. • An interrupt is generated in response to the request to stop output in the form of the POEGGn.IOCF flag.
External trigger output to the GPT	The GTETRGn pin is used for output as external triggers.
Noise removal	• Each GTETRGn pin has a digital noise filter. • Four types of sampling clock are selectable for the filter.
TrustZone Filter	• Security attribution can be set for each groups.

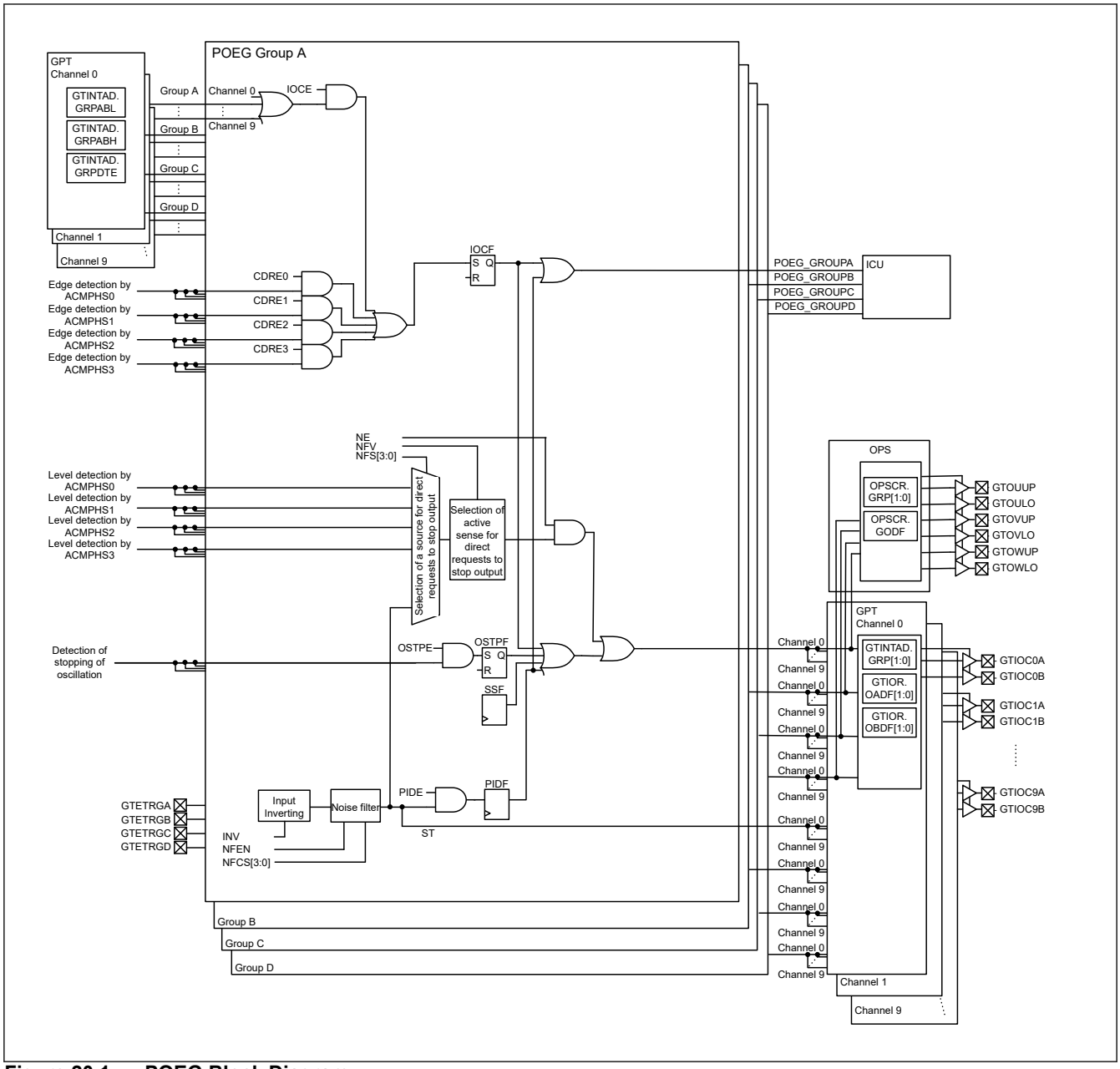


Figure 20.1 POEG Block Diagram

Table 20.2 POEG I/O pins

Pin name	I/O	Function
GTETRGA	Input	Output disable detection signal for GPT output pin and GPT external trigger input pin A
GTETRGB	Input	Output disable detection signal for GPT output pin and GPT external trigger input pin B
GTETRGC	Input	Output disable detection signal for GPT output pin and GPT external trigger input pin C
GTETRGD	Input	Output disable detection signal for GPT output pin and GPT external trigger input pin D

20.2 Register Descriptions

20.2.1 POEGGn : POEG Group n Setting Register (n = A to D)

Base address: POEG = 0x4008_A000

Offset address: 0x000 (POEGGA)
0x100 (POEGGB)
0x200 (POEGGC)
0x300 (POEGGD)

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	CDRE3	CDRE2	CDRE1	CDRE0	—	OSTPE	IOCE	PIDE	SSF	OSTPF	IOCF	PIDF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PIDF	Port Input Detection Flag 0: The selected input level was not detected on the GTETRGn pin 1: The selected input level was detected on the GTETRGn pin	R/W*1
1	IOCF	GPT or ACMPHS Output Stop Request Detection Flag 0: Neither stopping of GPT output nor a comparator edge was detected 1: Either stopping of GPT output or comparator edge was detected	R/W*1
2	OSTPF	Oscillation Stop Detection Flag 0: Stopping of oscillation was not detected 1: Stopping of oscillation was detected	R/W*1
3	SSF	Software Stop Flag 0: Software has not stopped output 1: Software has stopped output	R/W
4	PIDE	Port Input Detection Enable 0: Detection of input levels on the corresponding GTETRGn pin is disabled 1: Detection of input levels on the corresponding GTETRGn pin is enabled	R/W*2
5	IOCE	GPT Output Stop Request Enable 0: Detection of stopping of output from the GPT is disabled 1: Detection of stopping of output from the GPT is enabled	R/W*2
6	OSTPE	Enable Stopping Output on Stopping of Oscillation 0: Detection of stopping of oscillation is disabled 1: Detection of stopping of oscillation is enabled	R/W*2
7	—	This bit is read as 0. The write value should be 0.	R/W
8	CDRE0	ACMPHS0 Enable 0: Comparator edge detection 0 is disabled 1: Comparator edge detection 0 is enabled	R/W*2
9	CDRE1	ACMPHS1 Enable 0: Comparator edge detection 1 is disabled 1: Comparator edge detection 1 is enabled	R/W*2
10	CDRE2	ACMPHS2 Enable 0: Comparator edge detection 2 is disabled 1: Comparator edge detection 2 is enabled	R/W*2
11	CDRE3	ACMPHS3 Enable 0: Comparator edge detection 3 is disabled 1: Comparator edge detection 3 is enabled	R/W*2
15:12	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
16	ST	GTETRGn Input Status Flag 0: The corresponding external trigger for output to the GPT is 0 1: The corresponding external trigger for output to the GPT is 1	R
27:17	—	These bits are read as 0. The write value should be 0.	R/W
28	INV	GTETRGn Input Inverting 0: Input on the GTETRGn pin is not inverted 1: Input on the GTETRGn pin is inverted	R/W
29	NFEN	Noise filter Enable 0: Digital noise filter on the GTETRGn pin is disabled 1: Digital noise filter on the GTETRGn pin is enabled	R/W
31:30	NFCS[1:0]	Noise filter Clock Select 0 0: Samples the input level of GTETRGn pin three times per PCLKB/1 clock 0 1: Samples the input level of GTETRGn pin three times per PCLKB/8 clock 1 0: Samples the input level of GTETRGn pin three times per PCLKB/32 clock 1 1: Samples the input level of GTETRGn pin three times per PCLKB/128 clock	R/W

Note 1. Only 0 can be written to clear the flag.
Note 2. Can be modified only once after a reset.

The POEGGn register (n = A to D) controls requests for stopping output and an external trigger for the GPT based in response to the detection of various signals.

SSF Flag (Software Stop Flag)

Writing 1 to the SSF flag leads to a request to stop output being issued to the GPT, and writing 0 to the flag releases the GPT from the request to stop output. In addition, requests to stop output issued by software can be monitored by reading the flag.

20.2.2 GTONCWPn : GPT Output Stopping Control Group n Write Protection Register (n = A to D)

Base address: POEG = 0x4008_A000

Offset address: 0x040 (GTONCWPA)
0x140 (GTONCWPB)
0x240 (GTONCWPC)
0x340 (GTONCWPD)

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

Bit	Symbol	Function	R/W
0	WP	Register Writing Disable 0: Writing to the GTONCCRn register is enabled 1: Writing to the GTONCCRn register is disabled	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	PRKEY[7:0]	Key Code These bits control whether or not writing new values to the GTONCWPn register is possible. To write to the GTONCWPn register, write 0xA5 to the 8 higher-order bits and write any value to make up a 16-bit unit to the 8 lowerorder bits at the same time. These bits are read as 0x00.	R/W

The GTONCWPn register (n = A to D) enables writing to the GTONCCRn register in order to avoid incorrect writing to registers.

20.2.3 GTONCCRn : GPT Output Stopping Control Group n Controlling Register (n = A to D)

Base address: POEG = 0x4008_A000

Offset address: 0x044 (GTONCCRA)
0x144 (GTONCCRB)
0x244 (GTONCCRC)
0x344 (GTONCCRD)

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

Bit	Symbol	Function	R/W
0	NE	Direct Stopping Request Setting 0: The signal for detection is not set as a direct stopping request signal 1: The signal for detection is set as a direct stopping request signal	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
7:4	NFS[3:0]	Direct Stopping Request Selection Other settings are prohibited. 0x0: Comparator level detection 0 0x1: Comparator level detection 1 0x2: Comparator level detection 2 0x4: Comparator level detection 3 0x7: GTETRGn pin input level detection (n = A to D) Others: Setting prohibited	R/W
8	NFV	Direct Stopping Request Active Sense 0: Stopping output is requested when the output stopping detection signal is 0 1: Stopping output is requested when the output stopping detection signal is 1	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W

The GTONCCRn register (n = A to D) set up requests for stopping output in response to detected signals.

NE bit (Direct Stopping Request Setting)

Writing 1 to the NE bit leads to direct output of the stopping request signal in response to the signal for detection selected by the NFS[3:0] bits.

20.3 Operation

20.3.1 Request to Stop Output in Response to Detection of Input Level on the Corresponding GTETRGn Pin (n = A to D)

There are two types of output stopping request: requests in response to setting of the POEGGn.PIDF flag (n = A to D), and requests directly in response to the detected signal.

1. To request stopping of output in response to setting of the PIDF flag, the input set in the POEGGn register (inversion or non-inversion set in the INV bit, filtering enabled or disabled by the NFEN bit, sample clock for filtering set in the NFCS[1:0] bits) is detected while the POEGGn.PIDE bit is 1. After the POEGGn.PIDF flag is set to 1 in response to this, the request to disable output is issued per group to each channel of the GPT. To de-assert the request signal for stopping output, clear the POEGGn.PIDF flag. For cancelling requests to stop output, see section 20.3.6. Cancelling Requests to Stop Output.
2. To request stopping of output in direct response to detection of the input level of a selected signal, the input set in the POEGGn (inversion or non-inversion set in the INV bit, filtering enabled or disabled by the NFEN bit, sample clock for filtering set in the NFCS[1:0] bits) and GTONCCRn registers (active sense set in the NFV bit) to the GTETRGn pin selected in the GTONCCRn.NFS[3:0] bits is detected while the GTONCCRn.NE bit is 1, and a request to stop output is then directly issued per channel group of the GPT. The signal to request stopping of output is de-asserted when the

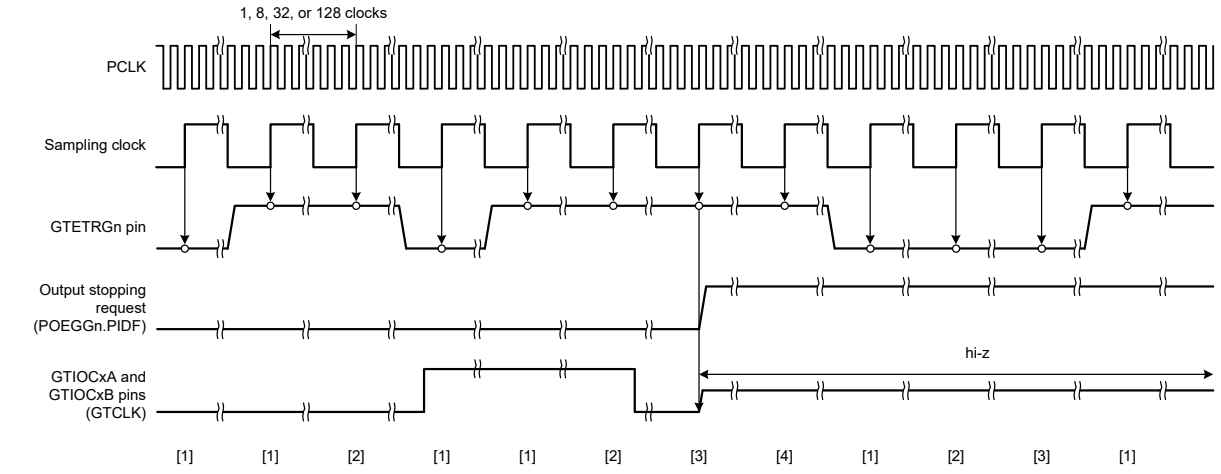
detected input level does not match the conditions for issuing a request. For details, see section 20.3.7. Requests to Stop Output in Response to Detected Signals and Cancelling the Requests.

20.3.1.1 Digital Noise Filter

Each GTETRGn pin input has a digital noise filter. Figure 20.2 shows the operation example in detection of the high level with the use of the filter. When the digital noise filter is enabled (POEGGn.NFEN bit = 1) and the high level is detected 3 consecutive times with the sampling clock cycle selected by the POEGGn.NFCS[1:0] bits while inverted or non-inverted according to the setting of the POEGGn.INV bit, this is regarded as detection of the high level, and the request to stop output is issued to the GPT.

At this time, if the low level is detected even once in the sequence, it is not regarded as detection of the high level. In addition, changes in the levels on pins GTETRGA through GTETRGD are ignored while the sampling clock is not being output.

Digital noise filters can be used with requests to stop output in response to setting of the POEGGn.PIDF flags (n = A to D), requests to stop output in direct response to a detected signal, and the output of external triggers to the GPT.



Note: Each channel output can be set by the GPT32x.
Note: The polarity can be inverted by setting the POEGGn.INV bit.
Note: PCLK = PCLKB

Figure 20.2 Example for Operation of Digital Noise Filter

20.3.2 Requests to Stop Output in Response to Detection of Output Stopping from GPT

When any among a dead-time error or simultaneous high- or low-level output from the GPT is detected, the corresponding POEGGn.IOCF flag is set to 1 and the request to stop output per group is issued to each channel in the GPT. The POEGGn.IOCF flag is used to indicate both edge detection by the comparator and requests to stop output. To cancel a request to stop output, clear the given POEGGn.IOCF flag. For details, see section 20.3.6. Cancelling Requests to Stop Output.

To detect any among a dead-time error or simultaneous high- or low-level output from the GPT, detection of output stopping must be permitted in the GRPDTE, GRPABH, and GRPABL bits of the given GPT32n.GTINTAD register. Specify the group of the GPT for which stopping is to be detected in the GPT32n.GTINTAD.GRP[1:0] bits. For details, see section 21.2.15. GTINTAD : General PWM Timer Interrupt Output Setting Register.

20.3.3 Request to Stop Output in Response to Comparator Detection

A request to stop output can be issued to the GPT in response to detection by a comparator. There are two types of request to stop output: requests output in response to setting of a POEGGn.IOCF flag (n = A to D) due to edge detection by the comparator and requests that are directly output in response to level detection by the comparator.

1. To stop output in response to setting of an IOCF flag, the corresponding edge is detected by the comparator while the POEGGn.CDRE[3:0] bits are 1. The POEGGn.IOCF flag is then set to 1, which leads to a request to stop output from

the given group for each channel of the GPT. The POEGGn.IOCF flag is used for both comparator edge detection and the request to stop output. To cancel the request to stop output, clear the POEGGn.IOCF flag. For details, see section 20.3.6. Cancelling Requests to Stop Output.

2. The output can also be stopped by using a detected signal as a direct request to stop output. When the comparator level detection signal selected in the GTONCCRn.NFS[3:0] bits matches the level set in the GTONCCRn.NFV bit, the request to stop output per group is issued to each channel of the GPT. The request to stop output is released when the comparator level detection does not match the issuing conditions. For details, see section 20.3.7. Requests to Stop Output in Response to Detected Signals and Cancelling the Requests.

20.3.4 Requests to Stop Output by Oscillation Stop Detection

When the circuit for detecting stopping of oscillation by the main clock oscillator detects the stopping of the oscillation while a POEGGn.OSTPE bit is 1, the POEGGn.OSTPF flag is set to 1 and a request to stop output per group is issued to each channel of the GPT. To cancel the request to stop output, clear the POEGGn.OSTPF flag. For details, see section 20.3.6. Cancelling Requests to Stop Output.

20.3.5 Requests to Stop Output by a Register

Writing 1 to the software stop flag (POEGGn.SSF) leads to a request to stop output per group to each channel in the GPT. To release from the request to disable output, clear the POEGGn.SSF flag. For details, see section 20.3.6. Cancelling Requests to Stop Output.

20.3.6 Cancelling Requests to Stop Output

Requests to stop output are cancelled in any of the following three ways.

1. Cancellation by a reset (return to the initial state)
2. Cancellation by clearing all flags in the POEGGn register
3. Cancellation in response to direct input of a detected signal

(1) Cancellation by a reset

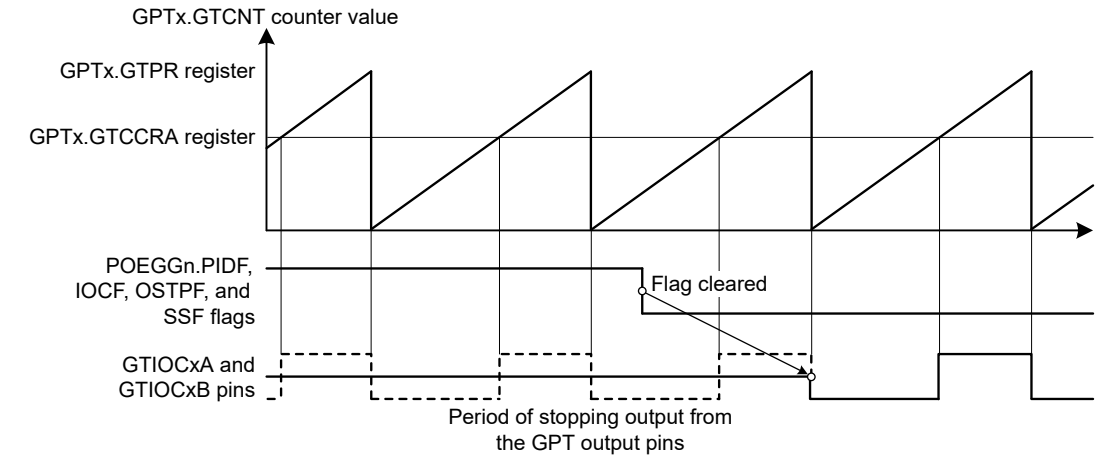
Any type of reset can cancel a request to stop output. For details, see section 5, Resets.

(2) Cancellation by clearing all flags in the POEGGn register

- POEGGn.PIDF
- POEGGn.IOCF
- POEGGn.OSTPF
- POEGGn.SSF

Cancellation of the request is taken into the GPT at the end of the cycle of counting by the GPT, and the output pins are released from being stopped no less than 3 GTCLK cycles from that time. Figure 20.3 shows the timing of release from the output-stopped state. To clear each of the flags, read the status flag for each source to check that the source condition is not being detected, and then write 0. Sources other than the edge detection by the comparators cannot be cleared even if the flag is cleared in the detected state. Since the comparators detect edges in edge detection, writing 0 in the detected state clears the flag, and the flag is not set until the relevant source generates the next edge. Status flags for the respective sources are listed below.

Detection of input level	POEGGn.ST (GTETRGn input status flag)
Comparator edge detection	ACMPHSn.CMPMON.CMPMON (comparator output monitoring flag)
Oscillation stop detection	OSTDSR.OSTDF (oscillation stop detection flag)
Stopping detection from the GPT	GPT32n.GTST.DTEF (dead time error flag) GPT32n.GTST.OABLF (same time output level low flag) GPT32n.GTST.OABHF (same time output level high flag)



Note: GPTx indicates GPT32x.

Figure 20.3 Timing of Re-enabling Output from the GPT Output Pins Following Cancellation of a Request to Stop Output

(3) Cancellation in response to direct input of a detected signal

For details, see section 20.3.7. Requests to Stop Output in Response to Detected Signals and Cancelling the Requests.

20.3.7 Requests to Stop Output in Response to Detected Signals and Cancelling the Requests

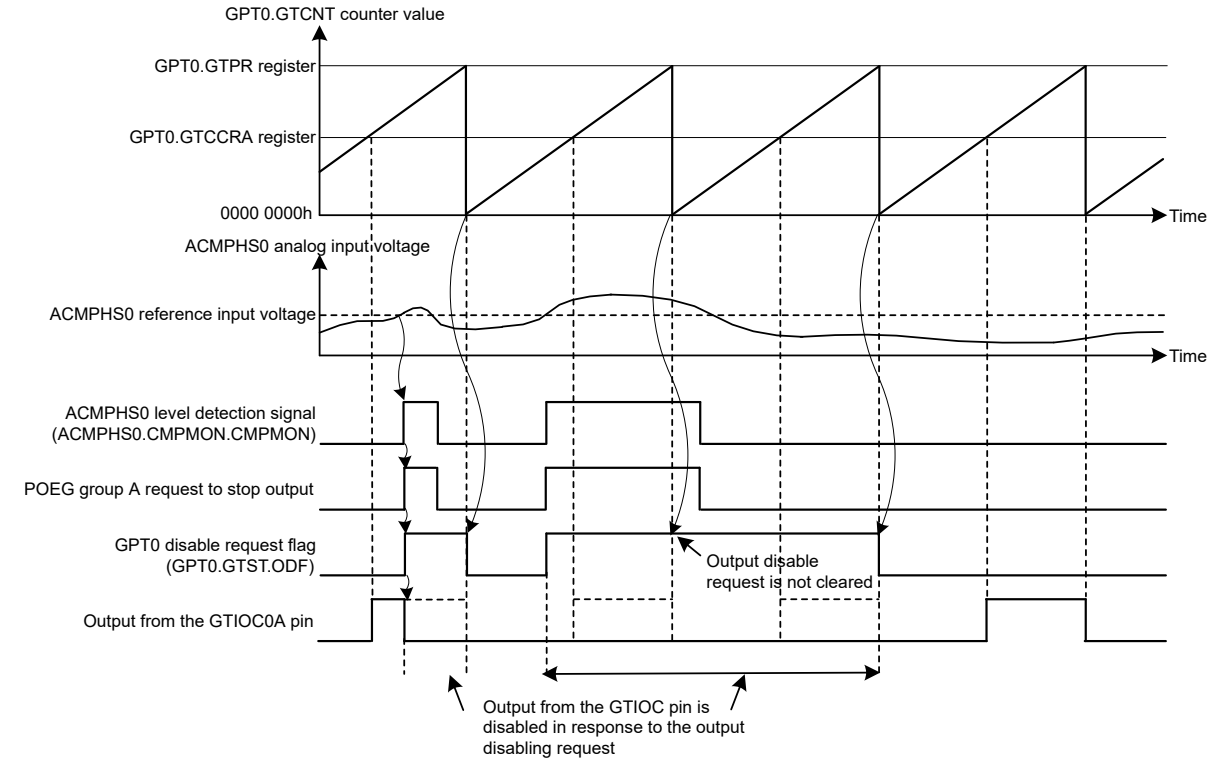
The input level detection signals on the GTETRGn pin (n = A to D) and comparator level detection signals on the ACMPHSm (m = 0 to 3) can be used as direct requests to stop output in response to detected signals. The sources of the signals for detection are selected in the GTONCCRn.NFS[3:0] bits, and the active senses of the signals for detection are set in the GTONCCRn.NFV bit. Setting a GTONCCRn.NE bit to activate direct requests to stop output results in the issuing of a request to stop output to the GPT when the selected source signal for output stopping detection is generated.

The request to stop output is released when the input level detection signals on the GTETRGn pin or comparator level detection does not match the issuing conditions. To cancel a request, confirm that the values of the PIDF and IOCF flags in the POEGGn register have become 0.

Figure 20.4 shows the operation of a request to stop output issued in response to level detection by the ACMPHS0. The example shows the case where the comparator detects the analog input voltage on ACMPHS0 becoming higher than the reference voltage while the GPT is producing PWM waveforms on the GTIOC0A pin on a cyclic-counting basis, and the level-detection signal from the comparator is input to the POEG. The POEG outputs a request to stop output to the GPT on the basis of detection as described above, and the GPT remains stopped until the end of the cycle of counting, even if the request to stop output is cancelled before then.

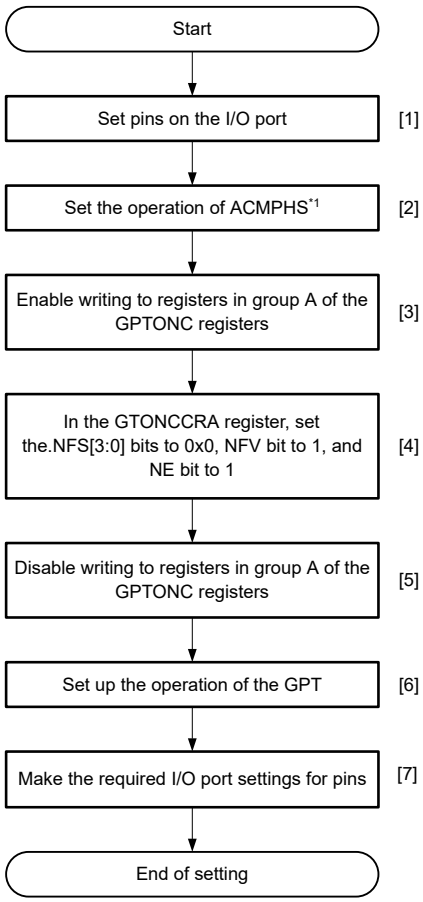
If the request to stop output has not been cancelled at the end of the cycle of counting cycle, output from the GPT remains stopped until the end of the next cycle of counting, and so on.

The output-stopped state can be checked by reading the ODF flag in General PWM Timer Status Register (GTST) of the GPT. For details, see the description of the ODF flag in section 21.2.16. GTST : General PWM Timer Status Register. Figure 20.5 shows the procedure for setting requests to stop output in response to detected signals.



Note: GPT0 indicates GPT320.

Figure 20.4 Example of Operation to Stop Output from the GTIOC Pin In Response to Level Detection by a Comparator



[1] Make the I/O port settings.
Set the corresponding pins of the I/O port to general purpose I/O port pins in order to use the ACMPHS0 analog input pin function and GPT output pin function.

[2] Set the operation of ACMPHS0 to output a level detection signal to the POEG in response to comparing the analog input voltage and reference input voltage.

[3] Write 0xA500 to the GTONCWPA register to enable writing to the GTONCCPRA register.

[4] Select level detection for the comparator by setting the GTONCCRA.NFS[3:0] bits to 0x0. Set the GTONCCRA.NFV bit to 1 so that the request to stop output is sent when the selected source indicates 1.
Set the GTONCCRA.NE bit to 1 to enable the output of requests to stop output.

[5] Write 0xA501 to the GTONCWPA register to disable writing to the GTONCCPRA register.

[6] Set the GPT for the output of a PWM waveform on its GTIOCn pin.

[7] Set the I/O port in order to use the required analog input pin of ACMPHS0 and output pin of the GPT.

The settings described above lead to the output of requests to stop output in response to comparison of the analog input voltage and reference input voltage while a PWM waveform is being output from the GPT pin, stopping output from the GPT pin.

Note 1. In this example, the comparator output is non-inverted.

Figure 20.5 Example of Settings to Stop Output on the GTIOC Pin in Response to Level Detection by a Comparator

20.4 Interrupt Sources

The POEG generates interrupts for the interrupt controller when any of the following is detected.

- Detection of level of input, indicated by the POEGGn.PIDF flag
- Detection of stopping of output from the GPT, indicated by the POEGGn.IOCF flag
- Detection of edges by the comparator, indicated by the POEGGn.IOCF flag

Table 20.3 shows interrupt sources and conditions.

Table 20.3 Interrupt sources and conditions (1 of 2)

Interrupt source	Symbol	Corresponding flag	Trigger condition
POEG Group A Interrupt	POEG_GROUPA	POEGGA.IOCF	Detection of stopping of output from the GPT
			Detection of edges by the comparator
		POEGGA.PIDF	Detection of the level input from the GTETRGA pin

Table 20.3 Interrupt sources and conditions (2 of 2)

Interrupt source	Symbol	Corresponding flag	Trigger condition
POEG Group B Interrupt	POEG_GROUPB	POEGGB.IOCF	Detection of stopping of output from the GPT
			Detection of edges by the comparator
		POEGGB.PIDF	Detection of the level input from the GTETRGB pin
POEG Group C Interrupt	POEG_GROUPC	POEGGC.IOCF	Detection of stopping of output from the GPT
			Detection of edges by the comparator
		POEGGC.PIDF	Detection of the level input from the GTETRGC pin
POEG Group D Interrupt	POEG_GROUPD	POEGGD.IOCF	Detection of stopping of output from the GPT
			Detection of edges by the comparator
		POEGGD.PIDF	Detection of the level input from the GTETRGD pin

20.5 External Trigger Output to GPT

Detection of the level input from the GTETRGD pin can be monitored by the POEGGn.ST flag via active-sense selection and digital noise filtering. The GPT can function as follows in response to external trigger signals.

- Count start
- Count stop
- Counter clearing
- Up-counting
- Down-counting
- Input capture

For details of the above functions, see section 21, General PWM Timer (GPT).

20.6 Usage Notes

20.6.1 Transitions to Low-Power Modes

When the POEG is used, it should not be transitioned to software standby or deep software standby modes. In this mode, the stopping of output cannot be requested since the POEG is stopped.

20.6.2 Setting the Function for Stopping the Module

The module-stop control register can be set to enable or disable operation of the POEG. The POEG is stopped immediately after a reset. Registers of the POEG become accessible following release from the module-stopped state. For details, see section 10, Low Power Modes.

20.6.3 Duplication of Requests to Stop Output

While either the PIDF or IOCF flag in the POEGGn register is 1, cancellation of requests to stop by the detection signal set in the GTONCCRn register does not work due to the request to stop still being output because of the value of the flag. That is, note that requests to stop output will not be cancelled when a corresponding flag is set stop output in response to detection. Request signals to stop output in response to flag setting are obtained as the logical OR of the corresponding detection signals for stopping output.

21. General PWM Timer (GPT)

21.1 Overview

The General PWM Timer (GPT) is a 32-bit timer with GPT32 × 10 channels. PWM waveforms can be generated by controlling the up-counter, down-counter, or the up- and down-counter. In addition, PWM waveforms can be generated for controlling brushless DC motors. The GPT can also be used as a general-purpose timer.

This GPT provides the high resolution PWM waveform generation function for the channel 0 to 3, the A/D conversion start request function, the asymmetric automatic dead time setting function, and the enhanced interrupt skipping function. In addition, it is enhanced with the external pulse width measuring function for the channel 0 to 3, the additional PWM modes for the channel 4 to 9, the extended buffer transfer function, the GTCPPO pin output function, and the inter-channel cooperation function.

Table 21.1 lists the GPT specifications, Table 21.2 shows the GPT functions, and Figure 21.1 shows a block diagram.

Table 21.1 GPT specifications

Item	Description
Functions	• 32 bits × 10 channels (GPT32n (n = 0 to 9)) • Up-counting or down-counting (saw waves) or up/down-counting (triangle waves) for each counter • Clock sources independently selectable for each channel • Two input/output pins per channel • Two output compare/input capture registers per channel • For the two output compare/input capture registers of each channel, four registers are provided as buffer registers and are capable of operating as comparison registers when buffering is not in use • In output compare operation, buffer switching can be at crests or troughs, enabling the generation of laterally asymmetric PWM waveforms • Registers for setting up frame cycles in each channel with capability for generating interrupts at overflow or underflow • Generation of dead times in PWM operation • Generation of high accuracy duty in the vicinity of duty 0% and 100% PWM waveform • In output compare operation, setting compare register is immediately used to generate PWM waveform with dead times • Synchronous starting, stopping and clearing counters for arbitrary channels • Count start, count stop, count clear, up-count, down-count, or input capture operation in response to a maximum of 8 ELC events • Count start, count stop, count clear, up-count, down-count, or input capture operation in response to the status of two input pins • Count start, count stop, count clear, up-count, down-count, or input capture operation in response to a maximum of 4 external triggers • Output pin disable function by dead time error and detected short-circuits between output pins • A/D conversion start request generation function • PWM waveform for controlling brushless DC motors can be generated • Compare match A to F event, overflow/underflow event, and input UVW edge event can be output to the ELC • Enables the noise filter for input capture and input UVW • Period count function • External pulse width measuring function • Logical operation between the channel output • Synchronous counter clearing/counter setting/input capture among channels • Bus clock: PCLKA, Core clock: GTCLK*1 • Frequency ratio: PCLKA:PCLKD = 1:N (N = 1/2/4/8/16/32/64) (when using synchronous clock), PCLKA ≤ GTCLK (when using asynchronous clock)

Note 1. GPT core clock (GTCLK) is PCLKD when synchronous clock is selected, and GPTCLK when asynchronous clock is selected. See Figure 21.3.

Table 21.2 GPT functions (1 of 2)

Parameter		Description
Count clock		GTCLK GTCLK/2 GTCLK/4 GTCLK/8 GTCLK/16 GTCLK/32 GTCLK/64 GTCLK/128 GTCLK/256 GTCLK/512 GTCLK/1024 GTETRGA, GTETRGB, GTETRGC, GTETRGD
Output compare/input capture registers (GTCCR)		GTCCRA GTCCRB
Compare/buffer registers		GTCCRC GTCCRD GTCCRE GTCCRF
Cycle setting register		GTPR
Cycle setting buffer register		GTPBR GTPDBR
I/O pins		GTIOCnA GTIOCnB (n = 0 to 9)
External trigger input pin*1		GTETRGA GTETRGB GTETRGC GTETRGD
Counter clear sources		GTPR register compare match Input capture Input pin status ELC event input GTETRGn (n = A to D) pin input GTCCR register compare match Other channel's counter clear sources
Period count function		Available (GPT32n (n = 0 to 3))
Compare match output	Low output	Available
	High output	Available
	Toggle output	Available
Input capture function		Available
Automatic addition of dead time		Available
PWM mode		Available
High accuracy PWM waveform		Available (GPT32n (n = 0 to 3))
Phase count function		Available (GPT32n (n = 0 to 3))
External pulse width measuring function		Available (GPT32n (n = 0 to 3))
Buffer operation		Double buffer Simultaneous operation disable control for multiple channels Buffer operation by counter clearing/compare match
One-shot operation		Available
DMAC/DTC activation		All the interrupt sources
A/D conversion start request		Compare match of GTADTRA or GTADTRB register
Brushless DC motor control function		Available

Table 21.2 GPT functions (2 of 2)

Parameter	Description
Interrupt sources	11 sources ● GTCCRA compare match/input capture(GPTn_CCMPA) ● GTCCRB compare match/input capture(GPTn_CCMPB) ● GTCCRC compare match(GPTn_CMPC) ● GTCCRD compare match(GPTn_CMPD) ● GTCCRE compare match(GPTn_CMPE) ● GTCCRF compare match(GPTn_CMPF) ● GTADTRA compare match (GPTn_ADTRGA) ● GTADTRB compare match (GPTn_ADTRGB) ● GTCNT overflow (GTPR compare match) (GPTn_OVF) ● GTCNT underflow (GPTn_UDF) ● GTPC count stop(GPTx_PC) (x = 0 to 3)
Interrupt skipping function	● Skipping of interrupts of GTCNT counter overflow (GTPR register compare match) (GTPn_OVF) and GTCNT counter underflow (GTPn_UDF) (interlocked with other interrupts and A/D conversion start requests) ● Skipping of GTADTRA and GTADTRB register compare match (GPT32y (y = 4 to 9)) ● Buffer operation skipping function
Event linking (ELC) function	Available*2
Noise filtering function	Available
Logical operation between the channel output	Available
Synchronous counter clearing/counter setting/input capture	Available

Note 1. GTETRGn connects to GPT through the POEG module. Therefore, to use the GPT function, supply the POEG clock by clearing the MSTPCRD.MSTPDn (n = 11 to 14) bit.
Note 2. See section 21.6. Operations Linked by ELC.

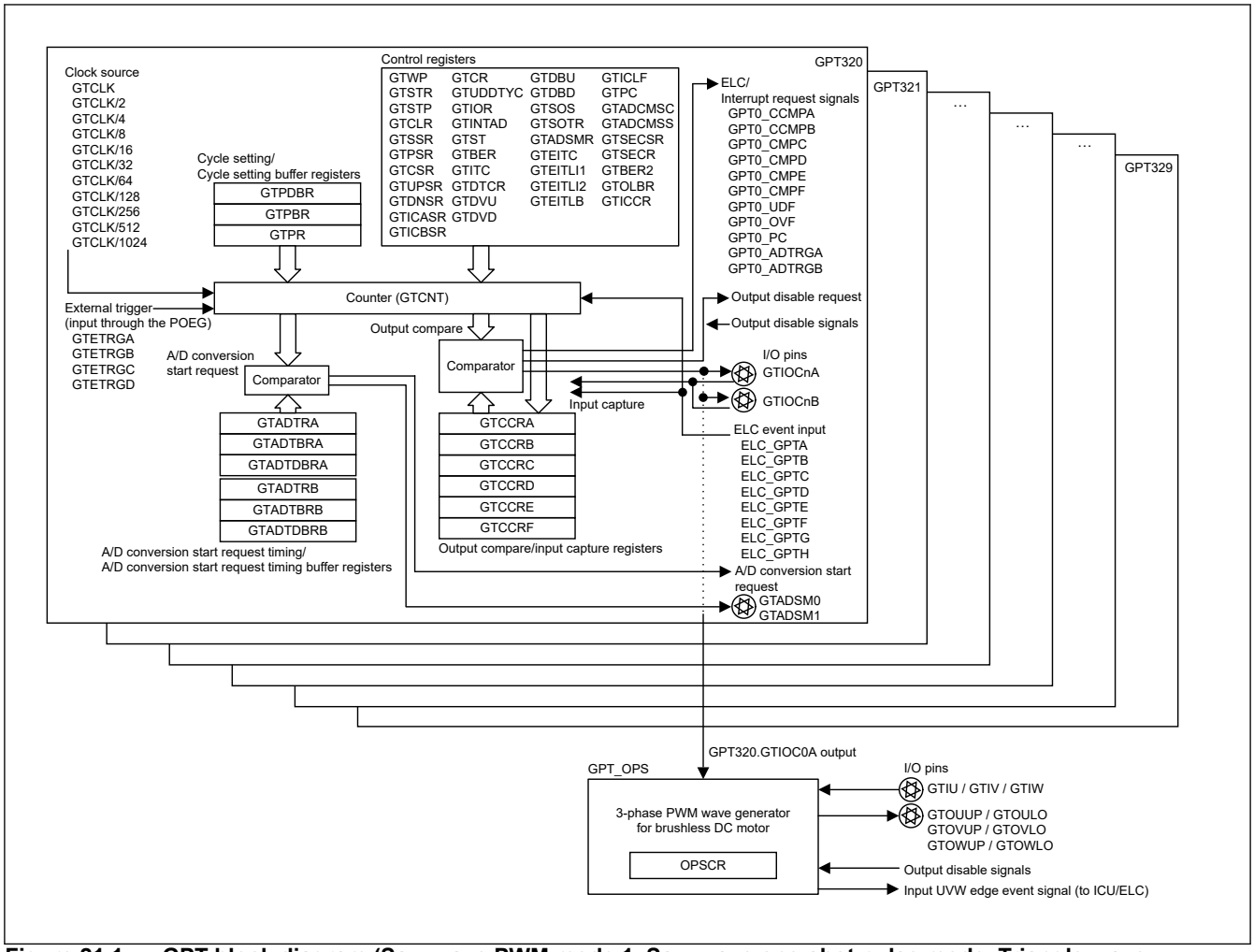


Figure 21.1 GPT block diagram (Saw-wave PWM mode 1, Saw-wave one-shot pulse mode, Triangle-wave PWM mode1,2,3)

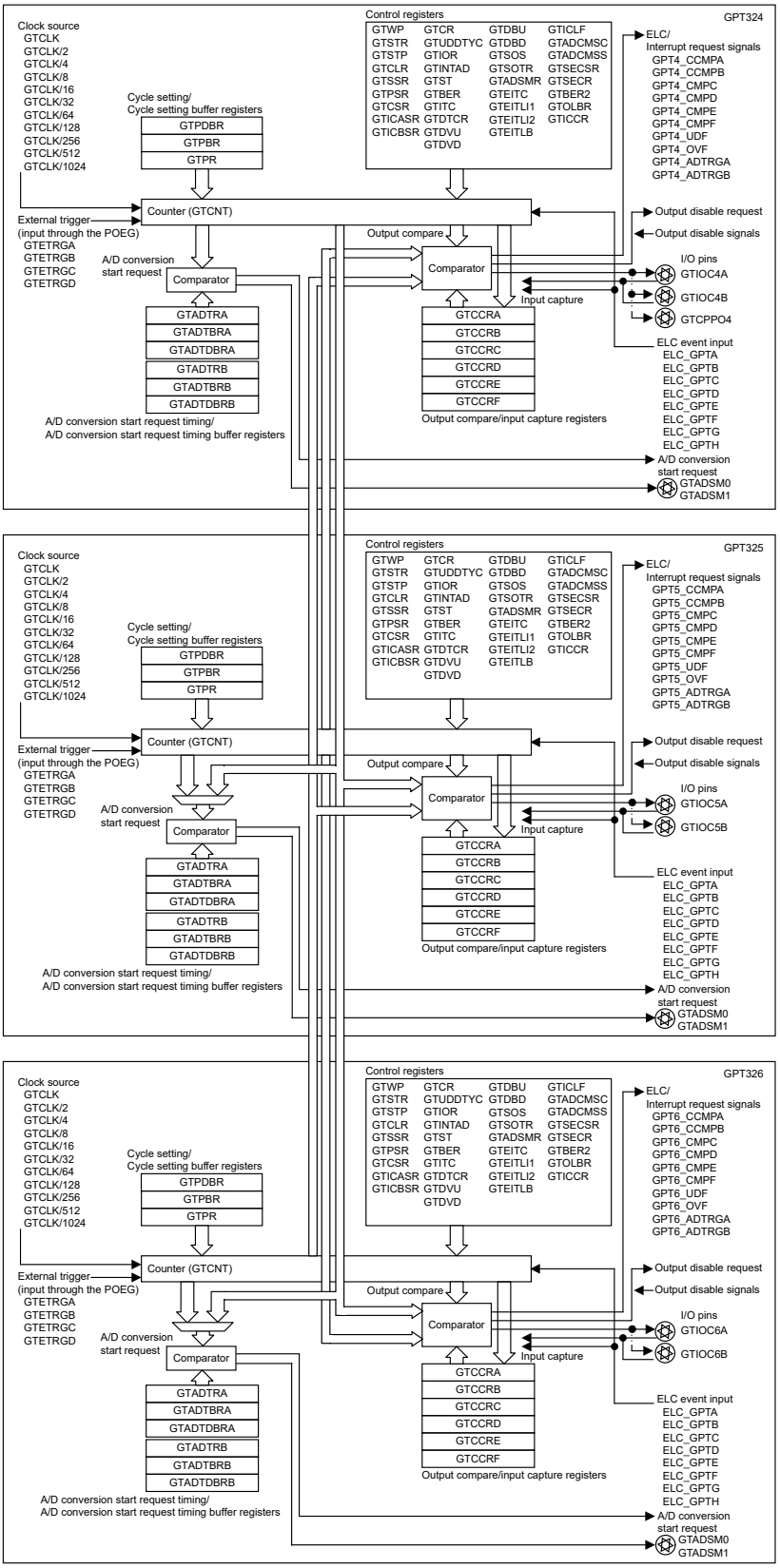


Figure 21.2 GPT block diagram (Saw-wave PWM mode 1, 2, Saw-wave one-shot pulse mode, Triangle-wave PWM mode1,2,3, Complementary PWM mode 1, 2, 3, 4)

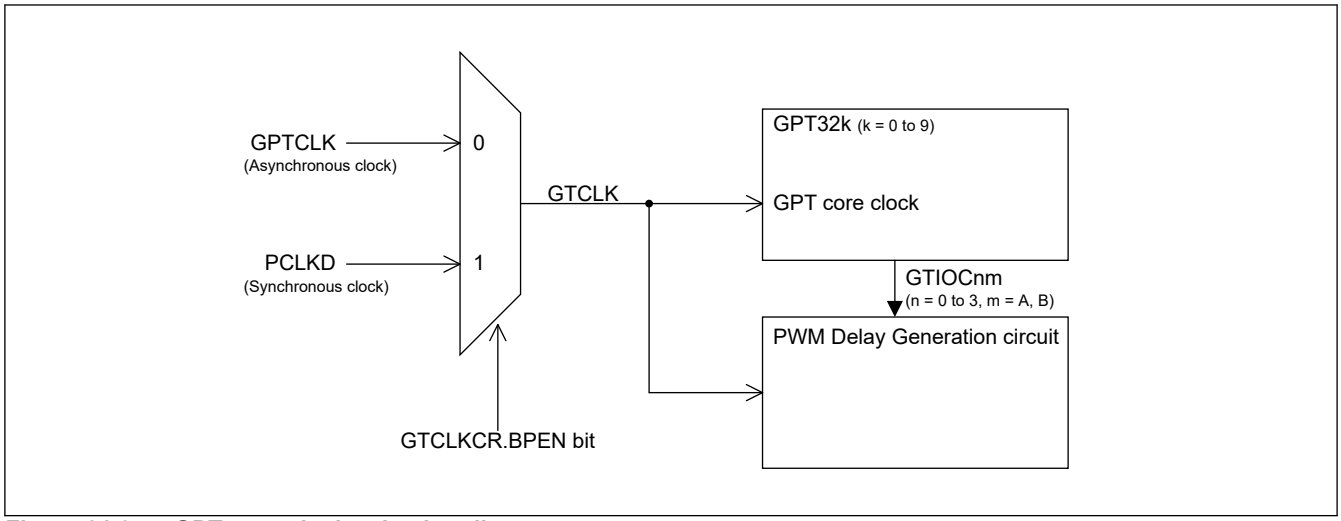


Figure 21.3 GPT core clock selection diagram

In this specification, three consecutive channels to configure complementary PWM mode is defined as complementary PWM mode channel group. The lowest position channel of complementary PWM mode channel group is defined as master channel. The second channel is defined as slave channel 1. The highest position channel is defined as slave channel 2.

Figure 21.4 shows an example using multiple GPTs.

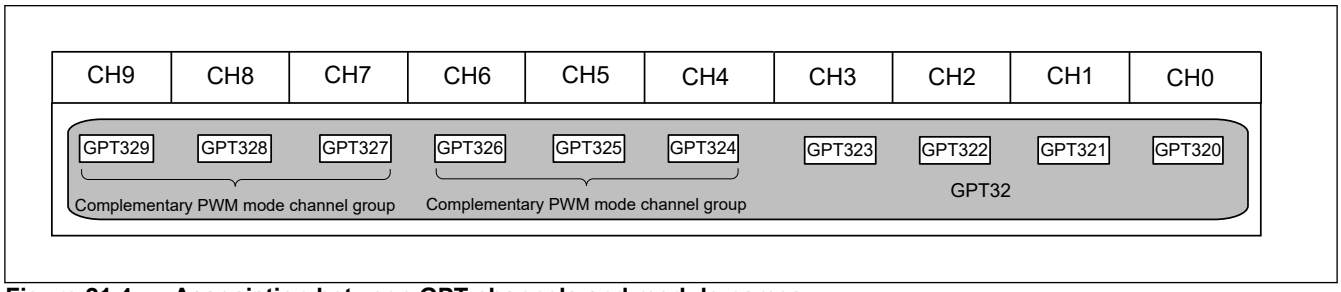


Figure 21.4 Association between GPT channels and module names

Table 21.3 lists the I/O pins.

Table 21.3 GPT I/O pins

Channel	Pin name	I/O	Function
Common	GTETRGx	Input	External trigger input pin x (input through the POEG)
	GTADSM0	Output	A/D conversion start request monitor 0 output pin
	GTADSM1	Output	A/D conversion start request monitor 1 output pin
GPT32n	GTIOCnA	I/O	GTCCRA register input capture input/output compare output/PWM output pin
	GTIOCnB	I/O	GTCCRB register input capture input/output compare output/PWM output pin
	GTCPPOk	Output	Toggle output synchronized with PWM period
GPT_OPS	GTIU	Input	Hall sensor input pin U
	GTIV	Input	Hall sensor input pin V
	GTIW	Input	Hall sensor input pin W
	GTOUUP	Output	3-phase PWM output for BLDC motor control (positive U-phase)
	GTOULO	Output	3-phase PWM output for BLDC motor control (negative U-phase)
	GTOVUP	Output	3-phase PWM output for BLDC motor control (positive V-phase)
	GTOVLO	Output	3-phase PWM output for BLDC motor control (negative V-phase)
	GTOWUP	Output	3-phase PWM output for BLDC motor control (positive W-phase)
	GTOWLO	Output	3-phase PWM output for BLDC motor control (negative W-phase)

Note: x: A to D

n: 0 to 9
k: 0 to 4, 7

21.2 Register Descriptions

21.2.1 GTWP : General PWM Timer Write-Protection Register

Base address: GPT32n = 0x4016_9000 + 0x0100 × n (n = 0 to 9)

Offset address: 0x00

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PRKEY[7:0]								—	—	—	CMNWP	CLRWP	STPWP	STRWP	WP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	WP	Register Write Disable 0: Write to the register enabled 1: Write to the register disabled	R/W
1	STRWP	GTSTR.CSTRT Bit Write Disable 0: Write to the bit is enabled 1: Write to the bit is disabled	R/W
2	STPWP	GTSTP.CSTOP Bit Write Disable 0: Write to the bit is enabled 1: Write to the bit is disabled	R/W
3	CLRWP	GTCLR.CCLR Bit Write Disable 0: Write to the bit is enabled 1: Write to the bit is disabled	R/W
4	CMNWP	Common Register Write Disabled 0: Write to the register is enabled 1: Write to the register is disabled	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
15:8	PRKEY[7:0]	GTWP Key Code When 0xA5 is written to these bits, writing to the WP, STRWP, STPWP, CLRWP, and CMNWP bits are permitted. These bits are read as 0.	W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

GTWP enables or disables writing to registers to prevent accidental modification. Protection by the GTWP register is only for the writes by the CPU. GTWP does not protect registers from updates that occur in association with CPU writes.

WP bit (Register Write Disable)

The following is a list of write enabled or disabled registers:

GTSSR, GTPSR, GTCSR, GTUPSR, GTDNSR, GTICASR, GTICBSR, GTCR, GTUDDTYC, GTIOR, GTINTAD, GTST, GTBER, GTITC, GTCNT, GTCCRA, GTCCRB, GTCCRC, GTCCRD, GTCCRE, GTCCRF, GTPR, GTPBR, GTPDBR, GTADTRA, GTADTBRA, GTADTDBRA, GTADTRB, GTADTBRB, GTADTDBRB, GTDTCR, GTDVU, GTDVD, GTDBU, GTDBD, GTSOS, GTSOTR, GTADSMR, GTEITC, GTEITLI1, GTEITLI2, GTEITLB, GTICLF, GTPC, GTADCMSC, GTADCMSS, GTBER2, GTOLBR, GTICCR.

STRWP bit (GTSTR.CSTRT Bit Write Disable)

The STRWP bit enables or disables starting the updating of counter values by writing to the CSTRTn bit (n = 0 to 9) corresponding to a channel number in the GTSTR register.

The bit position of each CSTRTn bit in the GTSTR register is allocated to the channel with the corresponding number, and writing to the GTSTR register for any channel results in writing to the registers of all channels. The STRWP bit for each channel does not control writing but only controls updating of the CSTRT bit for the corresponding channel when simultaneously writing to all channels.

Therefore, when writing to the CSTRT bits of a channel for which the setting of the STRWP bit is 1(disabling writing), the CSTRT bit for the given channel is not updated, but the CSTRT bits corresponding to channel for which the setting of the STRWP bit is 0 (enabling writing) are updated. For example, when the setting of the GPT320.GTWP.STRWP bit is 0 (enabling writing), writing 1 to the GPT321.GTSTR.CSTRT0 bit when its current setting is 0 causes the value to be updated, and the GPT320.GTCNT counter starts to run. When the setting of the GPT320.GTWP.STRWP bit is 1 (disabling writing), writing 1 to the GPT321.GTSTR.CSTRT0 bit when its current setting is 0 leaves the bit with the value 0, and the GPT320.GTCNT counter does not run.

If you want to protect all bits in the GTSTR register from being updated, set the STRWP bits of all channels to 1.

STPWP bit (GTSTP.CSTOP Bit Write Disable)

The STPWP bit enables or disables starting the updating of counter values by writing to the CSTOPn bit (n = 0 to 9) corresponding to a channel number in the GTSTP register.

The bit position of each CSTOPn bit in the GTSTP registers is allocated to the channel with the corresponding number, and the writing to the GTSTP register for any channel results in writing to the registers of all channels. The STPWP bit for each channel does not control writing but only controls updating of the CSTOP bit for the corresponding channel when simultaneously writing to all channels.

Therefore, when writing to the CSTOP bits of a channel for which the setting of the STPWP bit is 1(disabling writing), the CSTOP bit for the given channel is not updated, but the CSTOP bits corresponding to channel for which the setting of the STPWP bit is 0 (enabling writing) are updated. For example, when the setting of the GPT320.GTWP.STPWP bit is 0 (enabling writing), writing 1 to the GPT321.GTSTP.CSTOP0 bit when its current setting is 0 causes the value to be updated, and the GPT320.GTCNT counter is stopped. When the setting of the GPT320.GTWP.STPWP bit is 1 (disabling writing), writing 1 to the GPT321.GTSTP.CSTOP0 bit when its current setting is 0 leaves the bit with the value 0, and the GPT320.GTCNT counter is not stopped.

If you want to protect all bits in the GTSTP register from being updated, set the STPWP bits of all channels to 1.

CLRWP bit (GTCLR.CCLR Bit Write Disable)

CLRWP bit enables or disables starting the updating of counter values by writing to the CCLRn bit (n = 0 to 9) corresponding to a channel number in the GTCLR register.

The bit position of each CCLRn bit in the GTCLR registers is allocated to the channel with the corresponding number, and the writing to the GTCLR register for any channel results in writing to the registers of all channels. The CLRWP bit for each channel does not control writing but only controls updating of the CCLR bit for the corresponding channel when simultaneously writing to all channels.

Therefore, when writing to the CCLR bits of a channel for which the setting of the CLRWP bit is 1(disabling writing), the CCLR bit for the given channel is not updated, but the CCLR bits corresponding to channel for which the setting of the CLRWP bit is 0 (enabling writing) are updated. For example, when the setting of the GPT320.GTWP.CLRWP bit is 0 (enabling writing), writing 1 to the GPT321.GTCLR.CCLR0 bit when its current setting is 0 causes the value to be updated, and the GPT320.GTCNT counter is cleared. When the setting of the GPT320.GTWP.CLRWP bit is 1 (disabling writing), writing 1 to the GPT321.GTCLR.CCLR0 bit when its current setting is 0 leaves the bit with the value 0, and the GPT320.GTCNT counter is not cleared.

If you want to protect all bits in the GTCLR register from being updated, set the CLRWP bits of all channels to 1.

CMNWP bit (Common Register Write Disabled)

CMNWP bit enables or disables starting the updating of counter values by writing to the SECSELn bit (n = 0 to 9) corresponding to a channel number in the GTSECSR register or to the GTSECR register.

The bit position of each SECSEL bit in the GTSECSR registers is allocated to the channel with the corresponding number, and the writing to the GTSECSR register for any channel results in writing to the registers of all channels. Writing to the GTSECR register of any channel leads to writing to the registers of all channels. The CMNWP bit for each channel does not control writing but only controls updating of the SECSEL bit and the GTSECR register value for the corresponding channel when simultaneously writing to all channels.

Therefore, when writing to the SECSEL bit and the GTSECR register value of a channel for which the setting of the CMNWP bit is 1(disabling writing), the SECSEL bit and the GTSECR register value for the given channel is not updated, but the SECSEL bit and the GTSECR register value corresponding to channel for which the setting of the CMNWP bit is 0 (enabling writing) are updated.

For example, when the setting of the GPT320.GTWP.CMNWP bit is 0 (enabling writing), writing to the GPT321.GTSECSR.SECSEL0 bit causes the value of the GPT320.GTSECSR.SECSEL0 bit to be updated. In the same way, writing to the GPT321.GTSECR register updates the value of the GPT320.GTSECR register. When the setting of the GPT320.GTWP.CMNWP bit is 1 (disabling writing), writing to the GPT321.GTSECSR.SECSEL0 bit does not cause the value of the GPT320.GTSECSR.SECSEL0 bit to be updated. In the same way, writing to the GPT321.GTSECR register does not update the value of the GPT320.GTSECR register.

If you want to protect all bits in the GTSECSR and GTSECR registers from being updated, set the CMNWP bits of all channels to 1.

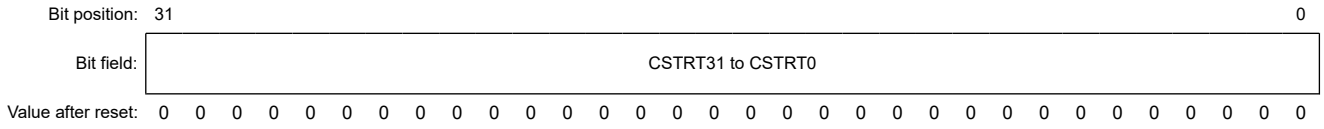
PRKEY[7:0] bit (GTWP Key Code)

This bit controls whether the WP, STRWP, STPWP, CLRWP, and CMNWP bits can be overwritten.

21.2.2 GTSTR : General PWM Timer Software Start Register

Base address: GPT32n = 0x4016_9000 + 0x0100 × n (n = 0 to 9)

Offset address: 0x04



Bit	Symbol	Function	R/W
31:0	CSTRT0 to CSTRT31*1	Channel n GTCNT Count Start (n is the same as the bit position value) 0: GTCNT counter is not started 1: GTCNT counter is started	R/W

Note 1. The bits that can be used vary depending on the product. The n in CSTRTn is the same as the GPT channel number. For this product, n is 0 to 9.

The GTSTR starts the GTCNT counter operation for each channel n, where n = 0 to 9.

The GTSTR bit number represents the channel number. The GTSTR register of each channel is shared by all of the channels. The GTCNT counter starts for the channel associated with the GTSTR bit number where 1 is written. Writing 0 has no effect on the status of GTCNT counter and the value of GTSTR register.

In complementary PWM mode, writing to the CSTRTn bit of master channel is only valid. The bit on slave channels reflects the bit value of master channel.

For the association between module names and channel numbers, see Figure 21.4.

CSTRTn bits (Channel n GTCNT Count Start (n = 0 to 9))

The CSTRTn bits start channel n of the GTCNT counter operation. Writing to the GTSTR.CSTRTn bit (n = 0 to 9) has no effect unless the GTSSR.CSTRT bit is set to 1.

The read data shows the counter status of each channel (GTCR.CST bit). A value of 0 means the counter is stopped and 1 means the counter is running.

21.2.3 GTSTP : General PWM Timer Software Stop Register

Base address: GPT32n = 0x4016_9000 + 0x0100 × n (n = 0 to 9)

Offset address: 0x08

Bit position:	31																												0
Bit field:	CSTOP31 to CSTOP0																												
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
31:0	CSTOP0 to CSTOP31*1	Channel n GTCNT Count Stop (n is the same as the bit position value) 0: GTCNT counter is not stopped 1: GTCNT counter stopped	R/W

Note 1. The bits that can be used vary depending on the product. The n in CSTOPn is the same as the GPT channel number. For this product, n is 0 to 9.

The GTSTP stops the GTCNT counter operation for each channel n, where n = 0 to 9.

The GTSTP bit number represents the channel number. The GTSTP register of each channel is shared by all the channels. The GTCNT counter stops for the channel associated with the GTSTP bit number where 1 is written. Writing 0 has no effect on the status of the GTCNT counter and the value of GTSTP register.

In complementary PWM mode, writing to the CSTOPn bit of master channel is only valid. The bit on slave channels reflects the bit value of master channel.

For the association between module names and channel numbers, see Figure 21.4.

CSTOPn bits (Channel n GTCNT Count Stop (n = 0 to 9))

The CSTOPn bits stop channel n of the GTCNT counter operation. Writing to the GTSTP.CSTOPn bit (n = 0 to 9) has no effect unless the GTPSR.CSTOP bit is set to 1. The read data shows the counter status of each channel (invert of GTCR.CST bit). A value of 0 means the counter is running and 1 means the counter is stopped.

21.2.4 GTCLR : General PWM Timer Software Clear Register

Base address: GPT32n = 0x4016_9000 + 0x0100 × n (n = 0 to 9)

Offset address: 0x0C

Bit position:	31																												0
Bit field:	CCLR31 to CCLR0																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	CCLR0 to CCLR31*1	Channel n GTCNT Count Clear (n : the same as bit position value) 0: GTCNT counter is not cleared 1: GTCNT counter is cleared	W

Note 1. The bits that can be used vary depending on the product. The n of CCLRn is the same as the GPT channel number. For this product, n is 0 to 9.

The GTCLR is a write-only register that clears the GTCNT counter operation for each channel n, where n = 0 to 9.

The GTCLR bit number represents the channel number. The GTCLR register of each channel is shared by all the channels. The GTCNT counter is cleared for the channel associated with the GTCLR bit number where 1 is written. Writing 0 has no effect on the status of GTCNT counter.

In complementary PWM mode, writing to the CCLRn bit of master channel is only valid. The bit on slave channels reflects the bit value of master channel.

For the association between module names and channel numbers, see Figure 21.4.

CCLRn bits (Channel n GTCNT Count Clear (n = 0 to 9))

When the counting direction flag is set for decrement (GTST.TUCF flag = 0) with saw-wave mode selected in the GTCR.MD[2:0] bits, the value of the GTCNT counter becomes that of the corresponding GTPR register in response to writing 1 to the CCLRn bit. The value of the counter becomes 0x0000 0000 with other settings. These bits are read as 0.

21.2.5 GTSSR : General PWM Timer Start Source Select Register

Base address: GPT32n = 0x4016_9000 + 0x0100 × n (n = 0 to 9)

Offset address: 0x10

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CSTRT	—	—	—	—	—	—	—	SSELCH	SSELCG	SSELCF	SSELCE	SSELCD	SSELCC	SSELCB	SSELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SSCBFAH	SSCBFAL	SSCBRAH	SSCBRAL	SSCAFBH	SSCAFBL	SSCARBH	SSCARBL	SSGTRGDF	SSGTRGDR	SSGTRGCF	SSGTRGCR	SSGTRGBF	SSGTRGBR	SSGTRGAF	SSGTRGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SSGTRGAR	GTETRGA Pin Rising Input Source Counter Start Enable 0: Counter start disabled on the rising edge of GTETRGA input 1: Counter start enabled on the rising edge of GTETRGA input	R/W*1
1	SSGTRGAF	GTETRGA Pin Falling Input Source Counter Start Enable 0: Counter start disabled on the falling edge of GTETRGA input 1: Counter start enabled on the falling edge of GTETRGA input	R/W*1
2	SSGTRGBR	GTETRGB Pin Rising Input Source Counter Start Enable 0: Counter start disabled on the rising edge of GTETRGB input 1: Counter start enabled on the rising edge of GTETRGB input	R/W*1
3	SSGTRGBF	GTETRGB Pin Falling Input Source Counter Start Enable 0: Counter start disabled on the falling edge of GTETRGB input 1: Counter start enabled on the falling edge of GTETRGB input	R/W*1
4	SSGTRGCR	GTETRGC Pin Rising Input Source Counter Start Enable 0: Counter start disabled on the rising edge of GTETRGC input 1: Counter start enabled on the rising edge of GTETRGC input	R/W*1
5	SSGTRGCF	GTETRGC Pin Falling Input Source Counter Start Enable 0: Counter start disabled on the falling edge of GTETRGC input 1: Counter start enabled on the falling edge of GTETRGC input	R/W*1
6	SSGTRGDR	GTETRGD Pin Rising Input Source Counter Start Enable 0: Counter start disabled on the rising edge of GTETRGD input 1: Counter start enabled on the rising edge of GTETRGD input	R/W*1
7	SSGTRGDF	GTETRGD Pin Falling Input Source Counter Start Enable 0: Counter start disabled on the falling edge of GTETRGD input 1: Counter start enabled on the falling edge of GTETRGD input	R/W*1
8	SSCARBL	GTIOCnA Pin Rising Input during GTIOCnB Value Low Source Counter Start Enable 0: Counter start disabled on the rising edge of GTIOCnA input when GTIOCnB input is 0 1: Counter start enabled on the rising edge of GTIOCnA input when GTIOCnB input is 0	R/W
9	SSCARBH	GTIOCnA Pin Rising Input during GTIOCnB Value High Source Counter Start Enable 0: Counter start disabled on the rising edge of GTIOCnA input when GTIOCnB input is 1 1: Counter start enabled on the rising edge of GTIOCnA input when GTIOCnB input is 1	R/W

Bit	Symbol	Function	R/W
10	SSCAFBL	GTIOCnA Pin Falling Input during GTIOCnB Value Low Source Counter Start Enable 0: Counter start disabled on the falling edge of GTIOCnA input when GTIOCnB input is 0 1: Counter start enabled on the falling edge of GTIOCnA input when GTIOCnB input is 0	R/W
11	SSCAFBH	GTIOCnA Pin Falling Input during GTIOCnB Value High Source Counter Start Enable 0: Counter start disabled on the falling edge of GTIOCnA input when GTIOCnB input is 1 1: Counter start enabled on the falling edge of GTIOCnA input when GTIOCnB input is 1	R/W
12	SSCBRAL	GTIOCnB Pin Rising Input during GTIOCnA Value Low Source Counter Start Enable 0: Counter start disabled on the rising edge of GTIOCnB input when GTIOCnA input is 0 1: Counter start enabled on the rising edge of GTIOCnB input when GTIOCnA input is 0	R/W
13	SSCBRAH	GTIOCnB Pin Rising Input during GTIOCnA Value High Source Counter Start Enable 0: Counter start disabled on the rising edge of GTIOCnB input when GTIOCnA input is 1 1: Counter start enabled on the rising edge of GTIOCnB input when GTIOCnA input is 1	R/W
14	SSCBFAL	GTIOCnB Pin Falling Input during GTIOCnA Value Low Source Counter Start Enable 0: Counter start disabled on the falling edge of GTIOCnB input when GTIOCnA input is 0 1: Counter start enabled on the falling edge of GTIOCnB input when GTIOCnA input is 0	R/W
15	SSCBFAH	GTIOCnB Pin Falling Input during GTIOCnA Value High Source Counter Start Enable 0: Counter start disabled on the falling edge of GTIOCnB input when GTIOCnA input is 1 1: Counter start enabled on the falling edge of GTIOCnB input when GTIOCnA input is 1	R/W
16	SSELCA	ELC_GPTA Event Source Counter Start Enable 0: Counter start disabled at the ELC_GPTA input 1: Counter start enabled at the ELC_GPTA input	R/W*1
17	SSELCB	ELC_GPTB Event Source Counter Start Enable 0: Counter start disabled at the ELC_GPTB input 1: Counter start enabled at the ELC_GPTB input	R/W*1
18	SSELCC	ELC_GPTC Event Source Counter Start Enable 0: Counter start disabled at the ELC_GPTC input 1: Counter start enabled at the ELC_GPTC input	R/W*1
19	SSELCD	ELC_GPTD Event Source Counter Start Enable 0: Counter start disabled at the ELC_GPTD input 1: Counter start enabled at the ELC_GPTD input	R/W*1
20	SSELCE	ELC_GPTE Event Source Counter Start Enable 0: Counter start disabled at the ELC_GPTE input 1: Counter start enabled at the ELC_GPTE input	R/W*1
21	SSELCF	ELC_GPTF Event Source Counter Start Enable 0: Counter start disabled at the ELC_GPTF input 1: Counter start enabled at the ELC_GPTF input	R/W*1
22	SSELCG	ELC_GPTG Event Source Counter Start Enable 0: Counter start disabled at the ELC_GPTG input 1: Counter start enabled at the ELC_GPTG input	R/W*1
23	SSELCH	ELC_GPTH Event Source Counter Start Enable 0: Counter start disabled at the ELC_GPTH input 1: Counter start enabled at the ELC_GPTH input	R/W*1
30:24	—	These bits are read as 0. The write value should be 0.	R/W
31	CSTRT	Software Source Counter Start Enable 0: Counter start disabled by the GTSTR register 1: Counter start enabled by the GTSTR register	R/W*1

Note 1. When in complementary PWM mode, no matter which register of master channel/slave channel 1/slave channel 2 is written, three channels are written at the same time.

The GTSSR sets the source to start the GTCNT counter.

Input from GTETRGn (n = A to D) pins are input to the GPT through the POEG. Set the polarity of these signals with the POEG.

SSGTRGAR bit (GTETRGA Pin Rising Input Source Counter Start Enable)

The SSGTRGAR bit enables or disables the GTCNT counter start on the rising edge of the GTETRGA pin input.

SSGTRGAF bit (GTETRGA Pin Falling Input Source Counter Start Enable)

The SSGTRGAF bit enables or disables the GTCNT counter start on the falling edge of the GTETRGA pin input.

SSGTRGBR bit (GTETRGB Pin Rising Input Source Counter Start Enable)

The SSGTRGBR bit enables or disables the GTCNT counter start on the rising edge of the GTETRGB pin input.

SSGTRGBF bit (GTETRGB Pin Falling Input Source Counter Start Enable)

The SSGTRGBF bit enables or disables the GTCNT counter start on the falling edge of the GTETRGB pin input.

SSGTRGCR bit (GTETRGC Pin Rising Input Source Counter Start Enable)

The SSGTRGCR bit enables or disables the GTCNT counter start on the rising edge of the GTETRGC pin input.

SSGTRGCF bit (GTETRGC Pin Falling Input Source Counter Start Enable)

The SSGTRGCF bit enables or disables the GTCNT counter start on the falling edge of the GTETRGC pin input.

SSGTRGDR bit (GTETRGD Pin Rising Input Source Counter Start Enable)

The SSGTRGDR bit enables or disables the GTCNT counter start on the rising edge of the GTETRGD pin input.

SSGTRGDF bit (GTETRGD Pin Falling Input Source Counter Start Enable)

The SSGTRGDF bit enables or disables the GTCNT counter start on the falling edge of the GTETRGD pin input.

SSCARBL bit (GTIOCnA Pin Rising Input during GTIOCnB Value Low Source Counter Start Enable)

The SSCARBL bit enables or disables the GTCNT counter start on the rising edge of the GTIOCnA pin input, when GTIOCnB input is 0.

In complementary PWM mode, this setting is invalid.

SSCARBH bit (GTIOCnA Pin Rising Input during GTIOCnB Value High Source Counter Start Enable)

The SSCARBH bit enables or disables the GTCNT counter start on the rising edge of the GTIOCnA pin input, when GTIOCnB input is 1.

In complementary PWM mode, this setting is invalid.

SSCAFBL bit (GTIOCnA Pin Falling Input during GTIOCnB Value Low Source Counter Start Enable)

The SSCAFBL bit enables or disables the GTCNT counter start on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 0.

In complementary PWM mode, this setting is invalid.

SSCAFBH bit (GTIOCnA Pin Falling Input during GTIOCnB Value High Source Counter Start Enable)

The SSCAFBH bit enables or disables the GTCNT counter start on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 1.

In complementary PWM mode, this setting is invalid.

SSCBRAL bit (GTIOCnB Pin Rising Input during GTIOCnA Value Low Source Counter Start Enable)

The SSCBRAL bit enables or disables the GTCNT counter start on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 0.

In complementary PWM mode, this setting is invalid.

SSCBRAH bit (GTIOCnB Pin Rising Input during GTIOCnA Value High Source Counter Start Enable)

The SSCBRAH bit enables or disables the GTCNT counter start on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 1.

In complementary PWM mode, this setting is invalid.

SSCBFAL bit (GTIOCnB Pin Falling Input during GTIOCnA Value Low Source Counter Start Enable)

The SSCBFAL bit enables or disables the GTCNT counter start on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 0.

In complementary PWM mode, this setting is invalid.

SSCBFAH bit (GTIOCnB Pin Falling Input during GTIOCnA Value High Source Counter Start Enable)

The SSCBFAH bit enables or disables the GTCNT counter start on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 1.

In complementary PWM mode, this setting is invalid.

SSELCm bit (ELC_GPTm Event Source Counter Start Enable) (m = A to H)

The SSELCm bit enables or disables the GTCNT counter start at the ELC_GPTm event input.

CSTRT bit (Software Source Counter Start Enable)

The CSTRT bit enables or disables the GTCNT counter start by GTSTR register.

21.2.6 GTPSR : General PWM Timer Stop Source Select Register

Base address: GPT32n = 0x4016_9000 + 0x0100 × n (n = 0 to 9)

Offset address: 0x14

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CSTOP	—	—	—	—	—	—	—	PSELCH	PSELCG	PSELCF	PSELCE	PSELCD	PSELCC	PSELCB	PSELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PSCBFAH	PSCBFAL	PSCBRAH	PSCBRAL	PSCAFBH	PSCAFBL	PSCARBH	PSCARBL	PSGTRGDF	PSGTRGDR	PSGTRGCF	PSGTRGCR	PSGTRGBF	PSGTRGBR	PSGTRGAF	PSGTRGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PSGTRGAR	GTETRGA Pin Rising Input Source Counter Stop Enable 0: Counter stop disabled on the rising edge of GTETRGA input 1: Counter stop enabled on the rising edge of GTETRGA input	R/W*1
1	PSGTRGAF	GTETRGA Pin Falling Input Source Counter Stop Enable 0: Counter stop disabled on the falling edge of GTETRGA input 1: Counter stop enabled on the falling edge of GTETRGA input	R/W*1
2	PSGTRGBR	GTETRGB Pin Rising Input Source Counter Stop Enable 0: Counter stop disabled on the rising edge of GTETRGB input 1: Counter stop enabled on the rising edge of GTETRGB input	R/W*1
3	PSGTRGBF	GTETRGB Pin Falling Input Source Counter Stop Enable 0: Counter stop disabled on the falling edge of GTETRGB input 1: Counter stop enabled on the falling edge of GTETRGB input	R/W*1
4	PSGTRGCR	GTETRGC Pin Rising Input Source Counter Stop Enable 0: Counter stop disabled on the rising edge of GTETRGC input 1: Counter stop enabled on the rising edge of GTETRGC input	R/W*1
5	PSGTRGCF	GTETRGC Pin Falling Input Source Counter Stop Enable 0: Counter stop disabled on the falling edge of GTETRGC input 1: Counter stop enabled on the falling edge of GTETRGC input	R/W*1

Bit	Symbol	Function	R/W
6	PSGTRGDR	GTETRGD Pin Rising Input Source Counter Stop Enable 0: Counter stop disabled on the rising edge of GTETRGD input 1: Counter stop enabled on the rising edge of GTETRGD input	R/W*1
7	PSGTRGDF	GTETRGD Pin Falling Input Source Counter Stop Enable 0: Counter stop disabled on the falling edge of GTETRGD input 1: Counter stop enabled on the falling edge of GTETRGD input	R/W*1
8	PSCARBL	GTIOCnA Pin Rising Input during GTIOCnB Value Low Source Counter Stop Enable 0: Counter stop disabled on the rising edge of GTIOCnA input when GTIOCnB input is 0 1: Counter stop enabled on the rising edge of GTIOCnA input when GTIOCnB input is 0	R/W
9	PSCARBH	GTIOCnA Pin Rising Input during GTIOCnB Value High Source Counter Stop Enable 0: Counter stop disabled on the rising edge of GTIOCnA input when GTIOCnB input is 1 1: Counter stop enabled on the rising edge of GTIOCnA input when GTIOCnB input is 1	R/W
10	PSCAFBL	GTIOCnA Pin Falling Input during GTIOCnB Value Low Source Counter Stop Enable 0: Counter stop disabled on the falling edge of GTIOCnA input when GTIOCnB input is 0 1: Counter stop enabled on the falling edge of GTIOCnA input when GTIOCnB input is 0	R/W
11	PSCAFBH	GTIOCnA Pin Falling Input during GTIOCnB Value High Source Counter Stop Enable 0: Counter stop disabled on the falling edge of GTIOCnA input when GTIOCnB input is 1 1: Counter stop enabled on the falling edge of GTIOCnA input when GTIOCnB input is 1	R/W
12	PSCBRAL	GTIOCnB Pin Rising Input during GTIOCnA Value Low Source Counter Stop Enable 0: Counter stop disabled on the rising edge of GTIOCnB input when GTIOCnA input is 0 1: Counter stop enabled on the rising edge of GTIOCnB input when GTIOCnA input is 0	R/W
13	PSCBRAH	GTIOCnB Pin Rising Input during GTIOCnA Value High Source Counter Stop Enable 0: Counter stop disabled on the rising edge of GTIOCnB input when GTIOCnA input is 1 1: Counter stop enabled on the rising edge of GTIOCnB input when GTIOCnA input is 1	R/W
14	PSCBFAL	GTIOCnB Pin Falling Input during GTIOCnA Value Low Source Counter Stop Enable 0: Counter stop disabled on the falling edge of GTIOCnB input when GTIOCnA input is 0 1: Counter stop enabled on the falling edge of GTIOCnB input when GTIOCnA input is 0	R/W
15	PSCBFAH	GTIOCnB Pin Falling Input during GTIOCnA Value High Source Counter Stop Enable 0: Counter stop disabled on the falling edge of GTIOCnB input when GTIOCnA input is 1 1: Counter stop enabled on the falling edge of GTIOCnB input when GTIOCnA input is 1	R/W
16	PSELCA	ELC_GPTA Event Source Counter Stop Enable 0: Counter stop disabled at the ELC_GPTA input 1: Counter stop enabled at the ELC_GPTA input	R/W*1
17	PSELCB	ELC_GPTB Event Source Counter Stop Enable 0: Counter stop disabled at the ELC_GPTB input 1: Counter stop enabled at the ELC_GPTB input	R/W*1
18	PSELCC	ELC_GPTC Event Source Counter Stop Enable 0: Counter stop disabled at the ELC_GPTC input 1: Counter stop enabled at the ELC_GPTC input	R/W*1
19	PSELCD	ELC_GPTD Event Source Counter Stop Enable 0: Counter stop disabled at the ELC_GPTD input 1: Counter stop enabled at the ELC_GPTD input	R/W*1

Bit	Symbol	Function	R/W
20	PSELCE	ELC_GPTE Event Source Counter Stop Enable 0: Counter stop disabled at the ELC_GPTE input 1: Counter stop enabled at the ELC_GPTE input	R/W*1
21	PSELCF	ELC_GPTF Event Source Counter Stop Enable 0: Counter stop disabled at the ELC_GPTF input 1: Counter stop enabled at the ELC_GPTF input	R/W*1
22	PSELCG	ELC_GPTG Event Source Counter Stop Enable 0: Counter stop disabled at the ELC_GPTG input 1: Counter stop enabled at the ELC_GPTG input	R/W*1
23	PSELCH	ELC_GPTH Event Source Counter Stop Enable 0: Counter stop disabled at the ELC_GPTH input 1: Counter stop enabled at the ELC_GPTH input	R/W*1
30:24	—	These bits are read as 0. The write value should be 0.	R/W
31	CSTOP	Software Source Counter Stop Enable 0: Counter stop disabled by the GTSTP register 1: Counter stop enabled by the GTSTP register	R/W*1

Note 1. When in complementary PWM mode, no matter which register of master channel/slave channel 1/slave channel 2 is written, three channels are written at the same time.

The GTPSR sets the source to stop the GTCNT counter.

Inputs from GTETRGn (n = A to D) pins are input to the GPT through the POEG. Set the polarity of these signals with the POEG.

PSGTRGAR bit (GTETRGA Pin Rising Input Source Counter Stop Enable)

The PSGTRGAR bit enables or disables the GTCNT counter stop on the rising edge of the GTETRGA pin input.

PSGTRGAF bit (GTETRGA Pin Falling Input Source Counter Stop Enable)

The PSGTRGAF bit enables or disables the GTCNT counter stop on the falling edge of the GTETRGA pin input.

PSGTRGBR bit (GTETRGB Pin Rising Input Source Counter Stop Enable)

PSGTRGBR bit enables or disables the GTCNT counter stop on the rising edge of the GTETRGB pin input.

PSGTRGBF bit (GTETRGB Pin Falling Input Source Counter Stop Enable)

The PSGTRGBF bit enables or disables the GTCNT counter stop on the falling edge of the GTETRGB pin input.

PSGTRGCR bit (GTETRGC Pin Rising Input Source Counter Stop Enable)

PSGTRGCR bit enables or disables the GTCNT counter stop on the rising edge of the GTETRGC pin input.

PSGTRGCF bit (GTETRGC Pin Falling Input Source Counter Stop Enable)

The PSGTRGCF bit enables or disables the GTCNT counter stop on the falling edge of the GTETRGC pin input.

PSGTRGDR bit (GTETRGD Pin Rising Input Source Counter Stop Enable)

PSGTRGDR bit enables or disables the GTCNT counter stop on the rising edge of the GTETRGD pin input.

PSGTRGDF bit (GTETRGD Pin Falling Input Source Counter Stop Enable)

The PSGTRGDF bit enables or disables the GTCNT counter stop on the falling edge of the GTETRGD pin input.

PSCARBL bit (GTIOCnA Pin Rising Input during GTIOCnB Value Low Source Counter Stop Enable)

The PSCARBL bit enables or disables the GTCNT counter stop on the rising edge of the GTIOCnA pin input, when GTIOCnB input is 0.

In complementary PWM mode, this setting is invalid.

PSCARBH bit (GTIOCnA Pin Rising Input during GTIOCnB Value High Source Counter Stop Enable)

The PSCARBH bit enables or disables the GTCNT counter stop on the rising edge of the GTIOCnA pin input, when GTIOCnB input is 1.

In complementary PWM mode, this setting is invalid.

PSCAFBL bit (GTIOCnA Pin Falling Input during GTIOCnB Value Low Source Counter Stop Enable)

The PSCAFBL bit enables or disables the GTCNT counter stop on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 0.

In complementary PWM mode, this setting is invalid.

PSCAFBH bit (GTIOCnA Pin Falling Input during GTIOCnB Value High Source Counter Stop Enable)

The PSCAFBH bit enables or disables the GTCNT counter stop on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 1.

In complementary PWM mode, this setting is invalid.

PSCBRAL bit (GTIOCnB Pin Rising Input during GTIOCnA Value Low Source Counter Stop Enable)

The PSCBRAL bit enables or disables the GTCNT counter stop on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 0.

In complementary PWM mode, this setting is invalid.

PSCBRAH bit (GTIOCnB Pin Rising Input during GTIOCnA Value High Source Counter Stop Enable)

The PSCBRAH bit enables or disables the GTCNT counter stop on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 1.

In complementary PWM mode, this setting is invalid.

PSCBFAL bit (GTIOCnB Pin Falling Input during GTIOCnA Value Low Source Counter Stop Enable)

The PSCBFAL bit enables or disables the GTCNT counter stop on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 0.

In complementary PWM mode, this setting is invalid.

PSCBFAH bit (GTIOCnB Pin Falling Input during GTIOCnA Value High Source Counter Stop Enable)

The PSCBFAH bit enables or disables the GTCNT counter stop on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 1.

In complementary PWM mode, this setting is invalid.

PSELCm bit (ELCm Event Source Counter Stop Enable) (m = A to H)

The PSELCm bit enables or disables the GTCNT counter stop at the ELC_GPTm event input.

CSTOP bit (Software Source Counter Stop Enable)

The CSTOP bit enables or disables the GTCNT counter stop by the GTSTP register.

21.2.7 GTCSR : General PWM Timer Clear Source Select Register

Base address: GPT32n = 0x4016_9000 + 0x0100 × n (n = 0 to 9)

Offset address: 0x18

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CCLR	—	—	—	CP1CCE	CSCMSC[2:0]			CSELCH	CSELCG	CSELCF	CSELCE	CSELCD	CSELCC	CSELCB	CSELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CSCBFAH	CSCBFAL	CSCBRAH	CSCBRAL	CSCAFBH	CSCAFBL	CSCARBH	CSCARBL	CSGTRGDF	CSGTRGDR	CSGTRGCF	CSGTRGCR	CSGTRGBF	CSGTRGBR	CSGTRGAF	CSGTRGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CSGTRGAR	GTETRGA Pin Rising Input Source Counter Clear Enable 0: Counter clear disabled on the rising edge of GTETRGA input 1: Counter clear enabled on the rising edge of GTETRGA input	R/W*1
1	CSGTRGAF	GTETRGA Pin Falling Input Source Counter Clear Enable 0: Counter clear disabled on the falling edge of GTETRGA input 1: Counter clear enabled on the falling edge of GTETRGA input	R/W*1
2	CSGTRGBR	GTETRGB Pin Rising Input Source Counter Clear Enable 0: Disable counter clear on the rising edge of GTETRGB input 1: Enable counter clear on the rising edge of GTETRGB input	R/W*1
3	CSGTRGBF	GTETRGB Pin Falling Input Source Counter Clear Enable 0: Counter clear disabled on the falling edge of GTETRGB input 1: Counter clear enabled on the falling edge of GTETRGB input	R/W*1
4	CSGTRGCR	GTETRGC Pin Rising Input Source Counter Clear Enable 0: Disable counter clear on the rising edge of GTETRGC input 1: Enable counter clear on the rising edge of GTETRGC input	R/W*1
5	CSGTRGCF	GTETRGC Pin Falling Input Source Counter Clear Enable 0: Counter clear disabled on the falling edge of GTETRGC input 1: Counter clear enabled on the falling edge of GTETRGC input	R/W*1
6	CSGTRGDR	GTETRGD Pin Rising Input Source Counter Clear Enable 0: Disable counter clear on the rising edge of GTETRGD input 1: Enable counter clear on the rising edge of GTETRGD input	R/W*1
7	CSGTRGDF	GTETRGD Pin Falling Input Source Counter Clear Enable 0: Counter clear disabled on the falling edge of GTETRGD input 1: Counter clear enabled on the falling edge of GTETRGD input	R/W*1
8	CSCARBL	GTIOCnA Pin Rising Input during GTIOCnB Value Low Source Counter Clear Enable 0: Counter clear disabled on the rising edge of GTIOCnA input when GTIOCnB input is 0 1: Counter clear enabled on the rising edge of GTIOCnA input when GTIOCnB input is 0	R/W
9	CSCARBH	GTIOCnA Pin Rising Input during GTIOCnB Value High Source Counter Clear Enable 0: Counter clear disabled on the rising edge of GTIOCnA input when GTIOCnB input is 1 1: Counter clear enabled on the rising edge of GTIOCnA input when GTIOCnB input is 1	R/W
10	CSCAFBL	GTIOCnA Pin Falling Input during GTIOCnB Value Low Source Counter Clear Enable 0: Counter clear disabled on the falling edge of GTIOCnA input when GTIOCnB input is 0 1: Counter clear enabled on the falling edge of GTIOCnA input when GTIOCnB input is 0	R/W
11	CSCAFBH	GTIOCnA Pin Falling Input during GTIOCnB Value High Source Counter Clear Enable 0: Counter clear disabled on the falling edge of GTIOCnA input when GTIOCnB input is 1 1: Counter clear enabled on the falling edge of GTIOCnA input when GTIOCnB input is 1	R/W
12	CSCBRAL	GTIOCnB Pin Rising Input during GTIOCnA Value Low Source Counter Clear Enable 0: Counter clear disabled on the rising edge of GTIOCnB input when GTIOCnA input is 0 1: Counter clear enabled on the rising edge of GTIOCnB input when GTIOCnA input is 0	R/W
13	CSCBRAH	GTIOCnB Pin Rising Input during GTIOCnA Value High Source Counter Clear Enable 0: Counter clear disabled on the rising edge of GTIOCnB input when GTIOCnA input is 1 1: Counter clear enabled on the rising edge of GTIOCnB input when GTIOCnA input is 1	R/W
14	CSCBFAL	GTIOCnB Pin Falling Input during GTIOCnA Value Low Source Counter Clear Enable 0: Counter clear disabled on the falling edge of GTIOCnB input when GTIOCnA input is 0 1: Counter clear enabled on the falling edge of GTIOCnB input when GTIOCnA input is 0	R/W

Bit	Symbol	Function	R/W
15	CSCBFAH	GTIOCnB Pin Falling Input during GTIOCnA Value High Source Counter Clear Enable 0: Counter clear disabled on the falling edge of GTIOCnB input when GTIOCnA input is 1 1: Counter clear enabled on the falling edge of GTIOCnB input when GTIOCnA input is 1	R/W
16	CSELCA	ELC_GPTA Event Source Counter Clear Enable 0: Counter clear disabled at the ELC_GPTA input 1: Counter clear enabled at the ELC_GPTA input	R/W*1
17	CSELCB	ELC_GPTB Event Source Counter Clear Enable 0: Counter clear disabled at the ELC_GPTB input 1: Counter clear enabled at the ELC_GPTB input	R/W*1
18	CSELCC	ELC_GPTC Event Source Counter Clear Enable 0: Counter clear disabled at the ELC_GPTC input 1: Counter clear enabled at the ELC_GPTC input	R/W*1
19	CSELCD	ELC_GPTD Event Source Counter Clear Enable 0: Counter clear disabled at the ELC_GPTD input 1: Counter clear enabled at the ELC_GPTD input	R/W*1
20	CSELCE	ELC_GPTE Event Source Counter Clear Enable 0: Counter clear disabled at the ELC_GPTE input 1: Counter clear enabled at the ELC_GPTE input	R/W*1
21	CSELCF	ELC_GPTF Event Source Counter Clear Enable 0: Counter clear disabled at the ELC_GPTF input 1: Counter clear enabled at the ELC_GPTF input	R/W*1
22	CSELCG	ELC_GPTG Event Source Counter Clear Enable 0: Counter clear disabled at the ELC_GPTG input 1: Counter clear enabled at the ELC_GPTG input	R/W*1
23	CSELCH	ELC_GPTH Event Source Counter Clear Enable 0: Counter clear disabled at the ELC_GPTH input 1: Counter clear enabled at the ELC_GPTH input	R/W*1
26:24	CSCMSC[2:0]	Compare Match/Input Capture/Synchronous counter clearing Source Counter Clear Enable 0 0 0: Counter clear disabled by Compare match/ Input capture/ Synchronous counter clearing group 0 0 1: Counter clear enabled at the GTCCRA register compare match/ Input capture 0 1 0: Counter clear enabled at the GTCCRB register compare match/ Input capture 0 1 1: Counter clear enabled at the GTCCRC register compare match 1 0 0: Counter clear enabled at the GTCCRD register compare match 1 0 1: Counter clear enabled at the GTCCRE register compare match 1 1 0: Counter clear enabled at the GTCCRF register compare match 1 1 1: Counter clear enabled at the synchronous counter clearing group	R/W
27	CP1CCE	Complementary PWM mode1 Crest Source Counter Clear Enable*2 0: Counter clear disabled at the crest of complementary PWM mode1 1: Counter clear enabled at the crest of complementary PWM mode1	R/W*1
30:28	—	These bits are read as 0. The write value should be 0.	R/W
31	CCLR	Software Source Counter Clear Enable 0: Counter clear disabled by the GTCLR register 1: Counter clear enabled by the GTCLR register	R/W*1

Note 1. When in complementary PWM mode, no matter which register of master channel/slave channel 1/slave channel 2 is written, three channels are written at the same time.
Note 2. This bit is only available in GPT324 to GPT329.
In GPT320 to GPT323, this bit is read as 0. The write value should be 0.

The GTCSR sets the source to clear the GTCNT counter.

Counter clearing can be executed whether the counter is running (GTCR.CST=1) or stopped (GTCR.CST=0).

Inputs from GTETRGn (n = A to D) pins are input to the GPT through the POEG. Set the polarity of these signals with the POEG.

CSGTRGAR bit (GTETRGA Pin Rising Input Source Counter Clear Enable)

The CSGTRGAR bit enables or disables the GTCNT counter clear on the rising edge of the GTETRGA pin input.

CSGTRGAF bit (GTETRGA Pin Falling Input Source Counter Clear Enable)

The CSGTRGAF bit enables or disables the GTCNT counter clear on the falling edge of the GTETRGA pin input.

CSGTRGBR bit (GTETRGB Pin Rising Input Source Counter Clear Enable)

The CSGTRGBR bit enables or disables the GTCNT counter clear on the rising edge of the GTETRGB pin input.

CSGTRGBF bit (GTETRGB Pin Falling Input Source Counter Clear Enable)

The CSGTRGBF bit enables or disables the GTCNT counter clear on the falling edge of the GTETRGB pin input.

CSGTRGCR bit (GTETRGC Pin Rising Input Source Counter Clear Enable)

The CSGTRGCR bit enables or disables the GTCNT counter clear on the rising edge of the GTETRGC pin input.

CSGTRGCF bit (GTETRGC Pin Falling Input Source Counter Clear Enable)

The CSGTRGCF bit enables or disables the GTCNT counter clear on the falling edge of the GTETRGC pin input.

CSGTRGDR bit (GTETRGD Pin Rising Input Source Counter Clear Enable)

The CSGTRGDR bit enables or disables the GTCNT counter clear on the rising edge of the GTETRGD pin input.

CSGTRGDF bit (GTETRGD Pin Falling Input Source Counter Clear Enable)

The CSGTRGDF bit enables or disables the GTCNT counter clear on the falling edge of the GTETRGD pin input.

CSCARBL bit (GTIOCnA Pin Rising Input during GTIOCnB Value Low Source Counter Clear Enable)

The CSCARBL bit enables or disables the GTCNT counter clear on the rising edge of the GTIOCnA pin input, when GTIOCnB input is 0.

In complementary PWM mode, this setting is invalid.

CSCARBH bit (GTIOCnA Pin Rising Input during GTIOCnB Value High Source Counter Clear Enable)

The CSCARBH bit enables or disables the GTCNT counter clear on the rising edge of the GTIOCnA pin input, when GTIOCnB input is 1.

In complementary PWM mode, this setting is invalid.

CSCAFBL bit (GTIOCnA Pin Falling Input during GTIOCnB Value Low Source Counter Clear Enable)

The CSCAFBL bit enables or disables the GTCNT counter clear on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 0.

In complementary PWM mode, this setting is invalid.

CSCAFBH bit (GTIOCnA Pin Falling Input during GTIOCnB Value High Source Counter Clear Enable)

The CSCAFBH bit enables or disables the GTCNT counter clear on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 1.

In complementary PWM mode, this setting is invalid.

CSCBRAL bit (GTIOCnB Pin Rising Input during GTIOCnA Value Low Source Counter Clear Enable)

The CSCBRAL bit enables or disables the GTCNT counter clear on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 0.

In complementary PWM mode, this setting is invalid.

CSCBRAH bit (GTIOCnB Pin Rising Input during GTIOCnA Value High Source Counter Clear Enable)

The CSCBRAH bit enables or disables the GTCNT counter clear on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 1.

In complementary PWM mode, this setting is invalid.

CSCBFAL bit (GTIOCnB Pin Falling Input during GTIOCnA Value Low Source Counter Clear Enable)

The CSCBFAL bit enables or disables the GTCNT counter clear on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 0.

In complementary PWM mode, this setting is invalid.

CSCBFAH bit (GTIOCnB Pin Falling Input during GTIOCnA Value High Source Counter Clear Enable)

The CSCBFAH bit enables or disables the GTCNT counter clear on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 1.

In complementary PWM mode, this setting is invalid.

CSELCm bit (ELCm Event Source Counter Clear Enable) (m = A to H)

The CSELCm bit enables or disables the GTCNT counter clear at the ELC_GPTm event input.

CSCMSC[2:0] bit (Compare Match/Input Capture/Synchronous counter clearing Source Counter Clear Enable)

Select enable or disable for the counter clear of the GTCNT counter by compare match/input capture/synchronous counter clearing group.

Since the compare match by the register that is performing the buffer operation (including the wave mode specific case) does not occur, the counter clear enable setting that makes the target register of the buffer operation the compare match factor is invalid.

In complementary PWM mode, the counter clear enable setting for compare match of the GTCCRB register, GTCCRE register, and GTCCRF register is invalid even when the buffer operation is not performed.

CP1CCE bit (Complementary PWM mode1 Crest Source Counter Clear Enable)

Select enable or disable for the counter clear at the crest of complementary PWM mode 1.

To enable this bit, do not set 1 to the PSYE bit of the GTIOR register.

It is valid only for the master channel in complementary PWM mode. The master channel setting also clears the GTCNT counter of the slave channel in complementary PWM mode.

CCLR bit (Software Source Counter Clear Enable)

The CCLR bit enables or disables the GTCNT counter clear by the GTCLR register.

21.2.8 GTUPSR : General PWM Timer Up Count Source Select Register

Base address: GPT32n = 0x4016_9000 + 0x0100 × n (n = 0 to 3)

Offset address: 0x1C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	USILVL[3:0]				USELCH	USELCG	USELCF	USELCE	USELCD	USELCC	USELCB	USELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	USCBFAH	USCBFAL	USCBRAH	USCBRAL	USCAFBH	USCAFBL	USCARBH	USCARBL	USGTRGDF	USGTRGDR	USGTRGCF	USGTRGCR	USGTRGBF	USGTRGBR	USGTRGAF	USGTRGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	USGTRGAR	GTETRGA Pin Rising Input Source Counter Count Up Enable 0: Counter count up disabled on the rising edge of GTETRGA input 1: Counter count up enabled on the rising edge of GTETRGA input	R/W
1	USGTRGAF	GTETRGA Pin Falling Input Source Counter Count Up Enable 0: Counter count up disabled on the falling edge of GTETRGA input 1: Counter count up enabled on the falling edge of GTETRGA input	R/W
2	USGTRGBR	GTETRGB Pin Rising Input Source Counter Count Up Enable 0: Counter count up disabled on the rising edge of GTETRGB input 1: Counter count up enabled on the rising edge of GTETRGB input	R/W

Bit	Symbol	Function	R/W
3	USGTRGBF	GTETRGB Pin Falling Input Source Counter Count Up Enable 0: Counter count up disabled on the falling edge of GTETRGB input 1: Counter count up enabled on the falling edge of GTETRGB input	R/W
4	USGTRGCR	GTETRGC Pin Rising Input Source Counter Count Up Enable 0: Counter count up disabled on the rising edge of GTETRGC input 1: Counter count up enabled on the rising edge of GTETRGC input	R/W
5	USGTRGCF	GTETRGC Pin Falling Input Source Counter Count Up Enable 0: Counter count up disabled on the falling edge of GTETRGC input 1: Counter count up enabled on the falling edge of GTETRGC input	R/W
6	USGTRGDR	GTETRGD Pin Rising Input Source Counter Count Up Enable 0: Counter count up disabled on the rising edge of GTETRGD input 1: Counter count up enabled on the rising edge of GTETRGD input	R/W
7	USGTRGDF	GTETRGD Pin Falling Input Source Counter Count Up Enable 0: Counter count up disabled on the falling edge of GTETRGD input 1: Counter count up enabled on the falling edge of GTETRGD input	R/W
8	USCARBL	GTIOCnA Pin Rising Input during GTIOCnB Value Low Source Counter Count Up Enable 0: Counter count up disabled on the rising edge of GTIOCnA input when GTIOCnB input is 0 1: Counter count up enabled on the rising edge of GTIOCnA input when GTIOCnB input is 0	R/W
9	USCARBH	GTIOCnA Pin Rising Input during GTIOCnB Value High Source Counter Count Up Enable 0: Counter count up disabled on the rising edge of GTIOCnA input when GTIOCnB input is 1 1: Counter count up enabled on the rising edge of GTIOCnA input when GTIOCnB input is 1	R/W
10	USCAFBL	GTIOCnA Pin Falling Input during GTIOCnB Value Low Source Counter Count Up Enable 0: Counter count up disabled on the falling edge of GTIOCnA input when GTIOCnB input is 0 1: Counter count up enabled on the falling edge of GTIOCnA input when GTIOCnB input is 0	R/W
11	USCAFBH	GTIOCnA Pin Falling Input during GTIOCnB Value High Source Counter Count Up Enable 0: Counter count up disabled on the falling edge of GTIOCnA input when GTIOCnB input is 1 1: Counter count up enabled on the falling edge of GTIOCnA input when GTIOCnB input is 1	R/W
12	USCBRAL	GTIOCnB Pin Rising Input during GTIOCnA Value Low Source Counter Count Up Enable 0: Counter count up disabled on the rising edge of GTIOCnB input when GTIOCnA input is 0 1: Counter count up enabled on the rising edge of GTIOCnB input when GTIOCnA input is 0	R/W
13	USCBRAH	GTIOCnB Pin Rising Input during GTIOCnA Value High Source Counter Count Up Enable 0: Counter count up disabled on the rising edge of GTIOCnB input when GTIOCnA input is 1 1: Counter count up enabled on the rising edge of GTIOCnB input when GTIOCnA input is 1	R/W
14	USCBFAL	GTIOCnB Pin Falling Input during GTIOCnA Value Low Source Counter Count Up Enable 0: Counter count up disabled on the falling edge of GTIOCnB input when GTIOCnA input is 0 1: Counter count up enabled on the falling edge of GTIOCnB input when GTIOCnA input is 0	R/W
15	USCBFAH	GTIOCnB Pin Falling Input during GTIOCnA Value High Source Counter Count Up Enable 0: Counter count up disabled on the falling edge of GTIOCnB input when GTIOCnA input is 1 1: Counter count up enabled on the falling edge of GTIOCnB input when GTIOCnA input is 1	R/W
16	USELCA	ELC_GPTA Event Source Counter Count Up Enable 0: Counter count up disabled at the ELC_GPTA input 1: Counter count up enabled at the ELC_GPTA input	R/W

Bit	Symbol	Function	R/W
17	USELCB	ELC_GPTB Event Source Counter Count Up Enable 0: Counter count up disabled at the ELC_GPTB input 1: Counter count up enabled at the ELC_GPTB input	R/W
18	USELCC	ELC_GPTC Event Source Counter Count Up Enable 0: Counter count up disabled at the ELC_GPTC input 1: Counter count up enabled at the ELC_GPTC input	R/W
19	USELCD	ELC_GPTD Event Source Counter Count Up Enable 0: Counter count up disabled at the ELC_GPTD input 1: Counter count up enabled at the ELC_GPTD input	R/W
20	USELCE	ELC_GPTE Event Source Counter Count Up Enable 0: Counter count up disabled at the ELC_GPTE input 1: Counter count up enabled at the ELC_GPTE input	R/W
21	USELCF	ELC_GPTF Event Source Counter Count Up Enable 0: Counter count up disabled at the ELC_GPTF input 1: Counter count up enabled at the ELC_GPTF input	R/W
22	USELCG	ELC_GPTG Event Source Counter Count Up Enable 0: Counter count up disabled at the ELC_GPTG input 1: Counter count up enabled at the ELC_GPTG input	R/W
23	USELCH	ELC_GPTH Event Source Counter Count Up Enable 0: Counter count up disabled at the ELC_GPTH input 1: Counter count up enabled at the ELC_GPTH input	R/W
27:24	USILVL[3:0]	External Input Level Source Count-Up Enable 0 0 0 0: Disables count-up by external input level 0 0 0 1: Setting prohibited 0 0 1 0: Enables count-up by GTIOCnA pin input level 0 0 0 1 1: Enables count-up by GTIOCnA pin input level 1 0 1 0 0: Enables count-up by GTIOCnB pin input level 0 0 1 0 1: Enables count-up by GTIOCnB pin input level 1 0 1 1 0: Setting prohibited 0 1 1 1: Setting prohibited 1 0 0 0: Enables count-up by GTETRGA pin input level 0 1 0 0 1: Enables count-up by GTETRGA pin input level 1 1 0 1 0: Enables count-up by GTETRGB pin input level 0 1 0 1 1: Enables count-up by GTETRGB pin input level 1 1 1 0 0: Enables count-up by GTETRGC pin input level 0 1 1 0 1: Enables count-up by GTETRGC pin input level 1 1 1 1 0: Enables count-up by GTETRGD pin input level 0 1 1 1 1: Enables count-up by GTETRGD pin input level 1	R/W
31:28	—	These bits are read as 0. The write value should be 0.	R/W

The GTUPSR sets the source to count up the GTCNT counter.

When at least one bit in the GTUPSR register is set to 1, the GTCNT counter is counted up by the source that is set to 1 in this register. In this case, GTCR.TPCS has no effect.

Number of increment in counting is one even when multiple sources are generated simultaneously.

Inputs from GTETRGn (n = A to D) pins are input to the GPT through the POEG. Set the polarity of these signals with the POEG.

USGTRGAR bit (GTETRGA Pin Rising Input Source Counter Count Up Enable)

The USGTRGAR bit enables or disables the GTCNT counter count up on the rising edge of the GTETRGA pin input.

USGTRGAF bit (GTETRGA Pin Falling Input Source Counter Count Up Enable)

The USGTRGAF bit enables or disables the GTCNT counter count up on the falling edge of the GTETRGA pin input.

USGTRGBR bit (GTETRGB Pin Rising Input Source Counter Count Up Enable)

The USGTRGBR bit enables or disables the GTCNT counter count up on the rising edge of the GTETRGB pin input.

USGTRGBF bit (GTETRGB Pin Falling Input Source Counter Count Up Enable)

The USGTRGBF bit enables or disables the GTCNT counter count up on the falling edge of the GTETRGB pin input.

USGTRGCR bit (GTETRGC Pin Rising Input Source Counter Count Up Enable)

The USGTRGCR bit enables or disables the GTCNT counter count up on the rising edge of the GTETRGC pin input.

USGTRGCF bit (GTETRGC Pin Falling Input Source Counter Count Up Enable)

The USGTRGCF bit enables or disables the GTCNT counter count up on the falling edge of the GTETRGC pin input.

USGTRGDR bit (GTETRGD Pin Rising Input Source Counter Count Up Enable)

The USGTRGDR bit enables or disables the GTCNT counter count up on the rising edge of the GTETRGD pin input.

USGTRGDF bit (GTETRGD Pin Falling Input Source Counter Count Up Enable)

The USGTRGDF bit enables or disables the GTCNT counter count up on the falling edge of the GTETRGD pin input.

USCARBL bit (GTIOCnA Pin Rising Input during GTIOCnB Value Low Source Counter Count Up Enable)

The USCARBL bit enables or disables GTCNT counter count up on the rising edge of GTIOCnA pin input, when GTIOCnB input is 0.

USCARBH bit (GTIOCnA Pin Rising Input during GTIOCnB Value High Source Counter Count Up Enable)

The USCARBH bit enables or disables the GTCNT counter count up on the rising edge of the GTIOCnA pin input, when GTIOCnB input is 1.

USCAFBL bit (GTIOCnA Pin Falling Input during GTIOCnB Value Low Source Counter Count Up Enable)

The USCAFBL bit enables or disables the GTCNT counter count up on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 0.

USCAFBH bit (GTIOCnA Pin Falling Input during GTIOCnB Value High Source Counter Count Up Enable)

The USCAFBH bit enables or disables the GTCNT counter count up on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 1.

USCBRAL bit (GTIOCnB Pin Rising Input during GTIOCnA Value Low Source Counter Count Up Enable)

The USCBRAL bit enables or disables the GTCNT counter count up on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 0.

USCBRAH bit (GTIOCnB Pin Rising Input during GTIOCnA Value High Source Counter Count Up Enable)

The USCBRAH bit enables or disables the GTCNT counter count up on the rising edge of the GTIOCnB pin input, when the GTIOCnA input is 1.

USCBFAL bit (GTIOCnB Pin Falling Input during GTIOCnA Value Low Source Counter Count Up Enable)

The USCBFAL bit enables or disables the GTCNT counter count up on the falling edge of the GTIOCnB pin input, when the GTIOCnA input is 0.

USCBFAH bit (GTIOCnB Pin Falling Input during GTIOCnA Value High Source Counter Count Up Enable)

The USCBFAH bit enables or disables the GTCNT counter count up on the falling edge of the GTIOCnB pin input, when the GTIOCnA input is 1.

USELCm bit (ELC_GPTm Event Source Counter Count Up Enable) (m = A to H)

The USELCm bit enables or disables the GTCNT counter count up at the ELC_GPTm event input.

USILVL[3:0] bit (External Input Level Source Count-Up Enable)

Select enable or disable for the count-up of the GTCNT counter by the GTIOCnA pin input level, GTIOCnB pin input level, and GTETRGA/GTETRGB/GTETRGC/GTETRGD input level.

21.2.9 GTDNSR : General PWM Timer Down Count Source Select Register

Base address: GPT32n = 0x4016_9000 + 0x0100 × n (n = 0 to 3)

Offset address: 0x20

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	DSILVL[3:0]				DSELCH	DSELCG	DSELCF	DSELCE	DSELCD	DSELCC	DSELCB	DSELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DSCBFAH	DSCBFAL	DSCBRAH	DSCBRAL	DSCAFBH	DSCAFBL	DSCARBH	DSCARBL	DSGTRGDF	DSGTRGDR	DSGTRGCF	DSGTRGCR	DSGTRGBF	DSGTRGBR	DSGTRGAF	DSGTRGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DSGTRGAR	GTETRGA Pin Rising Input Source Counter Count Down Enable 0: Counter count down disabled on the rising edge of GTETRGA input 1: Counter count down enabled on the rising edge of GTETRGA input	R/W
1	DSGTRGAF	GTETRGA Pin Falling Input Source Counter Count Down Enable 0: Counter count down disabled on the falling edge of GTETRGA input 1: Counter count down enabled on the falling edge of GTETRGA input	R/W
2	DSGTRGBR	GTETRGB Pin Rising Input Source Counter Count Down Enable 0: Counter count down disabled on the rising edge of GTETRGB input 1: Counter count down enabled on the rising edge of GTETRGB input	R/W
3	DSGTRGBF	GTETRGB Pin Falling Input Source Counter Count Down Enable 0: Counter count down disabled on the falling edge of GTETRGB input 1: Counter count down enabled on the falling edge of GTETRGB input	R/W
4	DSGTRGCR	GTETRGC Pin Rising Input Source Counter Count Down Enable 0: Counter count down disabled on the rising edge of GTETRGC input 1: Counter count down enabled on the rising edge of GTETRGC input	R/W
5	DSGTRGCF	GTETRGC Pin Falling Input Source Counter Count Down Enable 0: Counter count down disabled on the falling edge of GTETRGC input 1: Counter count down enabled on the falling edge of GTETRGC input	R/W
6	DSGTRGDR	GTETRGD Pin Rising Input Source Counter Count Down Enable 0: Counter count down disabled on the rising edge of GTETRGD input 1: Counter count down enabled on the rising edge of GTETRGD input	R/W
7	DSGTRGDF	GTETRGD Pin Falling Input Source Counter Count Down Enable 0: Counter count down disabled on the falling edge of GTETRGD input 1: Counter count down enabled on the falling edge of GTETRGD input	R/W
8	DSCARBL	GTIOCnA Pin Rising Input during GTIOCnB Value Low Source Counter Count Down Enable 0: Counter count down disabled on the rising edge of GTIOCnA input when GTIOCnB input is 0 1: Counter count down enabled on the rising edge of GTIOCnA input when GTIOCnB input is 0	R/W
9	DSCARBH	GTIOCnA Pin Rising Input during GTIOCnB Value High Source Counter Count Down Enable 0: Counter count down disabled on the rising edge of GTIOCnA input when GTIOCnB input is 1 1: Counter count down enabled on the rising edge of GTIOCnA input when GTIOCnB input is 1	R/W
10	DSCAFBL	GTIOCnA Pin Falling Input during GTIOCnB Value Low Source Counter Count Down Enable 0: Counter count down disabled on the falling edge of GTIOCnA input when GTIOCnB input is 0 1: Counter count down enabled on the falling edge of GTIOCnA input when GTIOCnB input is 0	R/W

Bit	Symbol	Function	R/W
11	DSCAFBH	GTIOCnA Pin Falling Input during GTIOCnB Value High Source Counter Count Down Enable 0: Counter count down disabled on the falling edge of GTIOCnA input when GTIOCnB input is 1 1: Counter count down enabled on the falling edge of GTIOCnA input when GTIOCnB input is 1	R/W
12	DSCBRAL	GTIOCnB Pin Rising Input during GTIOCnA Value Low Source Counter Count Down Enable 0: Counter count down disabled on the rising edge of GTIOCnB input when GTIOCnA input is 0 1: Counter count down enabled on the rising edge of GTIOCnB input when GTIOCnA input is 0	R/W
13	DSCBRAH	GTIOCnB Pin Rising Input during GTIOCnA Value High Source Counter Count Down Enable 0: Counter count down disabled on the rising edge of GTIOCnB input when GTIOCnA input is 1 1: Counter count down enabled on the rising edge of GTIOCnB input when GTIOCnA input is 1	R/W
14	DSCBFAL	GTIOCnB Pin Falling Input during GTIOCnA Value Low Source Counter Count Down Enable 0: Counter count down disabled on the falling edge of GTIOCnB input when GTIOCnA input is 0 1: Counter count down enabled on the falling edge of GTIOCnB input when GTIOCnA input is 0	R/W
15	DSCBFAH	GTIOCnB Pin Falling Input during GTIOCnA Value High Source Counter Count Down Enable 0: Counter count down disabled on the falling edge of GTIOCnB input when GTIOCnA input is 1 1: Counter count down enabled on the falling edge of GTIOCnB input when GTIOCnA input is 1	R/W
16	DSELCA	ELC_GPTA Event Source Counter Count Down Enable 0: Counter count down disabled at the ELC_GPTA input 1: Counter count down enabled at the ELC_GPTA input	R/W
17	DSELCB	ELC_GPTB Event Source Counter Count Down Enable 0: Counter count down disabled at the ELC_GPTB input 1: Counter count down enabled at the ELC_GPTB input	R/W
18	DSELCC	ELC_GPTC Event Source Counter Count Down Enable 0: Counter count down disabled at the ELC_GPTC input 1: Counter count down enabled at the ELC_GPTC input	R/W
19	DSELCD	ELC_GPTD Event Source Counter Count Down Enable 0: Counter count down disabled at the ELC_GPTD input 1: Counter count down enabled at the ELC_GPTD input	R/W
20	DSELCE	ELC_GPTE Event Source Counter Count Down Enable 0: Counter count down disabled at the ELC_GPTE input 1: Counter count down enabled at the ELC_GPTE input	R/W
21	DSELCF	ELC_GPTF Event Source Counter Count Down Enable 0: Counter count down disabled at the ELC_GPTF input 1: Counter count down enabled at the ELC_GPTF input	R/W
22	DSELCG	ELC_GPTG Event Source Counter Count Down Enable 0: Counter count down disabled at the ELC_GPTG input 1: Counter count down enabled at the ELC_GPTG input	R/W
23	DSELCH	ELC_GPTF Event Source Counter Count Down Enable 0: Counter count down disabled at the ELC_GPTF input 1: Counter count down enabled at the ELC_GPTF input	R/W

Bit	Symbol	Function	R/W
27:24	DSILVL[3:0]	External Input Level Source Count-Down Enable 0 0 0 0: Disables count-down by external input level 0 0 0 1: Setting prohibited 0 0 1 0: Enables count-down by GTIOCnA pin input level 0 0 0 1 1: Enables count-down by GTIOCnA pin input level 1 0 1 0 0: Enables count-down by GTIOCnB pin input level 0 0 1 0 1: Enables count-down by GTIOCnB pin input level 1 0 1 1 0: Setting prohibited 0 1 1 1: Setting prohibited 1 0 0 0: Enables count-down by GTETRGA pin input level 0 1 0 0 1: Enables count-down by GTETRGA pin input level 1 1 0 1 0: Enables count-down by GTETRGB pin input level 0 1 0 1 1: Enables count-down by GTETRGB pin input level 1 1 1 0 0: Enables count-down by GTETRGC pin input level 0 1 1 0 1: Enables count-down by GTETRGC pin input level 1 1 1 1 0: Enables count-down by GTETRGD pin input level 0 1 1 1 1: Enables count-down by GTETRGD pin input level 1	R/W
31:28	—	These bits are read as 0. The write value should be 0.	R/W

The GTDNSR sets the source to count down the GTCNT counter.

When at least one bit in the GTDNSR register is set to 1, the GTCNT counter is counted down by the source that is set to 1 in this register. In this case, GTCR.TPCS has no effect.

Number of decrement in counting is one even when multiple sources are generated simultaneously.

Inputs from GTETRGn (n = A to D) pins are input to the GPT through the POEG. Set the polarity of these signals with the POEG.

DSGTRGAR bit (GTETRGA Pin Rising Input Source Counter Count Down Enable)

The DSGTRGAR bit enables or disables the GTCNT counter count down on the rising edge of the GTETRGA pin input.

DSGTRGAF bit (GTETRGA Pin Falling Input Source Counter Count Down Enable)

The DSGTRGAF bit enables or disables the GTCNT counter count down on the falling edge of the GTETRGA pin input.

DSGTRGBR bit (GTETRGB Pin Rising Input Source Counter Count Down Enable)

The DSGTRGBR bit enables or disables the GTCNT counter count down on the rising edge of the GTETRGB pin input.

DSGTRGBF bit (GTETRGB Pin Falling Input Source Counter Count Down Enable)

The DSGTRGBF bit enables or disables the GTCNT counter count down on the falling edge of the GTETRGB pin input.

DSGTRGCR bit (GTETRGC Pin Rising Input Source Counter Count Down Enable)

The DSGTRGCR bit enables or disables the GTCNT counter count down on the rising edge of the GTETRGC pin input.

DSGTRGCF bit (GTETRGC Pin Falling Input Source Counter Count Down Enable)

The DSGTRGCF bit enables or disables the GTCNT counter count down on the falling edge of the GTETRGC pin input.

DSGTRGDR bit (GTETRGD Pin Rising Input Source Counter Count Down Enable)

The DSGTRGDR bit enables or disables the GTCNT counter count down on the rising edge of the GTETRGD pin input.

DSGTRGDF bit (GTETRGD Pin Falling Input Source Counter Count Down Enable)

The DSGTRGDF bit enables or disables the GTCNT counter count down on the falling edge of the GTETRGD pin input.

DSCARBL bit (GTIOCnA Pin Rising Input during GTIOCnB Value Low Source Counter Count Down Enable)

The DSCARBL bit enables or disables the GTCNT counter count down on the rising edge of the GTIOCnA pin input, when the GTIOCnB input is 0.

DSCARBH bit (GTIOCnA Pin Rising Input during GTIOCnB Value High Source Counter Count Down Enable)

The DSCARBH bit enables or disables the GTCNT counter count down on the rising edge of the GTIOCnA pin input, when GTIOCnB input is 1.

DSCAFBL bit (GTIOCnA Pin Falling Input during GTIOCnB Value Low Source Counter Count Down Enable)

The DSCAFBL bit enables or disables the GTCNT counter count down on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 0.

DSCAFBH bit (GTIOCnA Pin Falling Input during GTIOCnB Value High Source Counter Count Down Enable)

The DSCAFBH bit enables or disables the GTCNT counter count down on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 1.

DSCBRAL bit (GTIOCnB Pin Rising Input during GTIOCnA Value Low Source Counter Count Down Enable)

The DSCBRAL bit enables or disables the GTCNT counter count down on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 0.

DSCBRAH bit (GTIOCnB Pin Rising Input during GTIOCnA Value High Source Counter Count Down Enable)

The DSCBRAH bit enables or disables the GTCNT counter count down on the rising edge of GTIOCnB pin input, when GTIOCnA input is 1.

DSCBFAL bit (GTIOCnB Pin Falling Input during GTIOCnA Value Low Source Counter Count Down Enable)

The DSCBFAL bit enables or disables the GTCNT counter count down on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 0.

DSCBFAH bit (GTIOCnB Pin Falling Input during GTIOCnA Value High Source Counter Count Down Enable)

The DSCBFAH bit enables or disables the GTCNT counter count down on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 1.

DSELCm bit (ELC_GPTm Event Source Counter Count Down Enable) (m = A to H)

The DSELCm bit enables or disables the GTCNT counter count down at the ELC_GPTm event input.

DSILVL[3:0] bit (External Input Level Source Count-Down Enable)

Select enable or disable for the count-down of the GTCNT counter by the GTIOCnA pin input level, GTIOCnB pin input level, and GTETRGA/GTETRGB/GTETRGC/GTETRGD input level.

21.2.10 GTICASR : General PWM Timer Input Capture Source Select Register A

Base address: GPT32n = 0x4016_9000 + 0x0100 × n (n = 0 to 9)

Offset address: 0x24

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	ASOC	ASELCH	ASELCG	ASELCF	ASELCE	ASELCD	ASELCC	ASELCB	ASELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ASCBFAH	ASCBFAL	ASCBRAH	ASCBRAL	ASCAFBH	ASCAFBL	ASCARBH	ASCARBL	ASGTRGDF	ASGTRGDR	ASGTRGCF	ASGTRGCR	ASGTRGBF	ASGTRGBR	ASGTRGAF	ASGTRGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ASGTRGAR	GTETRGA Pin Rising Input Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the rising edge of GTETRGA input 1: GTCCRA input capture enabled on the rising edge of GTETRGA input	R/W
1	ASGTRGAF	GTETRGA Pin Falling Input Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the falling edge of GTETRGA input 1: GTCCRA input capture enabled on the falling edge of GTETRGA input	R/W
2	ASGTRGBR	GTETRGB Pin Rising Input Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the rising edge of GTETRGB input 1: GTCCRA input capture enabled on the rising edge of GTETRGB input	R/W
3	ASGTRGBF	GTETRGB Pin Falling Input Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the falling edge of GTETRGB input 1: GTCCRA input capture enabled on the falling edge of GTETRGB input	R/W
4	ASGTRGCR	GTETRGC Pin Rising Input Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the rising edge of GTETRGC input 1: GTCCRA input capture enabled on the rising edge of GTETRGC input	R/W
5	ASGTRGCF	GTETRGC Pin Falling Input Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the falling edge of GTETRGC input 1: GTCCRA input capture enabled on the falling edge of GTETRGC input	R/W
6	ASGTRGDR	GTETRGD Pin Rising Input Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the rising edge of GTETRGD input 1: GTCCRA input capture enabled on the rising edge of GTETRGD input	R/W
7	ASGTRGDF	GTETRGD Pin Falling Input Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the falling edge of GTETRGD input 1: GTCCRA input capture enabled on the falling edge of GTETRGD input	R/W
8	ASCARBL	GTIOCnA Pin Rising Input during GTIOCnB Value Low Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the rising edge of GTIOCnA input when GTIOCnB input is 0 1: GTCCRA input capture enabled on the rising edge of GTIOCnA input when GTIOCnB input is 0	R/W
9	ASCARBH	GTIOCnA Pin Rising Input during GTIOCnB Value High Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the rising edge of GTIOCnA input when GTIOCnB input is 1 1: GTCCRA input capture enabled on the rising edge of GTIOCnA input when GTIOCnB input is 1	R/W
10	ASCAFBL	GTIOCnA Pin Falling Input during GTIOCnB Value Low Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the falling edge of GTIOCnA input when GTIOCnB input is 0 1: GTCCRA input capture enabled on the falling edge of GTIOCnA input when GTIOCnB input is 0	R/W
11	ASCAFBH	GTIOCnA Pin Falling Input during GTIOCnB Value High Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the falling edge of GTIOCnA input when GTIOCnB input is 1 1: GTCCRA input capture enabled on the falling edge of GTIOCnA input when GTIOCnB input is 1	R/W
12	ASCBRAL	GTIOCnB Pin Rising Input during GTIOCnA Value Low Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the rising edge of GTIOCnB input when GTIOCnA input is 0 1: GTCCRA input capture enabled on the rising edge of GTIOCnB input when GTIOCnA input is 0	R/W

Bit	Symbol	Function	R/W
13	ASCBRAH	GTIOCnB Pin Rising Input during GTIOCnA Value High Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the rising edge of GTIOCnB input when GTIOCnA input is 1 1: GTCCRA input capture enabled on the rising edge of GTIOCnB input when GTIOCnA input is 1	R/W
14	ASCBFAL	GTIOCnB Pin Falling Input during GTIOCnA Value Low Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the falling edge of GTIOCnB input when GTIOCnA input is 0 1: GTCCRA input capture enabled on the falling edge of GTIOCnB input when GTIOCnA input is 0	R/W
15	ASCBFAH	GTIOCnB Pin Falling Input during GTIOCnA Value High Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the falling edge of GTIOCnB input when GTIOCnA input is 1 1: GTCCRA input capture enabled on the falling edge of GTIOCnB input when GTIOCnA input is 1	R/W
16	ASELCA	ELC_GPTA Event Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled at the ELC_GPTA input 1: GTCCRA input capture enabled at the ELC_GPTA input	R/W
17	ASELCB	ELC_GPTB Event Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled at the ELC_GPTB input 1: GTCCRA input capture enabled at the ELC_GPTB input	R/W
18	ASELCC	ELC_GPTC Event Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled at the ELC_GPTC input 1: GTCCRA input capture enabled at the ELC_GPTC input	R/W
19	ASELCD	ELC_GPTD Event Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled at the ELC_GPTD input 1: GTCCRA input capture enabled at the ELC_GPTD input	R/W
20	ASELCE	ELC_GPTE Event Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled at the ELC_GPTE input 1: GTCCRA input capture enabled at the ELC_GPTE input	R/W
21	ASELCF	ELC_GPTF Event Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled at the ELC_GPTF input 1: GTCCRA input capture enabled at the ELC_GPTF input	R/W
22	ASELCG	ELC_GPTG Event Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled at the ELC_GPTG input 1: GTCCRA input capture enabled at the ELC_GPTG input	R/W
23	ASELCH	ELC_GPTH Event Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled at the ELC_GPTH input 1: GTCCRA input capture enabled at the ELC_GPTH input	R/W
24	ASOC	Other channel Source GTCCRA Input Capture Enable 0: Disables GTCCRA input capture by other channel factor 1: Enables GTCCRA input capture by other channel factor	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

The GTICASR sets the source of input capture for GTCCRA.

When at least one bit among bits in the GTICASR register is set to 1, input capture operation making the GTCCRA register as an input capture register is performed.

Inputs from GTETRGn (n = A to D) pins are input to the GPT through the POEG. Set the polarity of these signals with the POEG.

ASGTRGAR bit (GTETRGA Pin Rising Input Source GTCCRA Input Capture Enable)

The ASGTRGAR bit enables or disables the input capture for GTCCRA on the rising edge of the GTETRGA pin input.

ASGTRGAF bit (GTETRGA Pin Falling Input Source GTCCRA Input Capture Enable)

The ASGTRGAF bit enables or disables the input capture for GTCCRA on the falling edge of the GTETRGA pin input.

ASGTRGBR bit (GTETRGB Pin Rising Input Source GTCCRA Input Capture Enable)

The ASGTRGBR bit enables or disables the input capture for GTCCRA on the rising edge of the GTETRGB pin input.

ASGTRGBF bit (GTETRGB Pin Falling Input Source GTCCRA Input Capture Enable)

The ASGTRGBF bit enables or disables the input capture for GTCCRA on the falling edge of the GTETRGB pin input.

ASGTRGCR bit (GTETRGC Pin Rising Input Source GTCCRA Input Capture Enable)

The ASGTRGCR bit enables or disables the input capture for GTCCRA on the rising edge of the GTETRGC pin input.

ASGTRGCF bit (GTETRGC Pin Falling Input Source GTCCRA Input Capture Enable)

The ASGTRGCF bit enables or disables the input capture for GTCCRA on the falling edge of the GTETRGC pin input.

ASGTRGDR bit (GTETRGD Pin Rising Input Source GTCCRA Input Capture Enable)

The ASGTRGDR bit enables or disables the input capture for GTCCRA on the rising edge of the GTETRGD pin input.

ASGTRGDF bit (GTETRGD Pin Falling Input Source GTCCRA Input Capture Enable)

The ASGTRGDF bit enables or disables the input capture for GTCCRA on the falling edge of the GTETRGD pin input.

ASCARBL bit (GTIOCnA Pin Rising Input during GTIOCnB Value Low Source GTCCRA Input Capture Enable)

The ASCARBL bit enables or disables the input capture for GTCCRA on the rising edge of the GTIOCnA pin input, when GTIOCnB input is 0.

ASCARBH bit (GTIOCnA Pin Rising Input during GTIOCnB Value High Source GTCCRA Input Capture Enable)

The ASCARBH bit enables or disables the input capture for GTCCRA on the rising edge of the GTIOCnA pin input, when GTIOCnB input is 1.

ASCAFBL bit (GTIOCnA Pin Falling Input during GTIOCnB Value Low Source GTCCRA Input Capture Enable)

The ASCAFBL bit enables or disables the input capture for GTCCRA on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 0.

ASCAFBH bit (GTIOCnA Pin Falling Input during GTIOCnB Value High Source GTCCRA Input Capture Enable)

The ASCAFBH bit enables or disables the input capture for GTCCRA on the falling edge of the GTIOCnA pin input, when the GTIOCnB input is 1.

ASCBRAL bit (GTIOCnB Pin Rising Input during GTIOCnA Value Low Source GTCCRA Input Capture Enable)

The ASCBRAL bit enables or disables the input capture for GTCCRA on the rising edge of the GTIOCnB pin input, when the GTIOCnA input is 0.

ASCBRAH bit (GTIOCnB Pin Rising Input during GTIOCnA Value High Source GTCCRA Input Capture Enable)

The ASCBRAH bit enables or disables the input capture for GTCCRA on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 1.

ASCBFAL bit (GTIOCnB Pin Falling Input during GTIOCnA Value Low Source GTCCRA Input Capture Enable)

The ASCBFAL bit enables or disables the input capture for GTCCRA on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 0.

ASCBFAH bit (GTIOCnB Pin Falling Input during GTIOCnA Value High Source GTCCRA Input Capture Enable)

The ASCBFAH bit enables or disables the input capture for GTCCRA on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 1.

ASELCm bit (ELC_GPTm Event Source Counter GTCCRA Input Capture Enable) (m = A to H)

The ASELCm bit enables or disables the input capture for GTCCRA at the ELC_GPTm event input.

ASOC bit (Other channel Source GTCCRA Input Capture Enable)

Select enable or disable for an input capture to the GTCCRA register by other channel factor.

Input capture of other channel factors is not subject to input capture factors to other channels set by the ICAFA and ICBFA bits of the GTICCR register.

21.2.11 GTICBSR : General PWM Timer Input Capture Source Select Register B

Base address: GPT32n = 0x4016_9000 + 0x0100 × n (n = 0 to 9)

Offset address: 0x28

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	BSOC	BSELCH	BSELCG	BSELCF	BSELCE	BSELCD	BSELCC	BSELCB	BSELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	BSCBFAH	BSCBFAL	BSCBRAH	BSCBRAL	BSCAFBH	BSCAFBL	BSCARBH	BSCARBL	BSGTRGDF	BSGTRGDR	BSGTRGCF	BSGTRGCR	BSGTRGBF	BSGTRGBR	BSGTRGAF	BSGTRGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BSGTRGAR	GTETRGA Pin Rising Input Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the rising edge of GTETRGA input 1: GTCCRB input capture enabled on the rising edge of GTETRGA input	R/W
1	BSGTRGAF	GTETRGA Pin Falling Input Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the falling edge of GTETRGA input 1: GTCCRB input capture enabled on the falling edge of GTETRGA input	R/W
2	BSGTRGBR	GTETRGB Pin Rising Input Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the rising edge of GTETRGB input 1: GTCCRB input capture enabled on the rising edge of GTETRGB input	R/W
3	BSGTRGBF	GTETRGB Pin Falling Input Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the falling edge of GTETRGB input 1: GTCCRB input capture enabled on the falling edge of GTETRGB input	R/W
4	BSGTRGCR	GTETRGC Pin Rising Input Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the rising edge of GTETRGC input 1: GTCCRB input capture enabled on the rising edge of GTETRGC input	R/W
5	BSGTRGCF	GTETRGC Pin Falling Input Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the falling edge of GTETRGC input 1: GTCCRB input capture enabled on the falling edge of GTETRGC input	R/W
6	BSGTRGDR	GTETRGD Pin Rising Input Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the rising edge of GTETRGD input 1: GTCCRB input capture enabled on the rising edge of GTETRGD input	R/W
7	BSGTRGDF	GTETRGD Pin Falling Input Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the falling edge of GTETRGD input 1: GTCCRB input capture enabled on the falling edge of GTETRGD input	R/W

Bit	Symbol	Function	R/W
8	BSCARBL	GTIOCnA Pin Rising Input during GTIOCnB Value Low Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the rising edge of GTIOCnA input when GTIOCnB input is 0 1: GTCCRB input capture enabled on the rising edge of GTIOCnA input when GTIOCnB input is 0	R/W
9	BSCARBH	GTIOCnA Pin Rising Input during GTIOCnB Value High Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the rising edge of GTIOCnA input when GTIOCnB input is 1 1: GTCCRB input capture enabled on the rising edge of GTIOCnA input when GTIOCnB input is 1	R/W
10	BSCAFBL	GTIOCnA Pin Falling Input during GTIOCnB Value Low Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the falling edge of GTIOCnA input when GTIOCnB input is 0 1: GTCCRB input capture enabled on the falling edge of GTIOCnA input when GTIOCnB input is 0	R/W
11	BSCAFBH	GTIOCnA Pin Falling Input during GTIOCnB Value High Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the falling edge of GTIOCnA input when GTIOCnB input is 1 1: GTCCRB input capture enabled on the falling edge of GTIOCnA input when GTIOCnB input is 1	R/W
12	BSCBRAL	GTIOCnB Pin Rising Input during GTIOCnA Value Low Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the rising edge of GTIOCnB input when GTIOCnA input is 0 1: GTCCRB input capture enabled on the rising edge of GTIOCnB input when GTIOCnA input is 0	R/W
13	BSCBRAH	GTIOCnB Pin Rising Input during GTIOCnA Value High Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the rising edge of GTIOCnB input when GTIOCnA input is 1 1: GTCCRB input capture enabled on the rising edge of GTIOCnB input when GTIOCnA input is 1	R/W
14	BSCBFAL	GTIOCnB Pin Falling Input during GTIOCnA Value Low Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the falling edge of GTIOCnB input when GTIOCnA input is 0 1: GTCCRB input capture enabled on the falling edge of GTIOCnB input when GTIOCnA input is 0	R/W
15	BSCBFAH	GTIOCnB Pin Falling Input during GTIOCnA Value High Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the falling edge of GTIOCnB input when GTIOCnA input is 1 1: GTCCRB input capture enabled on the falling edge of GTIOCnB input when GTIOCnA input is 1	R/W
16	BSELCA	ELC_GPTA Event Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled at the ELC_GPTA input 1: GTCCRB input capture enabled at the ELC_GPTA input	R/W
17	BSELCB	ELC_GPTB Event Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled at the ELC_GPTB input 1: GTCCRB input capture enabled at the ELC_GPTB input	R/W
18	BSELCC	ELC_GPTC Event Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled at the ELC_GPTC input 1: GTCCRB input capture enabled at the ELC_GPTC input	R/W
19	BSELCD	ELC_GPTD Event Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled at the ELC_GPTD input 1: GTCCRB input capture enabled at the ELC_GPTD input	R/W

Bit	Symbol	Function	R/W
20	BSELCE	ELC_GPTE Event Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled at the ELC_GPTE input 1: GTCCRB input capture enabled at the ELC_GPTE input	R/W
21	BSELCF	ELC_GPTF Event Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled at the ELC_GPTF input 1: GTCCRB input capture enabled at the ELC_GPTF input	R/W
22	BSELCG	ELC_GPTG Event Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled at the ELC_GPTG input 1: GTCCRB input capture enabled at the ELC_GPTG input	R/W
23	BSELCH	ELC_GPTH Event Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled at the ELC_GPTH input 1: GTCCRB input capture enabled at the ELC_GPTH input	R/W
24	BSOC	Other channel Source GTCCRB Input Capture Enable 0: Disables GTCCRB input capture by other channel factor 1: Enables GTCCRB input capture by other channel factor	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

The GTICBSR sets the source of input capture for GTCCRB.

When at least one bit among bits in the GTICBSR register is set to 1, input capture operation making the GTCCRB register as an input capture register is performed.

Inputs from GTETRGn (n = A to D) pins are input to the GPT through the POEG. Set the polarity of these signals with the POEG.

BSGTRGAR bit (GTETRGA Pin Rising Input Source GTCCRB Input Capture Enable)

The BSGTRGAR bit enables or disables the input capture for GTCCRB on the rising edge of the GTETRGA pin input.

BSGTRGAF bit (GTETRGA Pin Falling Input Source GTCCRB Input Capture Enable)

The BSGTRGAF bit enables or disables the input capture for GTCCRB on the falling edge of the GTETRGA pin input.

BSGTRGBR bit (GTETRGB Pin Rising Input Source GTCCRB Input Capture Enable)

The BSGTRGBR bit enables or disables the input capture for GTCCRB on the rising edge of GTETRGB pin input.

BSGTRGBF bit (GTETRGB Pin Falling Input Source GTCCRB Input Capture Enable)

The BSGTRGBF bit enables or disables the input capture for GTCCRB on the falling edge of the GTETRGB pin input.

BSGTRGCR bit (GTETRGC Pin Rising Input Source GTCCRB Input Capture Enable)

The BSGTRGCR bit enables or disables the input capture for GTCCRB on the rising edge of GTETRGC pin input.

BSGTRGCF bit (GTETRGC Pin Falling Input Source GTCCRB Input Capture Enable)

The BSGTRGCF bit enables or disables the input capture for GTCCRB on the falling edge of the GTETRGC pin input.

BSGTRGDR bit (GTETRGD Pin Rising Input Source GTCCRB Input Capture Enable)

The BSGTRGDR bit enables or disables the input capture for GTCCRB on the rising edge of GTETRGD pin input.

BSGTRGDF bit (GTETRGD Pin Falling Input Source GTCCRB Input Capture Enable)

The BSGTRGDF bit enables or disables the input capture for GTCCRB on the falling edge of the GTETRGD pin input.

BSCARBL bit (GTIOCnA Pin Rising Input during GTIOCnB Value Low Source GTCCRB Input Capture Enable)

The BSCARBL bit enables or disables the input capture for GTCCRB on the rising edge of the GTIOCnA pin input, when the GTIOCnB input is 0.

BSCARBH bit (GTIOCnA Pin Rising Input during GTIOCnB Value High Source GTCCRB Input Capture Enable)

The BSCARBH bit enables or disables the input capture for GTCCRB on the rising edge of the GTIOCnA pin input, when GTIOCnB input is 1.

BSCAFBL bit (GTIOCnA Pin Falling Input during GTIOCnB Value Low Source GTCCRB Input Capture Enable)

The BSCAFBL bit enables or disables the input capture for GTCCRB on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 0.

BSCAFBH bit (GTIOCnA Pin Falling Input during GTIOCnB Value High Source GTCCRB Input Capture Enable)

The BSCAFBH bit enables or disables the input capture for GTCCRB on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 1.

BSCBRAL bit (GTIOCnB Pin Rising Input during GTIOCnA Value Low Source GTCCRB Input Capture Enable)

The BSCBRAL bit enables or disables the input capture for GTCCRB on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 0.

BSCBRAH bit (GTIOCnB Pin Rising Input during GTIOCnA Value High Source GTCCRB Input Capture Enable)

The BSCBRAH bit enables or disables the input capture for GTCCRB on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 1.

BSCBFAL bit (GTIOCnB Pin Falling Input during GTIOCnA Value Low Source GTCCRB Input Capture Enable)

The BSCBFAL bit enables or disables the input capture for GTCCRB on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 0.

BSCBFAH bit (GTIOCnB Pin Falling Input during GTIOCnA Value High Source GTCCRB Input Capture Enable)

The BSCBFAH bit enables or disables the input capture for GTCCRB on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 1.

BSELCm bit (ELC_GPTm Event Source Counter GTCCRB Input Capture Enable) (m = A to H)

The BSELCm bit enables or disables the input capture for GTCCRB at the ELC_GPTm event input.

BSOC bit (Other channel Source GTCCRB Input Capture Enable)

Select enable or disable for an input capture to the GTCCRB register by other channel factor.

Input capture of other channel factors is not subject to input capture factors to other channels set by the ICAFB and ICBFB bits of the GTICCR register.

21.2.12 GTCR : General PWM Timer Control Register

Base address: GPT32n = 0x4016_9000 + 0x0100 × n (n = 0 to 9)

Offset address: 0x2C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	CKEG[1:0]		TPCS[3:0]				—	—	—	MD[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SSCEN	—	—	CPSCD	SSCGRP[1:0]		SCGTIOC	ICDS	—	—	—	—	—	—	—	CST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CST	Count Start 0: Count operation is stopped 1: Count operation is performed	R/W*1
7:1	—	These bits are read as 0. The write value should be 0.	R/W
8	ICDS	Input Capture Operation Select During Count Stop 0: Input capture is operated during count stop. 1: Input capture is not operated during count stop.	R/W
9	SCGTIOC	GTIOC input Source Synchronous Clear Enable 0: Disables to use the counter clear by GTIOC input as the clear factor for other channels 1: Enables to use the counter clear by GTIOC input as the clear factor for other channels	R/W
11:10	SSCGRP[1:0]	Synchronous Set/Clear Group Select 0 0: Select synchronous set/clear group A 0 1: Select synchronous set/clear group B 1 0: Select synchronous set/clear group C 1 1: Select synchronous set/clear group D	R/W*1
12	CPSCD	Complementary PWM Mode Synchronous Clear Disable*2 0: Enable synchronous counter clear by other channel other than the section of trough in complementary PWM mode 1: Disable synchronous counter clear by other channel other than the section of trough in complementary PWM mode	R/W*1
14:13	—	These bits are read as 0. The write value should be 0.	R/W
15	SSCEN	Synchronous Set/Clear Enable 0: Disable Synchronous set/clear of the GTCNT counter 1: Enable Synchronous set/clear of the GTCNT counter	R/W*1
19:16	MD[3:0]	Mode Select*3 0 0 0 0: Saw-wave PWM mode 1(single buffer or double buffer possible) 0 0 0 1: Saw-wave one-shot pulse mode (fixed buffer operation) 0 0 1 0: Saw-wave PWM mode 2(single buffer or double buffer possible) 0 0 1 1: Setting prohibited 0 1 0 0: Triangle-wave PWM mode 1 (32-bit transfer at trough) (single buffer or double buffer possible) 0 1 0 1: Triangle-wave PWM mode 2 (32-bit transfer at crest and trough) (single buffer or double buffer possible) 0 1 1 0: Triangle-wave PWM mode 3 (64-bit transfer at trough) (fixed buffer operation) 0 1 1 1: Setting prohibited 1 0 0 0: Setting prohibited 1 0 0 1: Setting prohibited 1 0 1 0: Setting prohibited 1 0 1 1: Setting prohibited 1 1 0 0: Complementary PWM mode 1(transfer at crest) 1 1 0 1: Complementary PWM mode 2(transfer at trough) 1 1 1 0: Complementary PWM mode 3(transfer at crest and trough) 1 1 1 1: Complementary PWM mode 4(immediate transfer)	R/W*1
22:20	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
26:23	TPCS[3:0]	Timer Prescaler Select 0 0 0 0: GTCLK/1 0 0 0 1: GTCLK/2 0 0 1 0: GTCLK/4 0 0 1 1: GTCLK/8 0 1 0 0: GTCLK/16 0 1 0 1: GTCLK/32 0 1 1 0: GTCLK/64 0 1 1 1: GTCLK/128 1 0 0 0: GTCLK/256 1 0 0 1: GTCLK/512 1 0 1 0: GTCLK/1024 1 0 1 1: Setting prohibited 1 1 0 0: GTETRGA (Via the POEG) 1 1 0 1: GTETRGB (Via the POEG) 1 1 1 0: GTETRGC (Via the POEG) 1 1 1 1: GTETRGD (Via the POEG)	R/W*1
28:27	CKEG[1:0]	Clock Edge Select 0 0: Select rising edge of GTETRG for clock count 0 1: Select falling edge of GTETRG for clock count Others: Select both edge of GTETRG for clock count	R/W*1
31:29	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. When in complementary PWM mode, no matter which register of master channel/slave channel 1/slave channel 2 is written, three channels are written at the same time.
Note 2. This bit is only available in GPT324 to GPT329.
In GPT320 to GPT323, this bit is read as 0. The write value should be 0.
Note 3. MD[3] bit is only available in GPT324 to GPT329. GPT320 to GPT323 only support Saw-wave PWM mode and Triangle-wave PWM mode except Saw-wave PWM mode 2.

The GTCR controls GTCNT.

Access in 8 bit unit to GTCR is prohibited.

CST bit (Count Start)

The CST bit controls the GTCNT counter start and stop.

[Setting conditions]

- The GTSTR value where the channel number associated with the bit number is set to 1 with the GTSSR.CSTRT bit at 1
- The ELC event input, the external trigger, or the GTIOCnA/GTIOCnB input that are enabled by GTSSR for the starting counter source, occurs (n = 0 to 9)
- 1 is written by software directly.

[Clearing conditions]

- The GTSTP value where the channel number associated with the bit number is set to 1 with the GTPSR.CSTOP bit at 1
- The ELC event input, the external trigger, or the GTIOCnA/GTIOCnB input enabled by GTPSR as the counter stop source, occurs (n = 0 to 9)
- 0 is written by software directly.
- When the period count function is finished while the GTPC.ASTP bit is 1.

ICDS bit (Input Capture Operation Select During Count Stop)

This bit selects input capture operation during count stop, when the input capture function is selected.

SCGTIOC bit (GTIOC input Source Synchronous Clear Enable)

Select enable or disable the use of the counter clear by GTIOCnA/GTIOCnB input pin selected by the GTCSR register as the counter clear factor for other channels.

SSCGRP[1:0] bits (Synchronous Set/Clear Group Select)

Select the channel group of synchronous set/clear.

In complementary PWM mode, slave channels are also controlled by setting the SSCGRP[1:0] bits of the master channel.

CPSCD bit (Complementary PWM Mode Synchronous Clear Disable)

Select disable or enable of counter clear when synchronous clear from other channel occurs except section of trough in complementary PWM mode.

The slave channel is also controlled by setting the CPSCD bit of the master channel.

SSCEN bit (Synchronous Set/Clear Enable)

Select disable or enable of Synchronous set/clear.

In complementary PWM mode, slave channels are also controlled by setting the SSCEN bits of the master channel.

MD[3:0] bits (Mode Select)

These bits select the GPT operating mode.

In complementary PWM mode, slave channels are also controlled by setting the MD bits of the master channel.

Only the MD[3:2] bit (MD[2] for GPT320 to GPT323) is valid at input capture. Counting in saw-wave mode is performed with 00 for the MD[3:2] bit (0 for the MD[2] of GPT320 to GPT323), and counting in triangle-wave mode is performed with 01 for the MD[3:2] bit (1 for MD[2] of GPT320 to GPT323), and counting in complementary PWM mode is performed with 1 for the MD[3].

The MD bits must be set while the GTCNT operation is stopped.

During the event count operation (when at least one bit among the bits for the GTUPSR and GTDNSR registers is set to 1), setting of the MD bits are ignored, where counting in saw-wave or triangle-wave or complementary PWM modes is not performed. Instead, up-counting or down-counting by a source set by the GTUPSR and GTDNSR registers is performed.

TPCS[3:0] bits (Timer Prescaler Select)

The TPCS[3:0] bits select the clock for GTCNT. A clock prescaler can be selected independently for each channel. The TPCS[3:0] bits must be set while the GTCNT operation is stopped.

CKEG[1:0] bits (Clock Edge Select)

When GTETRG input is selected by TPCS[3:0] bits, select the edge of GTETRG input used as the clock of GTCNT counter.

Set the CKEG[1:0] bits only when the GTCNT counter operation is stopped.

21.2.13 GTUDDTYC : General PWM Timer Count Direction and Duty Setting Register

Base address: GPT32n = 0x4016_9000 + 0x0100 × n (n = 0 to 9)

Offset address: 0x30

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

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

Bit	Symbol	Function	R/W
0	UD	Count Direction Setting 0: GTCNT counts down 1: GTCNT counts up	R/W
1	UDF	Forcible Count Direction Setting 0: Not forcibly set 1: Forcibly set	R/W

Bit	Symbol	Function	R/W
15:2	—	These bits are read as 0. The write value should be 0.	R/W
17:16	OADTY[1:0]	GTIOCnA Output Duty Setting 0 0: GTIOCnA pin duty depends on the compare match 0 1: GTIOCnA pin duty depends on the compare match 1 0: GTIOCnA pin duty 0% 1 1: GTIOCnA pin duty 100%	R/W
18	OADTYF	Forcible GTIOCnA Output Duty Setting 0: Not forcibly set 1: Forcibly set	R/W
19	OADTYR	GTIOCnA Output Value Selecting after Releasing 0%/100% Duty Setting 0: The function selected by the GTIOA[3:2] bits is applied to the output value when the duty cycle is set after release from the 0 or 100% duty-cycle setting. 1: The function selected by the GTIOA[3:2] bits is applied to the compare match output value which is masked after release from the 0 or 100% duty-cycle setting.	R/W
23:20	—	These bits are read as 0. The write value should be 0.	R/W
25:24	OBDTY[1:0]	GTIOCnB Output Duty Setting 0 0: GTIOCnB pin duty depends on the compare match 0 1: GTIOCnB pin duty depends on the compare match 1 0: GTIOCnB pin duty 0% 1 1: GTIOCnB pin duty 100%	R/W
26	OBDTYF	Forcible GTIOCnB Output Duty Setting 0: Not forcibly set 1: Forcibly set	R/W
27	OBDTYR	GTIOCnB Output Value Selecting after Releasing 0%/100% Duty Setting 0: The function selected by the GTIOB[3:2] bits is applied to the output value when the duty cycle is set after release from the 0 or 100% duty-cycle setting. 1: The function selected by the GTIOB[3:2] bits is applied to the compare match output value which is masked after release from the 0 or 100% duty-cycle setting.	R/W
31:28	—	These bits are read as 0. The write value should be 0.	R/W

The GTUDDTYC sets the direction in which the GTCNT counts (up-counting or down-counting), and sets the duty of the GTIOCnA/GTIOCnB pin output.

The setting is invalid during the event count operation, Saw-wave PWM mode 2, Complementary PWM mode.

Count Direction:

- In saw-wave mode.
  When the UD value is set to 0 during up-counting, the count direction changes at an overflow (the timing synchronous with count clock after the GTCNT value becomes the GTPR value). When the UD value is set to 1 during down-counting, the count direction changes at an underflow (the timing synchronous with count clock after the GTCNT value becomes 0).
  When the UD value changes from 1 to 0 with the UDF bit being 0 and while counting stops, the counter starts up-counting and the count direction changes at an overflow (the timing synchronous with count clock after the GTCNT value becomes the GTPR value). When the UD value changes from 0 to 1 with the UDF bit being 0 and while counting stops, the counter starts down-counting and the count direction changes at an underflow (the timing synchronous with count clock after the GTCNT value becomes 0).
  When the UDF bit is set to 1 while counting stops, the UD bit value is reflected in the count direction when counting starts.
- In triangle-wave mode.
  When the UD value changes during counting, the count direction does not change. When the UD value changes while the UDF bit is 0 and counting stops, the change is not reflected in the count direction when counting starts.
  When the UDF bit is set to 1 while counting is stopped, the UD value is reflected in the count direction when counting starts.

UD bit (Count Direction Setting)

The UD bit sets the count direction (up-counting or down-counting) for GTCNT.

UDF bit (Forcible Count Direction Setting)

The UDF bit forcibly sets the count direction when GTCNT starts operation as the UD value. Only 0 should be written to this bit during counter operation. When 1 is written to this bit while counting stops, return this bit to 0 before counting starts.

Output duty

- In saw-wave mode.
  When the OADTY/OBDTY value changes during up-counting, the duty is reflected at an overflow (GTCNT = GTPR).
  When the OADTY/OBDTY value is changed during down-counting, the duty is reflected at an underflow (GTCNT = 0).
  When the OADTY/OBDTY value is changed with the OADTYF/OBDTYF bit being 0 and while counting stops the output duty is not reflected at the starting counter operation. When the count direction is up, the output duty is reflected at an overflow (GTCNT = GTPR). When the count direction is down, the output duty is reflected at an underflow (GTCNT = 0).
  When the OADTY/OBDTY value is changed with the OADTYF/OBDTYF bit being 1 and while counting stops, the output duty is reflected at starting counter operation.
- In triangle-wave mode.
  When the OADTY/OBDTY value changes during counting, the duty is reflected at an underflow.
  When the OADTY/OBDTY value is changed with the OADTYF/OBDTYF bit being 0 and while counting stops, the output duty is not reflected at the starting counter operation. The output duty is reflected at an underflow.
  When the OADTY/OBDTY value is changed with the OADTYF/OBDTYF bit being 1 and while counting stops, the output duty is reflected at starting counter operation.

In both saw-wave mode and triangle-wave mode, when the OADTYF/OBDTYF bit is set back to 0 and the OADTY[1:0]/OBDTY[1:0] bits are set after setting the OADTYF/OBDTYF bit to 1 and setting the OADTY[1:0]/OBDTY[1:0] bits for the duty of first cycle while count operation is stopped, these duty-cycle set during stopping count operation are reflected in the first cycle and the second cycle after starting count operation.

OmDTY[1:0] bits (GTIOCnm Output Duty Setting) (m = A, B)

The OmDTY[1:0] bits set the output duty (0%, 100% or compare match control) of the GTIOCnm pin.

OmDTYF bit (Forcible GTIOCnm Output Duty Setting) (m = A, B)

The OmDTYF bit forcibly sets the output duty cycle to the OmDTY setting. Set this bit to 0 during counter operation.

OmDTYR bit (GTIOCnm Output Value Selecting after Releasing 0%/100% Duty Setting) (m = A, B)

The OmDTYR bit selects the value that is the object of output retained or toggled at cycle end, when the control changes from 0% or 100% duty setting to compare match for the GTIOCnm pin and GTIOR.GTIOm[3:2] bits are set to 00b (output retained at cycle end) or the GTIOR.GTIOm[3:2] bits are set to 11b (output toggled at cycle end).

The GPT internally continues to perform compare match operation during duty-cycle 0% or 100% operation. When the OmDTYR bit is 1, the value after the period has elapsed due this compare match operation is target for the GTIOm[3:2] bits.

21.2.14 GTIOR : General PWM Timer I/O Control Register

Base address: GPT32n = 0x4016_9000 + 0x0100 × n (n = 0 to 9)

Offset address: 0x34

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	NFCSB[1:0]		NFBEN	—	OBEOCD	OBDF[1:0]		OBE	OBHLD	OBDFLT	—	GTIOB[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	NFCSA[1:0]		NFAEN	PSYE	OAEOCD	OADF[1:0]		OAE	OAHLD	OADFLT	CPSCIR	GTIOA[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	GTIOA[4:0]	GTIOCnA Pin Function Select See Table 21.4 and Table 21.5.	R/W
5	CPSCIR*1	Complementary PWM Mode Initial Output at Synchronous Clear Disable 0: Output the initial value set by the GTIOR.GTIOA and GTIOB bits when synchronous clear occurs in Trough section of complementary PWM mode 1: Disable output the initial value	R/W
6	OADFLT	GTIOCnA Pin Output Value Setting at the Count Stop 0: The GTIOCnA pin outputs low when counting stops 1: The GTIOCnA pin outputs high when counting stops	R/W
7	OAHLD	GTIOCnA Pin Output Setting at the Start/Stop Count 0: The GTIOCnA pin output level at the start or stop of counting depends on the register setting 1: The GTIOCnA pin output level is retained at the start or stop of counting	R/W
8	OAE	GTIOCnA Pin Output Enable 0: Output is disabled 1: Output is enabled	R/W
10:9	OADF[1:0]	GTIOCnA Pin Disable Value Setting 0 0: None of the below options are specified 0 1: GTIOCnA pin is set to Hi-Z in response to controlling the output negation 1 0: GTIOCnA pin is set to 0 in response to controlling the output negation 1 1: GTIOCnA pin is set to 1 in response to controlling the output negation	R/W
11	OAEOCD*1	GTCCRA Compare Match Cycle End Output Invalidate 0: Validate GTIOA[3:2] setting 1: Invalidate GTIOA[3:2] setting (GTIOCnA pin output is retained)	R/W
12	PSYE	PWM Synchronous output Enable 0: Disable GTCPPOm pin output 1: Enable GTCPPOm pin output	R/W
13	NFAEN	Noise Filter A Enable 0: The noise filter for the GTIOCnA pin is disabled 1: The noise filter for the GTIOCnA pin is enabled	R/W
15:14	NFCSA[1:0]	Noise Filter A Sampling Clock Select 0 0: GTCLK/1 0 1: GTCLK/4 1 0: GTCLK/16 1 1: GTCLK/64	R/W
20:16	GTIOB[4:0]	GTIOCnB Pin Function Select See Table 21.4 and Table 21.5.	R/W
21	—	This bit is read as 0. The write value should be 0.	R/W
22	OBDFLT	GTIOCnB Pin Output Value Setting at the Count Stop 0: The GTIOCnB pin outputs low when counting stops 1: The GTIOCnB pin outputs high when counting stops	R/W
23	OBHLD	GTIOCnB Pin Output Setting at the Start/Stop Count 0: The GTIOCnB pin output level at the start/stop of counting depends on the register setting 1: The GTIOCnB pin output level is retained at the start/stop of counting	R/W
24	OBE	GTIOCnB Pin Output Enable 0: Output is disabled 1: Output is enabled	R/W
26:25	OBDF[1:0]	GTIOCnB Pin Disable Value Setting 0 0: None of the below options are specified 0 1: GTIOCnB pin is set to Hi-Z in response to controlling the output negation 1 0: GTIOCnB pin is set to 0 in response to controlling the output negation 1 1: GTIOCnB pin is set to 1 in response to controlling the output negation	R/W

Bit	Symbol	Function	R/W
27	OBEOCD*1	GTCCRB Compare Match Cycle End Output Invalidate 0: When Saw-wave PWM mode 1, validate GTIOB[3:2] setting When Saw-wave PWM mode 2, validate GTIOA[3:2] setting 1: When Saw-wave PWM mode 1, invalidate GTIOB[3:2] setting (GTIOCnB pin output is retained) When Saw-wave PWM mode 2, invalidate GTIOA[3:2] setting (GTIOCnA pin output is retained)	R/W
28	—	This bit is read as 0. The write value should be 0.	R/W
29	NFBEN	Noise Filter B Enable 0: The noise filter for the GTIOCnB pin is disabled 1: The noise filter for the GTIOCnB pin is enabled	R/W
31:30	NFCSB[1:0]	Noise Filter B Sampling Clock Select 0 0: GTCLK/1 0 1: GTCLK/4 1 0: GTCLK/16 1 1: GTCLK/64	R/W

Note 1. This bit is only available in GPT324 to GPT329.
In GPT320 to GPT323, this bit is read as 0. The write value should be 0.

The GTIOR sets the functions of the GTIOCnA, GTIOCnB and GTCPPOm pins. (n = 0 to 9, m = 0 to 4, 7)

GTIOA[4:0] bits (GTIOCnA Pin Function Select)

The GTIOA[4:0] bits select the GTIOCnA pin function. For details, see Table 21.4 and Table 21.5.

CPSCIR bit (Complementary PWM Mode Initial Output at Synchronous Clear Disable)

Select the output waveform when synchronous clear occurs in complementary PWM mode.

The initial output is disabled by this function only when synchronous clear occurs in the trough section in complementary PWM mode. If a synchronous clear occurs at any other time, the initial value set by the GTIOA[4]/GTIOB[4] bits is output regardless of the CPSCIR bit setting. In addition, the initial value set by the GTIOA[4]/GTIOB[4] bits is output even when the synchronous clear occurs in the trough section immediately after the count starts.

OADFLT bit (GTIOCnA Pin Output Value Setting at the Count Stop)

The OADFLT bit sets whether the GTIOCnA pin outputs high or low when counting stops.

OAHLD bit (GTIOCnA Pin Output Setting at the Start/Stop Count)

The OAHLD bit specifies whether the GTIOCnA pin output level is retained or the level at the start or stop of counting depends on the register setting.

When the OAHLD bit is set to 0:

- The value specified in bit [4] of the GTIOA[4:0] bits is output when counting starts
- The value specified in the OADFLT bit is output when counting stops
- If the OADFLT bit is modified while counting stops, the new value is immediately reflected in the output.

When the OAHLD bit is set to 1:

- The output is retained when counting starts or stops.

OAE bit (GTIOCnA Pin Output Enable)

The OAE bit disables or enables the GTIOCnA pin output.

When GTCCRA register is used as the input capture register (at least one bit in the GTICASR register is set to 1), the GTIOCnA pin does not output regardless of the OAE bit value.

OADF[1:0] bits (GTIOCnA Pin Disable Value Setting)

The OADF[1:0] bits select the output value of the GTIOCnA pin in response to a request to disable output from the POEG.

OAEOCD bit (GTCCRA Compare Match Cycle End Output Invalidate)

If the cycle end matches the GTCCRA compare match timing in Saw-wave PWM mode 1 and 2, select invalid/valid of the setting of GTIOA[3:2] bits. When 1 (disabled) is set, the GTIOCnA pin holds the output when the cycle end and the GTCCRA compare match timing match.

PSYE bit (PWM Synchronous output Enable)

This bit set Enable/disable of output signal from GTCPPOm pin synchronized with the PWM cycle that toggles at the crest/trough/GTCNT counter clear of complementary PWM mode and Triangle-wave mode or the end of the cycle of the Saw-wave mode.

The initial output of the GTCPPOn output pin is Low, and the count start results in High.

NFAEN bit (Noise Filter A Enable)

The NFAEN bit disables or enables the noise filter for input from the GTIOCnA pin. Because changing the value of the bit might lead to the internal generation of an unexpected edge, select the output compare function for the relevant pin in the GTIOR register before doing so.

NFCSA[1:0] bits (Noise Filter A Sampling Clock Select)

The NFCSA[1:0] bits set the sampling interval for the noise filter of the GTIOCnA pin. When setting these bits, wait for 2 cycles of the selected sampling interval before setting the input capture function.

GTIOB[4:0] bits (GTIOCnB Pin Function Select)

The GTIOB[4:0] bits select the GTIOCnB pin function. For details, see Table 21.4 and Table 21.5.

In Saw-wave PWM mode 2, only the GTIOB[1:0] bits are valid, and the GTIOCnA pin output is selected instead of the GTIOCnB pin by the GTCCRB register compare match.

OBDFLT bit (GTIOCnB Pin Output Value Setting at the Count Stop)

The OBDFLT bit sets whether the GTIOCnB pin outputs high or low when counting stops.

OBHLD bit (GTIOCnB Pin Output Setting at the Start/Stop Count)

The OBHLD bit specifies whether theGTIOCnB pin output level is retained or the level at the start or stop of counting depends on the register setting.

When the OBHLD bit is set to 0:

- The value specified in bit [4] of the GTIOB[4:0] bits is output when counting starts
- The value specified in the OBDFLT bit is output when counting stops
- If the OBDFLT bit is modified while counting stops, the new value is immediately reflected in the output.

When the OBHLD bit is set to 1:

- The output is retained when counting starts or stops.

OBE bit (GTIOCnB Pin Output Enable)

The OBE bit disables or enables the GTIOCnB pin output.

When GTCCRB register is used as the input capture register (at least one bit in the GTICBSR register is set to 1), the GTIOCnB pin does not output regardless of the OBE bit value.

OBDF[1:0] bits (GTIOCnB Pin Disable Value Setting)

The OBDF[1:0] bits select the output value of the GTIOCnB pin in response to a request to disable output from the POEG.

OBEOCD bit (GTCCRB Compare Match Cycle End Output Invalidate)

If the cycle end matches the GTCCRB compare match timing in Saw-wave PWM mode 1 and 2, select invalid/valid of the GTIOB[3:2] bits setting in saw-wave PWM mode 1 or the GTIOA[3:2] bit setting in saw-wave PWM mode 2. When 1 (disabled) is set, the GTIOCnB pin in saw-wave PWM mode 1 or the GTIOCnA pin in saw-wave PWM mode 2 holds the output when the cycle end and the GTCCRB compare match timing match.

NFBEN bit (Noise Filter B Enable)

The NFBEN bit disables or enables the noise filter for input from the GTIOCnB pin. Because changing the value of the bit might lead to the internal generation of an unexpected edge, select the output compare function for the relevant pin in the GTIOR register before doing so.

NFCSB[1:0] bits (Noise Filter B Sampling Clock Select)

The NFCSB[1:0] bits set the sampling interval for the noise filter of the GTIOCnB pin. When setting these bits, wait for 2 cycles of the selected sampling interval before setting the input capture function.

Table 21.4 Settings of GTIOA[4:0] and GTIOB[4:0] bits (Saw-wave mode, Triangle-wave mode)

GTIOA/GTIOB[4:0] bits					Function		
b4	b3	b2	b1	b0	b4*4	b3, b2*1 *2 *3 *4	b1, b0*2
0	0	0	0	0	Initial output is low	Output retained at cycle end	Output retained at GTCCRA/GTCCRB compare match
0	0	0	0	1			Low output at GTCCRA/GTCCRB compare match
0	0	0	1	0			High output at GTCCRA/GTCCRB compare match
0	0	0	1	1			Output toggled at GTCCRA/GTCCRB compare match
0	0	1	0	0		Low output at cycle end	Output retained at GTCCRA/GTCCRB compare match
0	0	1	0	1			Low output at GTCCRA/GTCCRB compare match
0	0	1	1	0			High output at GTCCRA/GTCCRB compare match
0	0	1	1	1			Output toggled at GTCCRA/GTCCRB compare match
0	1	0	0	0		High output at cycle end	Output retained at GTCCRA/GTCCRB compare match
0	1	0	0	1			Low output at GTCCRA/GTCCRB compare match
0	1	0	1	0			High output at GTCCRA/GTCCRB compare match
0	1	0	1	1			Output toggled at GTCCRA/GTCCRB compare match
0	1	1	0	0		Output toggled at cycle end	Output retained at GTCCRA/GTCCRB compare match
0	1	1	0	1			Low output at GTCCRA/GTCCRB compare match
0	1	1	1	0			High output at GTCCRA/GTCCRB compare match
0	1	1	1	1			Output toggled at GTCCRA/GTCCRB compare match
1	0	0	0	0	Initial output is high	Output retained at cycle end	Output retained at GTCCRA/GTCCRB compare match
1	0	0	0	1			Low output at GTCCRA/GTCCRB compare match
1	0	0	1	0			High output at GTCCRA/GTCCRB compare match
1	0	0	1	1			Output toggled at GTCCRA/GTCCRB compare match
1	0	1	0	0		Low output at cycle end	Output retained at GTCCRA/GTCCRB compare match
1	0	1	0	1			Low output at GTCCRA/GTCCRB compare match
1	0	1	1	0			High output at GTCCRA/GTCCRB compare match
1	0	1	1	1			Output toggled at GTCCRA/GTCCRB compare match
1	1	0	0	0		High output at cycle end	Output retained at GTCCRA/GTCCRB compare match
1	1	0	0	1			Low output at GTCCRA/GTCCRB compare match
1	1	0	1	0			High output at GTCCRA/GTCCRB compare match
1	1	0	1	1			Output toggled at GTCCRA/GTCCRB compare match
1	1	1	0	0		Output toggled at cycle end	Output retained at GTCCRA/GTCCRB compare match
1	1	1	0	1			Low output at GTCCRA/GTCCRB compare match
1	1	1	1	0			High output at GTCCRA/GTCCRB compare match
1	1	1	1	1			Output toggled at GTCCRA/GTCCRB compare match

Note 1. The cycle end means an overflow (GTCNT changes from GTPR to 0 in up-counting), an underflow (GTCNT changes from 0 to GTPR in down-counting), or counter clearing for saw-wave mode, and means a trough (GTCNT changes from 0 to 1) for triangle-wave mode.

Note 2. When the timing of a cycle end and the timing of a GTCCRA/GTCCRB compare match are the same in a compare-match operation, if the OAEOCD and OBEOCD bits are set to 0 and the end of cycle output is enabled, the b3 and b2 settings are given priority in saw-wave PWM mode, and the b1 and b0 settings are given priority in any other mode.
Note 3. In event count operation where at least one bit in GTUPSR or GTDNSR is set to 1, the setting of b3 and b2 is ignored.(GPT320 to GPT323)
Note 4. In Saw-wave PWM mode 2, GTIOB[4:2] bits is invalid. Since only GTIOCnA pins are output pins, set GTIOA[4] bit for initial output. Set GTIOA[3:2] bits for output at the end of a cycle.

Table 21.5 Settings of GTIOA[4:0] and GTIOB[4:0] bits (Complementary PWM mode)

GTIOA/GTIOB[4:0] bits*1 *2 *3					Function		
					Initial output, Active level	Up count Compare Match output	Down count Compare Match output
b4	b3	b2	b1	b0	b4	b3, b2	b1, b0
0	0	1	1	0	Initial output is Low. Active level is High.	Low output	High output
0	1	0	0	1		High output	Low output
1	0	1	1	0	Initial output is High. Active level is Low.	Low output	High output
1	1	0	0	1		High output	Low output

Note 1. In complementary PWM mode, the only values that can be set in the GTIOA[4:0] bits are 01001b, and 10110b. Setting other values is prohibited.
Note 2. In complementary PWM mode, the only values that can be set in the GTIOB[4:0] bits are 00110b, and 11001b. Setting other values is prohibited.
Note 3. In complementary PWM mode, setting GTIOB[4:0] bits does not use compare match of GTCCRB register. The combination of counter and register that is the target of compare match depends on the operation period of complementary PWM mode. For details, see section 21.3.3.7. Complementary PWM mode 1,2,3.

21.2.15 GTINTAD : General PWM Timer Interrupt Output Setting Register

Base address: GPT32n = 0x4016_9000 + 0x0100 × n (n = 0 to 9)

Offset address: 0x38

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	GRPABL	GRPABH	GRPDTE	—	—	GRP[1:0]		—	—	—	—	ADTRBDEN	ADTRBUEN	ADTRADEN	ADTRAUEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SCFPU	SCFPO	SCFF	SCFE	SCFD	SCFC	SCFB	SCFA	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0. The write value should be 0.	R/W
8	SCFA	GTCCRA Register Compare Match/Input Capture Source Synchronous Clear Enable 0: Disable use of GTCCRA register compare match/input capture as a clear factor for other channels. 1: Enable use of GTCCRA register compare match/input capture as a clear factor for other channels.	R/W
9	SCFB	GTCCRB Register Compare Match/Input Capture Source Synchronous Clear Enable 0: Disable use of GTCCRB register compare match/input capture as a clear factor for other channels. 1: Enable use of GTCCRB register compare match/input capture as a clear factor for other channels.	R/W
10	SCFC	GTCCRC Register Compare Match Source Synchronous Clear Enable 0: Disable use of GTCCRC register compare match as a clear factor for other channels. 1: Enable use of GTCCRC register compare match as a clear factor for other channels.	R/W

Bit	Symbol	Function	R/W
11	SCFD	GTCCRD Register Compare Match Source Synchronous Clear Enable 0: Disable use of GTCCRD register compare match as a clear factor for other channels. 1: Enable use of GTCCRD register compare match as a clear factor for other channels.	R/W
12	SCFE	GTCCRE Register Compare Match Source Synchronous Clear Enable 0: Disable use of GTCCRE register compare match as a clear factor for other channels. 1: Enable use of GTCCRE register compare match as a clear factor for other channels	R/W
13	SCFF	GTCCRF Register Compare Match Source Synchronous Clear Enable 0: Disable use of GTCCRF register compare match as a clear factor for other channels. 1: Enable use of GTCCRF register compare match as a clear factor for other channels	R/W
14	SCFPO	Overflow Source Synchronous Clear Enable 0: Disable use of overflow as a clear factor for other channels. 1: Enable use of overflow as a clear factor for other channels.	R/W
15	SCFPU	Underflow Source Synchronous Clear Enable 0: Disable use of underflow as a clear factor for other channels 1: Enable use of underflow as a clear factor for other channels	R/W
16	ADTRAUEN	GTADTRA Register Compare Match (Up-Counting) A/D Conversion Start Request Enable 0: A/D conversion start request is disabled. 1: A/D conversion start request is enabled.	R/W
17	ADTRADEN	GTADTRA Register Compare Match (Down-Counting) A/D Conversion Start Request Enable 0: A/D conversion start request is disabled. 1: A/D conversion start request is enabled.	R/W
18	ADTRBUEN	GTADTRB Register Compare Match (Up-Counting) A/D Conversion Start Request Enable 0: A/D conversion start request is disabled. 1: A/D conversion start request is enabled.	R/W
19	ADTRBDEN	GTADTRB Register Compare Match (Down-Counting) A/D Conversion Start Request Enable 0: A/D conversion start request is disabled. 1: A/D conversion start request is enabled.	R/W
23:20	—	These bits are read as 0. The write value should be 0.	R/W
25:24	GRP[1:0]	Output Disable Source Select 0 0: Group A output disable source is selected 0 1: Group B output disable source is selected 1 0: Group C output disable source is selected 1 1: Group D output disable source is selected	R/W
27:26	—	These bits are read as 0. The write value should be 0.	R/W
28	GRPDTE	Dead Time Error Output Disable Request Enable 0: Dead time error output disable request is disabled. 1: Dead time error output disable request is enabled.	R/W
29	GRPABH	Same Time Output Level High Disable Request Enable 0: Same time output level high disable request disabled 1: Same time output level high disable request enabled	R/W
30	GRPABL	Same Time Output Level Low Disable Request Enable 0: Same time output level low disable request disabled 1: Same time output level low disable request enabled	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

The GTINTAD enables or disables interrupt requests, A/D conversion start request, and output disable requests.

SCFA bit (GTCCRA Register Compare Match/Input Capture Source Synchronous Clear Enable)

This bit enables or disables use of GTCCRA register compare match/input capture as a clear factor for other channels.

SCFB bit (GTCCRB Register Compare Match/Input Capture Source Synchronous Clear Enable)

This bit enables or disables use of GTCCRB register compare match/input capture as a clear factor for other channels.

The setting is invalid in complementary PWM mode.

SCFC bit (GTCCRC Register Compare Match Source Synchronous Clear Enable)

This bit enables or disables use of GTCCRC register compare match as a clear factor for other channels.

The setting is invalid in complementary PWM mode.

SCFD bit (GTCCRD Register Compare Match Source Synchronous Clear Enable)

This bit enables or disables use of GTCCRD register compare match as a clear factor for other channels.

The setting is invalid in complementary PWM mode.

SCFE bit (GTCCRE Register Compare Match Source Synchronous Clear Enable)

This bit enables or disables use of GTCCRE register compare match as a clear factor for other channels.

The setting is invalid in complementary PWM mode.

SCFF bit (GTCCRF Register Compare Match Source Synchronous Clear Enable)

This bit enables or disables use of GTCCRF register compare match as a clear factor for other channels.

The setting is invalid in complementary PWM mode.

SCFPO bit (Overflow Source Synchronous Clear Enable)

This bit enables or disables use of overflow as a clear factor for other channels.

In complementary PWM mode, it is valid only for the master channel.

SCFPU bit (Underflow Source Synchronous Clear Enable)

This bit enables or disables use of underflow as a clear factor for other channels.

In complementary PWM mode, it is valid only for the master channel.

ADTRAUEN bit (GTADTRA Register Compare Match (Up-Counting) A/D Conversion Start Request Enable)

This bit enables or disables A/D conversion start requests generated by GTADTRA register compare matches during GTCNT counter up-counting.

In complementary PWM mode, it is valid only for the master channel.

The setting is invalid during the event count operation, and A/D conversion start request is not generated.

ADTRADEN bit (GTADTRA Register Compare Match (Down-Counting) A/D Conversion Start Request Enable)

This bit enables or disables A/D conversion start requests generated by GTADTRA register compare matches during GTCNT counter down-counting.

In complementary PWM mode, it is valid only for the master channel.

The setting is invalid during the event count operation, and A/D conversion start request is not generated.

ADTRBUEN bit (GTADTRB Register Compare Match (Up-Counting) A/D Conversion Start Request Enable)

This bit enables or disables A/D conversion start requests generated by GTADTRB register compare matches during GTCNT counter up-counting.

In complementary PWM mode, it is valid only for the master channel.

The setting is invalid during the event count operation, and A/D conversion start request is not generated.

ADTRBDEN bit (GTADTRB Register Compare Match (Down-Counting) A/D Conversion Start Request Enable)

This bit enables or disables A/D conversion start requests generated by GTADTRB register compare matches during GTCNT counter down-counting.

In complementary PWM mode, it is valid only for the master channel.

The setting is invalid during the event count operation, and A/D conversion start request is not generated.

GRP[1:0] bits (Output Disable Source Select)

These bits select the group of output disable request from GPT to POEG and the group of output disable for GTIOCnA pin and GTIOCnB pin from POEG to GPT.

The output disable request to POEG is output to the group selected in the GRP[1:0] bit, with dead-time errors, simultaneous high output, and simultaneous low output factors following their respective disable request enable bits.

GTST.ODF shows the request of the output disable source group that is selected with the GRP[1:0] bits. Set the GRP[1:0] bits when both GTIOR.OAE and GTIOR.OBE bits are 0.

GRPDTE bit (Dead Time Error Output Disable Request Enable)

This bit enables or disables the output disable request by a dead time error.

The dead time error output disable request is not generated during the event count operation.

GRPABH bit (Same Time Output Level High Disable Request Enable)

The GRPABH bit enables or disables the output disable request when the GTIOCnA pin and GTIOCnB pin output 1 at the same time.

GRPABL bit (Same Time Output Level Low Disable Request Enable)

The GRPABL bit enables or disables the output disable request when the GTIOCnA pin and GTIOCnB pin output 0 at the same time.

21.2.16 GTST : General PWM Timer Status Register

Base address: GPT32n = 0x4016_9000 + 0x0100 × n (n = 0 to 9)

Offset address: 0x3C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	PCF	OABLF	OABHF	DTEF	—	—	—	ODF	—	—	—	—	ADTRBDF	ADTRBUF	ADTRADF	ADTRAUF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TUCF	—	—	—	—	ITCNT[2:0]			TCFPU	TCFPO	TCFF	TCFE	TCFD	TCFC	TCFB	TCFA
Value after reset:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TCFA	Input Capture/Compare Match Flag A 0: No input capture/compare match of GTCCRA is generated 1: An input capture/compare match of GTCCRA is generated	R/W*1
1	TCFB	Input Capture/Compare Match Flag B 0: No input capture/compare match of GTCCRB is generated 1: An input capture/compare match of GTCCRB is generated	R/W*1
2	TCFC	Input Compare Match Flag C 0: No compare match of GTCCRC is generated 1: A compare match of GTCCRC is generated	R/W*1
3	TCFD	Input Compare Match Flag D 0: No compare match of GTCCRD is generated 1: A compare match of GTCCRD is generated	R/W*1

Bit	Symbol	Function	R/W
4	TCFE	Input Compare Match Flag E 0: No compare match of GTCCRE is generated 1: A compare match of GTCCRE is generated	R/W*1
5	TCFF	Input Compare Match Flag F 0: No compare match of GTCCRF is generated 1: A compare match of GTCCRF is generated	R/W*1
6	TCFPO	Overflow Flag 0: No overflow (crest) occurred 1: An overflow (crest) occurred	R/W*1
7	TCFPU	Underflow Flag 0: No underflow (trough) occurred 1: An underflow (trough) occurred	R/W*1
10:8	ITCNT[2:0]	GPTn_OVF/GPTn_UDF Interrupt Skipping Count Counter	R
14:11	—	These bits are read as 0. The write value should be 0.	R/W
15	TUCF	Count Direction Flag 0: GTCNT counter counts downward 1: GTCNT counter counts upward	R
16	ADTRAUF	GTADTRA Register Compare Match (Up-Counting) A/D Conversion Start Request Flag 0: No GTADTRA register compare match has occurred in up-counting. 1: A GTADTRA register compare match has occurred in up-counting.	R/W*1
17	ADTRADF	GTADTRA Register Compare Match (Down-Counting) A/D Conversion Start Request Flag 0: No GTADTRA register compare match has occurred in down-counting. 1: A GTADTRA register compare match has occurred in down-counting.	R/W*1
18	ADTRBUF	GTADTRB Register Compare Match (Up-Counting) A/D Conversion Start Request Flag 0: No GTADTRB register compare match has occurred in up-counting. 1: A GTADTRB register compare match has occurred in up-counting.	R/W*1
19	ADTRBDF	GTADTRB Register Compare Match (Down-Counting) A/D Conversion Start Request Flag 0: No GTADTRB register compare match has occurred in down-counting. 1: A GTADTRB register compare match has occurred in down-counting.	R/W*1
23:20	—	These bits are read as 0. The write value should be 0.	R/W
24	ODF	Output Disable Flag 0: No output disable request is generated 1: An output disable request is generated	R
27:25	—	These bits are read as 0. The write value should be 0.	R/W
28	DTEF	Dead Time Error Flag 0: No dead time error has occurred. 1: A dead time error has occurred.	R
29	OABHF	Same Time Output Level High Flag 0: No simultaneous generation of 1 both for the GTIOCA and GTIOCB pins has occurred. 1: A simultaneous generation of 1 both for the GTIOCA and GTIOCB pins has occurred.	R
30	OABLF	Same Time Output Level Low Flag 0: No simultaneous generation of 0 both for the GTIOCA and GTIOCB pins has occurred. 1: A simultaneous generation of 0 both for the GTIOCA and GTIOCB pins has occurred.	R
31	PCF*2	Period Count Function Finish Flag 0: No period count function finish has occurred 1: A period count function finish has occurred	R/W*1

Note 1. Only 0 can be written to this bit. Do not write 1.
When clearing the ADTRAUF, ADTRADF, ADTRBUF, or ADTRBDF flag, be sure to write 0 only to the target flag or flags for clearing and to write 1 to the other flags not for clearing.
Note 2. This bit is only available in GPT320 to GPT323.
In GPT324 to GPT329, this bit is read as 0. The write value should be 0.

The GTST indicates the status of the GPT.

TCFA flag (Input Capture/Compare Match Flag A)

The TCFA flag indicates the status for the input capture or compare match of GTCCRA.

[Setting conditions]

- GTCNT = GTCCRA, when the GTCCRA register functions as a compare match register
- GTCNT counter value is transferred to GTCCRA by the input capture signal when the GTCCRA register functions as an input capture register.

[Clearing condition]

- 0 is written to this flag.

TCFB flag (Input Capture/Compare Match Flag B)

The TCFB flag indicates the status for the input capture or compare match of GTCCRB.

[Setting conditions]

- GTCNT = GTCCRB, when the GTCCRB register functions as a compare match register
- GTCNT counter value is transferred to GTCCRB by the input capture signal when the GTCCRB register functions as an input capture register.

[Clearing condition]

- 0 is written to this flag.

TCFC flag (Input Compare Match Flag C)

The TCFC flag indicates the status for the compare match of GTCCRC.

When GTCCRC performs buffer operation, GTCCRC does not perform compare match.

[Setting condition]

- GTCNT = GTCCRC.

[Clearing condition]

- 0 is written to this flag.

[Not comparing condition]

- GTCR.MD[2:0] = 001b (saw-wave one-shot pulse mode)
- GTCR.MD[2:0] = 110b (triangle-wave PWM mode 3)
- GTBER.CCRA[1:0] = 01b, 10b, 11b (GTCCRC performs buffer operation).

TCFD flag (Input Compare Match Flag D)

The TCFD flag indicates the status for the compare match of GTCCRD.

When GTCCRD performs buffer operation, GTCCRD does not perform compare match.

[Setting condition]

- GTCNT = GTCCRD.

[Clearing condition]

- 0 is written to this flag.

[Not comparing condition]

- GTCR.MD[2:0] = 001b (saw-wave one-shot pulse mode)
- GTCR.MD[2:0] = 110b (Triangle-wave PWM mode 3)
- GTBER.CCRA[1:0] = 10b, 11b (GTCCRD performs buffer operation).

TCFE flag (Input Compare Match Flag E)

The TCFE flag indicates the status for the compare match of GTCCRE.

When GTCCRE performs buffer operation, GTCCRE does not perform compare match.

[Setting condition]

- GTCNT = GTCCRE.

[Clearing condition]

- 0 is written to this flag.

[Not comparing condition]

- GTCR.MD[2:0] = 001b (saw-wave one-shot pulse mode)
- GTCR.MD[2:0] = 110b (Triangle-wave PWM mode 3)
- GTBER.CCRB[1:0] = 01b, 10b, 11b (GTCCRE performs buffer operation).

TCFF flag (Input Compare Match Flag F)

The TCFF flag indicates the status for the compare match of GTCCRF.

When GTCCRF performs buffer operation, GTCCRF does not perform compare match.

[Setting condition]

- GTCNT = GTCCRF.

[Clearing condition]

- 0 is written to this flag.

[Not comparing condition]

- GTCR.MD[2:0] = 001b (saw-wave one-shot pulse mode)
- GTCR.MD[2:0] = 110b (Triangle-wave PWM mode 3)
- GTBER.CCRB[1:0] = 10b, 11b (GTCCRF performs buffer operation).

TCFPO flag (Overflow Flag)

The TCFPO flag indicates when an overflow or crest has occurred.

[Setting conditions]

- In saw-wave mode, an overflow (GTCNT changes from GTPR to 0 in up-counting) has occurred
- In triangle-wave mode, a crest (GTCNT changes from GTPR to GTPR - 1) has occurred
- In counting by hardware sources, an overflow (GTCNT changes from GTPR to 0 in up-counting) has occurred.

[Clearing condition]

- 0 is written to this flag.

TCFPU flag (Underflow Flag)

The TCFPU flag indicates when an underflow or trough has occurred.

[Setting conditions]

- In saw-wave mode, an underflow (GTCNT changes from 0 to GTPR in down-counting) has occurred
- In triangle-wave mode, a trough (GTCNT changes from 0 to 1) has occurred
- In counting by hardware sources, an underflow (GTCNT changes from 0 to GTPR in down-counting) has occurred.

[Clearing condition]

- 0 is written to this bit.

ITCNT[2:0] flag (GPTn_OVF/GPTn_UDF Interrupt Skipping Count Counter)

When the GPTn_OVF/GPTn_UDF interrupt skipping function is used (the GTITC.IVTC[1:0] bits are set to a value other than 00b), the counter is incremented by 1 every time the GPTn_OVF/GPTn_UDF interrupt source which is selected in GTITC.IVTC[1:0] is generated.

These bits are operated independently from the extended interrupt skipping by the GTEITC register.

[Clearing conditions]

- The GPTn_OVF/GPTn_UDF interrupt skipping function is not used (the GTITC.IVTT[2:0] bits are 000b when the IVTC[1:0] bits are 00b).
- The GPTn_OVF/GPTn_UDF interrupt skipping count matches the specified count (the ITCNT[2:0] bits match the skipping count specified by the IVTT[2:0] bits).
- When the count operation is stopped

TUCF flag (Count Direction Flag)

The TUCF flag indicates the count direction of GTCNT. In event count operation, this flag is set to 1 in up-counting and to 0 in down-counting.

ADTRAUF flag (GTADTRA Register Compare Match (Up-Counting) A/D Conversion Start Request Flag)

This status flag indicates generation of a GTADTRA register compare match in up-counting.

[Setting condition]

- The GTCNT counter matches the GTADTRA register in up-counting.

[Clearing condition]

- 0 is written to the ADTRAUF flag.

ADTRADF flag (GTADTRA Register Compare Match (Down-Counting) A/D Conversion Start Request Flag)

This status flag indicates generation of a GTADTRA register compare match in down-counting.

[Setting condition]

- The GTCNT counter matches the GTADTRA register in down-counting.

[Clearing condition]

- 0 is written to the ADTRADF flag.

ADTRBUF flag (GTADTRB Register Compare Match (Up-Counting) A/D Conversion Start Request Flag)

This status flag indicates generation of a GTADTRB register compare match in up-counting.

[Setting condition]

- The GTCNT counter matches the GTADTRB register in up-counting.

[Clearing condition]

- 0 is written to the ADTRBUF flag.

ADTRBDF flag (GTADTRB Register Compare Match (Down-Counting) A/D Conversion Start Request Flag)

This status flag indicates generation of a GTADTRB register compare match in down-counting.

[Setting condition]

- The GTCNT counter matches the GTADTRB register in down-counting.

[Clearing condition]

- 0 is written to the ADTRBDF flag.

ODF flag (Output Disable Flag)

The ODF flag shows the request of the output disable source group that is selected in the GRP[1:0] bits.

When output is disabled, an output disable control is not released within the same cycle in which an output disable request is negated. It is released in the next cycle.

DTEF flag (Dead Time Error Flag)

This flag indicates that the timer output toggle point after the automatic addition of dead time has exceeded the count period.

This flag returns to 0 when the timer output toggle point after the automatic addition of dead time is back within the period.

This flag can only be read from. (Writing 0 to clear the flag is not allowed.)

When the output disable request by the DTEF flag is enabled (when GTINTAD.GRPDTE bit is 1), the DTEF flag is output as the output disable request to the POEG. The GPT does not have a dead time error interrupt. Use the interrupt function in the POEG if this is necessary.

[Setting condition]

- When a change point of the waveform after the automatic setting of dead time has exceeded the count period (in the following cases).
  - Up-counting in triangle-wave mode:
    GTCCRA register – GTDVU register ≤ 0
  - Down-counting in triangle-wave mode:
    GTCCRA register – GTDVD register < 0
  - Up-counting in saw-wave one-shot pulse mode:
    GTCCRA register – GTDVU register < 0, or
    GTCCRA register + GTDVD register > GTPR register
  - Down-counting in saw-wave one-shot pulse mode:
    GTCCRA register + GTDVU register > GTPR register, or
    GTCCRA register - GTDVD register < 0

[Clearing condition]

- The timer output toggle point after the automatic addition of dead time is within the count period.

OABHF flag (Same Time Output Level High Flag)

The OABHF flag indicates that the GTIOCnA pin and GTIOCnB pin output 1 at the same time.

When the GTIOCnA or GTIOCnB pin outputs 0, this flag returns to 0. This flag is read only. Writing 0 to clear the flag is prohibited.

When the output disable request by the OABHF flag is enabled (GTINTAD.GRPABH = 1), the OABHF flag is output to POEG as an output disable request. The GPT does not have an interrupt to indicate that outputs have been simultaneous driven to the high level. Use the interrupt function in the POEG if this is necessary.

[Setting condition]

- The GTIOCnA and GTIOCnB pins output 1 at the same time when both OAE and OBE bits are set to 1.

[Clearing conditions]

- The GTIOCnA pin output value is different from the GTIOCnB pin output value when both OAE and OBE bits are set to 1
- The GTIOCnA and GTIOCnB pins output 0 at the same time when both OAE and OBE bits are set to 1
- Either the OAE bit or OBE bit is set to 0.

OABLF flag (Same Time Output Level Low Flag)

The OABLF flag indicates that the GTIOCnA and GTIOCnB pins output 0 at the same time.

When the GTIOCnA pin or GTIOCnB pin outputs 1, this flag returns to 0. This flag is read only. Writing 0 to clear the flag is prohibited.

When the output disable request by the OABLF flag is enabled (GTINTAD.GRPABL = 1), the OABLF flag is output to POEG as an output disable request. The GPT does not have an interrupt to indicate that outputs have been simultaneous driven to the low level. Use the interrupt function in the POEG if this is necessary.

[Setting condition]

- The GTIOCnA and GTIOCnB pins output 0 at the same time when both OAE and OBE bits are set to 1.

[Clearing conditions]

- The GTIOCnA pin output value is different from the GTIOCnB pin output value when both OAE and OBE bits are set to 1
- The GTIOCnA and GTIOCnB pins output 1 at the same time when both OAE and OBE bits are set to 1
- Either the OAE bit or the OBE bit is set to 0.

The compare-target signals to generate the OABHF/OABLF flag are the compare match outputs (PWM outputs) signals before they are masked by the output disable function. Even during the output disable condition, compare match operation continues internally, where the OABHF or OABLF flag is updated based on the operation results.

PCF flag (Period Count Function Finish Flag)

This bit is status flag of period count function finish.

[Setting condition]

- The GTPC.PCEN bit is 1 and the GTPC.PCNT counter is 1 at the end of cycle.
- The GTPC.PCEN bit is 1 and the GTPC.PCNT counter is 0 at the count clock.

[Clearing condition]

- 0 is written to this bit.

21.2.17 GTBER : General PWM Timer Buffer Enable Register

Base address: GPT32n = 0x4016_9000 + 0x0100 × n (n = 0 to 9)

Offset address: 0x40

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	ADTDB	ADTTB[1:0]		—	ADTDA	ADTTA[1:0]		—	CCRSWT	PR[1:0]		CCRB[1:0]		CCRA[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	DBRTECB	—	DBRTECA	—	—	—	—	BD3	BD2	BD1	BD0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BD0	GTCCR Buffer Operation Disable 0: Buffer operation is enabled 1: Buffer operation is disabled	R/W
1	BD1	GTPR Buffer Operation Disable 0: Buffer operation is enabled 1: Buffer operation is disabled	R/W*1
2	BD2	GTADTRA/GTADTRB Registers Buffer Operation Disable 0: Buffer operation is enabled 1: Buffer operation is disabled	R/W
3	BD3	GTDVU/GTDVD Registers Buffer Operation Disable 0: Buffer operation is enabled 1: Buffer operation is disabled	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
8	DBRTECA	GTCCRA Register Double Buffer Repeat Operation Enable 0: GTCCRA register double buffer repeat operation is disabled 1: GTCCRA register double buffer repeat operation is enabled	R/W
9	—	This bit is read as 0. The write value should be 0.	R/W
10	DBRTECB	GTCCRB Register Double Buffer Repeat Operation Enable 0: GTCCRB register double buffer repeat operation is disabled 1: GTCCRB register double buffer repeat operation is enabled	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W
17:16	CCRA[1:0]	GTCCRA Buffer Operation 0 0: No buffer operation 0 1: Single buffer operation (GTCCRA ↔GTCCRC) Others: Double buffer operation (GTCCRA ↔ GTCCRC ↔ GTCCRD)	R/W
19:18	CCRB[1:0]	GTCCRB Buffer Operation 0 0: No buffer operation 0 1: Single buffer operation (GTCCRB ↔ GTCCRE) Others: Double buffer operation (GTCCRB ↔ GTCCRE ↔ GTCCRF)	R/W
21:20	PR[1:0]	GTPR Buffer Operation 0 0: No buffer operation 0 1: Single buffer operation (GTPBR → GTPR) Others: Double buffer operation (GTPDBR → GTPBR → GTPR)	R/W
22	CCRSWT	GTCCRA and GTCCRB Forcible Buffer Operation Writing 1 to this bit forces a buffer transfer of GTCCRA and GTCCRB. This bit automatically returns to 0 after 1 is written. This bit is read as 0.	W
23	—	This bit is read as 0. The write value should be 0.	R/W
25:24	ADTTA[1:0]	GTADTRA Register Buffer Transfer Timing Select 0 0: In triangle wave or complementary PWM mode, no transfer. In saw-wave mode, no transfer. 0 1: In triangle wave or complementary PWM mode, transfer at crest. In saw-wave mode, transfer at underflow (in down-counting), overflow (in up-counting), or counter clearing. 1 0: In triangle wave or complementary PWM mode, transfer at trough. In saw-wave mode, transfer at underflow (in down-counting), overflow (in up-counting), or counter clearing. 1 1: In triangle wave or complementary PWM mode, transfer at both crest and trough. In saw-wave mode, transfer at underflow (in down-counting), overflow (in up-counting), or counter clearing.	R/W
26	ADTDA	GTADTRA Register Double Buffer Operation 0: Single buffer operation (GTADTBRA → GTADTRA) 1: Double buffer operation (GTADTDBRA → GTADTBRA → GTADTRA)	R/W
27	—	This bit is read as 0. The write value should be 0.	R/W
29:28	ADTTB[1:0]	GTADTRB Register Buffer Transfer Timing Select 0 0: In triangle wave or complementary PWM mode, no transfer. In saw-wave mode, no transfer. 0 1: In triangle wave or complementary PWM mode, transfer at crest. In saw-wave mode, transfer at underflow (in down-counting), overflow (in up-counting), or counter clearing. 1 0: In triangle wave or complementary PWM mode, transfer at trough. In saw-wave mode, transfer at underflow (in down-counting), overflow (in up-counting), or counter clearing. 1 1: In triangle wave or complementary PWM mode, transfer at both crest and trough. In saw-wave mode, transfer at underflow (in down-counting), overflow (in up-counting), or counter clearing.	R/W
30	ADTDB	GTADTRB Register Double Buffer Operation 0: Single buffer operation (GTADTBRB → GTADTRB) 1: Double buffer operation (GTADTDBRB → GTADTBRB → GTADTRB)	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

Note 1. When in complementary PWM mode, no matter which register of master channel/slave channel 1/slave channel 2 is written, three channels are written at the same time.

The GTBER register provides settings for the buffer operation. Set the GTBER register except the BDx (x = 0 to 3) bits while the GTCNT counter is stopped.

BD0 bit (GTCCR Buffer Operation Disable)

The BD0 bit disables the buffer operation using GTCCRA, GTCCRB, GTCCRC, GTCCRD, GTCCRE, and GTCCRF combined.

When GTDTCR.TDE is 1 and when BD0 is set to 0, GTCCRB does not perform buffer operation in Saw-wave one-shot pulse mode or Triangle-wave PWM mode. The GTCCRB register is automatically set to a compare match value for negative-phase waveform with dead time.

In complementary PWM mode, it is valid only for the buffer operation of GTCCRC register and GTCCRE register. The buffer operation of GTCCRA registers cannot be disabled. Buffer operation of GTCCRE register and GTCCRF register is enabled/disabled by CP3DB bit of GTBER2 register. No buffer transfer to GTCCRB is performed in complementary PWM mode.

A value for the BD0 bit in the channel related to the position of the bit written with 1 by the GTSECSR register can be set when 1 is written to the GTSECR.SBDCE or GTSECR.SBDCD.

When the DBRTECm (m = A, B) bit is 1, setting the BD0 bit to 1 while the mode of operation is saw-wave one-shot pulse mode or triangle-wave PWM mode 3 results in transfer from the intermediate buffer to the GTCCRm register.

BD1 bit (GTPR Buffer Operation Disable)

The BD1 bit disables the buffer operation using GTPR, GTPDBR, and GTPBR combined.

A value for the BD1 bit in the channel related to the position of the bit written with 1 by the GTSECSR register can be set when 1 is written to the GTSECR.SBDPE or GTSECR.SBDPD.

In complementary PWM mode, the slave channel is also controlled by setting the BD1 bit of the master channel.

BD2 bit (GTADTRA/GTADTRB Registers Buffer Operation Disable)

This bit disables buffer operation using the GTADTRA, GTADTBRA, and GTADTDBRA registers together and buffer operation using the GTADTRB, GTADTBRB, and GTADTDBRB registers together.

The setting is invalid during the event count operation, and the buffer operation using the GTADTRA and GTADTRB registers is not performed.

A value for the BD2 bit in the channel related to the position of the bit written with 1 by the GTSECSR register can be set when 1 is written to the SBDAE or SBDAD bit in the GTSECR register.

BD3 bit (GTDVU/GTDVD Registers Buffer Operation Disable)

In Saw-wave PWM mode 1, Saw-wave one-shot pulse mode or Triangle-wave PWM mode, this bit disables buffer operation using the GTDVU and GTDBU registers together and buffer operation using the GTDVD and GTDBD registers together.

Even though the BD3 bit is set to 0, buffer operation in the GTDVD register is not performed if the GTDTCR.TDFER bit is set to 1. Instead, the value in the GTDVU register is set automatically.

In Saw-wave PWM mode 2 or complementary PWM mode, this bit is invalid and GTDVU and GTDVD registers don’t perform buffer operation.

The setting is invalid during the event count operation, and the buffer operation using the GTDVU and GTDVD registers is not performed.

A value for the BD3 bit in the channel related to the position of the bit written with 1 by the GTSECSR register can be set when 1 is written to the SBDDE or SBDDD bit in the GTSECR register.

DBRTECm bits (GTCCRm Register Double Buffer Repeat Operation Enable) (m = A,B)

This setting enables the operation to repeat a transfer to the GTCCRm register from the intermediate buffer by cycle during the buffer transfer disable period when performing the double buffer operation using the GTCCRm register.

It is valid in saw-wave one-shot pulse mode and triangle-wave PWM mode 3.

The disabling period of buffer transfer indicates the period that buffer transfer is stopped by setting of the BD0 bit (CPU writing or simultaneous buffer operation control by the GTSECSR register) and the period for buffer transfer extended skipping (except the case for skipping by counting both crests and troughs) by the GTEITLB register.

When the DBRTECm bit is 1, writing by the CPU to the GTCCRm register sets the same value in temporary register x (x = C, E). The value of the GTCCRx (x = C, E) register is also transferred to the temporary register x (x = C, E) by forcible buffer transfer.

CCRA[1:0] bits (GTCCRA Buffer Operation)

The CCRA[1:0] bits set the buffer operation with GTCCRA, GTCCRC, and GTCCRD combined. When the buffer operation is restricted by the operating mode set in GTCR, the GTCR setting is given priority.

The buffer operation mode is fixed in saw-wave one-shot pulse mode or triangle-wave PWM mode 3 (64-bit transfer at trough), or compementary PWM mode.

CCRB[1:0] bits (GTCCRB Buffer Operation)

The CCRB[1:0] bits set the buffer operation using GTCCRB, GTCCRE, and GTCCRF combined. When the buffer operation is restricted by the operating mode set in GTCR, the GTCR setting is given priority.

The buffer operation mode is fixed in saw-wave one-shot pulse mode or triangle-wave PWM mode 3 (64-bit transfer at trough), or compementary PWM mode.

PR[1:0] bits (GTPR Buffer Operation)

The PR[1:0] bits set the buffer operation with GTPR, GTPDBR, and GTPBR combined.

The setting is invalid in complementary PWM mode. Buffer operation unique to complementary PWM mode is performed regardless of the setting value of the PR[1:0] bits.

CCRSWT bit (GTCCRA and GTCCRB Forcible Buffer Operation)

Writing 1 to the CCRSWT bit forces a buffer transfer of GTCCRA and GTCCRB. This bit automatically returns to 0 after the 1 is written. This bit is read as 0, and is valid only when counting is stopped with a compare match operation specified.

The setting is invalid in complementary PWM mode.

ADTTA[1:0] bits (GTADTRA Register Buffer Transfer Timing Select)

These bits set the transfer timing for buffer operation of the GTADTRA, GTADTBRA, and GTADTDBRA registers.

The setting is invalid during the event count operation.

ADTDA bit (GTADTRA Register Double Buffer Operation)

These bits set buffer operation with the GTADTRA, GTADTBRA, and GTADTDBRA registers combined.

The setting is invalid during the event count operation.

ADTTB[1:0] bits (GTADTRB Register Buffer Transfer Timing Select)

These bits set the transfer timing for buffer operation of the GTADTRB, GTADTBRB, and GTADTDBRB registers.

The setting is invalid during the event count operation.

ADTDB bit (GTADTRB Register Double Buffer Operation)

These bits set buffer operation with the GTADTRB, GTADTBRB, and GTADTDBRB registers combined.

The setting is invalid during the event count operation.

21.2.18 GTITC : General PWM Timer Interrupt and A/D Conversion Start Request Skipping Setting Register

Base address: GPT32n = 0x4016_9000 + 0x0100 × n (n = 0 to 9)

Offset address: 0x44

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	ADTBL	—	ADTAL	—	IVTT[2:0]			IVTC[1:0]		ITLF	ITLE	ITLD	ITLC	ITLB	ITLA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ITLA	GTCCRA Register Compare Match/Input Capture Interrupt Link 0: Not linked with GPTn_OVF/GPTn_UDF interrupt skipping function. 1: Linked with GPTn_OVF/GPTn_UDF interrupt skipping function.	R/W
1	ITLB	GTCCRB Register Compare Match/Input Capture Interrupt Link 0: Not linked with GPTn_OVF/GPTn_UDF interrupt skipping function. 1: Linked with GPTn_OVF/GPTn_UDF interrupt skipping function.	R/W
2	ITLC	GTCCRC Register Compare Match Interrupt Link 0: Not linked with GPTn_OVF/GPTn_UDF interrupt skipping function. 1: Linked with GPTn_OVF/GPTn_UDF interrupt skipping function.	R/W
3	ITLD	GTCCRD Register Compare Match Interrupt Link 0: Not linked with GPTn_OVF/GPTn_UDF interrupt skipping function. 1: Linked with GPTn_OVF/GPTn_UDF interrupt skipping function.	R/W
4	ITLE	GTCCRE Register Compare Match Interrupt Link 0: Not linked with GPTn_OVF/GPTn_UDF interrupt skipping function. 1: Linked with GPTn_OVF/GPTn_UDF interrupt skipping function.	R/W
5	ITLF	GTCCRF Register Compare Match Interrupt Link 0: Not linked with GPTn_OVF/GPTn_UDF interrupt skipping function. 1: Linked with GPTn_OVF/GPTn_UDF interrupt skipping function.	R/W
7:6	IVTC[1:0]	GPTn_OVF/GPTn_UDF Interrupt Skipping Function Select 0 0: Skipping is not performed. 0 1: Both overflow and underflow for saw waves*1 and crest for triangle waves and complementary PWM mode are counted and skipped. 1 0: Both overflow and underflow for saw waves*1 and trough for triangle waves and complementary PWM mode are counted and skipped. 1 1: Both overflow and underflow for saw waves*1 and both crest and trough for triangle waves and complementary PWM mode are counted and skipped.	R/W
10:8	IVTT[2:0]	GPTn_OVF/GPTn_UDF Interrupt Skipping Count Select 0 0 0: Skipping is not performed 0 0 1: Skipping count of 1 0 1 0: Skipping count of 2 0 1 1: Skipping count of 3 1 0 0: Skipping count of 4 1 0 1: Skipping count of 5 1 1 0: Skipping count of 6 1 1 1: Skipping count of 7	R/W
11	—	This bit is read as 0. The write value should be 0.	R/W
12	ADTAL	GTADTRA Register A/D Conversion Start Request Link 0: Not linked with GPTn_OVF/GPTn_UDF interrupt skipping function. 1: Linked with GPTn_OVF/GPTn_UDF interrupt skipping function.	R/W
13	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
14	ADTBL	GTADTRB Register A/D Conversion Start Request Link 0: Not linked with GPTn_OVF/GPTn_UDF interrupt skipping function. 1: Linked with GPTn_OVF/GPTn_UDF interrupt skipping function.	R/W
31:15	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Saw-wave PWM mode 2 is not the target of this function.

The GTITC register sets the skipping function for the GTCNT counter overflow (GTPR compare match) interrupt (GPTn_OVF) and underflow interrupt (GPTn_UDF) and also sets whether to link the other interrupts and A/D conversion start requests with the GPTn_OVF/GPTn_UDF interrupt skipping function. Note the output disable request to POEG cannot be linked with the GPTn_OVF/GPTn_UDF interrupt skipping function. Additionally, if the interrupt skipping function is performed, the change in the status flag is also skipped.

The setting is invalid during the event count operation.

The setting is operated independently from the extended interrupt skipping by the GTEITC register.

ITLA bit (GTCCRA Register Compare Match/Input Capture Interrupt Link)

This bit specifies whether to link the GTCCRA compare match/input capture interrupt (GPTn_CCMPA) with the GPTn_OVF/GPTn_UDF interrupt skipping function.

ITLB bit (GTCCRB Register Compare Match/Input Capture Interrupt Link)

This bit specifies whether to link the GTCCRB compare match/input capture interrupt (GPTn_CCMPB) with the GPTn_OVF/GPTn_UDF interrupt skipping function.

ITLC bit (GTCCRC Register Compare Match Interrupt Link)

This bit specifies whether to link the GTCCRC compare match interrupt (GPTn_CMPC) with the GPTn_OVF/GPTn_UDF interrupt skipping function.

ITLD bit (GTCCRD Register Compare Match Interrupt Link)

This bit specifies whether to link the GTCCRD compare match interrupt (GPTn_CMPD) with the GPTn_OVF/GPTn_UDF interrupt skipping function.

ITLE bit (GTCCRE Register Compare Match Interrupt Link)

This bit specifies whether to link the GTCCRE compare match interrupt (GPTn_CMPE) with the GPTn_OVF/GPTn_UDF interrupt skipping function.

ITLF bit (GTCCRF Register Compare Match Interrupt Link)

This bit specifies whether to link the GTCCRF compare match interrupt (GPTn_CMPF) with the GPTn_OVF/GPTn_UDF interrupt skipping function.

IVTC[1:0] bit (GPTn_OVF/GPTn_UDF Interrupt Skipping Function Select)

These bits set the skipping function for the GTPR compare match (GTCNT counter overflow) interrupt (GPTn_OVF) and GTCNT counter underflow interrupt (GPTn_UDF).

IVTT[2:0] bit (GPTn_OVF/GPTn_UDF Interrupt Skipping Count Select)

These bits set the skipping count for the GTPR compare match (GTCNT counter overflow) interrupt (GPTn_OVF) and GTCNT counter underflow interrupt (GPTn_UDF).

When modifying the IVTT[2:0] bits, first set the IVTC[1:0] bits to 00b.

ADTAL bit (GTADTRA Register A/D Conversion Start Request Link)

This bit specifies whether to link the GTADTRA A/D conversion start request, which is generated in response to a compare match with the GTCNT counter and the GTADTRA register, with the GPTn_OVF/GPTn_UDF interrupt skipping function.

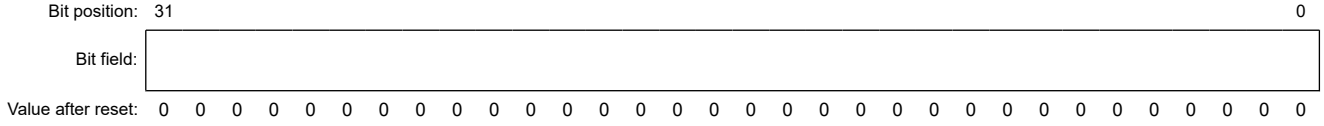
ADTBL bit (GTADTRB Register A/D Conversion Start Request Link)

This bit specifies whether to link the GTADTRB A/D conversion start request, which is generated in response to a compare match with the GTCNT counter and the GTADTRB register, with the GPTn_OVF/GPTn_UDF interrupt skipping function.

21.2.19 GTCNT : General PWM Timer Counter

Base address: GPT32n = 0x4016_9000 + 0x0100 × n (n = 0 to 9)

Offset address: 0x48

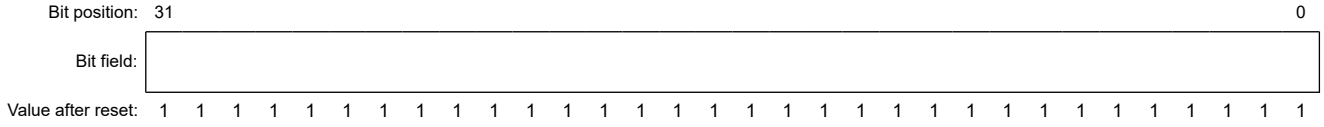


Bit	Symbol	Function	R/W
31:0	n/a	GTCNT is a 32-bit read/write counter for GPT32n (n = 0 to 9). GTCNT can only be written to after counting stops. Access in 8-bit or 16-bit units to the GTCNT counter is prohibited, and it should be accessed in 32-bit units. In saw waves or triangle waves, GTCNT must be set within the range of 0 ≤ GTCNT ≤ GTPR.	R/W

21.2.20 GTCCRk : General PWM Timer Compare Capture Register k (k = A to F)

Base address: GPT32n = 0x4016_9000 + 0x0100 × n (n = 0 to 9)

Offset address: 0x4C (GTCCRA)
0x50 (GTCCRB)
0x54 (GTCCRC)
0x58 (GTCCRE)
0x5C (GTCCRD)
0x60 (GTCCRF)

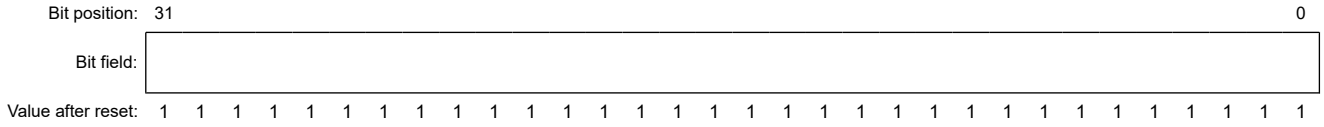


Bit	Symbol	Function	R/W
31:0	n/a	GTCCRk registers are read/write registers. Access in 8-bit or 16-bit units to the GTCCRk register is prohibited, and it should be accessed in 32-bit units. GTCCRA and GTCCRB are registers used for both output compare and input capture. GTCCRC and GTCCRE are compare match registers, and can also function as buffer registers for GTCCRA and GTCCRB. GTCCRD and GTCCRF are compare match registers, and can also function as buffer registers for GTCCRC and GTCCRE (double-buffer registers for GTCCRA and GTCCRB).	R/W

21.2.21 GTPR : General PWM Timer Cycle Setting Register

Base address: GPT32n = 0x4016_9000 + 0x0100 × n (n = 0 to 9)

Offset address: 0x64

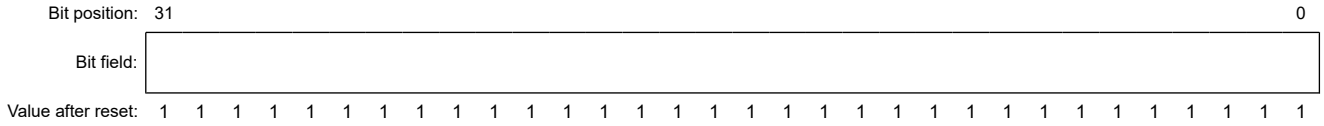


Bit	Symbol	Function	R/W
31:0	n/a	GTPR is a read/write register that sets the maximum count value of GTCNT. Access in 8-bit or 16-bit units to the GTPR register is prohibited, and it should be accessed in 32-bit units. The setting is invalid in the Saw-wave PWM mode 2. For Saw waves except the Saw-wave PWM mode 2, the value of (GTPR + 1) is the cycle. For triangle waves or complementary PWM mode, the value of (GTPR value × 2) is the cycle. When in complementary PWM mode, no matter which register of master channel/slave channel 1/slave channel 2 is written, three channels are written at the same time.	R/W

21.2.22 GTPBR : General PWM Timer Cycle Setting Buffer Register

Base address: GPT32n = 0x4016_9000 + 0x0100 × n (n = 0 to 9)

Offset address: 0x68

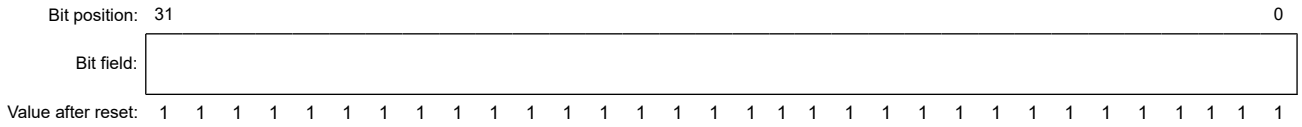


Bit	Symbol	Function	R/W
31:0	n/a	GTPBR is a read/write register that functions as a buffer register for GTPR. Access in 8-bit or 16-bit units to the GTPBR register is prohibited, and it should be accessed in 32-bit units. The setting is invalid in Saw-wave PWM mode 2. When in complementary PWM mode, no matter which register of master channel/slave channel 1/slave channel 2 is written, three channels are written at the same time.	R/W

21.2.23 GTPDBR : General PWM Timer Cycle Setting Double-Buffer Register

Base address: GPT32n = 0x4016_9000 + 0x0100 × n (n = 0 to 9)

Offset address: 0x6C

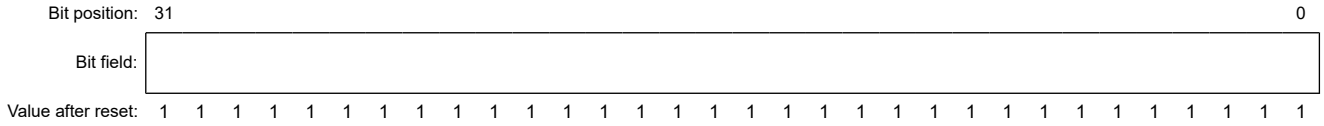


Bit	Symbol	Function	R/W
31:0	n/a	The buffer register for the GTPBR register (double buffer register for the GTPR register) Access in 8-bit or 16-bit units to the GTPDBR register is prohibited, and it should be accessed in 32-bit units. The setting is invalid in Saw-wave PWM mode 2. When in complementary PWM mode, no matter which register of master channel/slave channel 1/slave channel 2 is written, three channels are written at the same time.	R/W

21.2.24 GTADTRk : A/D Conversion Start Request Timing Register k (k = A, B)

Base address: GPT32n = 0x4016_9000 + 0x0100 × n (n = 0 to 9)

Offset address: 0x70 (GTADTRA)
0x7C (GTADTRB)



Bit	Symbol	Function	R/W
31:0	n/a	Set the timing of A/D conversion start request generation Access in 8-bit or 16-bit units to the GTADTRk register is prohibited, and it should be accessed in 32-bit units. When the GTADTRk register value matches the GTCNT counter value, an A/D conversion start request is generated. In complementary PWM mode, A/D conversion start request is generated when the GTCNT counter of the master channel matches this register.	R/W

21.2.25 GTADTBRk : A/D Conversion Start Request Timing Buffer Register k (k = A, B)

Base address: GPT32n = 0x4016_9000 + 0x0100 × n (n = 0 to 9)

Offset address: 0x74 (GTADTBRA)
0x80 (GTADTBRB)

Bit position:	31																												0
Bit field:																													
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
31:0	n/a	The buffer registers for the GTADTRk register Access in 8-bit or 16-bit units to the GTADTBRk register is prohibited, and it should be accessed in 32-bit units.	R/W

21.2.26 GTADTDBRk : A/D Conversion Start Request Timing Double-Buffer Register k (k = A, B)

Base address: GPT32n = 0x4016_9000 + 0x0100 × n (n = 0 to 9)

Offset address: 0x78 (GTADTDBRA)
0x84 (GTADTDBRB)

Bit position:	31																												0
Bit field:																													
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
31:0	n/a	The buffer registers for the GTADTBRk register (double buffer registers for the GTADTRk register) Access in 8-bit or 16-bit units to the GTADTDBRk register is prohibited, and it should be accessed in 32-bit units.	R/W

21.2.27 GTDTCR : General PWM Timer Dead Time Control Register

Base address: GPT32n = 0x4016_9000 + 0x0100 × n (n = 0 to 9)

Offset address: 0x88

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

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

Bit	Symbol	Function	R/W
0	TDE	Negative-Phase Waveform Setting 0: GTCCRB is set without using GTDVU and GTDVD 1: GTDVU and GTDVD are used to set the compare match value for negative-phase waveform with dead time automatically in GTCCRB	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
4	TDBUE	GTDVU Register Buffer Operation Enable 0: GTDVU register buffer operation is disabled 1: GTDVU register buffer operation is enabled	R/W
5	TDBDE	GTDVD Register Buffer Operation Enable 0: GTDVD register buffer operation is disabled 1: GTDVD register buffer operation is enabled	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
8	TDFER	GTDVD Register Setting 0: GTDVU and GTDVD registers are set separately. 1: The value written to GTDVU register is automatically set to GTDVD register.	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

GTDTCR enables automatic setting of a compare match value for negative-phase waveform with dead time. GPT has a dead time control function and GTDVU and GTDVD registers are used for setting dead time value.

This register is invalid in Saw-wave PWM mode 2 or complementary PWM mode.

The setting is invalid during the event count operation.

TDE bit (Negative-Phase Waveform Setting)

The TDE bit specifies whether to use GTDVU and GTDVD. When GTDVU and GTDVD are used, the compare match value for a negative-phase waveform with dead time obtained by the compare match value of a positive-phase waveform (GTCCRA) and the dead time value (GTDVU and GTDVD) is automatically set in GTCCRB.

The TDE bit setting is ignored in saw-wave PWM mode, and the GTCCRB is not automatic setting.

The GTCCRB value is automatically set and has the following upper and lower limit values. If the obtained GTCCRB value is not within the upper or lower limit, the following limit value is set in GTCCRB, and GTST.DTEF flag becomes 1. However, if the obtained GTCCRB value exceeds the upper limit in triangle-wave PWM mode, DTEF flag becomes 0.

- Triangle waves:
  Upper limit value: GTPR − 1
  Lower limit value: 1 in up-counting, 0 in down-counting
- Saw-wave one-shot pulse mode:
  Upper limit value: GTPR
  Lower limit value: 0.

TDBUE bit (GTDVU Register Buffer Operation Enable)

This bit enables buffer operation with the GTDVU and GTDBU registers combined.

The timing of buffer transfer is at troughs in triangle-wave mode, and at overflows or underflows in saw-wave mode.

TDBDE bit (GTDVD Register Buffer Operation Enable)

This bit enables buffer operation with the GTDVD and GTDBD registers combined.

The buffer transfer timing is the trough in triangle-wave mode, and at an overflow or underflow in saw-wave mode.

When this bit and the TDFER bit are set to 1 simultaneously, the TDFER bit setting is given priority.

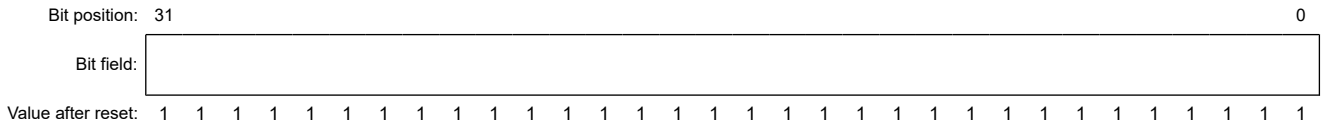
TDFER bit (GTDVD Register Setting)

This bit sets whether or not the value written to the GTDVU register is also set to the GTDVD register automatically.

21.2.28 GTDVk : General PWM Timer Dead Time Value Register k (k = U, D)

Base address: GPT32n = 0x4016_9000 + 0x0100 × n (n = 0 to 9)

Offset address: 0x8C (GTDVU)
0x90 (GTDVD)

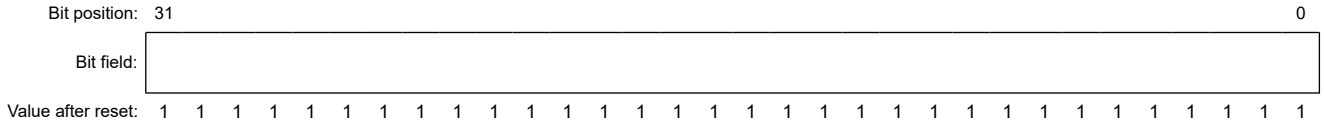


Bit	Symbol	Function	R/W
31:0	n/a	GTDVk is a read/write register that sets the dead time for generating PWM waveforms with dead time. Access in 8-bit or 16-bit units to the GTDVk register is prohibited, and it should be accessed in 32-bit units. The setting is invalid in Saw-wave PWM mode 2. In complementary PWM mode, the GTDVD register is invalid and the GTDVU register is used as the dead time value during both up-counting and down-counting. No matter which GTDVU register of master channel/slave channel 1/slave channel 2 is written, three channels are written at the same time. In Triangle-waves, The GTDVU register is used for up-counting. The GTDVD register is used for down-counting. In the case of a saw-waves, the GTDVU register controls the front dead time and the GTDVD register controls the rear dead time, regardless of whether the count is up or down. Setting a GTDVk value greater than or equal to GTPR is prohibited. In complementary PWM mode, set the GTDVU register to meet the following conditions: ● GTDVU > 0 ● GTDVU < GTPR/2 ● GTDVU + GTPR ≤ 0xffffffff When using the automatic dead time setting function, do not set a value that makes a change point of the waveform exceeding the count period. The change point of the negative-phase waveform, which is automatically calculated, is obtained by reading the GTCCRB register. When GTDVk is used, writing to GTCCRB is prohibited. Other than complementary PWM mode, when this register is set to 0, waveforms without dead time are output. When the GTDTCR.TDFER bit is 1, the writing to the GTDVD register has no effect. At this time, when the GTDVD register is read, the value for the GTDVU register is read. While GPT is running, changing the GTDVk values is prohibited. To change GTDVk to a new value, stop the GPT with the CST bit in the GTCR register.	R/W

21.2.29 GTDBk: General PWM Timer Dead Time Buffer Register k (k = U, D)

Base address: GPT32n = 0x4016_9000 + 0x0100 × n (n = 0 to 9)

Offset address: 0x94 (GTDBU)
0x98 (GTDBD)



Bit	Symbol	Function	R/W
31:0	n/a	The buffer register for the GTDVk register Access in 8-bit or 16-bit units to the GTDBk register is prohibited, and it should be accessed in 32-bit units.	R/W

21.2.30 GTSOS : General PWM Timer Output Protection Function Status Register

Base address: GPT32n = 0x4016_9000 + 0x0100 × n (n = 0 to 9)

Offset address: 0x9C

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

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

Bit	Symbol	Function	R/W
1:0	SOS[1:0]	Output Protection Function Status 0 0: Normal operation 0 1: Protected state (GTCCRA = 0 is set during transfer at trough or crest) 1 0: Protected state (GTCCRA ≥ GTPR is set during transfer at trough) 1 1: Protected state (GTCCRA ≥ GTPR is set during transfer at crest)	R
7:2	—	These bits are read as 0.	R
9:8	—	The read value is undefined.	R
31:10	—	These bits are read as 0.	R

The GTSOS register is a status register that indicates the status of the output protection function. The output protection function is enabled only when the dead time is automatically set (GTDTCR.TDE bit = 1) in triangle-wave mode.

SOS[1:0] bit (Output Protection Function Status)

This bit indicates the status of the output protection function in triangle-wave PWM mode. For details of the output protection function, see section 21.8.4. Output Protection Function for GTIOCnm Pin Output (n = 0 to 9; m = A, B).

21.2.31 GTSOTR : General PWM Timer Output Protection Function Temporary Release Register

Base address: GPT32n = 0x4016_9000 + 0x0100 × n (n = 0 to 9)

Offset address: 0xA0

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

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

Bit	Symbol	Function	R/W
0	SOTR	Output Protection Function Temporary Release 0: Protected state is not released 1: Protected state is released	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

The GTSOTR register temporarily releases the protected state of GTIOCnB pin output (n = 0 to 9) when output protection has been set.

The protected state can be released only for the case of GTSOS.SOS[1:0] bits are 10b (protected state in which GTCCRA register ≥ GTPR register has occurred during transfer at trough). The protected state cannot be released for any other case.

SOTR bit (Output Protection Function Temporary Release)

This bit sets whether to temporarily release the protected state of the GTIOCnB pin output in an output protected state.

After the SOTR bit has been set to 1, the output protection function is canceled from the first trough. After the SOTR bit has been set to 0, output protection is resumed from the first trough.

21.2.32 GTADSMR : General PWM Timer A/D Conversion Start Request Signal Monitoring Register

Base address: GPT32n = 0x4016_9000 + 0x0100 × n (n = 0 to 9)

Offset address: 0xA4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	ADSMEN1	—	—	—	—	—	—	ADSMS1[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	ADSMEN0	—	—	—	—	—	—	ADSMS0[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	ADSMS0[1:0]	A/D Conversion Start Request Signal Monitor 0 Selection 0 0: A/D conversion start request signal generated by the GTADTRA register during up-counting. 0 1: A/D conversion start request signal generated by the GTADTRA register during down-counting. 1 0: A/D conversion start request signal generated by the GTADTRB register during up-counting. 1 1: A/D conversion start request signal generated by the GTADTRB register during down-counting.	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
8	ADSMEN0	A/D Conversion Start Request Signal Monitor 0 Output Enabling 0: Output of A/D conversion start request signal monitor 0 is disabled. 1: Output of A/D conversion start request signal monitor 0 is enabled.	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W
17:16	ADSMS1[1:0]	A/D Conversion Start Request Signal Monitor 1 Selection 0 0: A/D conversion start request signal generated by the GTADTRA register during up-counting. 0 1: A/D conversion start request signal generated by the GTADTRA register during down-counting. 1 0: A/D conversion start request signal generated by the GTADTRB register during up-counting. 1 1: A/D conversion start request signal generated by the GTADTRB register during down-counting.	R/W
23:18	—	These bits are read as 0. The write value should be 0.	R/W
24	ADSMEN1	A/D Conversion Start Request Signal Monitor 1 Output Enabling 0: Output of A/D conversion start request signal monitor 1 is disabled. 1: Output of A/D conversion start request signal monitor 1 is enabled.	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

The GTADSMR register is used to control monitors for the A/D conversion start request signal that is synchronized with a frame period.

ADSMSk[1:0] bits (A/D Conversion Start Request Signal Monitor k Selection) (k = 0, 1)

These bits are used to select A/D conversion start request signal synchronized with a frame period which is monitored by the GTASMk pin.

In triangle-wave PWM mode or complementary PWM mode, the following settings are prohibited:

- Set ADSMSk[1:0] bit to 00b (A/D conversion start request during up-counting) when GTADTRA = 0
- Set ADSMSk[1:0] bit to 10b (A/D conversion start request during up-counting) when GTADTRB = 0
- Set ADSMSk[1:0] bit to 01b (A/D conversion start request during down-counting) when GTADTRA = GTPR
- Set ADSMSk[1:0] bit to 11b (A/D conversion start request during down-counting) when GTADTRB = GTPR

ADSMENk bit (A/D Conversion Start Request Signal Monitor k Output Enabling) (k = 0, 1)

This bit enables or disables the monitor output to the GTADSMk pin.

When the output is disabled, the GTADSMk pin goes to the low level.

When the bit is 1, the signal on the GTADSMk pin goes to the high level on assertion of the signal to request to the start of A/D conversion selected by the ADSMSk[1:0] bits and returns to the low level at the end of the current cycle of the timer for the channel that generated the given signal to request the start of A/D conversion. When the counter stops, the value when the counter stopped is retained for output. Set the ADSMENk bit to 0 to output the low level.

When a signal to request the start of A/D conversion is generated at the end of a timer period, the generation of this signal has priority in terms of monitoring output and the output remains at the high level till the end of the next period.

When the output of the same A/D conversion start request signal monitoring output is enabled for multiple channels, ORed signals will be output from the GPT.

21.2.33 GTEITC : General PWM Timer Extended Interrupt Skipping Counter Control Register

Base address: GPT32n = 0x4016_9000 + 0x0100 × n (n = 0 to 9)

Offset address: 0xA8

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	EITCNT2[3:0]				EITCNT2IV[3:0]				EIVTT2[3:0]				—	—	EIVTC2[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	EITCNT1[3:0]				—	—	—	—	EIVTT1[3:0]				—	—	EIVTC1[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	EIVTC1[1:0]	Extended Interrupt Skipping Counter 1 Count Source Select 0 0: Not counted (not skipped) 0 1: Counting both at overflow or underflow in saw-wave mode, and counting crests in triangle-wave mode or complementary PWM mode 1 0: Counting both at overflow or underflow in saw-wave mode, and counting troughs in triangle-wave mode or complementary PWM mode 1 1: Counting both at overflow or underflow in saw-wave mode, and counting both crests and troughs in triangle-wave mode or complementary PWM mode	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
7:4	EIVTT1[3:0]	Extended Interrupt Skipping 1 Skipping Count Setting Skipping count for the extended interrupt skipping 1	R/W
11:8	—	These bits are read as 0. The write value should be 0.	R/W
15:12	EITCNT1[3:0]	Extended Interrupt Skipping Counter 1	R

Bit	Symbol	Function	R/W
17:16	EIVTC2[1:0]	Extended Interrupt Skipping Counter 2 Count Source select 0 0: Not counted (not skipped) 0 1: Counting both at overflow or underflow in saw-wave mode, and counting crests in triangle-wave mode or complementary PWM mode 1 0: Counting both at overflow or underflow in saw-wave mode, and counting troughs in triangle-wave mode or complementary PWM mode 1 1: Counting both at overflow or underflow in saw-wave mode, and counting both crests and troughs in triangle-wave mode or complementary PWM mode	R/W
19:18	—	These bits are read as 0. The write value should be 0.	R/W
23:20	EIVTT2[3:0]	Extended Interrupt Skipping 2 Skipping Count Setting Skipping count for the extended interrupt skipping 2	R/W
27:24	EITCNT2IV[3:0]	Extended Interrupt Skipping Counter 2 Initial Value	R/W*1
31:28	EITCNT2[3:0]	Extended Interrupt Skipping Counter 2	R

Note 1. The EITCNT2IV[3:0] bits are only writable when the value other than 00b is written to the EIVTC2[1:0] bits that have been 00b.

GTEITC register sets the extended interrupt skipping function to skip the interrupts, A/D conversion start requests, and buffer transfers independently by counting at overflow and underflow in the GTCNT counter.

The setting is operated independently from the interrupt skipping by the GTITC register or the GTADCMSC register.

The setting is invalid during the event count operation.

Access in 8-bit units to GTEITC is prohibited.

EIVTCk[1:0] bits (Extended Interrupt Skipping Counter k Count Source Select) (k = 1, 2)

These bits select the way of counting for the extended interrupt skipping counter k.

Setting only with these bits does not skip the interrupts, A/D conversion start requests, and buffer transfers. Skipping function for the interrupt, A/D conversion start request, and buffer transfer, all of which are a target of skipping, is set individually with the GTEITLI1, GTEITLI2, and GTEITLB registers.

EIVTTk[3:0] bits (Extended Interrupt Skipping k Skipping Count Setting) (k = 1, 2)

A count for the period with continuous skipping is set as a skipping count, where a period is from a generation of a count source selected by the EIVTCk[1:0] bits to the next generation of the count source.

When the count source is generated while the EIVTTk[3:0] bits match the EITCNTk[3:0] bits, the EITCNTk[3:0] bits are cleared.

When these bits are 0x0, skipping is not performed.

EITCNT1[3:0] bits (Extended Interrupt Skipping Counter 1)

The counting is incremented by 1 every time a count source (overflow/underflow/crest/trough) selected by the EIVTC1[1:0] bits is generated.

Counting is performed periodically within the range between 0 and the EIVTT1[3:0] bits.

Even if the GTCNT counter is stopped, the value is not cleared, and value at stop of the GTCNT counter is retained.

[Clearing conditions]

- 00b is written to the EIVTC1[1:0] bits.
- 0x0 is written to the EIVTT1[3:0] bits.
- A count source (overflow/underflow/crest/trough) selected by the EIVTC1[1:0] bits is generated when the extended interrupt skipping 1 skipping count set by the EIVTT1[3:0] bits match the value for the EITCNT1[3:0]

EITCNT2IV[3:0] bit (Extended Interrupt Skipping Counter 2 Initial Value)

These bits are the value of the initial value for the extended interrupt skipping counter 2.

Writing to the EITCNT2IV[3:0] is performed only when the writing value to the EIVTC2[1:0] bits are other than 00b and when the GTEITC register is written by the access of upper 16 bits or 32 bits while the EITCNT2[3:0] bits are set not to count (EIVTC2[1:0] bits are 00b). When the EITCNT2IV[3:0] bits are written, the value written to the EITCNT2IV[3:0] bits is written to the EITCNT2[3:0] bits simultaneously.

The writing to the EITCNT2IV[3:0] bits are ignored when the EITCNT2[3:0] bits are set to count (EIVTC2[1:0] bits are other than 00b) or perform the setting of not to count (00b is written to the EIVTC2[1:0] bits).

The EITCNT2IV[3:0] bits are not reset by the writing 00b to the EIVTC2[1:0] bits.

EITCNT2[3:0] bit (Extended Interrupt Skipping Counter 2)

The counting is incremented by 1 every time a count source (overflow/underflow/crest/trough) selected by the EIVTC2[1:0] bits is generated.

Counting is performed periodically within the range between 0 and the EIVTT2[3:0] bits.

Even if the GTCNT counter is stopped, the value is not cleared, and value at stop of the GTCNT counter is retained.

Setting of the initial value for the EITCNT2[3:0] bits are performed only when the writing value to the EIVTC2[1:0] bits are other than 00b and when the GTEITC register is written by the access of upper 16 bits or 32 bits while the extended interrupt skipping counter 2 is set as not to count (EIVTC2[1:0] bits are 00b).

When the initial value is set, the written value to the EITCNT2IV[3:0] bits is written to the EITCNT2[3:0] bits as the initial value.

[Clearing condition]

- 00b is written to the EIVTC2[1:0] bits.
- 0x0 is written to the EIVTT2[3:0] bits.
- The value other than 00b is written to the EIVTC2[1:0] bits and the 0x0 is written to the EITCNT2IV[3:0] bits simultaneously while 00b is set to the EIVTC2[1:0] bits.
- A count source (overflow/underflow/crest/trough) selected by the EIVTC2[1:0] bits is generated when the extended interrupt skipping 2 skipping count set by the EIVTT2[3:0] bits match the value for the EITCNT2[3:0].

21.2.34 GTEITLI1 : General PWM Timer Extended Interrupt Skipping Setting Register 1

Base address: GPT32n = 0x4016_9000 + 0x0100 × n (n = 0 to 9)

Offset address: 0xAC

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	EITLU[2:0]			—	EITLV[2:0]			—	EITLF[2:0]			—	EITLE[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
2:0	EITLA[2:0]	GTCCRA Register Compare Match/Input Capture Interrupt Extended Skipping Function Select See Table 21.6.	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
6:4	EITLB[2:0]	GTCCRB Register Compare Match/Input Capture Interrupt Extended Skipping Function Select See Table 21.6.	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
10:8	EITLC[2:0]	GTCCRC Register Compare Match Interrupt Extended Skipping Function Select See Table 21.6.	R/W
11	—	This bit is read as 0. The write value should be 0.	R/W
14:12	EITLD[2:0]	GTCCRD Register Compare Match Interrupt Extended Skipping Function Select See Table 21.6.	R/W
15	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
18:16	EITLE[2:0]	GTCCRE Register Compare Match Interrupt Extended Skipping Function Select See Table 21.6.	R/W
19	—	This bit is read as 0. The write value should be 0.	R/W
22:20	EITLF[2:0]	GTCCRF Register Compare Match Interrupt Extended Skipping Function Select See Table 21.6.	R/W
23	—	This bit is read as 0. The write value should be 0.	R/W
26:24	EITLV[2:0]	Overflow Interrupt Extended Skipping Function Select See Table 21.6.	R/W
27	—	This bit is read as 0. The write value should be 0.	R/W
30:28	EITLU[2:0]	Underflow Interrupt Extended Skipping Function Select See Table 21.6.	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

The GTEITLI1 register sets the extended skipping function for interrupts such as compare match/input capture, overflow and underflow.

Setting only with this register does not perform skipping. The GTEITC register should be set so that a corresponding extended interrupt skipping counter operates the counting.

The setting is operated independently from the interrupt skipping by the GTITC register or the GTADCMSC register.

The setting is invalid during the event count operation.

EITLm[2:0] bits (GTCCRm Register Compare Match/Input Capture Interrupt Extended Skipping Function Select) (m = A, B)

These bits select the extended interrupt skipping function to skip the compare match/input capture interrupt (GPTn_CCMPm) in the GTCCRm register. See Table 21.6.

EITLx[2:0] bits (GTCCRx Register Compare Match Interrupt Extended Skipping Function Select) (x = C, D, E, F)

These bits select the extended interrupt skipping function to skip the compare match interrupt (GPTn_CMPx) in the GTCCRx register. See Table 21.6.

EITLV[2:0] bit (Overflow Interrupt Extended Skipping Function Select)

These bits select the extended interrupt skipping function to skip the interrupt at overflow(GPTn_OVF). See Table 21.6.

EITLU[2:0] bit (Underflow Interrupt Extended Skipping Function Select)

These bits select the extended interrupt skipping function to skip the interrupt at underflow (GPTn_UDF). See Table 21.6.

Table 21.6 Setting the Function select for the GTEITLI1 (1 of 2)

EITLy[2:0]	Function
0 0 0	Do not perform an extended interrupt skipping
0 0 1	Skip an interrupt in the period when the value for the extended interrupt skipping counter 1 is other than 0 (An interrupt is output in the period of the EITCNT1[3:0] bits = 0)
0 1 0	Skip an interrupt in the period when the value for the extended interrupt skipping counter 2 is other than 0 (An interrupt is output in the period of the EITCNT2[3:0] bits = 0)
0 1 1	Skip an interrupt in the period the value for the extended interrupt skipping counter 1 or 2 is other than 0 (An interrupt is output in the period of the EITCNT1[3:0] bits = 0 and the EITCNT2[3:0] bits = 0)
1 0 0	Setting prohibited
1 0 1	Skip an interrupt in the period when the value for the extended interrupt skipping counter 1 is other than the skipping count (An interrupt is output in the period of the EITCNT1[3:0] bits = the EIVTT1[3:0] bits)
1 1 0	Skip an interrupt in the period when the value for the extended interrupt skipping counter 2 is other than the skipping count (An interrupt is output in the period of the EITCNT2[3:0] bits = the EIVTT2[3:0] bits)

Table 21.6 Setting the Function select for the GTEITLI1 (2 of 2)

EITLy[2:0]	Function
1 1 1	Skip an interrupt in the period when the value for the extended interrupt skipping counter 1 or 2 is other than the skipping count (An interrupt is output in the period of the EITCNT1[3:0] bits = the EIVTT1[3:0] bits and the EITCNT2[3:0] bits = the EIVTT2[3:0] bits)

Note:
- y = A, B, C, D, E, F, V, U
- When the intended skipping counter is set as not to count (the EIVTCk[1:0] bits = 00b or the EIVTTk[3:0] bits = 0x0), skipping is not performed.(k = 1, 2)
- When the EITLy[2:0] bits are set to 011b or 111b, and when one of the skipping counter 1 or 2 is set as not to count, skipping is not performed.

21.2.35 GTEITLI2 : General PWM Timer Extended Interrupt Skipping Setting Register 2

Base address: GPT32n = 0x4016_9000 + 0x0100 × n (n = 0 to 9)

Offset address: 0xB0

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

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

Bit	Symbol	Function	R/W
2:0	EADTAL[2:0]	GTADTRA Register A/D Conversion Start Request Extended Skipping Function Select See Table 21.7.	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
6:4	EADTBL[2:0]	GTADTRB Register A/D Conversion Start Request Extended Skipping Function Select See Table 21.7.	R/W
31:7	—	These bits are read as 0. The write value should be 0.	R/W

The GTEITLI2 register sets the extended skipping function for A/D conversion start requests.

Setting only with this register does not perform skipping. The GTEITC register should be set so that a corresponding extended interrupt skipping counter operates the counting.

The setting is operated independently from the interrupt skipping by the GTITC register or the GTADCMSC register.

The setting is invalid during the event count operation.

EADTmL[2:0] bits (GTADTRm Register A/D Conversion Start Request Extended Skipping Function Select) (m = A, B)

These bits select the extended skipping function to skip A/D conversion start requests for the compare match in the GTADTRm register. See Table 21.7.

Table 21.7 Setting the Function Select for the GTEITLI2 Register (1 of 2)

EADTmL[2:0]	Function
0 0 0	Do not perform an extended interrupt skipping
0 0 1	Skip an A/D conversion start request in the period when the value for the extended interrupt skipping counter 1 is other than 0 (An A/D conversion start request is output in the period of the EITCNT1[3:0] bits = 0)
0 1 0	Skip an A/D conversion start request in the period when the value for the extended interrupt skipping counter 2 is other than 0 (An A/D conversion start request is output in the period of the EITCNT2[3:0] bits = 0)

Table 21.7 Setting the Function Select for the GTEITLI2 Register (2 of 2)

EADTmL[2:0]	Function
0 1 1	Skip an A/D conversion start request in the period when the value for the extended interrupt skipping counter 1 or 2 is other than 0 (An A/D conversion start request is output in the period of the EITCNT1[3:0] bits = 0 and the EITCNT2[3:0] bits = 0)
1 0 0	Setting prohibited
1 0 1	Skip an A/D conversion start request in the period when the value for the extended interrupt skipping counter 1 is other than the skipping count (An A/D conversion start request is output in the period of the EITCNT1[3:0] bits = the EIVTT1[3:0] bits)
1 1 0	Skip an A/D conversion start request in the period when the value for the extended interrupt skipping counter 2 is other than the skipping count (An A/D conversion start request is output in the period of the EITCNT2[3:0] bits = the EIVTT2[3:0] bits)
1 1 1	Skip an A/D conversion start request in the period when the value for the extended interrupt skipping counter 1 or 2 is other than the skipping count (An A/D conversion start request is output in the period of the EITCNT1[3:0] bits = the EIVTT1[3:0] bits and the EITCNT2[3:0] bits = the EIVTT2[3:0] bits)

Note:
- m = A, B
- When the intended skipping counter is set as not to count (the EIVTCk[1:0] bits = 00b or the EIVTTk[3:0] bits = 0x0), skipping is not performed. (k = 1, 2)
- When the EADTmL[2:0] bits are set to 011b or 111b, and when one of the skipping counter 1 or 2 is set as not to count, skipping is not performed.

21.2.36 GTEITLB : General PWM Timer Extended Buffer Transfer Skipping Setting Register

Base address: GPT32n = 0x4016_9000 + 0x0100 × n (n = 0 to 9)

Offset address: 0xB4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	EBTLDVD[2:0]			—	EBTLDVU[2:0]			—	EBTLADB[2:0]			—	EBTLADA[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
2:0	EBTLCA[2:0]	GTCCRA Register Buffer Transfer Extended Skipping Function Select See Table 21.8.	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
6:4	EBTLCB[2:0]	GTCCRB Register Buffer Transfer Extended Skipping Function Select See Table 21.8.	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
10:8	EBTLPR[2:0]	GTPR Register Buffer Transfer Extended Skipping Function Select See Table 21.8.	R/W*1
15:11	—	These bits are read as 0. The write value should be 0.	R/W
18:16	EBTLADA[2:0]	GTADTRA Register Buffer Transfer Extended Skipping Function Select See Table 21.8.	R/W
19	—	This bit is read as 0. The write value should be 0.	R/W
22:20	EBTLADB[2:0]	GTADTRB Register Buffer Transfer Extended Skipping Function Select See Table 21.8.	R/W
23	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
26:24	EBTLDVU[2:0]	GTDVU Register Buffer Transfer Extended Skipping Function Select See Table 21.8.	R/W
27	—	This bit is read as 0. The write value should be 0.	R/W
30:28	EBTLDVD[2:0]	GTDVD Register Buffer Transfer Extended Skipping Function Select See Table 21.8.	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

Note 1. When in complementary PWM mode, no matter which register of master channel/slave channel 1/slave channel 2 is written, three channels are written at the same time.

The GTEITLB register sets the extended skipping function for buffer transfers.

Setting only this register does not perform skipping. The GTEITC register should be set so that a corresponding extended interrupt skipping counter operates the counting.

The setting is operated independently from the interrupt skipping by the GTITC register.

The buffer transfer from the GTOLBR register to the GTIOR.GTIOA[4:0], GTIOB[4:0] bits is not target of the extended buffer transfer skipping function.

The setting is invalid during the event count operation.

EBTLCA[2:0] bit (GTCCRA Register Buffer Transfer Extended Skipping Function Select)

These bits select the extended buffer transfer skipping function to skip buffer transfers (transfers among the GTCCRA, GTCCRC, GTCCRD registers and a temporary register A) in the GTCCRA register. See Table 21.8.

An extended skipping of buffer transfers in the GTCCRA register is valid for forcible buffer transfers by the GTBER.CCRSWT bit while the count operation is stopped. Forcible buffer transfers in the GTCCRA register should be performed in the condition of not performing the extended buffer transfer skipping.

The buffer transfer between the GTCCRC, GTCCRE, and GTCCRA in complementary PWM mode cannot be skipped.

EBTLCB[2:0] bit (GTCCRB Register Buffer Transfer Extended Skipping Function Select)

These bits select the extended buffer transfer skipping function to skip buffer transfers (transfers among the GTCCRB, GTCCRE, GTCCRF registers and a temporary register B) in the GTCCRB register. See Table 21.8.

An extended skipping of buffer transfers in the GTCCRB register is valid for forcible buffer transfers by the GTBER.CCRSWT bit while the count operation is stopped. Forcible buffer transfers in the GTCCRB register should be performed in the condition of not performing the extended buffer transfer skipping.

EBTLPR[2:0] bit (GTPR Register Buffer Transfer Extended Skipping Function Select)

These bits select the extended buffer transfer skipping function to skip buffer transfers (transfers among the GTPR, GTPBR, and GTPDBR registers) in the GTPR register. See Table 21.8.

If the buffer transfer of GTPR is skipped in complementary PWM mode, GTEITC setting of slave channel should be matched to the master channel so that the buffer transfer timing of the slave channel matches the master channel.

EBTLADm[2:0] bits (GTADTRm Register Buffer Transfer Extended Skipping Function Select) (m = A, B)

These bits select the extended buffer transfer skipping function to skip buffer transfers (transfers among the GTADTRm, GTADTBRm, and GTADTDBRm registers) in the GTADTRm register. See Table 21.8.

EBTLDVm[2:0] bits (GTDVm Register Buffer Transfer Extended Skipping Function Select) (m = U, D)

These bits select the extended buffer transfer skipping function to skip buffer transfers (transfers between the GTDVm and GTDBm registers) in the GTDVm register. See Table 21.8.

Table 21.8 Setting the Function Select for the GTEITLB Register (1 of 2)

EBTLx[2:0] bits	Function
0 0 0	Do not perform an extended interrupt skipping
0 0 1	Skip a buffer transfer in the period when the value for the extended interrupt skipping counter 1 is other than 0 (Buffer is transferred in the period of the EITCNT1[3:0] bits = 0)

Table 21.8 Setting the Function Select for the GTEITLB Register (2 of 2)

EBTLx[2:0] bits	Function
0 1 0	Skip a buffer transfer in the period when the value for the extended interrupt skipping counter 2 is other than 0 (Buffer is transferred in the period of the EITCNT2[3:0] bits = 0)
0 1 1	Skip a buffer transfer in the period when the value for the extended interrupt skipping counter 1 or 2 is other than 0 (Buffer is transferred in the period of the EITCNT1[3:0] bits = 0 and the EITCNT2[3:0] bits = 0)
1 0 0	Setting prohibited
1 0 1	Skip a buffer transfer in the period when the value for the extended interrupt skipping counter 1 is other than the skipping count (Buffer is transferred in the period of the EITCNT1[3:0] bits = the EIVTT1[3:0] bits)
1 1 0	Skip a buffer transfer in the period when the value for the extended interrupt skipping counter 2 is other than the skipping count (Buffer is transferred in the period of the EITCNT2[3:0] bits = the EIVTT2[3:0] bits)
1 1 1	Skip a buffer transfer in the period when the value for the extended interrupt skipping counter 1 or 2 is other than the skipping count (Buffer is transferred in the period of the EITCNT1[3:0] bits = the EIVTT1[3:0] bits and the EITCNT2[3:0] bits = the EIVTT2[3:0] bits)

Note:
- x = CA, CB, PR, ADA, ADB, DVU, DVD
- When the intended skipping counter is set as not to count (EIVTCk[1:0]bits = 00b or EIVTTk[3:0] bits = 0x0), skipping is not performed. (k = 1, 2)
- When the EBTLx[2:0] bits are set to 011b or 111b, and when one of the skipping counter 1 or 2 is set as not to count, skipping is not performed.

21.2.37 GTICLF : General PWM Timer Inter Channel Logical Operation Function Setting Register

Base address: GPT32n = 0x4016_9000 + 0x0100 × n (n = 0 to 9)

Offset address: 0xB8

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	ICLFSELD[5:0]						—	ICLFB[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
2:0	ICLFA[2:0]	GTIOCnA Output Logical Operation Function Select 0 0 0: A (no delay) 0 0 1: NOT A (no delay) 0 1 0: C (1GTCLK delay) 0 1 1: NOT C (1GTCLK delay) 1 0 0: A AND C (1GTCLK delay)*2 1 0 1: A OR C (1GTCLK delay)*2 1 1 0: A EXOR C (1GTCLK delay)*2 1 1 1: A NOR C (1GTCLK delay)*2	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
9:4	ICLFSELC[5:0]	Inter Channel Signal C Select*1*2 0x00: GTIOC0A 0x01: GTIOC0B 0x02: GTIOC1A 0x03: GTIOC1B 0x04: GTIOC2A 0x05: GTIOC2B 0x06: GTIOC3A 0x07: GTIOC3B ⋮ ⋮ 0x3E: GTIOC31A 0x3F: GTIOC31B	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
18:16	ICLFB[2:0]	GTIOCnB Output Logical Operation Function Select 0 0 0: B (no delay) 0 0 1: NOT B (no delay) 0 1 0: D (1GTCLK delay) 0 1 1: NOT D (1GTCLK delay) 1 0 0: B AND D (1GTCLK delay)*3 1 0 1: B OR D (1GTCLK delay)*3 1 1 0: B EXOR D (1GTCLK delay)*3 1 1 1: B NOR D (1GTCLK delay)*3	R/W
19	—	This bit is read as 0. The write value should be 0.	R/W
25:20	ICLFSELD[5:0]	Inter Channel Signal D Select*1*3 0x00: GTIOC0A 0x01: GTIOC0B 0x02: GTIOC1A 0x03: GTIOC1B 0x04: GTIOC2A 0x05: GTIOC2B 0x06: GTIOC3A 0x07: GTIOC3B ⋮ ⋮ 0x3E: GTIOC31A 0x3F: GTIOC31B	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. The signal before performing output disable control is selected.
Note 2. When channel's own GTIOCnA is selected, C is treated as "1".
Note 3. When channel's own GTIOCnB is selected, D is treated as "1".

The GTICLF register sets the logical operation function between compare match outputs. The logical operation is performed with the signals that the duty 0%/100% control is performed after compare match control. (The output disable control is performed with the signal after logical operation.)

Access in 8-bit units to GTICLF is prohibited.

ICLFm[2:0] bit (GTIOCnm Output Logical Operation Function Select) (m = A, B)

These bits select the logical operation function between signals before performing output disable control for GTIOCnm. To prevent hazard to the GPT output, the signal after logical operation is latched with GTCLK. After latching, the output disable control is performed. When the logical operation function which causes the delay of 1 GTCLK is selected, the output enable signal is also delayed with 1 GTCLK and input to the output disable control.

When the same signal to operate logical function AND, OR, EXOR and NOR is selected, one signal is treated as "1".

ICLFSELk[5:0] bit (Inter Channel Signal k Select) (k = C, D)

These bits select the signal k that the logical operation is performed with the signal before performing output disable control for GTIOCnm.

21.2.38 GTPC : General PWM Timer Period Count Register

Base address: GPT32n = 0x4016_9000 + 0x0100 × n (n = 0 to 3)

Offset address: 0xBC

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	PCNT[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
0	PCEN	Period Count Function Enable 0: Period count function is disabled 1: Period count function is enabled	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
8	ASTP	Automatic Stop Function Enable 0: Automatic stop function is disabled 1: Automatic stop function is enabled	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W
27:16	PCNT[11:0]	Period Counter Counter for the number of period	R/W
31:28	—	These bits are read as 0. The write value should be 0.	R/W

The GTPC register counts the number of period.

PCEN bit (Period Count Function Enable)

This bit enables or disables period count function.

Writing is available when counting is both in progress and stopped.

When 1 is written to either the GTSECR.SPCE bit or the GTSECR.SPCD bit, the value is simultaneously set to the PCEN bit in the channels set to 1 by the GTSECSR register.

ASTP bit (Automatic Stop Function Enable)

This bit enables or disables the GTCNT counter automatic stopping after finishing counting the number of period.

When the PCEN bis is 0, writing is available.

When the PCEN bit is 1, writing is disabled.

When the PCEN bit is 1, the ASTP bit is 1, and the PCNT counter is stopped at PCNT = 0, the GTCNT counter is also stopped. When the ASTP bit is 0, the GTCNT counter continues to count.

PCNT[11:0] bit (Period Counter)

This counter counts the number of period.

When the PCEN bis is 0, writing the number of period is available.

When the PCEN bit is 1, writing is disabled, and down-counting is performed at the end of period. In saw-wave mode, the end of period refers to overflow, underflow, or counter clearing. In triangle-wave mode or complementary PWM mode, it refers to trough.

When the PCNT counter is 1 at the end of period, it becomes 0 and counting is stopped.

When the GTCNT counter is stopped while period count function is enabled, the PCNT counter keeps its value. When the GTCNT counter restarts counting and the PCEN bit is 1, the PCNT counter restarts down-counting from the hold value.

When the PCEN bit is changed from 0 to 1 while the PCNT counter is 0 and the ASTP bit is 1, the GTCNT counter is stopped at the count clock immediately after that.

21.2.39 GTADCMSC : General PWM Timer A/D Conversion Start Request Compare Match Skipping Control Register

Base address: GPT32n = 0x4016_9000 + 0x0100 × n (n = 4 to 9)

Offset address: 0xC0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	ADCMSCNT2[3:0]				ADCMSCNT2IV[3:0]				ADCMST2[3:0]				—	—	ADCMSC2[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ADCMSCNT1[3:0]				ADCMSCNT1IV[3:0]				ADCMST1[3:0]				—	—	ADCMSC1[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	ADCMSC1[1:0]	A/D Conversion Start Request Compare Match Skipping Counter 1 Count Source Select 0 0: Not counted (not skipped) 0 1: Counting GTADTRA register compare match 1 0: Counting GTADTRB register compare match 1 1: Counting both GTADTRA register compare match and GTADTRB register compare match	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
7:4	ADCMST1[3:0]	A/D Conversion Start Request Compare Match Skipping 1 Skipping Count Setting Skipping count for the A/D conversion start request compare match skipping 1.	R/W
11:8	ADCMSCNT1IV[3:0]	A/D Conversion Start Request Compare Match Skipping Counter 1 Initial Value	R/W*1
15:12	ADCMSCNT1[3:0]	A/D Conversion Start Request Compare Match Skipping Counter 1	R
17:16	ADCMSC2[1:0]	A/D Conversion Start Request Compare Match Skipping Counter 2 Count Source Select 0 0: Not counted (not skipped) 0 1: Counting GTADTRA register compare match 1 0: Counting GTADTRB register compare match 1 1: Counting both GTADTRA register compare match and GTADTRB register compare match	R/W
19:18	—	These bits are read as 0. The write value should be 0.	R/W
23:20	ADCMST2[3:0]	A/D Conversion Start Request Compare Match Skipping 2 Skipping Count Setting Skipping count for the A/D conversion start request compare match skipping 2	R/W
27:24	ADCMSCNT2IV[3:0]	A/D Conversion Start Request Compare Match Skipping Counter 2 Initial Value	R/W*2
31:28	ADCMSCNT2[3:0]	A/D Conversion Start Request Compare Match Skipping Counter 2	R

Note 1. Writing is possible only when the ADCMSC1[1:0] bits are 00b and a value other than 00b is written to the ADCMSC1[1:0] bits.
Note 2. Writing is possible only when the ADCMSC2[1:0] bits are 00b and a value other than 00b is written to the ADCMSC2[1:0] bits.

The GTADCMSC register is a register that controls the skipping counter of the A/D conversion start request compare match skipping function that counts compare matches of GTADTRA register and GTADTRB register, and skipping A/D conversion start request and buffer transfer independently.

This register setting is operated independently from the interrupt skipping by the GTITC register or GTEITC register.

Access in 8-bit units to GTADCMSC is prohibited.

ADCMSCk[1:0] bits (A/D Conversion Start Request Compare Match Skipping Counter k Count Source Select) (k = 1, 2)

These bits select the way of counting for the A/D conversion start request compare match skipping counter k.

Setting only with these bits does not skip the A/D conversion start requests and buffer transfers. Skipping function for the A/D conversion start request and buffer transfer which are a target of skipping is set individually with the GTADCMSS register.

ADCMSTk[3:0] bits (A/D Conversion Start Request Compare Match Skipping k Skipping Count Setting) (k = 1, 2)

A count for the period with continuous skipping is set as a skipping count, where a period is from a generation of a count source selected by the ADCMSCk[1:0] bits to the next generation of the count source.

When the count source is generated while the ADCMSTk[3:0] bits match the ADCMSCNTk[3:0] bits, the ADCMSCNTk[3:0] bits are cleared.

When these bits are 0x0, skipping is not performed.

ADCMSCNTkIV[3:0] bits (A/D Conversion Start Request Compare Match Skipping Counter k Initial Value) (k = 1, 2)

These bits are the value of the initial value for the A/D conversion start request compare match skipping counter k.

Writing to the ADCMSCNTkIV[3:0] is performed only when the writing value to the ADCMSCk[1:0] bits are other than 00b and when the GTADCMSC register is written by the access of 16 bits or 32 bits while the ADCMSCNTk[3:0] bits are set not to count (ADCMSCk[1:0] bits are 00b). When the ADCMSCNTkIV[3:0] bits are written, the value written to the ADCMSCNTkIV[3:0] bits is written to the ADCMSCNTk[3:0] bits simultaneously.

The writing to the ADCMSCNTkIV[3:0] bits are ignored when the ADCMSCNTk[3:0] bits are set to count (ADCMSCk[1:0] bits are other than 00b) or perform the setting of not to count (00b is written to the ADCMSCk[1:0] bits).

The ADCMSCNTkIV[3:0] bits are not reset by the writing 00b to the ADCMSCk[1:0] bits.

ADCMSCNTk[3:0] bits (A/D Conversion Start Request Compare Match Skipping Counter k) (k=1,2)

The counting is incremented by 1 every time a count source selected by the ADCMSCk[1:0] bits is generated.

Counting is performed periodically within the range between 0 and the ADCMSTk[3:0] bits.

Even if the GTCNT counter is stopped, the value is not cleared, and value at stop of the GTCNT counter is retained.

Setting of the initial value for the ADCMSCNTk[3:0] bits are performed only when the writing value to the ADCMSCk[1:0] bits are other than 00b and when the GTADCMSC register is written by the access of 16 bits or 32 bits while the A/D conversion start request compare match skipping counter k is set as not to count (ADCMSCk[1:0] bits are 00b).

When the initial value is set, the written value to the ADCMSCNTkIV[3:0] bits is written to the ADCMSCNTk[3:0] bits as the initial value.

[Clearing condition]

- 00b is written to the ADCMSCk[1:0] bits.
- 0x0 is written to the ADCMSTk[3:0] bits.
- The value other than 00b is written to the ADCMSCk[1:0] bits and the 0x0 is written to the ADCMSCNTkIV[3:0] bits simultaneously while 00b is set to the ADCMSCk[1:0] bits.
- A count source selected by the ADCMSCk[1:0] bits is generated when the extended interrupt skipping 2 skipping count set by the ADCMSTk[3:0] bits match the value for the ADCMSCNTk[3:0].

21.2.40 GTADCMSS : General PWM Timer A/D Conversion Start Request Compare Match Skipping Setting Register

Base address: GPT32n = 0x4016_9000 + 0x0100 × n (n = 4 to 9)

Offset address: 0xC4

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

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

Bit	Symbol	Function	R/W
2:0	ADCMSAL[2:0]	GTADTRA Register A/D Conversion Start Request Compare Match Skipping Function Select See Table 21.9.	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
6:4	ADCMSBL[2:0]	GTADTRB Register A/D Conversion Start Request Compare Match Skipping Function Select See Table 21.9.	R/W
15:7	—	These bits are read as 0. The write value should be 0.	R/W
18:16	ADCMBSA[2:0]	GTADTRA Register Buffer Transfer by A/D Conversion Start Request Compare Match Skipping Function Select See Table 21.10.	R/W
19	—	This bit is read as 0. The write value should be 0.	R/W
22:20	ADCMBSB[2:0]	GTADTRB Register Buffer Transfer by A/D Conversion Start Request Compare Match Skipping Function Select See Table 21.10.	R/W
31:23	—	These bits are read as 0. The write value should be 0.	R/W

The GTADCMSS register is a register that selects the A/D conversion start request compare match skipping function or the GTADTRm (m = A, B) register buffer transfer by A/D conversion start request compare match skipping function.

Setting only with this register does not perform skipping. The GTADCMSC register should be set so that a corresponding A/D conversion start request compare match skipping counter operates the counting.

This register setting is operated independently from the interrupt skipping by the GTITC register or GTEITC register.

ADCMSmL[2:0] bits (GTADTRm Register A/D Conversion Start Request Compare Match Skipping Function Select) (m = A, B)

These bits select the A/D conversion start request compare match skipping function of the GTADTRm (m = A, B) register.Table 21.9.

ADCMBSm[2:0] bits (GTADTRm Register Buffer Transfer by A/D Conversion Start Request Compare Match Skipping Function Select) (m = A, B)

These bits select the GTADTRm (m = A, B) register buffer transfer by A/D conversion start request compare match skipping function which skips the buffer transfer of GTADTRm (m = A, B) register(Transfer between GTADTRm register, GTADTBRm register, GTADTDBRm register). See Table 21.10.

Table 21.9 Setting of the GTADTRm Register A/D Conversion Start Request Compare Match Skipping Function Select bit (m = A ,B) (1 of 2)

ADCMSmL[2:0]	Function
0 0 0	Do not perform an A/D conversion start request compare match skipping

Table 21.9 Setting of the GTADTRm Register A/D Conversion Start Request Compare Match Skipping Function Select bit (m = A ,B) (2 of 2)

ADCMSmL[2:0]	Function
0 0 1	Skip an A/D conversion start request in the period when the value for the A/D conversion start request compare match skipping counter 1 is other than 0 (An A/D conversion start request is output in the period of the ADCMSCNT1[3:0] bits = 0)
0 1 0	Skip an A/D conversion start request in the period when the value for the A/D conversion start request compare match skipping counter 2 is other than 0 (An A/D conversion start request is output in the period of the ADCMSCNT2[3:0] bits = 0)
0 1 1	Skip an A/D conversion start request in the period when the value for the A/D conversion start request compare match skipping counter 1 or 2 is other than 0 (An A/D conversion start request is output in the period of the ADCMSCNT1[3:0] bits = 0 and the ADCMSCNT2[3:0] bits = 0)
1 0 0	Setting prohibited
1 0 1	Skip an A/D conversion start request in the period when the value for the A/D conversion start request compare match skipping counter 1 is other than the skipping count (An A/D conversion start request is output in the period of the ADCMSCNT1[3:0] bits = the ADCMST1[3:0] bits)
1 1 0	Skip an A/D conversion start request in the period when the value for the A/D conversion start request compare match skipping counter 2 is other than the skipping count (An A/D conversion start request is output in the period of the ADCMSCNT2[3:0] bits = the ADCMST2[3:0] bits)
1 1 1	Skip an A/D conversion start request in the period when the value for the A/D conversion start request compare match skipping counter 1 or 2 is other than the skipping count (An A/D conversion start request is output in the period of the ADCMSCNT1[3:0] bits = the ADCMST1[3:0] bits and the ADCMSCNT2[3:0] bits = the ADCMST2[3:0] bits)

Note:
- m = A, B
- When the intended skipping counter is set as not to count (the ADCMSCk[1:0] bits = 00b or the ADCMSTk[3:0] bits = 0x0), skipping is not performed. (k = 1, 2)
- When the ADCMSmL[2:0] bits are set to 011b or 111b, and when one of the skipping counters is set as not to count, skipping is not performed.

Table 21.10 Setting of the GTADTRm Register Buffer Transfer by A/D Conversion Start Request Compare Match Skipping Function Select bit (m = A, B)

ADCMBSm[2:0]	Function
0 0 0	Do not perform an GTADTRm register buffer transfer by A/D conversion start request compare match skipping
0 0 1	Skip a buffer transfer in the period when the value for the A/D conversion start request compare match skipping counter 1 is other than 0 (Buffer is transferred in the period of the ADCMSCNT1[3:0] bits = 0)
0 1 0	Skip a buffer transfer in the period when the value for the A/D conversion start request compare match skipping counter 2 is other than 0 (Buffer is transferred in the period of the ADCMSCNT2[3:0] bits = 0)
0 1 1	Skip a buffer transfer in the period when the value for the A/D conversion start request compare match skipping counter 1 or 2 is other than 0 (Buffer is transferred in the period of the ADCMSCNT1[3:0] bits = 0 and the ADCMSCNT2[3:0] bits = 0)
1 0 0	Setting prohibited
1 0 1	Skip a buffer transfer in the period when the value for the A/D conversion start request compare match skipping counter 1 is other than the skipping count (Buffer is transferred in the period of the ADCMSCNT1[3:0] bits = the ADCMST1[3:0] bits)
1 1 0	Skip a buffer transfer in the period when the value for the A/D conversion start request compare match skipping counter 2 is other than the skipping count (Buffer is transferred in the period of the ADCMSCNT2[3:0] bits = the ADCMST2[3:0] bits)
1 1 1	Skip a buffer transfer in the period when the value for the A/D conversion start request compare match skipping counter 1 or 2 is other than the skipping count (A Buffer is transferred in the period of the ADCMSCNT1[3:0] bits = the ADCMST1[3:0] bits and the ADCMSCNT2[3:0] bits = the ADCMST2[3:0] bits)

Note:
- m = A, B
- When the intended skipping counter is set as not to count (the ADCMSCk[1:0] bits = 00b or the ADCMSTk[3:0] bits = 0x0), skipping is not performed. (k = 1, 2)
- When the ADCMSmL[2:0] bits are set to 011b or 111b, and when one of the skipping counters is set as not to count, skipping is not performed.

21.2.41 GTSECSR : General PWM Timer Operation Enable Bit Simultaneous Control Channel Select Register

Base address: GPT32n = 0x4016_9000 + 0x0100 × n (n = 0 to 9)

Offset address: 0xD0

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	SECSEL9	SECSEL8	SECSEL7	SECSEL6	SECSEL5	SECSEL4	SECSEL3	SECSEL2	SECSEL1	SECSEL0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SECSEL0	Channel 0 Operation Enable Bit Simultaneous Control Channel Select 0: Disable simultaneous control 1: Enable simultaneous control	R/W
1	SECSEL1	Channel 1 Operation Enable Bit Simultaneous Control Channel Select 0: Disable simultaneous control 1: Enable simultaneous control	R/W
2	SECSEL2	Channel 2 Operation Enable Bit Simultaneous Control Channel Select 0: Disable simultaneous control 1: Enable simultaneous control	R/W
3	SECSEL3	Channel 3 Operation Enable Bit Simultaneous Control Channel Select 0: Disable simultaneous control 1: Enable simultaneous control	R/W
4	SECSEL4	Channel 4 Operation Enable Bit Simultaneous Control Channel Select 0: Disable simultaneous control 1: Enable simultaneous control	R/W
5	SECSEL5	Channel 5 Operation Enable Bit Simultaneous Control Channel Select 0: Disable simultaneous control 1: Enable simultaneous control	R/W
6	SECSEL6	Channel 6 Operation Enable Bit Simultaneous Control Channel Select 0: Disable simultaneous control 1: Enable simultaneous control	R/W
7	SECSEL7	Channel 7 Operation Enable Bit Simultaneous Control Channel Select 0: Disable simultaneous control 1: Enable simultaneous control	R/W
8	SECSEL8	Channel 8 Operation Enable Bit Simultaneous Control Channel Select 0: Disable simultaneous control 1: Enable simultaneous control	R/W
9	SECSEL9	Channel 9 Operation Enable Bit Simultaneous Control Channel Select 0: Disable simultaneous control 1: Enable simultaneous control	R/W
31:10	—	These bits are read as 0. The write value should be 0.	R/W

The GTSECSR register selects an intended channel n (n = 0 to 9) for updating an operation enable bit by the GTSECR register. A bit position for the GTSECSR register indicates a channel number. The GTSECSR register of each channel is a common register, and writing 1 to a bit in the GTSECSR register in any channel and updating it changes a channel, related to the position of the bit written with 1 by the GTSECSR register, to be simultaneously controlled of the operation enable bit by the GTSECR register.

Access in 8-bit or 16-bit units to GTSECSR is prohibited, and it should be accessed in 32-bit units.

SECSELn bit (Operation Enable Bit Simultaneous Control Channel Select) (n = 0 to 9)

This bit enables or disables the simultaneous control of operation enable in channel n.

When the bit is set to 1, the simultaneous control is enabled, and disabled when the bit is 0.

21.2.42 GTSECR : General PWM Timer Operation Enable Bit Simultaneous Control Register

Base address: GPT32n = 0x4016_9000 + 0x0100 × n (n = 0 to 9)

Offset address: 0xD4

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	SBDDD	SBDAD	SBDPD	SBDCD	—	—	—	—	SBDDE	SBDAE	SBDPE	SBDCE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SBDCE	GTCCR Register Buffer Operation Simultaneous Enable 0: Disable simultaneous enabling GTCCR buffer operations 1: Enable GTCCR register buffer operations simultaneously	R/W
1	SBDPE	GTPR Register Buffer Operation Simultaneous Enable 0: Disable simultaneous enabling GTPR buffer operations 1: Enable GTPR register buffer operations simultaneously	R/W*1
2	SBDAE	GTADTR Register Buffer Operation Simultaneous Enable 0: Disable simultaneous enabling GTADTR buffer operations 1: Enable GTADTR register buffer operations simultaneously	R/W
3	SBDDE	GTDV Register Buffer Operation Simultaneous Enable 0: Disable simultaneous enabling GTDV buffer operations 1: Enable GTDV register buffer operations simultaneously	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
8	SBDCD	GTCCR Register Buffer Operation Simultaneous Disable 0: Disable simultaneous disabling GTCCR buffer operations 1: Disable GTCCR register buffer operations simultaneously	R/W
9	SBDPD	GTPR Register Buffer Operation Simultaneous Disable 0: Disable simultaneous disabling GTPR buffer operations 1: Disable GTPR register buffer operations simultaneously	R/W*1
10	SBDAD	GTADTR Register Buffer Operation Simultaneous Disable 0: Disable simultaneous disabling GTADTR buffer operations 1: Disable GTADTR register buffer operations simultaneously	R/W
11	SBDDD	GTDV Register Buffer Operation Simultaneous Disable 0: Disable simultaneous disabling GTDV buffer operations 1: Disable GTDV register buffer operations simultaneously	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W
16	SPCE	Period Count Function Simultaneous Enable*2 0: Disable simultaneous enabling period count function 1: Enable period count function simultaneously	R/W
17	SSCE	Synchronous Set/Clear Simultaneous Enable 0: Disable simultaneous enabling synchronous set/clear 1: Enable synchronous set/clear simultaneously	R/W*1
23:18	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
24	SPCD	Period Count Function Simultaneous Disable*2 0: Disable simultaneous disabling period count function 1: Disable period count function simultaneously	R/W
25	SSCD	Synchronous Set/Clear Simultaneous Disable 0: Disable simultaneous disabling synchronous set/clear 1: Disable synchronous set/clear simultaneously	R/W*1
31:26	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. When in complementary PWM mode, no matter which register of master channel/slave channel 1/slave channel 2 is written, three channels are written at the same time.
Note 2. This bit is only available in GPT320 to GPT323.
In GPT324 to GPT329, this bit is read as 0. The write value should be 0.

The GTSECR register simultaneously updates the value for operation enable bits of a channel set by the GTSECSR register.

Writing 1 to a bit in the GTSECR register in any channel and updating it updates an operation enable bit for all channels, related to the position of the bit written with 1 by the all GTSECSR registers.

Setting enable and disable bits for the same operation enable bit to 1 in the GTSECR is prohibited.

A bit written to 1 is automatically cleared. When the GTSECR is read, 0 is read.

Access in 8-bit or 16-bit units to the GTSECR register is prohibited, and it should be accessed in 32-bit units.

SBDCE bit (GTCCR Register Buffer Operation Simultaneous Enable)

When 1 is written to this bit, 0 is simultaneously set to a GTBER.BD[0] bit in the channels set to 1 by the GTSECSR register, and buffer operations using the GTCCRA, GTCCRC, and GTCCRD registers and using the GTCCRB, GTCCRE, and GTCCRF registers are enabled.

Simultaneous setting of SBDCE and SBDCD bits to 1 is prohibited.

SBDPE bit (GTPR Register Buffer Operation Simultaneous Enable)

When 1 is written to this bit, 0 is simultaneously set to a GTBER.BD[1] bit in the channels set to 1 by the GTSECSR register, and buffer operations using the GTPR, GTPDBR, and GTPBR registers are enabled.

Simultaneous setting of SBDPE and SBDPD bits to 1 is prohibited.

SBDAE bit (GTADTR Register Buffer Operation Simultaneous Enable)

When 1 is written to this bit, 0 is simultaneously set to a GTBER.BD[2] bit in the channels set to 1 by the GTSECSR register, and buffer operations using the GTADTRA, GTADTBRA, and GTADTDBRA registers and using the GTADTRB, GTADTBRB, and GTADTDBRB registers are enabled.

Simultaneous setting of SBDAE and SBDAD bits to 1 is prohibited.

SBDDE bit (GTDV Register Buffer Operation Simultaneous Enable)

When 1 is written to this bit, 0 is simultaneously set to a GTBER.BD[3] bit in the channels set to 1 by the GTSECSR register, and buffer operations using the GTDVU and GTDBU registers and using the GTDVD and GTDBD registers are enabled.

Simultaneous setting of SBDDE and SBDDD bits to 1 is prohibited.

SBDCD bit (GTCCR Register Buffer Operation Simultaneous Disable)

When 1 is written to this bit, 1 is simultaneously set to a GTBER.BD[0] bit in the channels set to 1 by the GTSECSR register, and buffer operations using the GTCCRA, GTCCRC, and GTCCRD registers and using the GTCCRB, GTCCRE, and GTCCRF registers are disabled.

Simultaneous setting of SBDCE and SBDCD bits to 1 is prohibited.

SBDPD bit (GTPR Register Buffer Operation Simultaneous Disable)

When 1 is written to this bit, 1 is simultaneously set to a GTBER.BD[1] bit in the channels set to 1 by the GTSECSR register, and buffer operations using the GTPR, GTPDBR, and GTPBR registers are disabled.

Simultaneous setting of SBDPE and SBDPD bits to 1 is prohibited.

SBDAD bit (GTADTR Register Buffer Operation Simultaneous Disable)

When 1 is written to this bit, 1 is simultaneously set to a GTBER.BD[2] bit in the channels set to 1 by the GTSECSR register, and buffer operations using the GTADTRA, GTADTBRA, and GTADTDBRA registers and using the GTADTRB, GTADTBRB, and GTADTDBRB registers are disabled.

Simultaneous setting of SBDAE and SBDAD bits to 1 is prohibited.

SBDDD bit (GTDV Register Buffer Operation Simultaneous Disable)

When 1 is written to this bit, 1 is simultaneously set to a GTBER.BD[3] bit in the channels set to 1 by the GTSECSR register, and buffer operations using the GTDVU and GTDBU registers and using the GTDVD and GTDBD registers are disabled.

Simultaneous setting of SBDDE and SBDDD bits to 1 is prohibited.

SPCE bit (Period Count Function Simultaneous Enable)

When 1 is written to this bit, 1 is simultaneously set to GTPC.PCEN bit in the channels set to 1 by the GTSECSR register, and period count function is enabled.

Simultaneous setting of SPCE and SPCD bits to 1 is prohibited.

SSCE bit (Synchronous Set/Clear Simultaneous Enable)

When 1 is written to this bit, 1 is simultaneously set to GTCR.SSCEN bit in the channels set to 1 by the GTSECSR register and Enable synchronous set/clear function.

Simultaneous setting of SSCE and SSCD bits to 1 is prohibited.

SPCD bit (Period Count Function Simultaneous Disable)

When 1 is written to this bit, 0 is simultaneously set to GTPC.PCEN bit in the channels set to 1 by the GTSECSR register, and period count function is disabled.

Simultaneous setting of SPCE and SPCD bits to 1 is prohibited.

SSCD bit (Synchronous Set/Clear Simultaneous Disable)

When 1 is written to this bit, 0 is simultaneously set to GTCR.SSCEN bit in the channels set to 1 by the GTSECSR register and Disable synchronous set/clear function.

Simultaneous setting of SSCE and SSCD bits to 1 is prohibited.

21.2.43 GTBER2 : General PWM Timer Buffer Enable Register 2

Base address: GPT32n = 0x4016_9000 + 0x0100 × n (n = 0 to 9)

Offset address: 0xE0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	OLTTB[1:0]		OLTTA[1:0]		CPBTD	CP3DB	—		CPTDV	CPTADB	CPTADA	CPTPR	CPTCB	CPTCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	CMTADB	CMTADA	—	CMTCB[1:0]		CMTCA[1:0]		—	—	CCTDV	CCTADB	CCTADA	CCTPR	CCTCB	CCTCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CCTCA	Counter Clear Source GTCCRA Register Buffer Transfer Disable 0: Enable GTCCRA register buffer transfer by counter clear 1: Disable GTCCRA register buffer transfer by counter clear	R/W
1	CCTCB	Counter Clear Source GTCCRB Register Buffer Transfer Disable 0: Enable GTCCRB register buffer transfer by counter clear 1: Disable GTCCRB register buffer transfer by counter clear	R/W

Bit	Symbol	Function	R/W
2	CCTPR	Counter Clear Source GTPR Register Buffer Transfer Disable 0: Enable GTPR register buffer transfer by counter clear 1: Disable GTPR register buffer transfer by counter clear	R/W
3	CCTADA	Counter Clear Source GTADTRA Register Buffer Transfer Disable 0: Enable GTADTRA register buffer transfer by counter clear 1: Disable GTADTRA register buffer transfer by counter clear	R/W
4	CCTADB	Counter Clear Source GTADTRB Register Buffer Transfer Disable 0: Enable GTADTRB register buffer transfer by counter clear 1: Disable GTADTRB register buffer transfer by counter clear	R/W
5	CCTDV	Counter Clear Source GTDVU/GTDVD Register Buffer Transfer Disable 0: Enable GTDVU/GTDVD register buffer transfer by counter clear 1: Disable GTDVU/GTDVD register buffer transfer by counter clear	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
9:8	CMTCA[1:0]	Compare Match Source GTCCRA Register Buffer Transfer Enable 0 0: Disable GTCCRA register Buffer Transfer by compare match of GTCCRA register and GTCCRB register 0 1: Enable GTCCRA register Buffer Transfer by compare match of GTCCRA register 1 0: Enable GTCCRA register Buffer Transfer by compare match of GTCCRB register 1 1: Enable GTCCRA register Buffer Transfer by compare match of GTCCRA register and GTCCRB register	R/W
11:10	CMTCB[1:0]	Compare Match Source GTCCRB Register Buffer Transfer Enable 0 0: Disable GTCCRB register Buffer Transfer by compare match of GTCCRA register and GTCCRB register 0 1: Enable GTCCRB register Buffer Transfer by compare match of GTCCRA register 1 0: Enable GTCCRB register Buffer Transfer by compare match of GTCCRB register 1 1: Enable GTCCRB register Buffer Transfer by compare match of GTCCRA register and GTCCRB register	R/W
12	—	This bit is read as 0. The write value should be 0.	R/W
13	CMTADA	Compare Match Source GTADTRA Register Buffer Transfer Enable 0: Disable GTADTRA register buffer transfer by compare match of GTADTRA register 1: Enable GTADTRA register buffer transfer by compare match of GTADTRA register	R/W
14	CMTADB	Compare Match Source GTADTRB Register Buffer Transfer Enable 0: Disable GTADTRB register buffer transfer by compare match of GTADTRB register 1: Enable GTADTRB register buffer transfer by compare match of GTADTRB register	R/W
15	—	This bit is read as 0. The write value should be 0.	R/W
16	CPTCA	Overflow/Underflow Source GTCCRA Register Buffer Transfer Disable 0: Enable GTCCRA register buffer transfer by overflow/underflow 1: Disable GTCCRA register buffer transfer by overflow/underflow	R/W
17	CPTCB	Overflow/Underflow Source GTCCRB Register Buffer Transfer Disable 0: Enable GTCCRB register buffer transfer by overflow/underflow 1: Disable GTCCRB register buffer transfer by overflow/underflow	R/W
18	CPTPR	Overflow/Underflow Source GTPR Register Buffer Transfer Disable 0: Enable GTPR register buffer transfer by overflow/underflow 1: Disable GTPR register buffer transfer by overflow/underflow	R/W
19	CPTADA	Overflow/Underflow Source GTADTRA Register Buffer Transfer Disable 0: Enable GTADTRA register buffer transfer by overflow/underflow 1: Disable GTADTRA register buffer transfer by overflow/underflow	R/W
20	CPTADB	Overflow/Underflow Source GTADTRB Register Buffer Transfer Disable 0: Enable GTADTRB register buffer transfer by overflow/underflow 1: Disable GTADTRB register buffer transfer by overflow/underflow	R/W
21	CPTDV	Overflow/Underflow Source GTDVU/GTDVD Register Buffer Transfer Disable 0: Enable GTDVU/GTDVD register buffer transfer by overflow/underflow 1: Disable GTDVU/GTDVD register buffer transfer by overflow/underflow	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
24	CP3DB	Complementary PWM mode 3,4 Double Buffer select*1 0: Disable double buffer function in complementary PWM mode 3, 4 1: Enable double buffer function in complementary PWM mode 3, 4	R/W
25	CPBTD	Complementary PWM mode Buffer Transfer Disable*1 0: Enable buffer transfer from temporary register to GTCCRC and GTPBR register 1: Disable buffer transfer from temporary register to GTCCRC and GTPBR register	R/W
27:26	OLTTA[1:0]	GTIOCnA Output Level Buffer Transfer Timing Select*1 0 0: No transfer 0 1: Triangle waves, complementary PWM mode: Transfer at crest Saw waves: Transfer at the end of period 1 0: Triangle waves, complementary PWM mode: Transfer at trough Saw waves: Transfer by compare match of GTCCRA register 1 1: Triangle waves, complementary PWM mode: Transfer at both crest and trough Saw waves: Setting prohibited	R/W
29:28	OLTTB[1:0]	GTIOCnB Output Level Buffer Transfer Timing Select*1 0 0: No transfer 0 1: Triangle waves, complementary PWM mode: Transfer at crest Saw waves: Transfer at the end of period 1 0: Triangle waves, complementary PWM mode: Transfer at trough Saw waves: Transfer by compare match of GTCCRB register 1 1: Triangle waves, complementary PWM mode: Transfer at both crest and trough Saw waves: Setting prohibited	R/W
31:30	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. This bit is only available in GPT324 to GPT329.
In GPT320 to GPT323, this bit is read as 0. The write value should be 0.

The GTBER2 register makes settings for buffer operation.

Set the CP3DB bit and OLTTm[1:0] (m = A, B) bits when the GTCNT counter is stopped.

CCTCA bit (Counter Clear Source GTCCRA Register Buffer Transfer Disable)

This bit disables buffer transfer by counter clear using the GTCCRA, GTCCRC, and GTCCRD registers together.

This bit is effective when the GTBER.BD[0] bit is 0 (buffer operation enabled) and the buffer operation is selected by the GTBER.CCRA[1:0] bits with saw-waves.

The setting is invalid in triangle waves or complementary PWM mode. Even if the CCTCA bit is 0, buffer transfer is not performed by counter clear.

If there is a conflict with the CMTCA bit setting, the CCTCA bit setting has priority.

The setting is invalid during the event count operation.

CCTCB bit (Counter Clear Source GTCCRB Register Buffer Transfer Disable)

This bit disables buffer transfer by counter clear using the GTCCRB, GTCCRE, and GTCCRF registers together.

This bit is effective when the GTBER.BD[0] bit is 0 (buffer operation enabled) and the buffer operation is selected by the GTBER.CCRB[1:0] bits with saw-waves.

The setting is invalid in triangle waves or complementary PWM mode. Even if the CCTCB bit is 0, buffer transfer is not performed by counter clear.

If there is a conflict with the CMTCB bit setting, the CCTCB bit setting has priority.

The setting is invalid during the event count operation.

CCTPR bit (Counter Clear Source GTPR Register Buffer Transfer Disable)

This bit disables buffer transfer by counter clear using the GTPR, GTPBR, and GTPDBR registers together.

This bit is effective when the GTBER.BD[1] bit is 0 (buffer operation enabled) and the buffer operation is selected by the GTBER.PR[1:0] bits with saw-waves.

The setting is invalid in triangle waves or complementary PWM mode. Even if the CCTPR bit is 0, buffer transfer is not performed by counter clear.

The setting is invalid during the event count operation.

CCTADm bit (Counter Clear Source GTADTRm Register Buffer Transfer Disable) (m = A, B)

This bit disables buffer transfer by counter clear using the GTADTRm, GTADTBRm and GTADTDBRm registers together.

This bit is effective when the GTBER.BD[2] bit is 0 (buffer operation enabled) and the buffer operation is selected by the ADTTm[1:0] bits with saw-waves.

The setting is invalid in triangle waves or complementary PWM mode. Even if the CCTADm bit is 0, buffer transfer is not performed by counter clear.

If there is a conflict with the CMTADm bit setting, the CCTADm bit setting has priority.

The setting is invalid during the event count operation.

CCTDV bit (Counter Clear Source GTDVU/GTDVD Register Buffer Transfer Disable)

This bit disables buffer transfer by counter clear using the GTDVU and GTDBU or GTDVD and GTDBD registers together.

This bit is effective when the GTBER.BD[3] bit is 0 (buffer operation enabled) and the buffer operation is selected by the GTDTCR.TDBUE or GTDTCR.TDBDE bit with saw-waves.

The setting is invalid in triangle waves or complementary PWM mode. Even if the CCTDV bit is 0, buffer transfer is not performed by counter clear.

The setting is invalid during the event count operation.

CMTCA[1:0] bit (Compare Match Source GTCCRA Register Buffer Transfer Enable)

This bit enables buffer transfer by compare match of GTCCRA using the GTCCRA, GTCCRC, and GTCCRD registers together.

This bit is effective when the GTBER.BD[0] bit is 0 (buffer operation enabled) and the buffer operation is selected by the GTBER.CCRA[1:0] bits with saw-waves.

The setting is invalid in triangle waves or complementary PWM mode.

If there is a conflict with the CCTCA bit setting, the CCTCA bit setting has priority.

The setting is invalid during the event count operation.

CMTCB[1:0] bit (Compare Match Source GTCCRB Register Buffer Transfer Enable)

This bit enables buffer transfer by compare match of GTCCRB using the GTCCRB, GTCCRE, and GTCCRF registers together.

This bit is effective when the GTBER.BD[0] bit is 0 (buffer operation enabled) and the buffer operation is selected by the GTBER.CCRB[1:0] bits with saw-waves.

The setting is invalid in triangle waves or complementary PWM mode.

If there is a conflict with the CCTCB bit setting, the CCTCB bit setting has priority.

The setting is invalid during the event count operation.

CMTADm bit (Compare Match Source GTADTRm Register Buffer Transfer Enable) (m = A, B)

This bit enables buffer transfer by compare match of GTADTRm using the GTADTRm, GTADTBRm and GTADTDBRm registers together.

This bit is effective when the GTBER.BD[2] bit is 0 (buffer operation enabled) and the buffer operation is selected by the ADTTm[1:0] bits with saw-waves.

The setting is invalid in triangle waves or complementary PWM mode.

If there is a conflict with the CCTADm bit setting, the CCTADm bit setting has priority.

The setting is invalid during the event count operation.

CPTCA bit (Overflow/Underflow Source GTCCRA Register Buffer Transfer Disable)

This bit disables buffer transfer by overflow/underflow in saw-waves using the GTCCRA, GTCCRC, and GTCCRD registers together.

This bit is effective when the CCTCA bit is 0 (GTCCRA register buffer transfer by counter clear is enabled), the GTBER.BD[0] bit is 0 (buffer operation enabled) and the buffer operation is selected by the GTBER.CCRA[1:0] bits with saw-waves.

The setting is invalid in triangle waves or complementary PWM mode.

The setting is invalid during the event count operation.

CPTCB bit (Overflow/Underflow Source GTCCRB Register Buffer Transfer Disable)

This bit disables buffer transfer by overflow/underflow in saw-waves using the GTCCRB, GTCCRE and GTCCRF registers together.

This bit is effective when the CCTCB bit is 0 (GTCCRB register buffer transfer by counter clear is enabled), the GTBER.BD[0] bit is 0 (buffer operation enabled) and the buffer operation is selected by the GTBER.CCRB[1:0] bits with saw-waves.

The setting is invalid in triangle waves or complementary PWM mode.

The setting is invalid during the event count operation.

CPTPR bit (Overflow/Underflow Source GTPR Register Buffer Transfer Disable)

This bit disables buffer transfer by overflow/underflow in saw-waves using the GTPR, GTPBR, and GTPDBR registers together.

This bit is effective when the CCTPR bit is 0 (GTPR register buffer transfer by counter clear is enabled), the GTBER.BD[1] bit is 0 (buffer operation enabled) and the buffer operation is selected by the GTBER.PR[1:0] bits with saw-waves.

The setting is invalid in triangle waves or complementary PWM mode.

The setting is invalid during the event count operation.

CPTADm bit (Overflow/Underflow Source GTADTRm Register Buffer Transfer Disable) (m = A, B)

This bit disables buffer transfer by overflow/underflow in saw-waves using the GTADTRm, GTADTBRm and GTADTDBRm registers together.

This bit is effective when the CCTADm bit is 0 (GTADTRm register buffer transfer by counter clear is enabled), the GTBER.BD[2] bit is 0 (buffer operation enabled) and the buffer operation is selected by the ADTTm[1:0] bits with saw-waves.

The setting is invalid in triangle waves or complementary PWM mode.

The setting is invalid during the event count operation.

CPTDV bit (Overflow/Underflow Source GTDVU/GTDVD Register Buffer Transfer Disable)

This bit disables buffer transfer by overflow/underflow in saw-waves using the GTDVU and GTDBU or GTDVD and GTDBD registers together.

This bit is effective when the CCTDV bit is 0 (GTDVU/GTDVD register buffer transfer by counter clear is enabled), the GTBER.BD[3] bit is 0 (buffer operation enabled) and the buffer operation is selected by the GTDTCR.TDBUE or GTDTCR.TDBDE bit with saw-waves.

The setting is invalid in triangle waves or complementary PWM mode.

The setting is invalid during the event count operation.

CP3DB bit (Complementary PWM mode 3,4 Double Buffer select)

This bit enables buffer transfer using the GTCCRA, GTCCRE and GTCCRF registers together in complementary PWM mode 3,4.

CPBTD bit (Complementary PWM mode Buffer Transfer Disable)

This bit disables buffer transfer from temporary register(temporary register A and temporary register P) to the GTCCRC and GTPBR registers during timer counting in complementary PWM mode 1, 2 and 3. When CP3DB bit is 1, the buffer transfer from temporary register B to GTCCRE register is also disabled. The setting is invalid in complementary PWM mode 4.

OLTTm[1:0] bits (GTIOCnA Output Level Buffer Transfer Timing Select) (m = A, B)

This bit set the timing of buffer transfer from GTOLBR.GTIOmB[4:0] bits to GTIOR.GTIOm[4:0] bits.

21.2.44 GTOLBR : General PWM Timer Output Level Buffer Register

Base address: GPT32n = 0x4016_9000 + 0x0100 × n (n = 4 to 9)

Offset address: 0xE4

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

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

Bit	Symbol	Function	R/W
4:0	GTIOAB[4:0]	GTIOA buffer bits	R/W
15:5	—	These bits are read as 0. The write value should be 0.	R/W
20:16	GTIOBB[4:0]	GTIOB buffer bits	R/W
31:21	—	These bits are read as 0. The write value should be 0.	R/W

The GTOLBR register is a buffer register for the GTIOR.GTIOA[4:0] bits and GTIOR.GTIOB[4:0] bits.

GTIOmB[4:0] bits (GTIOm buffer bits) (m = A, B)

These bits are buffer bits of GTIOR.GTIOm[4:0] bits.

These bits are transferred to the GTIOR.GTIOm[4:0] bits at the transfer timing selected by the GTBER2.OLTTm[1:0] (m = A, B) bits.

21.2.45 GTICCR : General PWM Timer Inter Channel Cooperation Input Capture Control Register

Base address: GPT32n = 0x4016_9000 + 0x0100 × n (n = 0 to 9)

Offset address: 0xEC

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	ICBGRP[1:0]		—	—	—	—	—	ICBCLK	ICBFPU	ICBFPO	ICBFF	ICBFE	ICBFD	ICBFC	ICBFB	ICBFA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ICAGRP[1:0]		—	—	—	—	—	ICACLK	ICAFPU	ICAFPO	ICAFF	ICAFE	ICAFD	ICAFC	ICAFB	ICAFA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ICAFA	Forwarding GTCCRA register Compare Match/Input Capture to Other Channel GTCCRA Input Capture Source Enable 0: Disable forwarding GTCCRA register compare match/input capture to GTCCRA input capture source of other channels 1: Enable forwarding GTCCRA register compare match/input capture to GTCCRA input capture source of other channels	R/W

Bit	Symbol	Function	R/W
1	ICAFB	Forwarding GTCCRB register Compare Match/Input Capture to Other Channel GTCCRA Input Capture Source Enable 0: Disable forwarding GTCCRB register compare match/input capture to GTCCRA input capture source of other channels 1: Enable forwarding GTCCRB register compare match/input capture to GTCCRA input capture source of other channels	R/W
2	ICAFC	Forwarding GTCCRC register Compare Match to Other Channel GTCCRA Input Capture Source Enable 0: Disable forwarding GTCCRC register compare match to GTCCRA input capture source of other channels 1: Enable forwarding GTCCRC register compare match to GTCCRA input capture source of other channels	R/W
3	ICAFD	Forwarding GTCCRD register Compare Match to Other Channel GTCCRA Input Capture Source Enable 0: Disable forwarding GTCCRD register compare match to GTCCRA input capture source of other channels 1: Enable forwarding GTCCRD register compare match to GTCCRA input capture source of other channels	R/W
4	ICAFE	Forwarding GTCCRE register Compare Match to Other Channel GTCCRA Input Capture Source Enable 0: Disable forwarding GTCCRE register compare match to GTCCRA input capture source of other channels 1: Enable forwarding GTCCRE register compare match to GTCCRA input capture source of other channels	R/W
5	ICAFF	Forwarding GTCCRF register Compare Match to Other Channel GTCCRA Input Capture Source Enable 0: Disable forwarding GTCCRF register compare match to GTCCRA input capture source of other channels 1: Enable forwarding GTCCRF register compare match to GTCCRA input capture source of other channels	R/W
6	ICAFPO	Forwarding Overflow to Other Channel GTCCRA Input Capture Source Enable 0: Disable forwarding overflow in saw-waves or the crest in triangle-waves or complementary PWM mode to GTCCRA input capture source of other channels 1: Enable forwarding overflow in saw-waves or the crest in triangle-waves or complementary PWM mode to GTCCRA input capture source of other channels	R/W
7	ICAFPU	Forwarding Underflow to Other Channel GTCCRA Input Capture Source Enable 0: Disable forwarding underflow in saw-waves or the trough in triangle-waves or complementary PWM mode to GTCCRA input capture source of other channels 1: Enable forwarding underflow in saw-waves or the trough in triangle-waves or complementary PWM mode to GTCCRA input capture source of other channels	R/W
8	ICACLK	Forwarding Count Clock to Other Channel GTCCRA Input Capture Source Enable 0: Disable forwarding count clock to GTCCRA input capture source of other channels 1: Enable forwarding count clock to GTCCRA input capture source of other channels	R/W
13:9	—	These bits are read as 0. The write value should be 0.	R/W
15:14	ICAGRP[1:0]	GTCCRA Input Capture Group Select 0 0: Select group A 0 1: Select group B 1 0: Select group C 1 1: Select group D	R/W
16	ICBFA	Forwarding GTCCRA register Compare Match/Input Capture to Other Channel GTCCRB Input Capture Source Enable 0: Disable forwarding GTCCRA register compare match/input capture to GTCCRB input capture source of other channels 1: Enable forwarding GTCCRA register compare match/input capture to GTCCRB input capture source of other channels	R/W

Bit	Symbol	Function	R/W
17	ICBFB	Forwarding GTCCRB register Compare Match/Input Capture to Other Channel GTCCRB Input Capture Source Enable 0: Disable forwarding GTCCRB register compare match/input capture to GTCCRB input capture source of other channels 1: Enable forwarding GTCCRB register compare match/input capture to GTCCRB input capture source of other channels	R/W
18	ICBFC	Forwarding GTCCRC register Compare Match to Other Channel GTCCRB Input Source Capture Enable 0: Disable forwarding GTCCRD register compare match to GTCCRB input capture source of other channels 1: Enable forwarding GTCCRD register compare match to GTCCRB input capture source of other channels	R/W
19	ICBFD	Forwarding GTCCRD register Compare Match to Other Channel GTCCRB Input Capture Source Enable 0: Disable forwarding GTCCRD register compare match to GTCCRB input capture source of other channels 1: Enable forwarding GTCCRD register compare match to GTCCRB input capture source of other channels	R/W
20	ICBFE	Forwarding GTCCRE register Compare Match to Other Channel GTCCRB Input Capture Source Enable 0: Disable forwarding GTCCRE register compare match to GTCCRB input capture source of other channels 1: Enable forwarding GTCCRE register compare match to GTCCRB input capture source of other channels	R/W
21	ICBFF	Forwarding GTCCRF register Compare Match to Other Channel GTCCRB Input Capture Source Enable 0: Disable forwarding GTCCRF register compare match to GTCCRB input capture source of other channels 1: Enable forwarding GTCCRF register compare match to GTCCRB input capture source of other channels	R/W
22	ICBFPO	Forwarding Overflow to Other Channel GTCCRB Input Capture Source Enable 0: Disable forwarding overflow in saw-waves or the crest in triangle-waves or complementary PWM mode to GTCCRB input capture source of other channels 1: Enable forwarding overflow in saw-waves or the crest in triangle-waves or complementary PWM mode to GTCCRB input capture source of other channels	R/W
23	ICBFPU	Forwarding Underflow to Other Channel GTCCRB Input Capture Source Enable 0: Disable forwarding underflow in saw-waves or the trough in triangle-waves or complementary PWM mode to GTCCRB input capture source of other channels 1: Enable forwarding underflow in saw-waves or the trough in triangle-waves or complementary PWM mode to GTCCRB input capture source of other channels	R/W
24	ICBCLK	Forwarding Count Clock to Other Channel GTCCRB Input Capture Source Enable 0: Disable forwarding count clock to GTCCRB input capture source of other channels 1: Enable forwarding count clock to GTCCRB input capture source of other channels	R/W
29:25	—	These bits are read as 0. The write value should be 0.	R/W
31:30	ICBGRP[1:0]	GTCCRB Input Capture Group Select 0 0: Select group A 0 1: Select group B 1 0: Select group C 1 1: Select group D	R/W

The GTICCR register is a register that controls input capture by inter channel cooperation.

For channels that perform input capture by inter channel cooperation, the forwarding enable bit of the input capture source corresponding to the GTCCRA register or GTCCRB register where the input capture occurs is invalid.

ICAFm bit (Forwarding GTCCRm register Compare Match/Input Capture to Other Channel GTCCRA Input Capture Source Enable) (m = A, B)

Enables/disables the use of compare match/input capture of GTCCRm register as the input capture source of other channel's GTCCRA register.

ICAFx bit (Forwarding GTCCRx register Compare Match to Other Channel GTCCRA Input Capture Source Enable) (x = C, D, E, F)

Enables/disables the use of compare match of GTCCRx register as the input capture source of other channel's GTCCRA register.

ICAFPO bit (Forwarding Overflow to Other Channel GTCCRA Input Capture Source Enable)

Enable/disable to use the overflow of saw-waves, the crest of triangle-waves or complementary PWM mode as the input capture source of other channel's GTCCRA register.

ICAFPU bit (Forwarding Underflow to Other Channel GTCCRA Input Capture Source Enable)

Enable/disable to use the underflow of saw-waves, the trough of triangle-waves or complementary PWM mode as the input capture source of other channel's GTCCRA register.

ICACLK bit (Forwarding Count Clock to Other Channel GTCCRA Input Capture Source Enable)

Enable/disable to use count clock as the input capture source of other channel's GTCCRA register.

ICAGRP[1:0] bit (GTCCRA Input Capture Group Select)

Select the group of input capture by inter channel cooperation for GTCCRA register.

For channels that accept input capture of the GTCCRA register due to input capture sources from other channels, set the GTICASR.ASOC bit to 1 and select the group of inter channel cooperation with the ICAGRP[1:0] bits.

ICBFm bit (Forwarding GTCCRm register Compare Match/Input Capture to Other Channel GTCCRB Input Capture Source Enable) (m = A, B)

Enables/disables the use of compare match/input capture of GTCCRm register as the input capture source of other channel's GTCCRB register.

ICBFx bit (Forwarding GTCCRx register Compare Match to Other Channel GTCCRB Input Capture Source Enable) (x = C, D, E, F)

Enables/disables the use of compare match of GTCCRx register as the input capture source of other channel's GTCCRB register.

ICBFPO bit (Forwarding Overflow to Other Channel GTCCRB Input Capture Source Enable)

Enable/disable to use the overflow of saw-waves, the crest of triangle-waves or complementary PWM mode as the input capture source of other channel's GTCCRB register.

ICBFPU bit (Forwarding Underflow to Other Channel GTCCRB Input Capture Source Enable)

Enable/disable to use the underflow of saw-waves, the trough of triangle-waves or complementary PWM mode as the input capture source of other channel's GTCCRB register.

ICBCLK bit (Forwarding Count Clock to Other Channel GTCCRB Input Capture Source Enable)

Enable/disable to use count clock as the input capture source of other channel's GTCCRB register.

ICBGRP[1:0] bit (GTCCRB Input Capture Group Select)

Select the group of input capture by inter channel cooperation for GTCCRB register.

For channels that accept input capture of the GTCCRB register due to input capture sources from other channels, set the GTICBSR.BSOC bit to 1 and select the group of inter channel cooperation with the ICBGRP[1:0] bits.

21.2.46 OPSCR : Output Phase Switching Control Register

Base address: GPT_OPS = 0x4016_9A00

Offset address: 0x00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	NFCS[1:0]		NFEN	—	—	GODF	GRP[1:0]		—	—	ALIGN	RV	INV	N	P	FB
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	EN	—	W	V	U	—	WF	VF	UF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	UF	Input Phase Soft Setting These bits set the input phase from software settings. Setting these bits is valid when OPSCR.FB = 1.	R/W
1	VF		R/W
2	WF		R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	U	Input U-Phase Monitor This bit monitors the state of the input phase. OPSCR.FB = 0 : External input that are synchronized by GTCLK OPSCR.FB = 1 : Software settings (UF)	R
5	V	Input V-Phase Monitor This bit monitors the state of the input phase. OPSCR.FB = 0 : External input that are synchronized by GTCLK OPSCR.FB = 1 : Software settings (VF)	R
6	W	Input W-Phase Monitor This bit monitors the state of the input phase. OPSCR.FB = 0 : External input that are synchronized by GTCLK OPSCR.FB = 1 : Software settings (WF)	R
7	—	This bit is read as 0. The write value should be 0.	R/W
8	EN	Output Phase Enable 0: Do not output (Hi-Z external pin) 1: Output*1	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W
16	FB	External Feedback Signal Enable This bit selects the input phase from software settings and external input. 0: Select the external input 1: Select the soft setting (OPSCR.UF, VF, WF)	R/W
17	P	Positive-Phase Output (P) Control 0: Level signal output 1: PWM signal output	R/W
18	N	Negative-Phase Output (N) Control 0: Level signal output 1: PWM signal output	R/W
19	INV	Output Phase Invert Control 0: Positive logic (active-high) output 1: Negative logic (active-low) output	R/W
20	RV	Output Phase Rotation Direction Reversal Control 0: Positive rotation 1: Reverse rotation	R/W
21	ALIGN	Input Phase Alignment 0: Input phase aligned to GTCLK 1: Input phase aligned to the falling edge of PWM	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
25:24	GRP[1:0]	Output Disabled Source Selection 0 0: Select group A output disable source 0 1: Select group B output disable source 1 0: Select group C output disable source 1 1: Select group D output disable source	R/W
26	GODF	Group Output Disable Function 0: This bit function is ignored 1: Group disable clears the OPSCR.EN bit*1	R/W
28:27	—	These bits are read as 0. The write value should be 0.	R/W
29	NFEN	External Input Noise Filter Enable 0: Do not use a noise filter on the external input 1: Use a noise filter on the external input	R/W
31:30	NFCS[1:0]	External Input Noise Filter Clock Selection Noise filter sampling clock setting of the external input. 0 0: GTCLK/1 0 1: GTCLK/4 1 0: GTCLK/16 1 1: GTCLK/64	R/W

Note 1. When OPSCR.GODF = 1 and the signal value selected by the OPSCR.GRP[1:0] bit is high, the OPSCR.EN bit is set to 0.

The OPSCR register sets the output of the signal waveform required for brushless DC motor control.

UF , VF , WF bits (Input Phase Soft Setting)

The UF , VF , WF bits set the input phase from the software settings. When OPSCR.FB bit is 1, these bits are valid. The set value of the UF /VF /WF takes the place of the U/V/W external input.

U, V, W bits (Input Phase Monitor)

When the OPSCR.FB bit is 0, external inputs that are synchronized by GTCLK are monitored by these bits. When the OPSCR.FB bit is 1, the OPSCR.U, OPSCR.V, and OPSCR.W bits can read the OPSCR.UF , OPSCR.VF , and OPSCR.WF bits.

EN bit (Output Phase Enable)

The EN bit controls the output enable signal of output phase (positive phase/negative phase).

When the OPSCR.EN bit is 1, the signal waveform is output.

When the OPSCR.EN bit is 0, first set OPSCR.FB, OPSCR.UF /VF /WF (software setting is selected), OPSCR.P/N, OPSCR.INV, OPSCR.RV, OPSCR.ALIGN, OPSCR.GRP[1:0], OPSCR.GODF, OPSCR.NFEN, OPSCR.NFCS. Then, set the EN bit to 1. The EN bit should be set when output disable request does not occur from POEG. Also when OPSCR.GODF is 1 and the signal value selected in the OPSCR.GRP[1:0] bit is high, the OPSCR.EN bit is set to 0. Even if 1 is written by software, the EN bit remains at 0.

For the return, after clearing the Output Disable Request by software, set the EN bit to 1.

Priority order of the EN bit is as follows (when the conflict occurs).

When writing 1 by software and clearing to 0 by the Output Disable Request conflict for the EN bit, clearing to 0 by the Output Disable Request is enabled.

FB bit (External Feedback Signal Enable)

The FB bit selects the input phase from the software settings (OPSCR.UF, VF, WF) and external input such as a Hall element.

P bit (Positive-Phase Output (P) Control)

The P bit selects one of the level signal output or PWM signal output for the positive-phase output (GTOUUP, GTOVUP and GTOWUP pins).

N bit (Negative-Phase Output (N) Control)

The N bit selects one of the level signal output or PWM signal output for the negative-phase output (GTOULO, GTOVLO and GTOWLO pins).

INV bit (Output Phase Invert Control)

The INV bit selects one of the positive logic (active-high) output or negative logic (active-low) output for the output phase.

RV bit (Output Phase Rotation Direction Reversal Control)

The RV bit reverses the direction of rotation of the motor by inverting the input phase.

ALIGN bit (Input Phase Alignment)

The ALIGN bit selects the GTCLK or PWM for the sampling of the input phase (input phase is specified in the OPSCR.FB bit).

When OPSCR.ALIGN bit is 0, input phase is aligned to GTCLK.

Note: When the chopping is performed, there are cases where the PWM width of output is shorter than the width of the PWM used to chop just before or after switching of output phase, depending on the phase difference between the phase output switch timing and the phase of PWM.

When OPSCR.ALIGN bit is 1, input phase is aligned with the falling edge of PWM.

GRP[1:0] bit (Output Disabled Source Selection)

The GRP[1:0] bit selects the output disable source.

The GRP bits should be set when GODF bit is 0. If GRP bits select a POEG except for the connected groups, the status of output pin never change to disable.

GODF bit (Group Output Disable Function)

When OPSCR.GODF is 1 and the signal value selected by the OPSCR.GRP[1:0] bit is high, the OPSCR.EN bit is set to 0.

When OPSCR.GODF bit is 0, this bit is ignored.

The GODF bit should be set when output disable request does not occur from POEG.

NFEN bit (External Input Noise Filter Enable)

The NFEN bit selects the noise filter for external input. When OPSCR.NFEN bit is 0, a noise filter for the external input is not used.

Note: Set this bit during the EN bit is 0 to avoid generation of unintentional internal edge caused by switching this bit.

NFCS[1:0] bits (External Input Noise Filter Clock Selection)

The NFCS[1:0] bits select the clock for the external input noise filter. When the OPSCR.NFEN bit is 1, noise filter sampling clock setting of the external input is enabled.

1. Set the NFCS[1:0].
2. Wait for 2 cycles.
3. Set the OPSCR.EN bit to 1.

21.2.47 GTCLKCR : General PWM Timer Clock Control Register

Base address: GPT_GTCLK = 0x4016_9B00

Offset address: 0x00

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

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

Bit	Symbol	Function	R/W
0	BPEN	Synchronization Circuit Bypass Enable 0: In case of using Bus Clock and GPT Core Clock asynchronously 1: In case of using Bus Clock and GPT Core Clock synchronously	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

The GTCLKCR register is a register that controls the clock.

Set first of initial setting after resetting.

If MSTPCRE.MSTPE31 bit is 0, changing this register is prohibited.

BPEN bit (Synchronization Circuit Bypass Enable)

Enable or disable the synchronization bypass function between the bus clock (PCLKA) and GPT core clock (GTCLK).

Set 0 when using an asynchronous clock (GPTCLK) or 1 when using a synchronous clock (PCLKD) as the core clock.

21.3 Operation

21.3.1 Basic Operation

Each channel has a 32-bit timer that performs a periodic count operation using the count clock and hardware sources. The count function provides both up-counting and down-counting. The GTPR or GTCCRm (m = A to F) controls the count cycle.

When the GTCNT counter value matches the value in GTCCRA or GTCCRB, the output from the associated GTIOCnA or GTIOCnB can be changed (n = 0 to 9). GTCCRA or GTCCRB can be used as an input capture register with hardware resources.

GTCCRC and GTCCRD can function as buffer registers for GTCCRA. GTCCRE and GTCCRF can function as buffer registers for GTCCRB.

21.3.1.1 Counter operation

(1) Counter start and stop

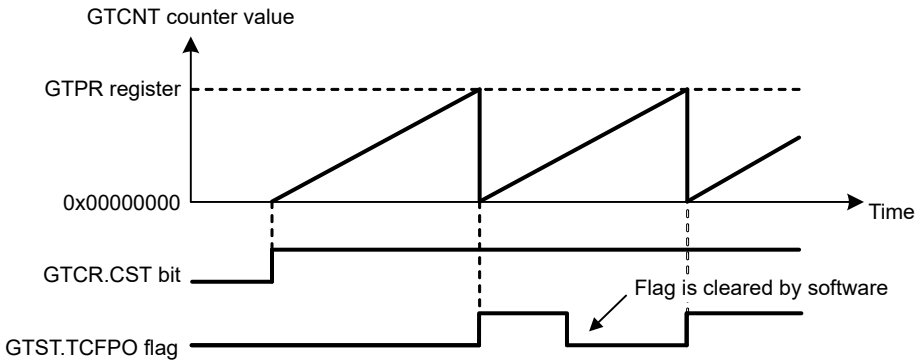
The counter of each channel starts the count operation when GTCR.CST is set to 1, and stops counting when the bit is set to 0. The GTCR.CST bit value is changed by the following sources:

- Writing to GTCR register
- Writing 1 to the bit in GTSTR associated with the GPT channel number when the GTSSR.CSTRT bit set to 1
- Writing 1 to the bit in GTSTP associated with the GPT channel number when the GTPSR.CSTOP bit set to 1
- The hardware source selected in the GTSSR register
- The hardware source selected in the GTPSR register
- Completion of the period count function while the GTPC.ASTP bit is 1

(2) Periodic count operation in up-counting by count clock

The GTCNT counter in each channel starts up-counting when the associated GTCR.CST bit is set to 1 with GTUPSR and GTDNSR registers set to 0x00000000. When the GTCNT value changes from the GTPR value to 0 (overflow) or the GTCCRm (m = A to F) value selected by the GTCSR.CSCMSC[2:0] bits to 0 in Saw-wave PWM mode 2, the GTST.TCFPO flag is set to 1, and the overflow interrupt(GPTn_OVF) is also generated. After GTCNT overflows, up-counting resumes from 0x00000000.

Figure 21.5 shows an example of a periodic count operation in up-counting by the count clock.



Note: The base address of registers are the same.

Figure 21.5 Example of periodic count operation in up-counting by the count clock

Table 21.11 shows an example for setting periodic count operation in up-counting by the count clock.

Table 21.11 Example for setting a periodic count operation in up-counting by the count clock

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits or GTCR.MD[3:0] bits. In Figure 21.5, 000b or 0000b (saw-wave PWM mode 1) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 21.5, after 11b is set in the GTUDDTYC[1:0] bits, 01b is set in the GTUDDTYC[1:0] bits (up-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Other than the saw-wave PWM mode2, set the cycle in the GTPR register. In the saw-wave PWM mode2, select the counter clear source compare match register GTCCRx (x = A to F) by GTCSR.CSCMSC[2:0] bits and set the cycle in that register.
5	Set initial value for counter	Set the initial value in the GTCNT counter. In Figure 21.5, 0x00000000 is set.
6	Start count operation	Set the GTCR.CST bit to 1 to start count operation.

(3) Periodic count operation in down-counting by count clock

The GTCNT counter in each channel can perform down-counting by setting GTUDDTYC.UD with GTUPSR and GTDNSR registers set to 0x00000000. When GTCNT changes from 0 to the GTPR value (underflow), GTST.TCFPU is set to 1, and the underflow interrupt(GPTn_UDF) is also generated. After the GTCNT counter underflows, down-counting resumes from the GTPR value.

Figure 21.6 shows an example of periodic count operation in down-counting by the count clock.

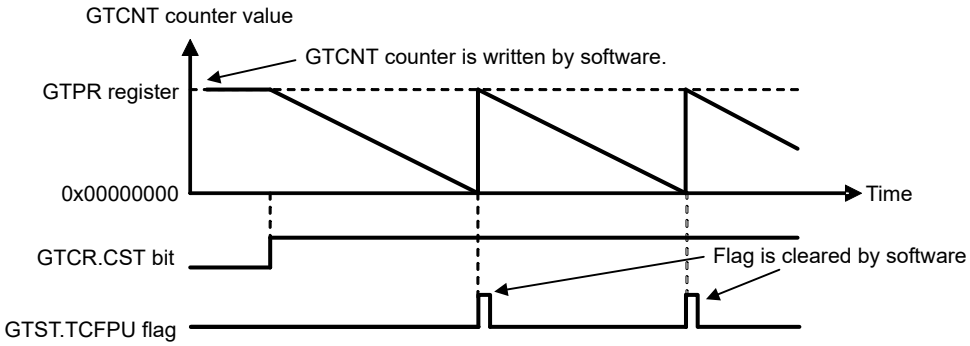


Figure 21.6 Example of periodic count operation in down-counting by the count clock

Table 21.12 shows an example for setting periodic count operation in down-counting by the count clock.

Table 21.12 Example for setting periodic count operation in down-counting by count clock

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits or GTCR.MD[3:0] bits. In Figure 21.6, 000b or 0000b (saw-wave PWM mode 1) is set.
2	Set count direction	Select the count direction with the GTUDDTYC register. In Figure 21.6, after 10b is set in the GTUDDTYC[1:0] bits, 00b is set in the GTUDDTYC[1:0] bits (down-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Set the cycle in the GTPR register.
5	Set initial value for counter	Set the initial value in the GTCNT counter. In Figure 21.6, the GTPR register value is set.
6	Start count operation	Set the GTCR.CST bit to 1 to start count operation. In Figure 21.6, 1 is set in the CST bit.

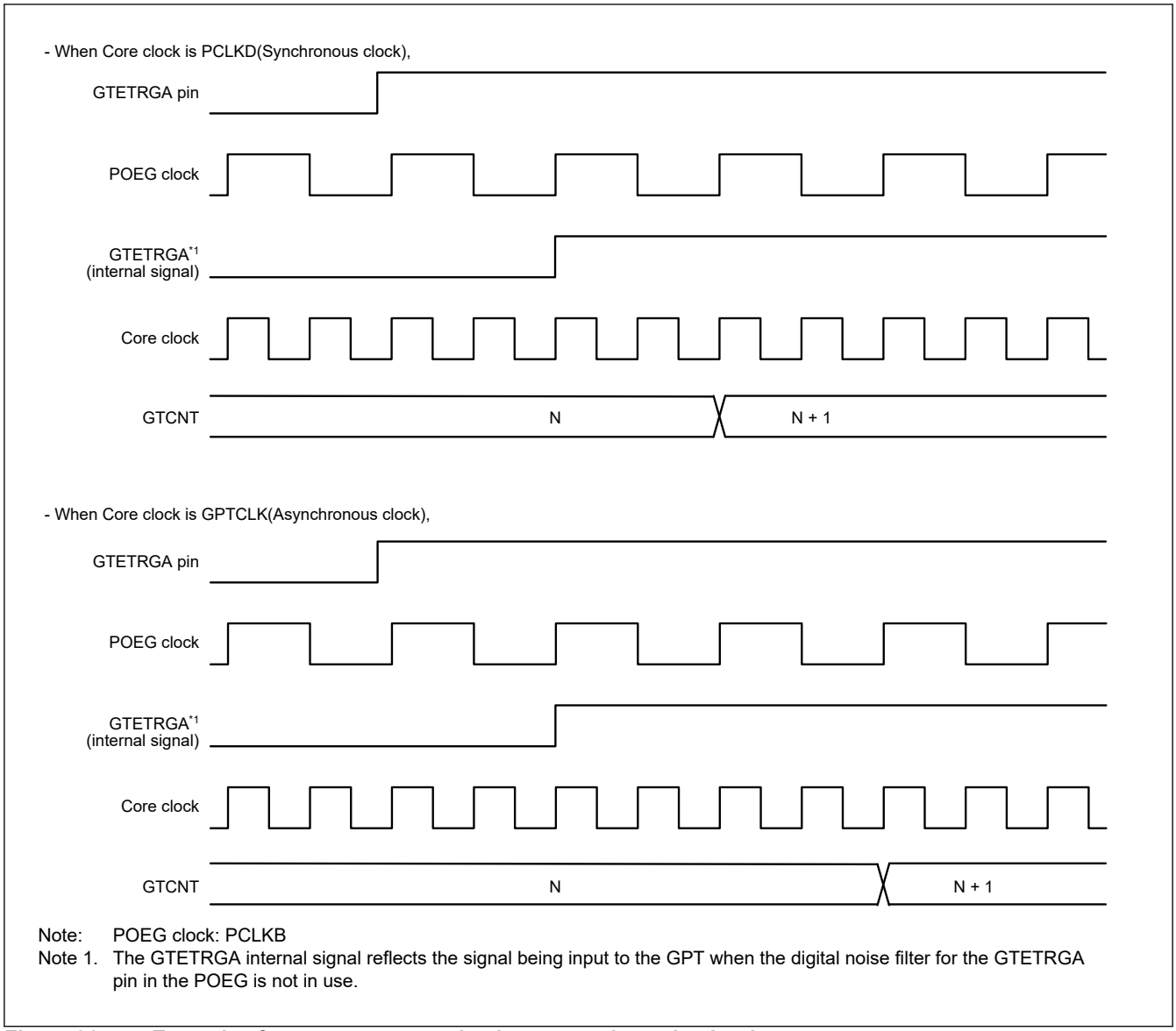
(4) Event count operation in up-counting using hardware sources

The GTCNT counter in each channel can perform up-counting using hardware sources as set in GTUPSR.

When GTUPSR is set to enable, the count clock selected in GTCR.TPCS[3:0] and the count direction selected in GTUDDTYC.UD are ignored. If up-counting and down-counting using hardware sources occur at the same time, the GTCNT counter value does not change. The overflow behavior when up-counting using hardware sources is the same as when up-counting by the count clock.

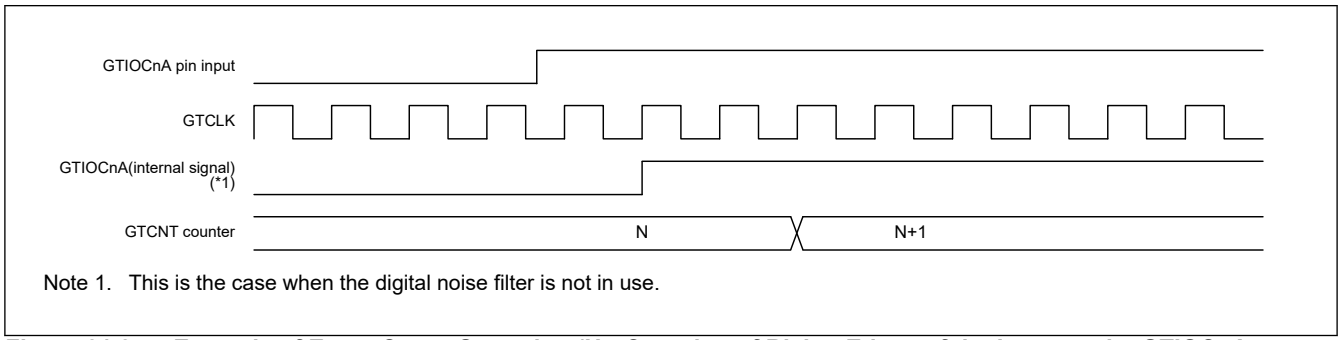
If you are using a hardware source to count up, set the GTCR.CST bit to 1 to enable the counting operation. After GTCR.CST is set to 1, the counter cannot count up for 1 clock cycle as specified in GTCR.TPCS[3:0] because the count operation is synchronized by the count clock selected in GTCR.TPCS[3:0]. Set GTCR.TPCS[3:0] to 000b to count up with a 1 GTCLK delay after GTCR.CST is set to 1.

Figure 21.7 and Figure 21.8 show an example of an event count operation in up-counting by a hardware resource (the rising edge of GTETRGA pin input and the rising edge of GTIOCnA pin input).



Note: POEG clock: PCLKB
Note 1. The GTETRGA internal signal reflects the signal being input to the GPT when the digital noise filter for the GTETRGA pin in the POEG is not in use.

Figure 21.7 Example of event count operation in up-counting using hardware sources



Note 1. This is the case when the digital noise filter is not in use.

Figure 21.8 Example of Event Count Operation (Up-Counting of Rising Edges of the Input on the GTIOCnA Pin)

Figure 21.9 shows an example of event count operation by ELC_GPTA event input.

This is an example of event count operation of GPT321.GTCNT counter. An event signal is output to the ELC after matches in comparison with the GPT320.GTCCRA register. This is selected as a trigger for output to the GPT321 by the ELC as ELC_GPTA.

ELC passes the event signal output from GPT320 without delay to GPT321.

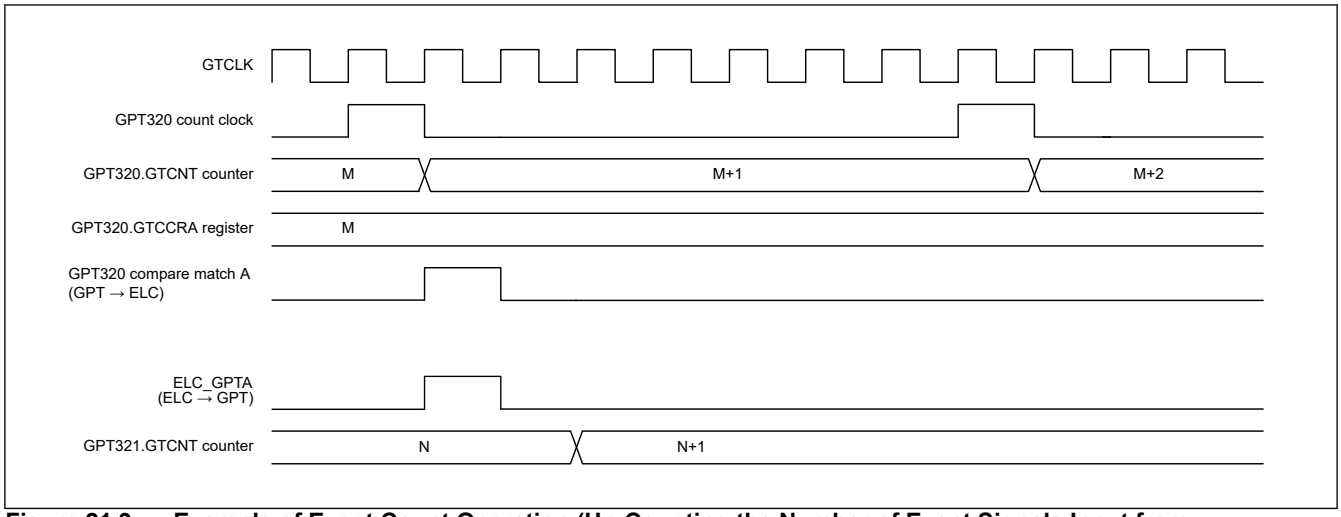


Figure 21.9 Example of Event Count Operation (Up-Counting the Number of Event Signals Input from ELC_GPTA)

Table 21.13 shows an example for setting event count operation in up-counting by a hardware source.

Table 21.13 Example for setting an event count operation in up-counting using hardware sources

No.	Step Name	Description
1	Set count source	Select the counting-up hardware source with the GTUPSR register.
2	Set cycle	Set the cycle in the GTPR register.
3	Set initial value for counter	Set the initial value in the GTCNT counter.
4	Start count operation	Set the GTCR.CST bit to 1 to start count operation.

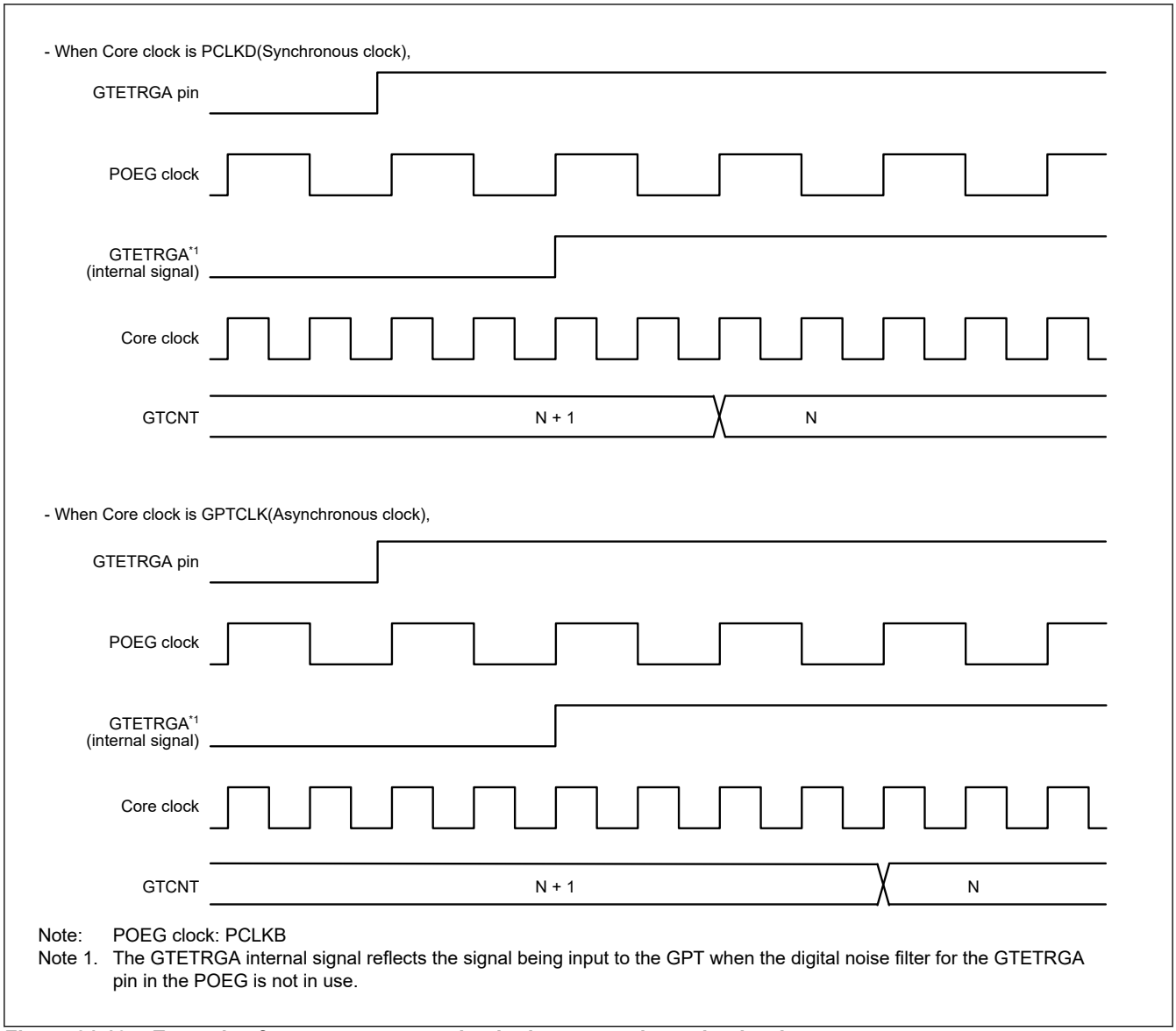
(5) Event count operation in down-counting using hardware sources

The GTCNT counter in each channel can perform down-counting using hardware sources set in the GTDNSR.

When GTDNSR is set to enable, the count clock selected in GTCR.TPCS[3:0] and the count direction selected in GTUDDTYC.UD are ignored. If up-counting and down-counting using hardware sources occur at the same time, the GTCNT counter value does not change. The underflow behavior when down-counting using hardware sources is the same as when down-counting by the count clock.

When GTCR.CST bit is set to 1 to count down using hardware sources, the count operation is enabled. After GTCR.CST is set to 1, the counter cannot count down for 1 clock cycle as specified in GTCR.TPCS[3:0] because the count operation is synchronized with the count clock selected in GTCR.TPCS[3:0]. Set GTCR.TPCS[3:0] to 000b to count down with a 1 GTCLK delay after GTCR.CST is set to 1.

Figure 21.10 shows an example of a event count operation in down-counting by a hardware resource (rising edge of GTETRGA pin).



Note: POEG clock: PCLKB
Note 1. The GTETRGA internal signal reflects the signal being input to the GPT when the digital noise filter for the GTETRGA pin in the POEG is not in use.

Figure 21.10 Example of event count operation in down-counting using hardware sources

Table 21.14 shows an example for setting a periodic count operation in down-counting using a hardware resource.

Table 21.14 Example for setting an event count operation in down-counting using hardware sources

No.	Step Name	Description
1	Set count source	Select the counting-down hardware source with the GTDNSR register.
2	Set cycle	Set the cycle in the GTPR register.
3	Set initial value for counter	Set the initial value in the GTCNT counter.
4	Start count operation	Set the GTCR.CST bit to 1 to start count operation.

(6) Counter clear operation

The counter of each channel is cleared by following sources:

- Writing 0 to GTCNT register
- Writing 1 to the bit in GTCLR associated with the GPT channel number when the GTCSR.CCLR bit set to 1
- The hardware source selected in GTCSR register.

Writing to the GTCNT register is prohibited during count operation. Write access during counting (when CST = 1) is disabled. The GTCNT counter can be cleared both by writing 1 to the GTCLR and by the clear request of hardware sources, whether GTCNT is counting (GTCR.CST is 1) or not (GTCR.CST is 0).

When the count direction flag is set as decrement (GTST.TCUF flag = 0) in saw-wave mode (Except saw-wave PWM mode 2) selected with GTCR.MD[2:0] bits or GTCR.MD[3:0] bits, the GTCNT register is set to the value of the GTPR register when writing 1 to the GTCLR register and when clearing by hardware sources are performed.

When not in saw-waves mode (except saw-wave PWM mode 2) and down-counting, the GTCNT register is set to 0 when writing 1 to the GTCLR register and when clearing by hardware sources are performed.

In event count operation when at least 1 bit in the GTUPSR or GTDNSR is set to 1, after clear sources occur, both writing to GTCLR register and clearing by hardware sources are performed immediately to synchronize with GTCLK. If other settings are used, clear is synchronized with the counter clock selected in GTCR.TPCS[3:0].

21.3.1.2 Waveform output by compare match

Compare match means that the GTCNT counter value matches the value of GTCCRA or GTCCRB. When a compare match occurs, the compare match flag is generated synchronously with the count clock, including the event count. At the same time, the GPT can output low, high, or toggled output from the associated GTIOCnA or GTIOCnB output pin (n = 0 to 9). In addition, the GTIOCnA or GTIOCnB pin output can be low, high, or toggled at the cycle end which is determined by GTPR or GTCCRm selected as a counter clear source by the GTCSR.CSCMSC[2:0] bits in saw-wave PWM mode 2 (n = 0 to 9, m = A to F).

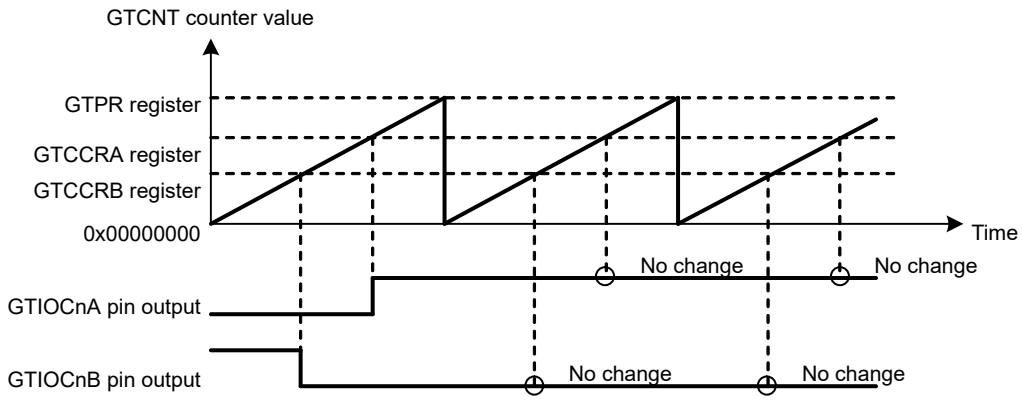
The cycle end is:

- For saw waves (except saw-wave PWM mode 2) in up-counting – when GTCNT changes from the GTPR value to 0 (overflow)
- For saw waves (except saw-wave PWM mode 2) in down-counting – when GTCNT changes from 0 to GTPR value (underflow)
- For saw-wave PWM mode 2 and GTCCRm register (m = A to F) selected as a counter clear source by the GTCSR.CSCMSC[2:0] bits – When the GTCNT counter value changes from the GTCCRm register value to 0x0000 0000
- For saw waves – when the GTCNT counter is cleared
- For triangle waves or complementary PWM mode – when the GTCNT changes from 0 to 1 (trough).

(1) Low output and high output

Figure 21.11 shows an example of low output and high output operation by a compare match of GTCCRA and GTCCRB.

In this example, the GTCNT counter performs up-counting, and settings are made so that high is output from the GTIOCnA pin by a GTCCRA compare match, and low is output from the GTIOCnB pin by a GTCCRB compare match. The pin level does not change when the specified level and pin level match.



[Setting examples]
GTIOR.GTIOA[4:0] bits: Initial output is low, high output at compare match, output retained at cycle end
GTIOR.GTIOB[4:0] bits: Initial output is high, low output at compare match, output retained at cycle end

Note: The base address of registers are the same.

Figure 21.11 Example of low output and high output operation

Table 21.15 shows an example for setting low output and high output operation.

Table 21.15 Example for setting low output and high output operation

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits or GTCR.MD[3:0] bits. In Figure 21.11, 000b or 0000b (saw-wave PWM mode 1) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 21.11, after 11b is set in the GTUDDTYC[1:0] bits, 01b is set in the GTUDDTYC[1:0] bits (up-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Other than the saw-wave PWM mode2, set the cycle in the GTPR register. In the saw-wave PWM mode2, select the counter clear source compare match register GTCCRx (x = A to F) by GTCSR.CSCMSC[2:0] bits and set the cycle in that register.
5	Set initial value for counter	Set the initial value in the GTCNT counter.
6	Set GTIOCnm pin function	Set the GTIOCnm pin function with the GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register. In Figure 21.11, GTIOA[4:0] = 00010b, GTIOB[4:0] = 10001b.
7	Enable GTIOCnm pin output*1	Set to enable the GTIOCnm pin output with the OAE and OBE bits in the GTIOR register.
8	Set compare match value*1	Set compare match values in the GTCCRA and GTCCRB registers.
9	Start count operation	Set the GTCR.CST bit to 1 to start count operation.

Note: n: 0 to 9
m: A, B

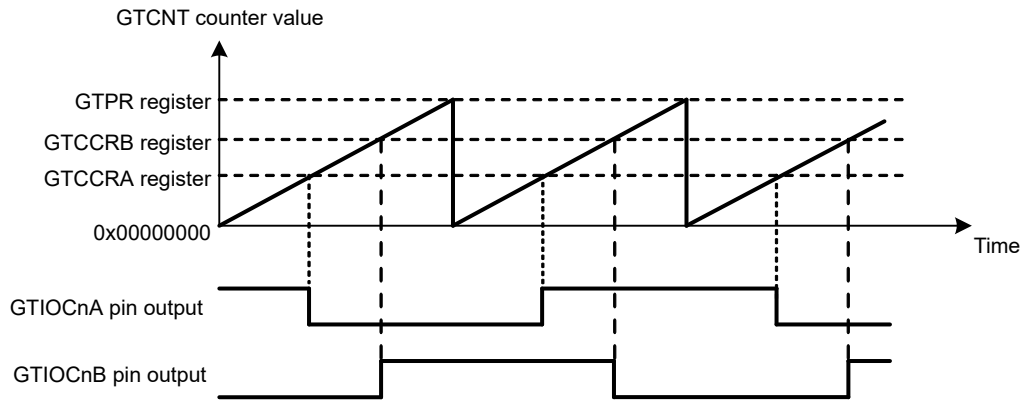
Note 1. When using PWM Delay Generation circuit, change the order of setting for the GTIOCnm pin output enable and setting for a compare match value.

(2) Toggled output

Figure 21.12 and Figure 21.13 show examples of toggled output operation by compare matches of GTCCRA and GTCCRB.

In Figure 21.12, the GTCNT counter performs up-counting, and settings are made so that the GTIOCnA pin output by a GTCCRA compare match and GTIOCnB pin output by a GTCCRB compare match are toggled.

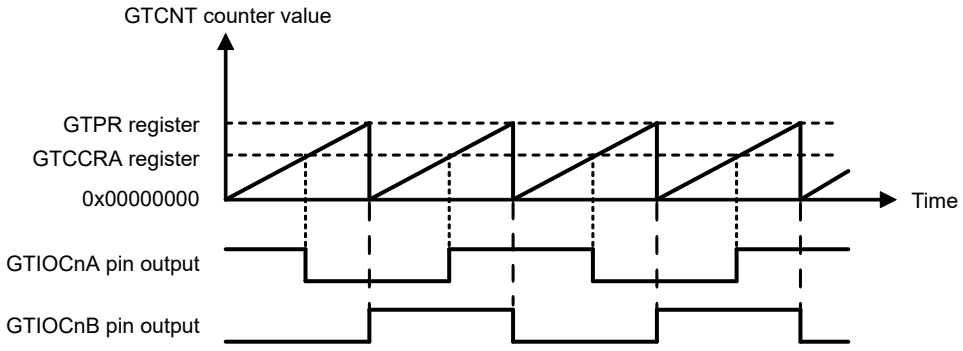
In Figure 21.13, the GTCNT counter performs up-counting, and settings are made so that a GTCCRA compare match toggles the GTIOCnA pin output level and a cycle end toggles the GTIOCnB pin output level.



[Setting examples]
GTIOR.GTIOA[4:0] bits: Initial output is high, output toggled at compare match, output retained at cycle end
GTIOR.GTIOB[4:0] bits: Initial output is low, output toggled at compare match, output retained at cycle end

Note: The base address of registers are the same.

Figure 21.12 Example of toggled output operation (1)



[Setting examples]
GTIOR.GTIOA[4:0] bits: Initial output is high, output toggled at compare match, output retained at cycle end
GTIOR.GTIOB[4:0] bits: Initial output is low, output retained at compare match, output toggled at cycle end

Note: The base address of registers are the same.

Figure 21.13 Example of toggled output operation (2)

Table 21.16 shows an example for setting toggled output operation.

Table 21.16 Example for setting toggled output operation (1 of 2)

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits or GTCR.MD[3:0] bits. In Figure 21.12 and Figure 21.13, 000b or 0000b (saw-wave PWM mode 1) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 21.12 and Figure 21.13, after 11b is set in the GTUDDTYC[1:0] bits, 01b is set in the GTUDDTYC[1:0] bits (up-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Other than the saw-wave PWM mode2, set the cycle in the GTPR register. In the saw-wave PWM mode2, select the counter clear source compare match register GTCCRx (x = A to F) by GTCSR.CSCMSC[2:0] bits and set the cycle in that register.
5	Set initial value for counter	Set the initial value in the GTCNT counter.

Table 21.16 Example for setting toggled output operation (2 of 2)

No.	Step Name	Description
6	Set GTIOCnm pin function	Set the GTIOCnm pin function with the GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register. In Figure 21.12, GTIOA[4:0] = 10011b, GTIOB[4:0] = 00011b, and in Figure 21.13, GTIOA[4:0] = 10011b, GTIOB[4:0] = 01100b.
7	Enable GTIOCnm pin output*1	Set to enable the GTIOCnm pin output with the OAE and OBE bits in the GTIOR register.
8	Set compare match value*1	Set compare match values in the GTCCRA and GTCCRB registers.
9	Start count operation	Set the GTCR.CST bit to 1 to start count operation.

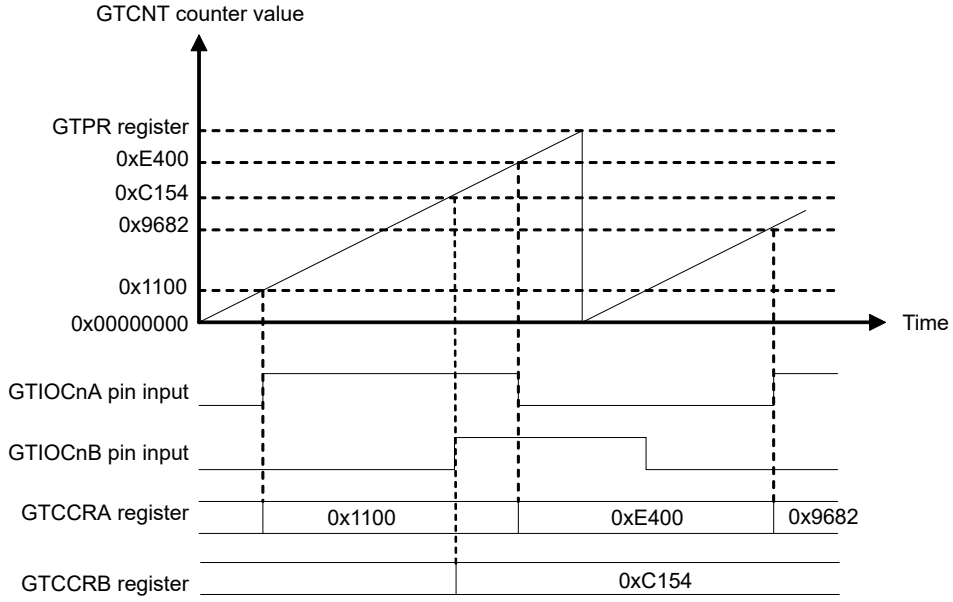
Note: n: 0 to 9
m: A, B
Note 1. When using PWM Delay Generation circuit, change the order of setting for the GTIOCnm pin output enable and setting for a compare match value.

21.3.1.3 Input Capture Function

The GTCNT counter value can be transferred to either GTCCRA or GTCCRB on detection of the hardware source that is set in GTICASR and GTICBSR. In complementary PWM mode, GTCCRA and GTCCRB registers do not function as input capture registers.

Figure 21.14 shows an example of the input capture function.

In this example, the GTCNT counter performs up-counting by the count clock, and settings are made so that an input capture is performed to GTCCRA at both edges of the GTIOCnA input pin and to GTCCRB on the rising edge of the GTIOCnB input pin.



Note: The base address of registers are the same.
Note: [Setting examples]
GTICASR setting: input capture at both edges
GTICBSR setting: input capture at the rising edge

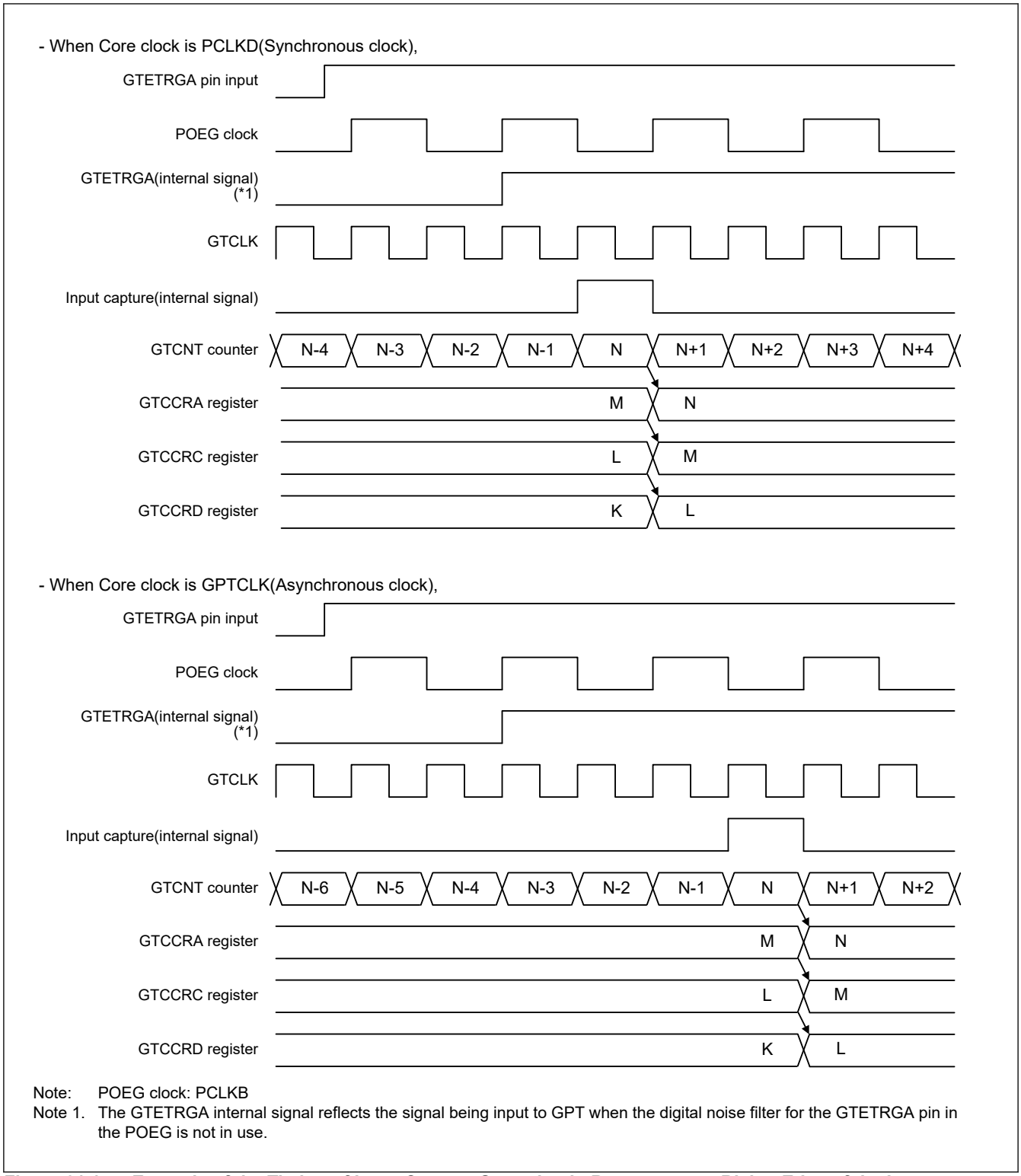
Figure 21.14 Example of input capture operation

Table 21.17 and Table 21.21 show the example for setting an input capture operation with count operation by the count clock.

Table 21.17 Example for setting input capture operation

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits or GTCR.MD[3:0] bits. In Figure 21.14, 000b or 0000b (saw-wave PWM mode 1) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 21.14, after 11b is set in the GTUDDTYC[1:0] bits, 01b is set in the GTUDDTYC[1:0] bits (up-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Other than the saw-wave PWM mode2, set the cycle in the GTPR register. In the saw-wave PWM mode2, select the counter clear source compare match register GTCCRx (x = A to F) by GTCSR.CSCMSC[2:0] bits and set the cycle in that register.
5	Set initial value for counter	Set the initial value in the GTCNT counter.
6	Select input capture source	Select the input capture source in the GTICASR and GTICBSR registers. In Figure 21.14, GTICASR = 0x00000F00, GTICBSR = 0x00003000. When input capture by other channel sources is used, set the GTICmSR.mSOC bit (m = A or B) to 1 and allow input capture by other channel sources, and select the group that cooperates with other channel sources by the GTICCR register.
7	Start count operation	Set the GTCR.CST bit to 1 to start count operation.

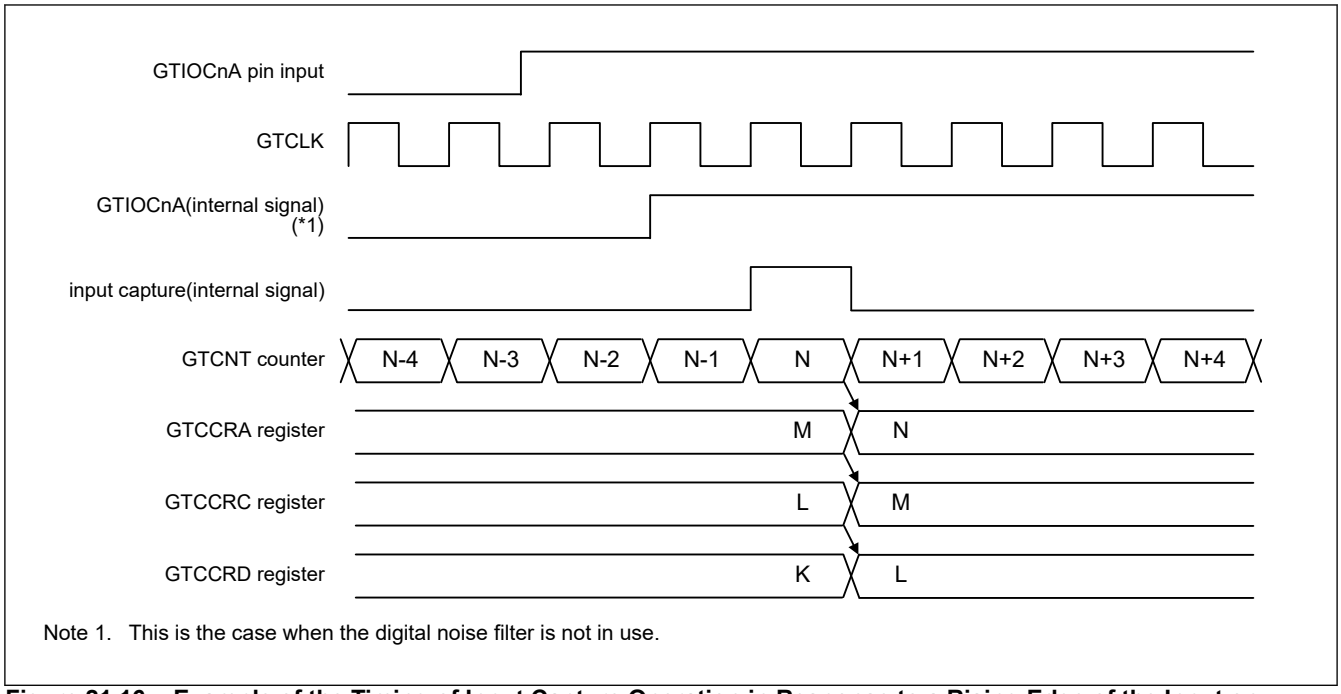
Figure 21.15 shows an example of the timing of input capture operation in response to a rising edge of the input on the GTETRGA pin.



Note: POEG clock: PCLKB
Note 1. The GTETRGA internal signal reflects the signal being input to GPT when the digital noise filter for the GTETRGA pin in the POEG is not in use.

Figure 21.15 Example of the Timing of Input Capture Operation in Response to a Rising Edge of the Input on the GTETRGA Pin

Figure 21.16 shows an example of the timing of input capture operation in response to a rising edge of the input on the GTIOCnA pin.



Note 1. This is the case when the digital noise filter is not in use.

Figure 21.16 Example of the Timing of Input Capture Operation in Response to a Rising Edge of the Input on the GTIOCnA Pin

Figure 21.17 shows an example of the timing of input capture operation in response to ELC_GPTA event input.

This is an example of capture of the counter value by the GPT321.GTCCRA register in response to an input signal. An event signal is output to the ELC after matches in comparison with the GPT320.GTCCRA register. This is selected as a trigger for output to the GPT321 by the ELC as ELC_GPTA.

ELC passes the event signal output from GPT320 without delay to GPT321.

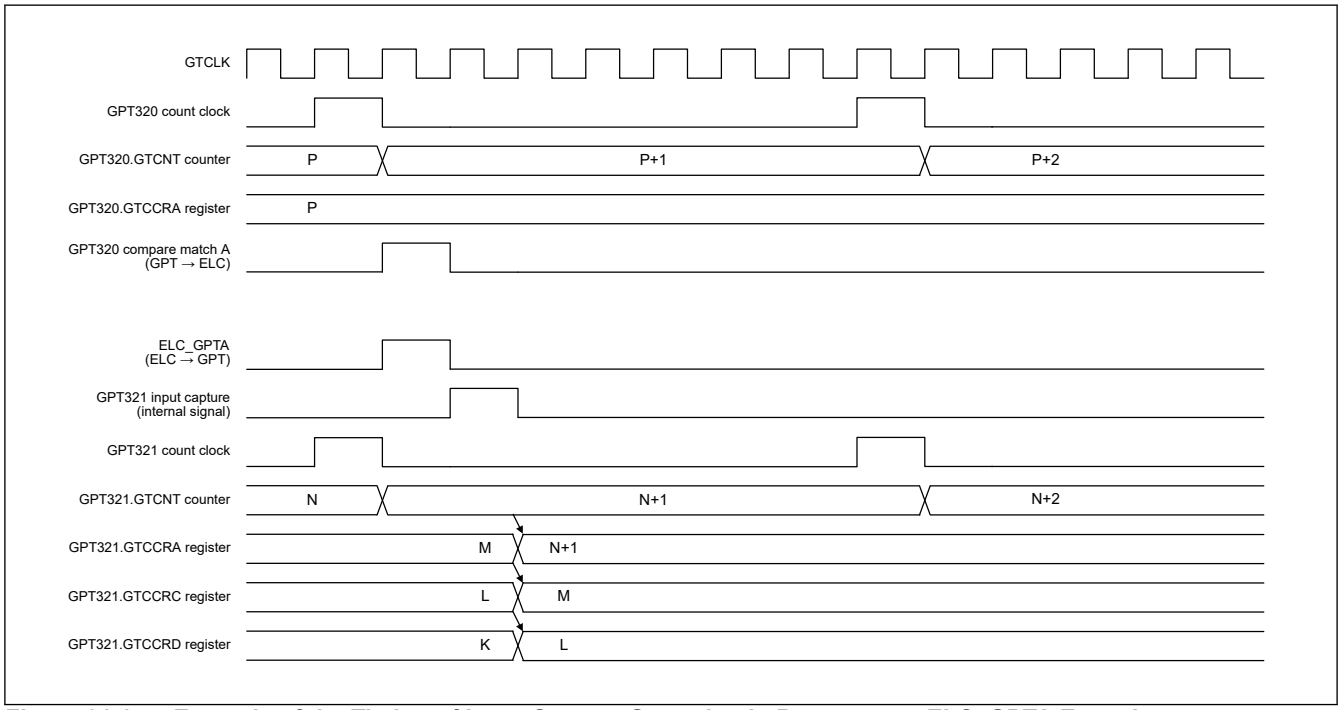


Figure 21.17 Example of the Timing of Input Capture Operation in Response to ELC_GPTA Event Input

Figure 21.18 shows an example of the timing of input capture operation in response to count clock from other channel.

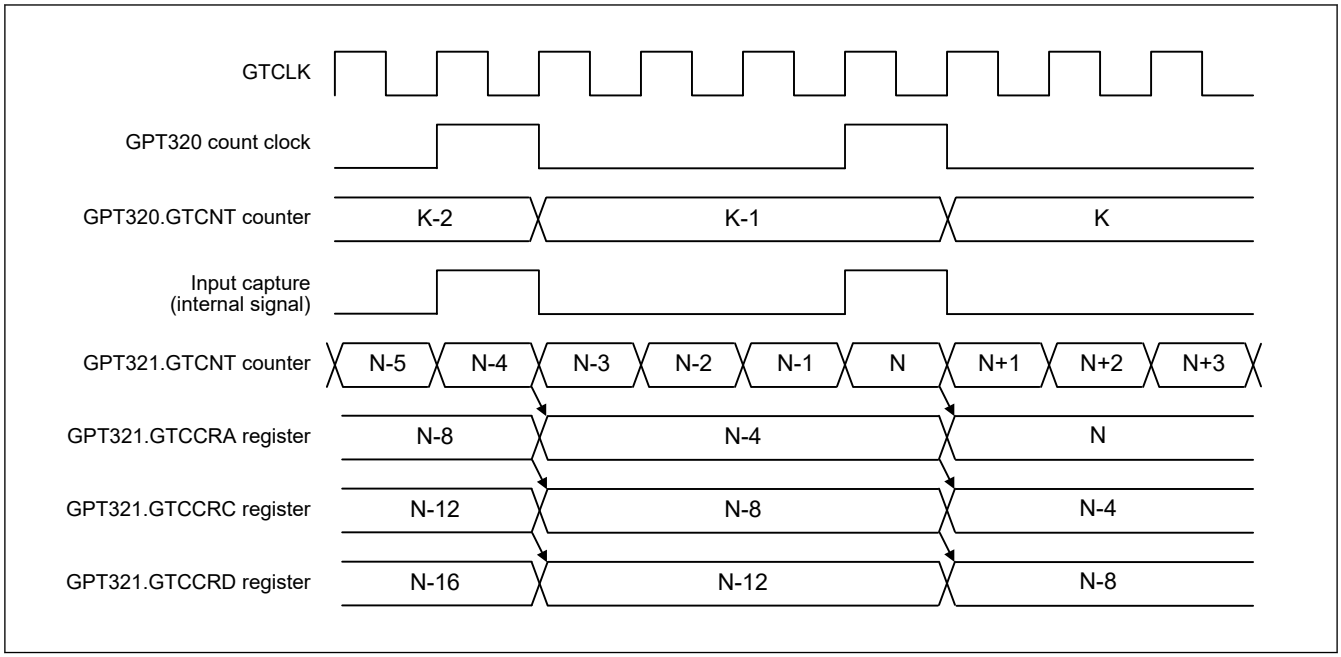


Figure 21.18 Example of the Timing of Input Capture Operation in Response to Count Clock from Other Channel

21.3.2 Buffer Operation

The following buffer operations can be set with GTBER:

- GTPR, GTPBR, and GTPDBR
- GTCCRA, GTCCRC, and GTCCRD
- GTCCRB, GTCCRE, and GTCCRF
- GTADTRA, GTADTBRA, and GTADTDBRA
- GTADTRB, GTADTBRB, and GTADTDBRB

The following buffer operation is enabled by setting GTDTCR:

- GTDVU and GTDBU
- GTDVD and GTDBD

The following buffer operation is enabled by setting GTBER2:

- GTCCRA, GTCCRE, and GTCCRF (in complementary PWM mode 3, 4)
- GTOLBR.GTIOAB[4:0] bits and GTIOR.GTIOA[4:0] bits
- GTOLBR.GTIOBB[4:0] bits and GTIOR.GTIOB[4:0] bits

21.3.2.1 GTPR Register Buffer Operation

GTPBR can function as a buffer register for GTPR.

GTPDBR register can function as a buffer register for the GTPBR register (double-buffer register for the GTPR register).

In complementary PWM mode, the buffer transfer from the GTPDBR register to temporary register P is performed only in the master channel (GPT32n). The temporary register P is transferred to each GTPBR register of master channel, slave channel 1 (GPT32n+1), and slave channel 2 (GPT32n+2). Transfer from the GTPBR register to the GTPR register is made concurrently in three channels. Therefore, the same value is stored in the same register of the three channels. The GTPR register of the master channel represents the GTCNT counter's (GTCNTn) period of the master channel. In the slave channels, periods are controlled using the GTPR register value and the GTDVU register value.

The setting is invalid in the Saw-wave PWM mode 2.

The buffer transfer is performed at an overflow (during up-counting) or an underflow (during down-counting) in saw-wave mode or in event count, and at a trough in triangle-wave mode.

In saw-wave mode or in event count, the buffer transfer is performed when the following counter clear operations occur during counting:

- Clear by hardware sources (the clear source is selected in GTCSR register)
- Clear by software (when GTCSR.CCLR bit is 1 and GTCLR.CCLRn bit is set to 1, n = 0 to 9).

In the case of saw waves, the buffer transfer by clearing counter can be prohibited by GTBER2.CCTPR bit.

Table 21.18 shows the buffer transfer timing in the complementary PWM mode.

Table 21.18 GTPR Buffer Transfer Timing in Complementary PWM Mode

	Complementary PWM mode 1	Complementary PWM mode 2	Complementary PWM mode 3, 4
GTPDBR ↓ Temporary register P	After one GTCLK cycle from GTCCRD register write of slave channel 2 (GPT32n+2)	After one GTCLK cycle from GTCCRD register write of slave channel 2 (GPT32n+2)	After one GTCLK cycle from GTCCRD register write of slave channel 2 (GPT32n+2)
Temporary register P ↓ GTPBR	(1) When data is transferred to temporary register P during up-counting middle section: After one GTCLK cycle from data transfer to temporary register P (2) When data is transferred to temporary register P during a section other than up-counting middle section: At the end of trough section	(1) When data is transferred to temporary register P during down-counting middle section: After one GTCLK cycle from data transfer to temporary register P (2) When data is transferred to temporary register P during a section other than down-counting middle section: At the end of crest section	(1) When data is transferred to temporary register P during middle section: After one GTCLK cycle from data transfer to temporary register P (2) When data is transferred to temporary register P during a section other than middle section: At the end of crest/ trough sections
GTPBR ↓ GTPR	At the end of crest section Synchronous clear	At the end of trough section Synchronous clear	At the end of crest section At the end of trough section Synchronous clear

To set GTPR to function as a buffer, set the GTBER.PR bit to 1. To set GTPR not to function as a buffer, set the GTBER.PR bit to 0.

To set the GTPR register to function as double buffer, set the GTBER.PR[1:0] bits to 10b or 11b. For single buffer operation, set 01b. Not to function as buffer, set 00b.

In complementary PWM mode, complementary PWM mode-specific buffer operation is performed regardless of the GTBER.PR[1:0] bit setting.

Figure 21.19 to Figure 21.24 show examples of GTPR buffer operation and Table 21.19 shows an example for setting GTPR buffer operation. For details of operation settings in complementary PWM mode, see section 21.3.3.7. Complementary PWM mode 1,2,3.

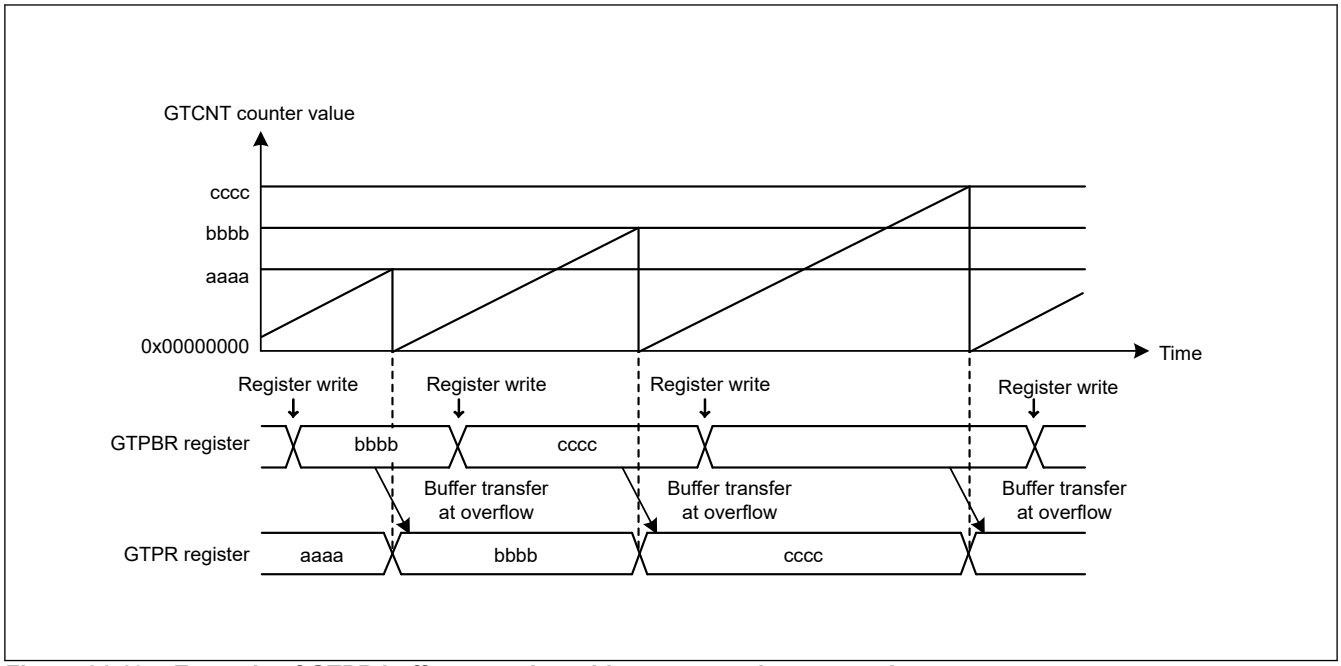


Figure 21.19 Example of GTPR buffer operation with saw waves in up-counting

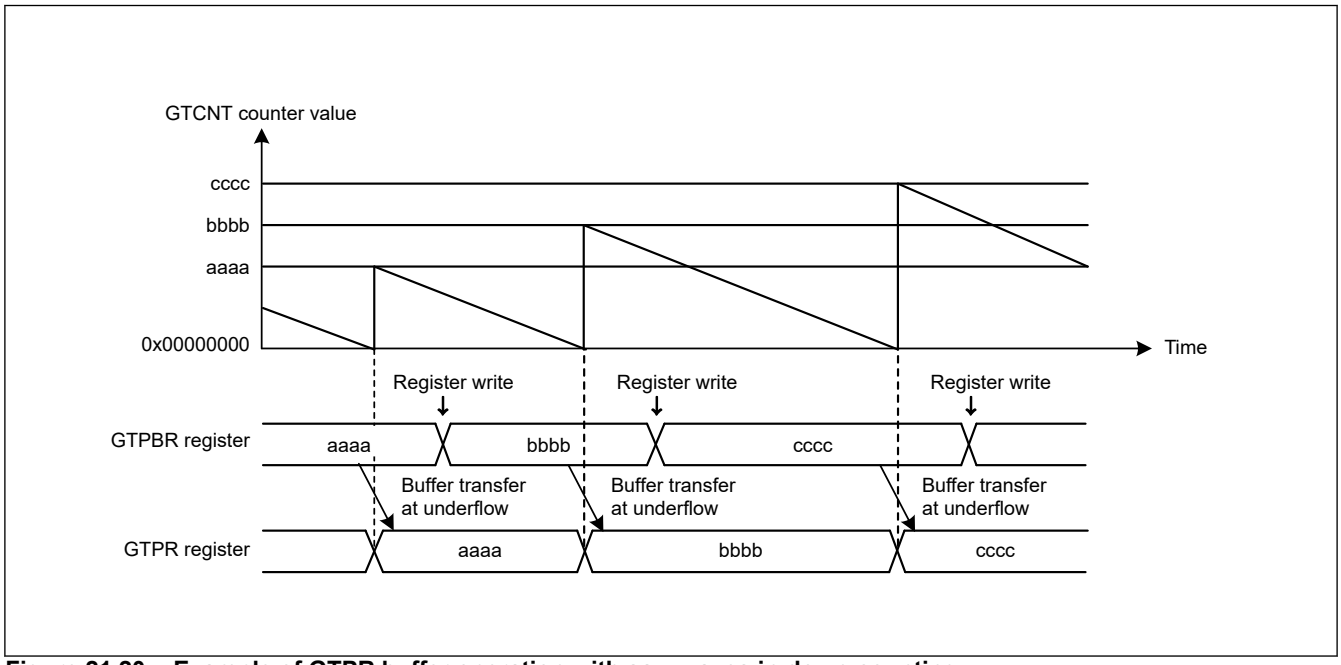


Figure 21.20 Example of GTPR buffer operation with saw waves in down-counting

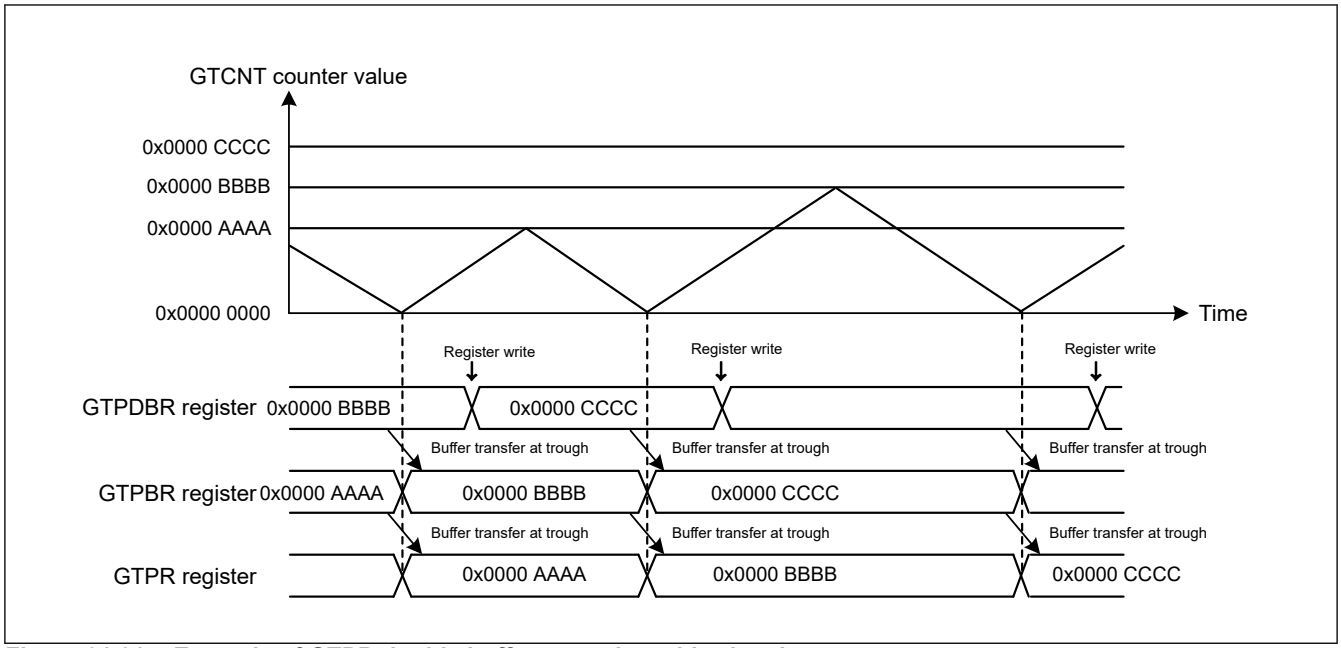


Figure 21.21 Example of GTPR double buffer operation with triangle waves

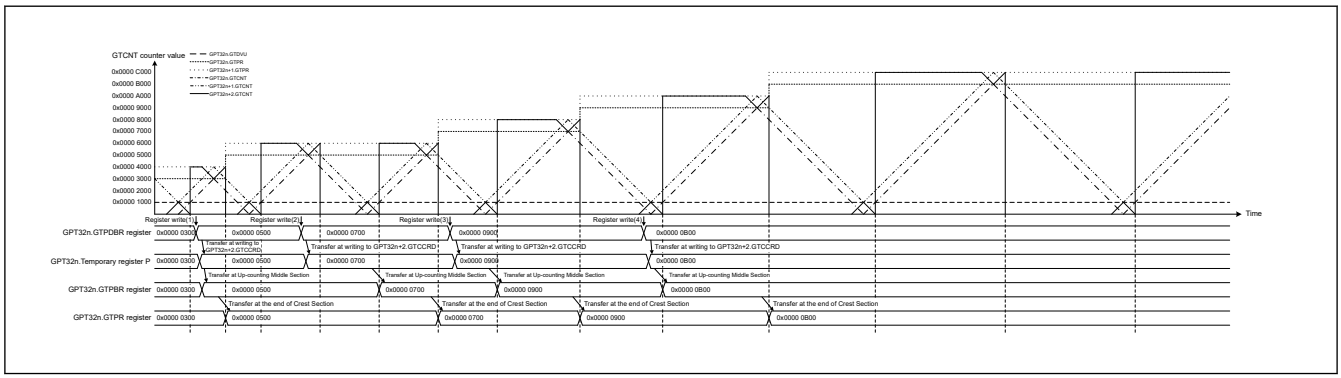


Figure 21.22 Example of GTPR double buffer operation with complementary PWM mode 1

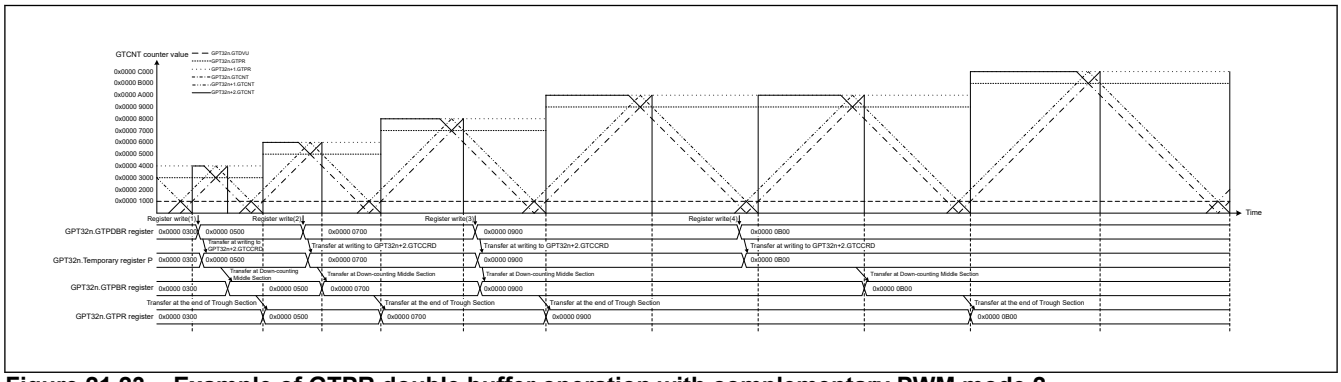


Figure 21.23 Example of GTPR double buffer operation with complementary PWM mode 2

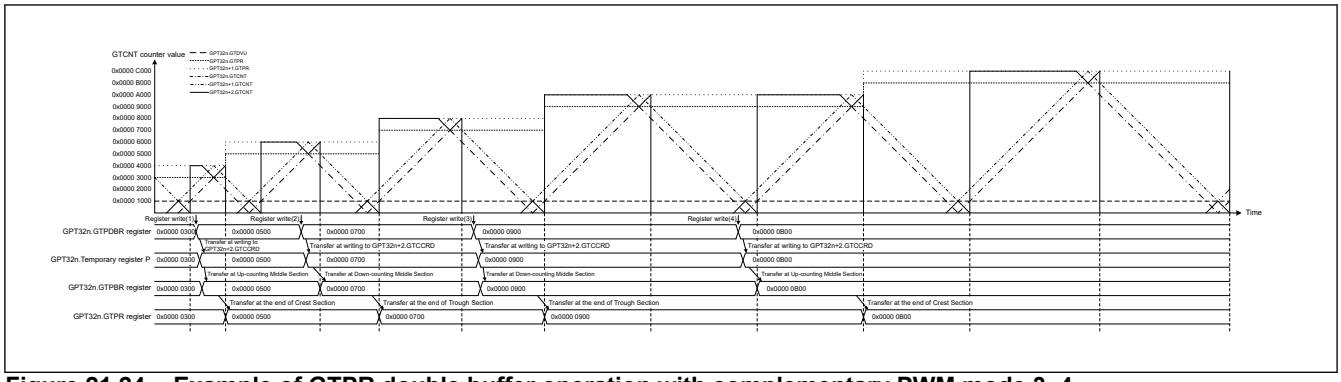


Figure 21.24 Example of GTPR double buffer operation with complementary PWM mode 3, 4

Table 21.19 Example for setting GTPR register buffer operation

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits or GTCR.MD[3:0] bits. In Figure 21.19 and Figure 21.20, 000b or 0000b (saw-wave PWM mode 1) is set, and in Figure 21.21, 100b or 0100b (triangle-wave PWM mode 1) is set. In Figure 21.22 to Figure 21.24, 11xxb (complementary PWM mode) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 21.19, after 11b is set in the GTUDDTYC[1:0] bits, 01b is set in the GTUDDTYC[1:0] bits (up-counting). In Figure 21.20, after 10b is set in the GTUDDTYC[1:0] bits, 00b is set in the GTUDDTYC[1:0] bits (down-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Set the cycle in the GTPR register.
5	Set initial value for counter	Set the initial value in the GTCNT counter.
6	Set buffer operation	Set buffer operation with the GTBER.PR[1:0] bits. In Figure 21.19 and Figure 21.20, PR[1:0] = 01b. In Figure 21.21, PR[1:0] = 1xb. In Figure 21.22 to Figure 21.24, no PR[1:0] bits setting required.
7	Set buffer value	For buffer operation, set a value in one cycle after the current cycle in the GTPBR register. For double buffer operation, also set a value of the period for the cycle after the next cycle in the GTPDBR register.
8	Start count operation	Set the GTCR.CST bit to 1 to start count operation.
9	Set buffer value for each cycle	For buffer operation, set a value in one cycle after the current cycle in the GTPBR register. For double buffer operation, also set a value of the period for the cycle after the next cycle in the GTPDBR register. In complementary PWM mode 1 (crest transfer), when the GTPDBR register is set in the up-counting trough or middle section, a value of the period for the next cycle is set. In other sections, a value of the period for the cycle after the next cycle is set. In complementary PWM mode 2 (trough transfer), when the GTPDBR register is set in the up-counting sect ion or the down-counting crest or middle sections, a value of the period for the next cycle is set. In other sections, a value of the period for the cycle after the next cycle is set. In complementary PWM mode 3 or 4 (crest/trough immediate transfer), when the GTPDBR register is set in a section other than the down-counting trough section, a value of the period for the next cycle is set. In the down-counting trough section, a value of the period for the cycle after the next cycle is set.

21.3.2.2 Buffer Operation for GTCCRA and GTCCRB Registers

GTCCRC can function as the GTCCRA buffer register and GTCCRD can function as the GTCCRC buffer register (double-buffer register for GTCCRA). Similarly, GTCCRE can function as the GTCCRB buffer register and GTCCRF can function as the GTCCRE buffer register (double-buffer register for GTCCRB).

To set GTCCRA or GTCCRB to function as a double buffer, set GTBER.CCRA[1:0] or GTBER.CCRB[1:0] to 10b or 11b. For single buffer operation, set 01b. To set GTCCRA or GTCCRB to not function as a buffer, set 00b.

In saw-wave one-shot pulse mode, triangle-wave PWM mode 3, complementary PWM mode, the buffer operations that specific each PWM output operation mode are performed regardless of the setting of GTBER.CCRA [1:0] bits and GTBER.CCRB [1:0] bits.

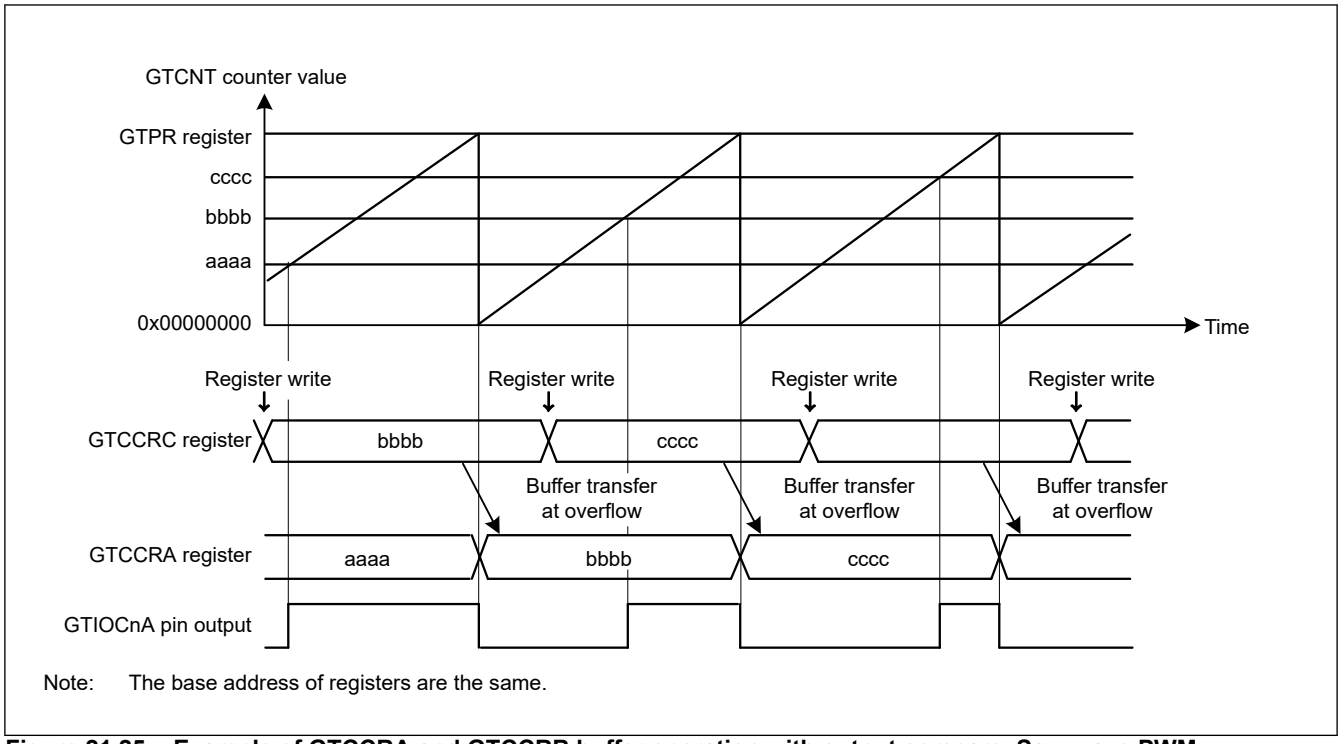
(1) When GTCCRA or GTCCRB Functions as Output Compare Register

In saw-wave one-shot pulse mode, triangle-wave PWM mode 3, complementary PWM mode, the buffer operations that specific each PWM output operation mode are performed regardless of the setting of GTBER.CCRA [1:0] bits and GTBER.CCRB [1:0] bits. For details, see section 21.3.3. PWM Output Operating Mode. Other than above PWM output operation mode, Buffer transfer occurs in the following situations:

- Buffer transfer by overflow or underflow
  Buffer transfer is performed at an overflow (during up-counting) or an underflow (during down-counting) in saw-wave mode or in event count operation. In triangle-wave mode, buffer transfer is performed at a trough (triangle-wave PWM mode 1) or a crest and trough (triangle-wave PWM mode 2).
- Buffer transfer by counter clear
  In saw-wave mode or in event count operation, during counting, buffer transfer (which is the same as an overflow during up-counting or an underflow during down-counting) is performed by the counter clear sources similar to the case shown in section 21.3.2.1. GTPR Register Buffer Operation.
  In saw-wave, the buffer transfer of GTCCRm register by counter clearing is possible to be disabled with GTBER2.CCTCm bit (m = A, B).
  In triangle-wave mode, buffer transfer is not performed by the counter clear.
- Buffer transfer by compare match
  In saw-wave, the buffer transfer by the compare match of GTCCRm register enabled by GTBER2.CMTCm (m = A, B) bit is performed.
- Forcible buffer transfer
  When GTBER.CCRSWT bit is set to 1 while the count operation is stopped, the GTCCRA and the GTCCRB register buffer transfer are performed forcibly in saw-wave mode, in event count operation and in triangle-wave mode. Additionally buffer transfer from the GTCCRD register to temporary register A and from the GTCCRF register to temporary register B are performed in saw-wave one-shot pulse mode or triangle-wave PWM mode 3.

When the setting of the GTBER.DBRTECm (m = A, B) bit is 1 in saw-wave one-shot pulse mode or triangle-wave PWM mode 3, transfer from the intermediate buffers to the GTCCRm (m = A, B) registers is repeated on a cyclic basis even while buffer transfer is disabled by the setting of the GTBER.BD[0] bit or buffer transfer extended skipping function (function for repeated double-buffer operation while buffer transfer is disabled). For details, see section 21.8.2.2. Repeated Double-Buffered Operation When Disabling GTCCR Buffer Transfer.

Figure 21.25 to Figure 21.28 show examples of GTCCRA and GTCCRB buffer operation and Table 21.20 shows an example for setting GTCCRA and GTCCRB buffer operation.



Note: The base address of registers are the same.

Figure 21.25 Example of GTCCRA and GTCCRB buffer operation with output compare, Saw-wave PWM mode 1 in up- counting, high output at GTCCRA compare match, and low output at cycle end

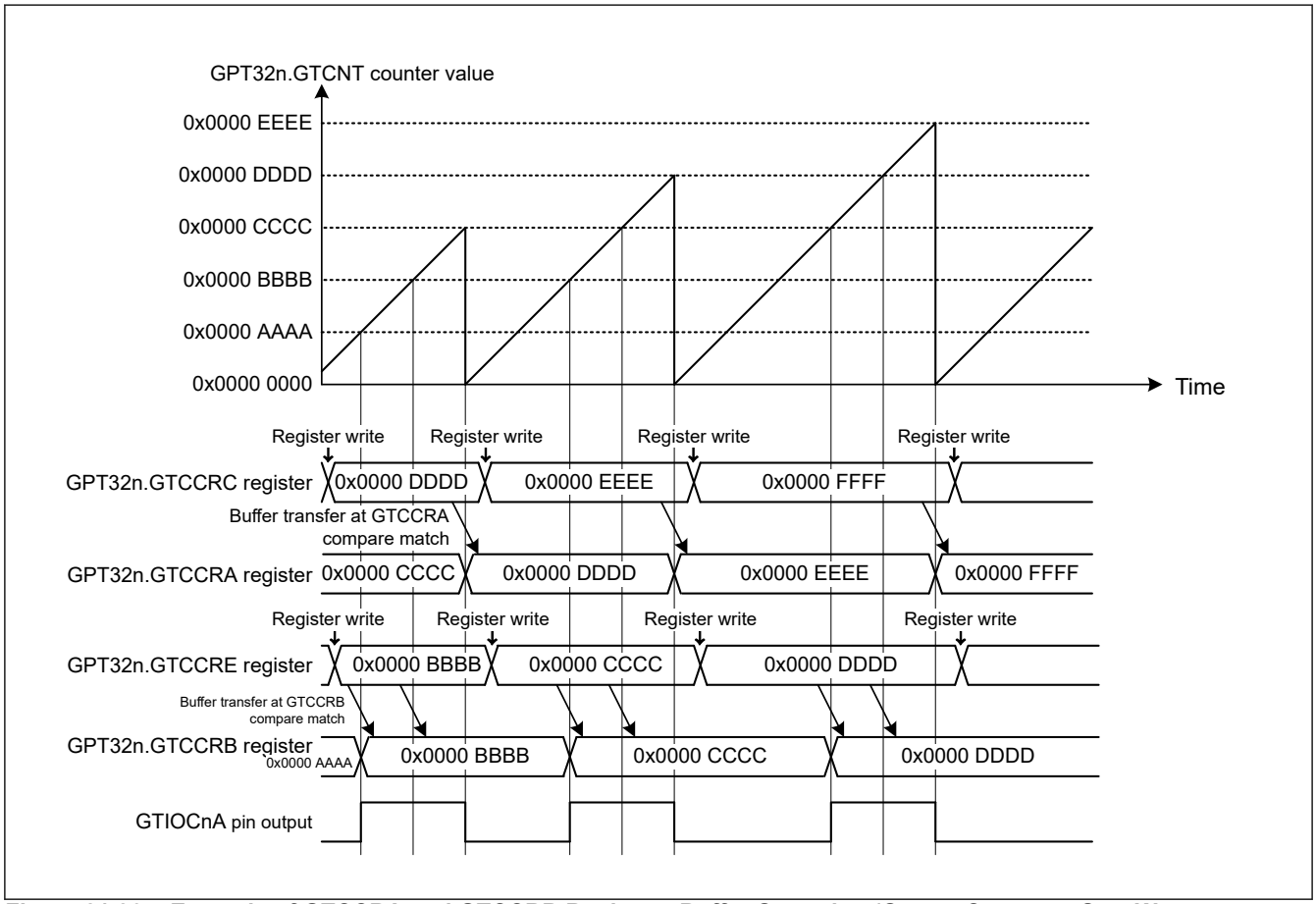
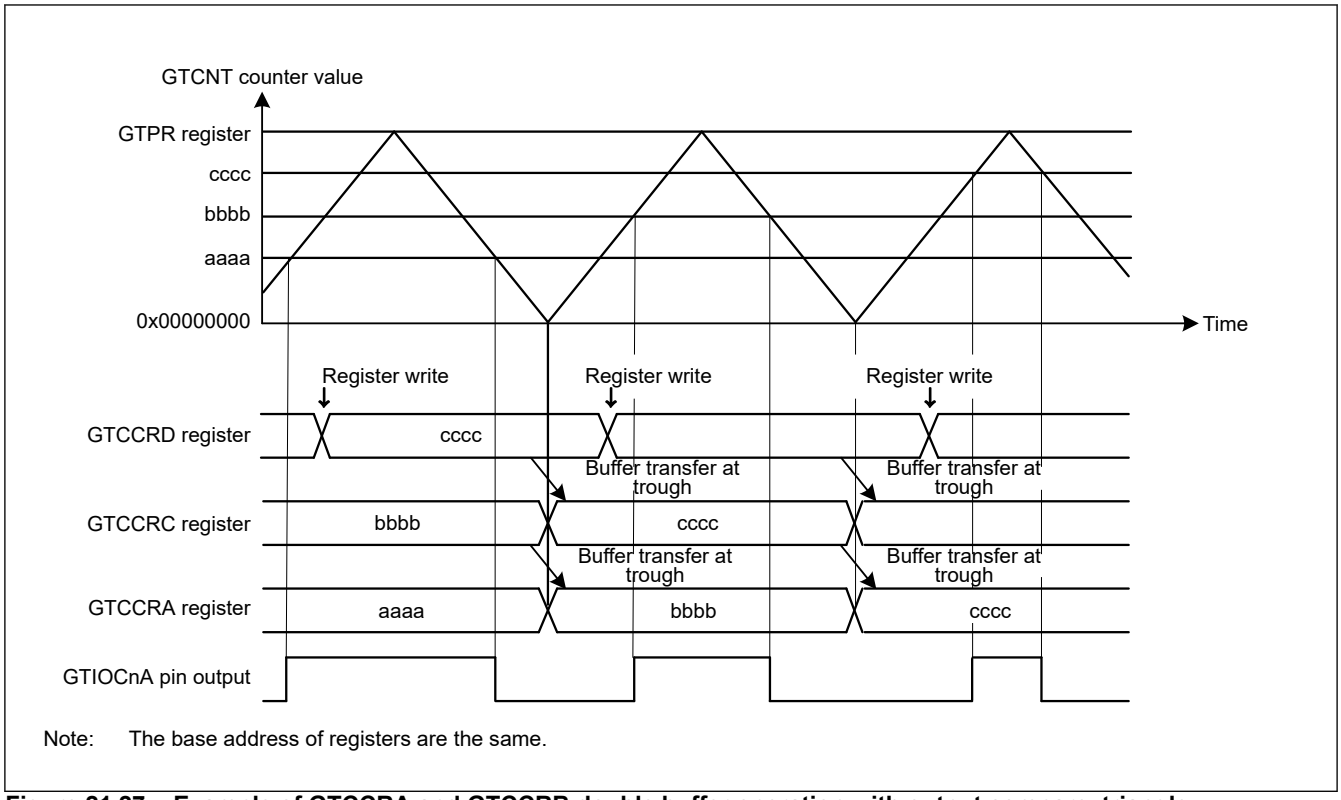
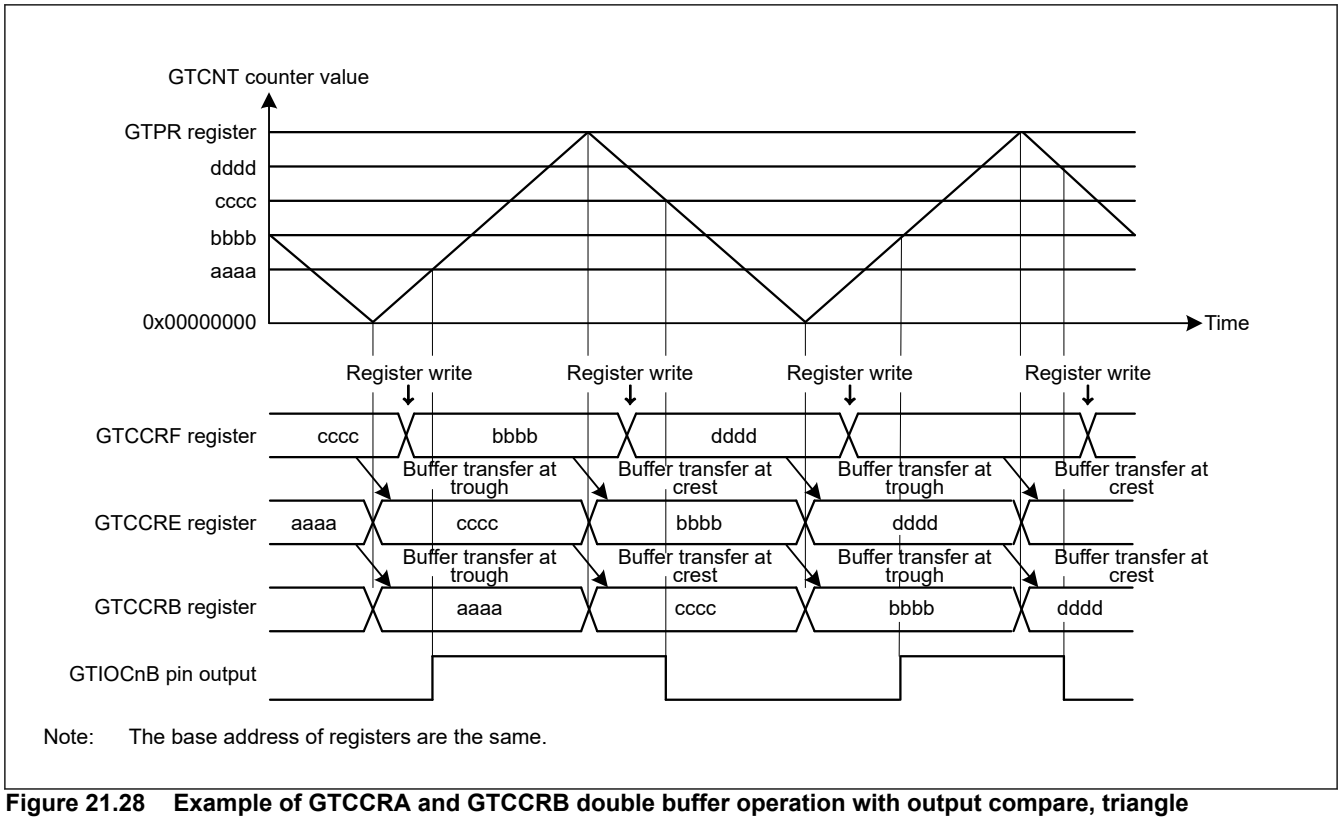


Figure 21.26 Example of GTCCRA and GTCCRB Registers Buffer Operation (Output Compare, Saw-Wave PWM Mode 2, Buffer Transfer at GTCCRA Register Compare Match, Counter Clearing, Low Output, Buffer Transfer at GTCCRB Register Compare Match, High Output) (n = 4 to 9)



Note: The base address of registers are the same.

Figure 21.27 Example of GTCCRA and GTCCRB double buffer operation with output compare, triangle waves, buffer operation at trough, output toggled at GTCCRA compare match, and output retained at cycle end



Note: The base address of registers are the same.

Figure 21.28 Example of GTCCRA and GTCCRB double buffer operation with output compare, triangle waves, buffer operation at both troughs and crests, output toggled at GTCCRB compare match, and output retained at cycle end

Table 21.20 Example for setting GTCCRA and GTCCRB buffer operation for output compare

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits or GTCR.MD[3:0] bits. In Figure 21.25, 000b or 0000b (saw-wave PWM mode 1) is set, in Figure 21.26, 0010b (saw-wave PWM mode 2) is set, in Figure 21.27, 100b or 0100b (triangle-wave PWM mode 1) is set, and in Figure 21.28, 101b or 0101b (triangle-wave PWM mode 2) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 21.25, after 11b is set in the GTUDDTYC[1:0] bits, 01b is set in the GTUDDTYC[1:0] bits (up-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Other than the saw-wave PWM mode2, set the cycle in the GTPR register. In the saw-wave PWM mode2, select the counter clear source compare match register GTCCRx (x = A to F) by GTCSR.CSCMSC[2:0] bits and set the cycle in that register.
5	Set initial value for counter	Set the initial value in the GTCNT counter.
6	Set GTIOCnm pin function	Set the GTIOCnm pin function with the GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register. In Figure 21.25, GTIOA[4:0] = 00110b, in Figure 21.26, GTIOA[4:0] = 00101b and GTIOB[1:0] = 10b, in Figure 21.27, GTIOA[4:0] = 00011b, and in Figure 21.28, GTIOB[4:0] = 00011b.
7	Enable GTIOCnm pin output*1	Set to enable the GTIOCnm pin output with the OAE and OBE bits in the GTIOR register.
8	Set buffer operation	Set buffer operation with the CCRA[1:0] and CCRB[1:0] bits in the GTBER register. In Figure 21.25, CCRA[1:0] = 01b, in Figure 21.26, CCRA[1:0] = 01b and CCRB[1:0] = 01b, in Figure 21.27, CCRA[1:0] = 1xb, and in Figure 21.28, CCRB[1:0] = 1xb.
9	Set compare match value*1	Set the GTIOCnA pin transition in the GTCCRA register and the GTIOCnB pin transition in the GTCCRB register.
10	Set buffer value	For buffer operation, set the GTIOCnA and GTIOCnB pins transitions in 1 cycle after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or half cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTCCRC and GTCCRE registers, respectively. For double buffer operation, also set the GTIOCnA and GTIOCnB pins transitions in 2 cycles after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or 1 cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTCCRD and GTCCRF registers, respectively.
11	Start count operation	Set the GTCR.CST bit to 1 to start count operation.
12	Set buffer value for each cycle	For buffer operation, set the GTIOCnA and GTIOCnB pins transitions in 1 cycle after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or half cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTCCRC and GTCCRE registers, respectively. For double buffer operation, also set the GTIOCnA and GTIOCnB pins transitions in 2 cycles after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or 1 cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTCCRD and GTCCRF registers, respectively.

Note: n: 0 to 9
m: A, B

Note 1. When using PWM Delay Generation circuit, change the order of setting for the GTIOCnm pin output enable and setting for a compare match value.

(2) When GTCCRA or GTCCRB Functions as Input Capture Register

When an input capture is generated, the GTCNT counter value is transferred to GTCCRA and GTCCRB and the stored GTCCRA and GTCCRB register values are transferred to the buffer registers. In input capture operation, the buffer transfer is not performed by the counter clear.

Figure 21.29 and Figure 21.30 show examples of GTCCRA and GTCCRB buffer operation and Table 21.21 shows an example for setting GTCCRA and GTCCRB buffer operation.

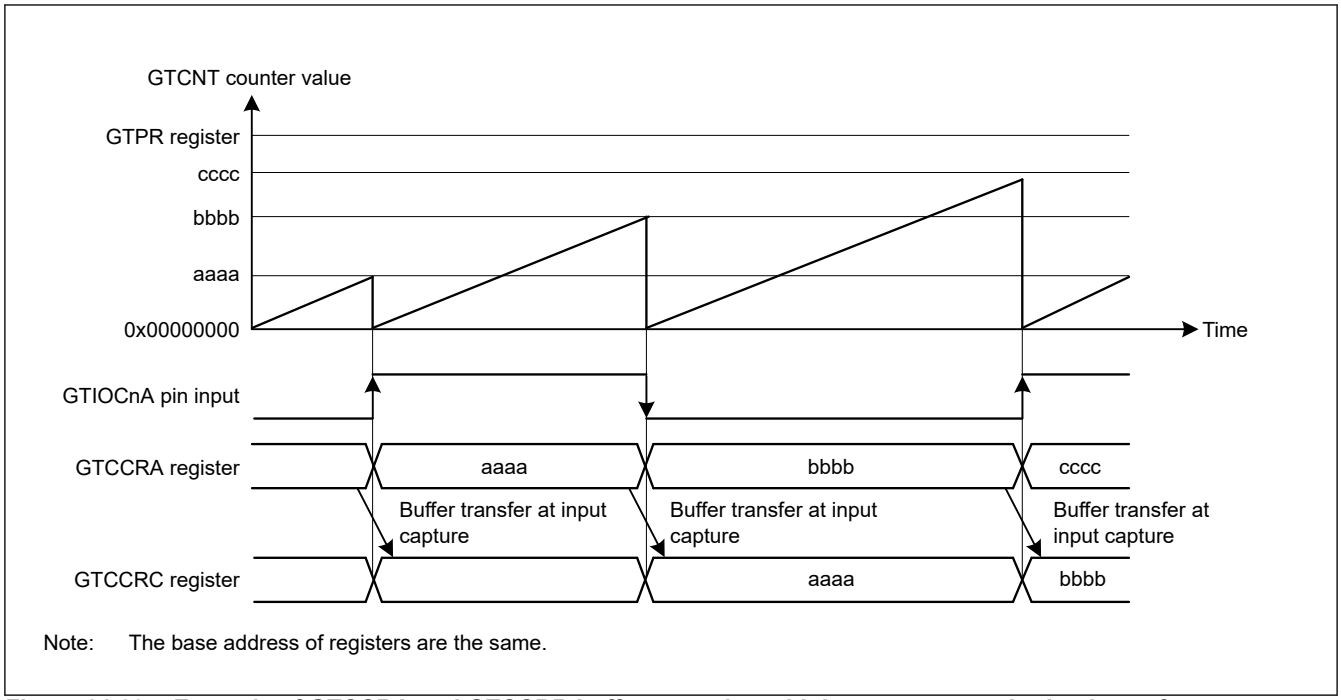


Figure 21.29 Example of GTCCRA and GTCCRB buffer operation with input capture at both edges of GTIOCnA input, saw waves in up-counting, and GTCNT counter cleared at both edges of GTIOCnA input

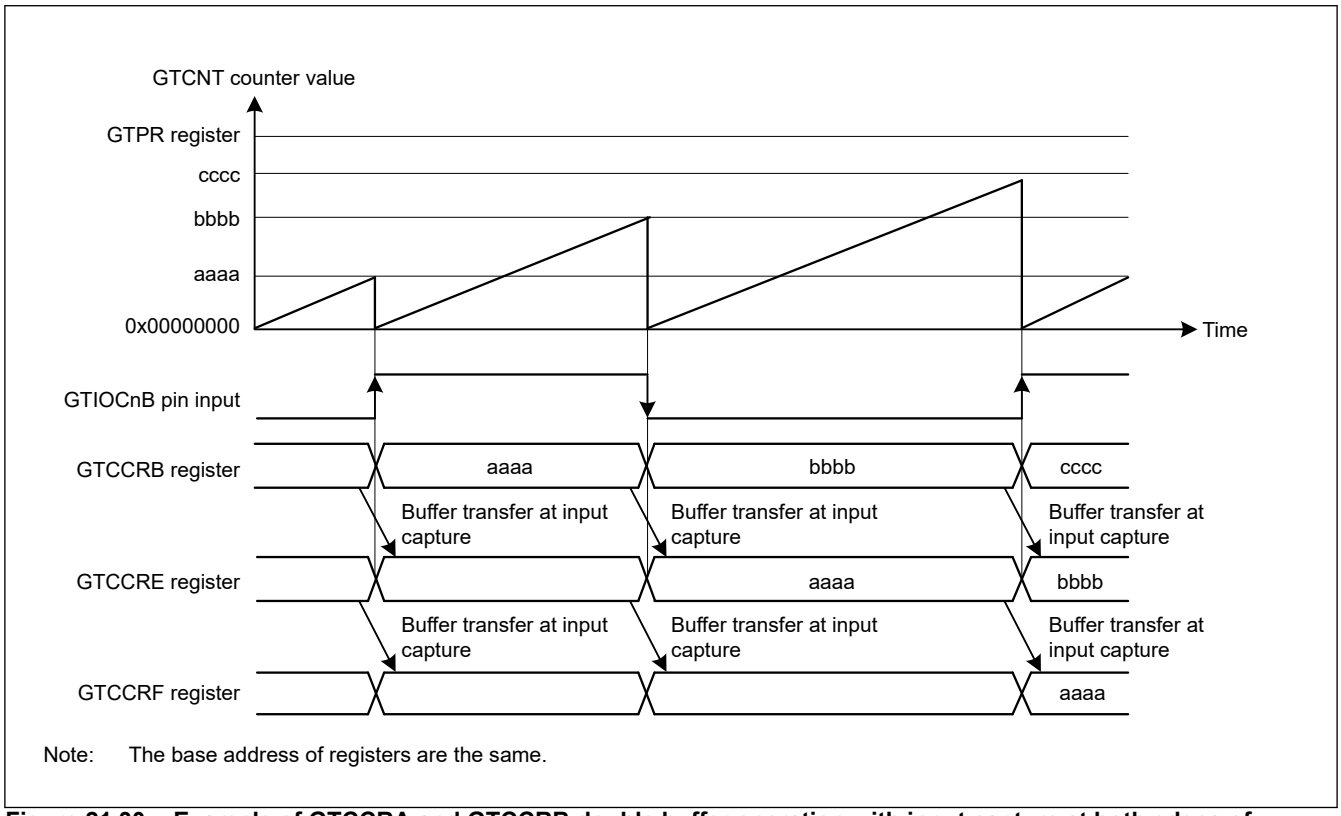


Figure 21.30 Example of GTCCRA and GTCCRB double buffer operation with input capture at both edges of GTIOCnB input, saw waves in up-counting, and GTCNT counter cleared at both edges of GTIOCnB input

Table 21.21 Example for setting GTCCRA and GTCCRB buffer operation for input capture

No.	Step Name	Description
1	Set operating mode and counter clear sources	Set the operating mode with the GTCR.MD[2:0] bits or GTCR.MD[3:0] bits and count clear source with the GTCSR register. In Figure 21.29, MD[2:0] = 000b or MD[3:0] = 0000b (saw-wave PWM mode 1) and GTCSR = 0x00000F00, and in Figure 21.30, MD[2:0] = 000b or MD[3:0] = 0000b (saw-wave PWM mode 1) and GTCSR = 0x0000F000.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 21.29, after 11b is set in the GTUDDTYC[1:0] bits, 01b is set in the GTUDDTYC[1:0] bits (up-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Other than the saw-wave PWM mode2, set the cycle in the GTPR register. In the saw-wave PWM mode2, select the counter clear source compare match register GTCCRx (x = A to F) by GTCSR.CSCMSC[2:0] bits and set the cycle in that register.
5	Set initial value for counter	Set the initial value in the GTCNT counter.
6	Select input capture source	Select input capture source in the GTICASR register and GTICBSR register. In Figure 21.29, GTICASR = 0x00000F00, and in Figure 21.30, GTICBSR = 0x0000F000. To perform input capture with sources of other channels, select a group that performs inter-channel cooperation by the GTICCR.ICmGRP bit (m = A or B). For the output-side channel of the input capture sources, set the GTICCR register to enables the input capture sources to output to other channels. For input-side channels, set the GTIOmSR.mSOC bit (m = A or B) to 1 to enable input capture with sources of other channels.
7	Set buffer operation	Set buffer operation with the CCRA and CCRB bits in the GTBER register. In Figure 21.29, CCRA[1:0] = 01b, and in Figure 21.30, CCRB[1:0] = 1xb.
8	Start count operation	Set the GTCR.CST bit to 1 to start count operation.

21.3.2.3 Buffer Operation for GTADTRA and GTADTRB Registers

The GTADTBRA register can function as the GTADTRA buffer register and the GTADTDBRA register can function as the GTADTBRA buffer register (double buffer register for the GTADTRA register). Similarly, the GTADTBRB register can function as the GTADTRB buffer register and the GTADTDBRB register can function as the GTADTBRB buffer register (double buffer register for the GTADTRB register).

To set the GTADTRA or GTADTRB register to function as a double buffer, set the GTBER.ADTDA or ADTDB bit to 1. For single buffer operation, set 0. Not to function as buffer, set the GTBER.ADTTA[1:0] or ADTTB[1:0] bits to 00b.

The buffer transfer timing can be set with the ADTTA[1:0] and ADTTB[1:0] bits to an overflow (in up-counting) or an underflow (in down-counting) in saw-wave mode, with ADTTA[1:0] and ADTTB[1:0] bits to 01b for a crest, to 10b for a trough, or to 11b for both crest and trough in triangle-wave mode or complementary PWM mode.

In saw-wave mode, when the ADTTA[1:0] and ADTTB[1:0] bits are set to value other than 00b and in count operation, the buffer transfer, by similar counter clearing sources in section 21.3.2.1. GTPR Register Buffer Operation, is performed in the same way at an overflow (in up-counting) or an underflow (in down-counting).

In complementary PWM mode, the buffer transfer is performed after one GTCLK cycle from GTCCRD register write of slave channel 2.

In saw-wave mode, the buffer transfer of GTADTRm register by counter clearing can be disabled by GTBER2.CCTADm (m = A, B) bit setting.

In saw-wave mode, the buffer transfer of GTADTRm register by its own compare match can be enabled by the GTBER2.CMTADm (m = A, B) bit setting.

Figure 21.31 to Figure 21.35 show examples of buffer operation of the GTADTRA and GTADTRB registers, and Table 21.22 shows an example for setting buffer operation of the GTADTRA and GTADTRB registers.

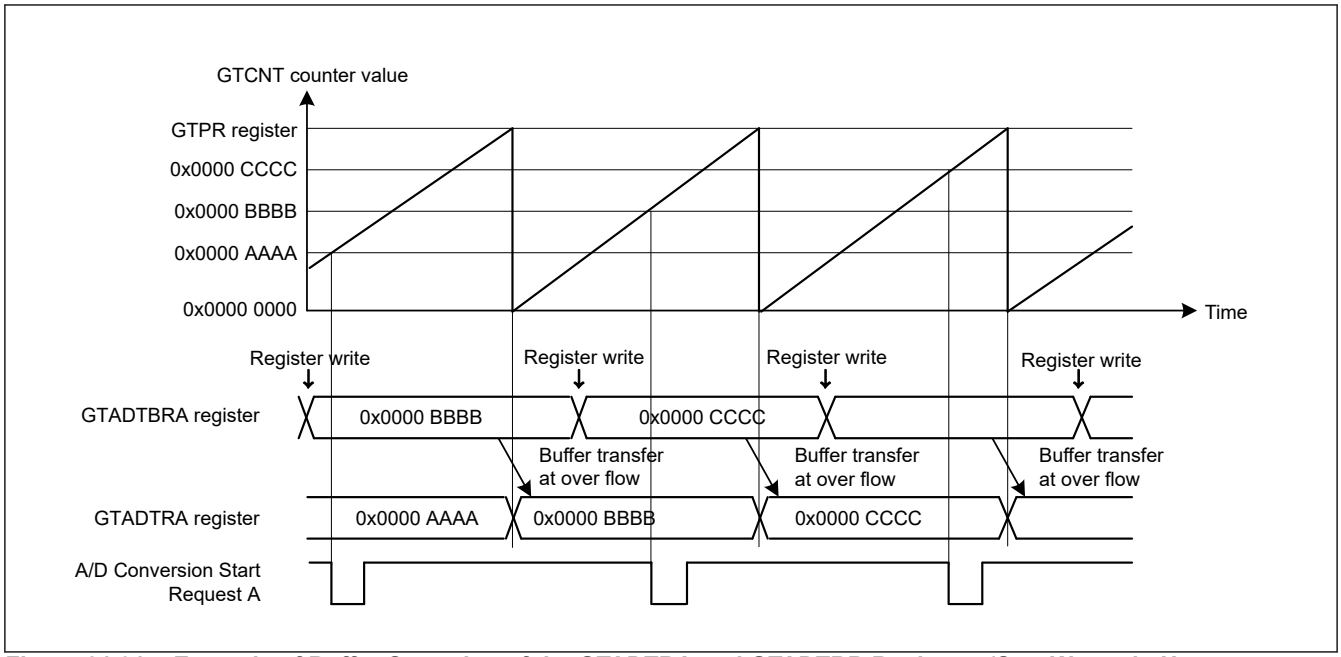


Figure 21.31 Example of Buffer Operation of the GTADTRA and GTADTRB Registers (Saw Waves in Up-Counting, A/D Conversion Start Request Generated by Up-Counting)

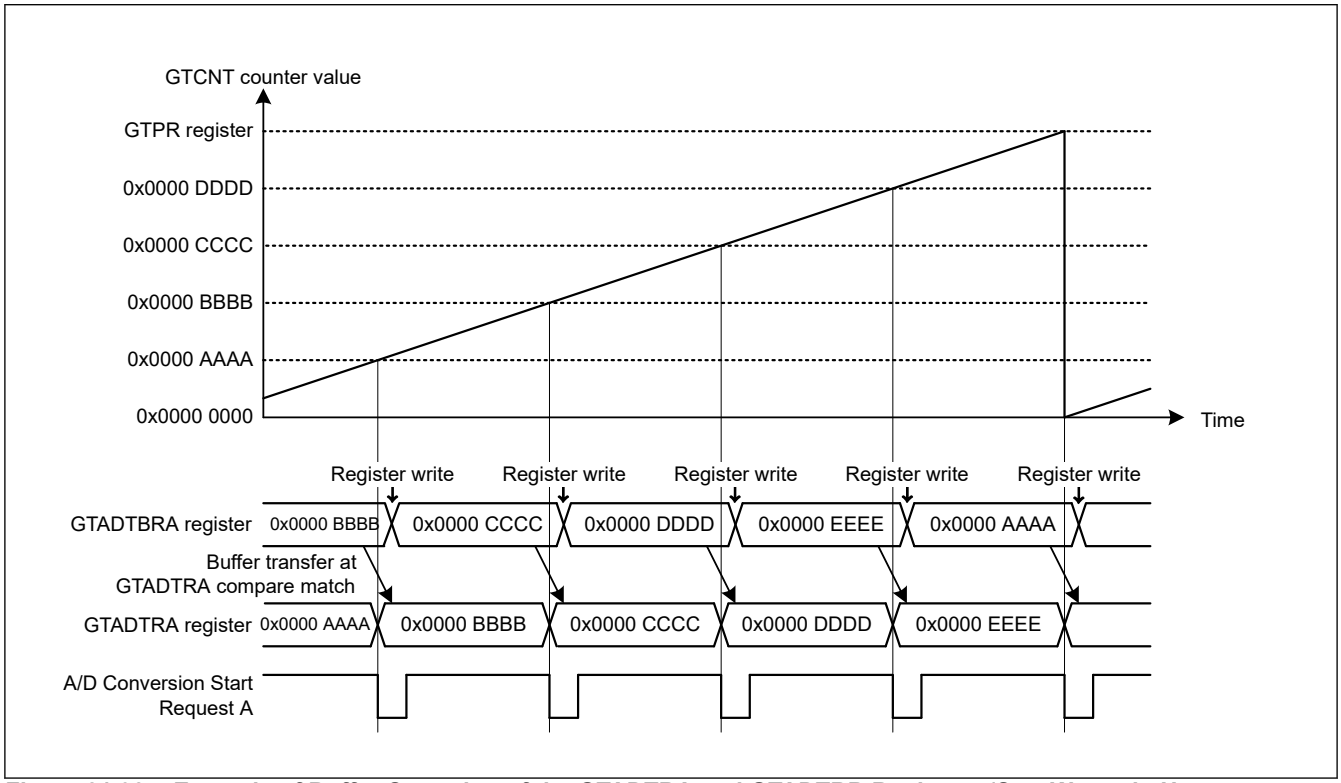


Figure 21.32 Example of Buffer Operation of the GTADTRA and GTADTRB Registers (Saw Waves in Up-Counting, Buffer Transfer at GTADTRA Compare Match, A/D Conversion Start Request Generated by Up-Counting)

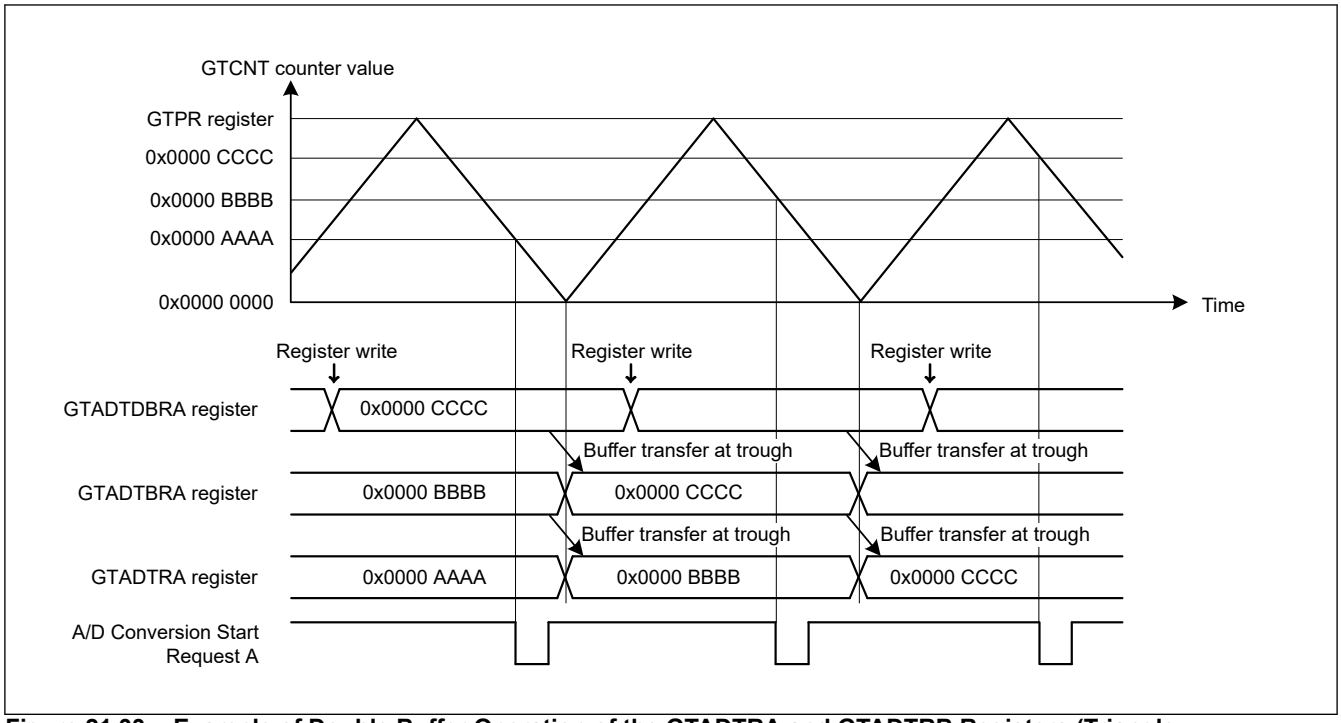


Figure 21.33 Example of Double Buffer Operation of the GTADTRA and GTADTRB Registers (Triangle Waves, Buffer Transfer at Troughs, A/D Conversion Start Request Generated by Down-Counting)

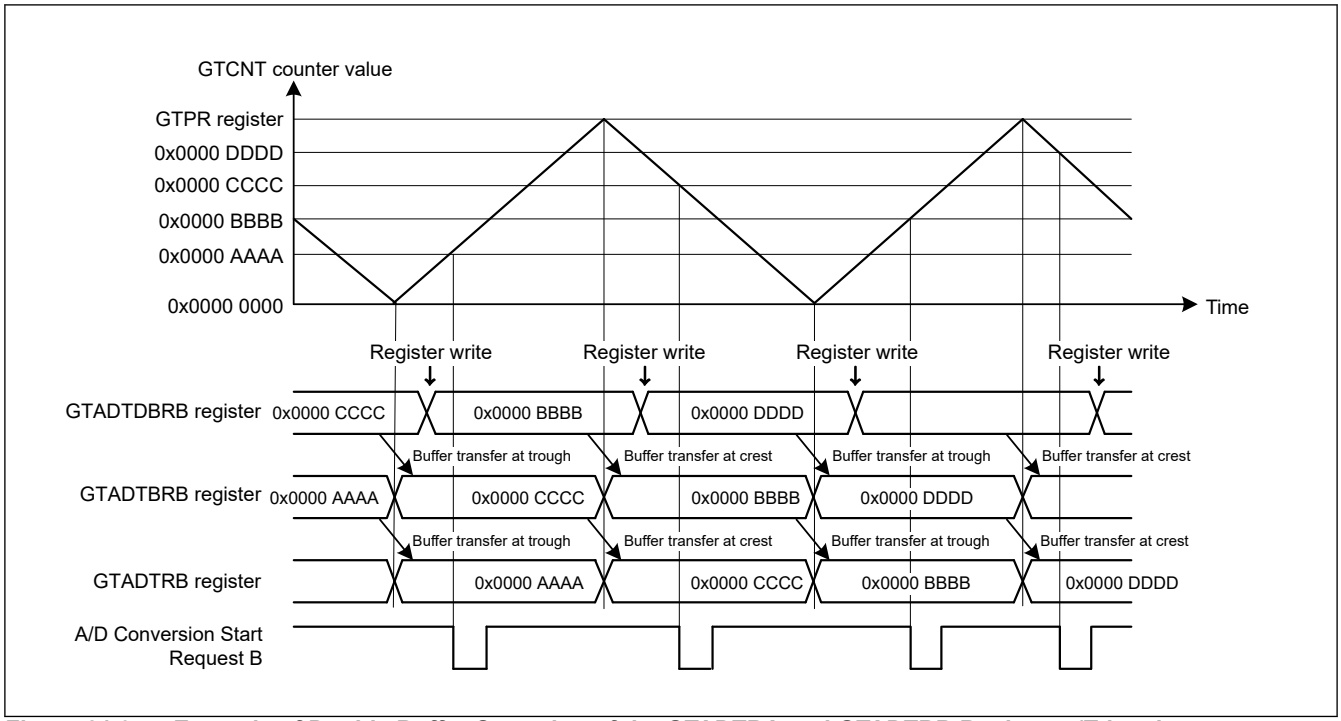


Figure 21.34 Example of Double Buffer Operation of the GTADTRA and GTADTRB Registers (Triangle Waves, Buffer Transfer at Both Troughs and Crests, A/D Conversion Start Request Generated by Both Up- and Down-Counting)

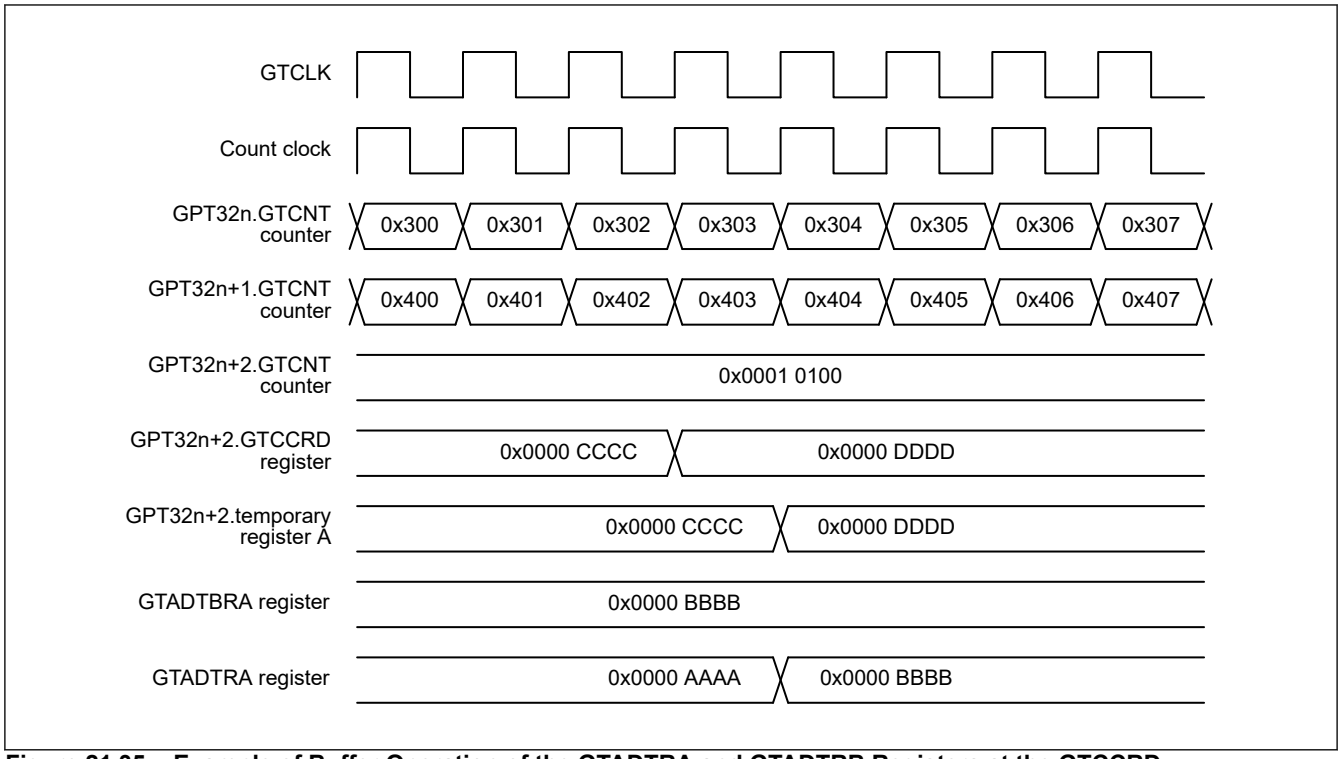


Figure 21.35 Example of Buffer Operation of the GTADTRA and GTADTRB Registers at the GTCCRD Register of Slave Channel 2 Updating in Complementary PWM Mode

Table 21.22 Example for Setting Buffer Operation of the GTADTRA and GTADTRB registers (1 of 2)

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits or GTCR.MD[3:0] bits. (In Figure 21.31 and Figure 21.32, 000b or 0000b (saw-wave PWM mode 1) is set, in Figure 21.33 and Figure 21.34, 100b, 101b, 110b, 0100b, 0101b, or 0110b (triangle-wave PWM mode) is set.)
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. (In Figure 21.31 and Figure 21.32, after 11b is set in the GTUDDTYC[1:0] bits, 01b is set in the GTUDDTYC[1:0] bits (up-counting).)
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Other than the saw-wave PWM mode 2, set the cycle in the GTPR register. In the saw-wave PWM mode 2, select the counter clear source compare match register GTCCRx (x = A to F) by GTCSR.CSCMSC[2:0] bits and set the cycle in that register.
5	Set initial value for counter	Set the initial value in the GTCNT counter.
6	Set buffer operation	Set buffer operation with the ADTTA[1:0], ADTTB[1:0], ADTDA, and ADTDB bits in GTBER register. To perform buffer transfer at a compare match of the ADTRAm (m = A, B) register, set the GTBER2.CMTADm bit to 1. (In Figure 21.31, ADTTA[1:0] bits = 01b, 10b, or 11b and ADTDA bit = 0, in Figure 21.32, CMTADA bit = 1, in Figure 21.33, ADTTA[1:0] bits = 10b and ADTDA bit = 1, and in Figure 21.34, ADTTB[1:0] bits = 11b and ADTDB bit = 1.)
7	Set compare match value	Set the A/D conversion start request point in the GTADTRA and GTADTRB registers.
8	Set buffer value	For buffer operation, set the A/D conversion start request point in one cycle after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or half cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTADTBRA and GTADTBRB registers. For double buffer operation, also set the A/D conversion start request point in two cycles after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or one cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTADTDBRA and GTADTDBRB registers.

Table 21.22 Example for Setting Buffer Operation of the GTADTRA and GTADTRB registers (2 of 2)

No.	Step Name	Description
9	Enable A/D conversion start request	Set to enable A/D conversion start request with the ADTRAUEN, ADTRADEN, ADTRBUEN, and ADTRBDEN bits in the GTINTAD register. (In Figure 21.31 and Figure 21.32, ADTRAUEN bit = 1, in Figure 21.33, ADTRADEN bit = 1, and in Figure 21.34, ADTRBUEN bit = 1 and ADTRBDEN bit = 1.)
10	Start count operation	Set the GTCR.CST bit to 1 to start count operation.
11	Set buffer value of each cycle	For buffer operation, set the A/D conversion start request point in one cycle after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or half cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTADTBRA and GTADTBRB registers. For double buffer operation, also set the A/D conversion start request point in two cycles after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or one cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTADTDBRA and GTADTDBRB registers.

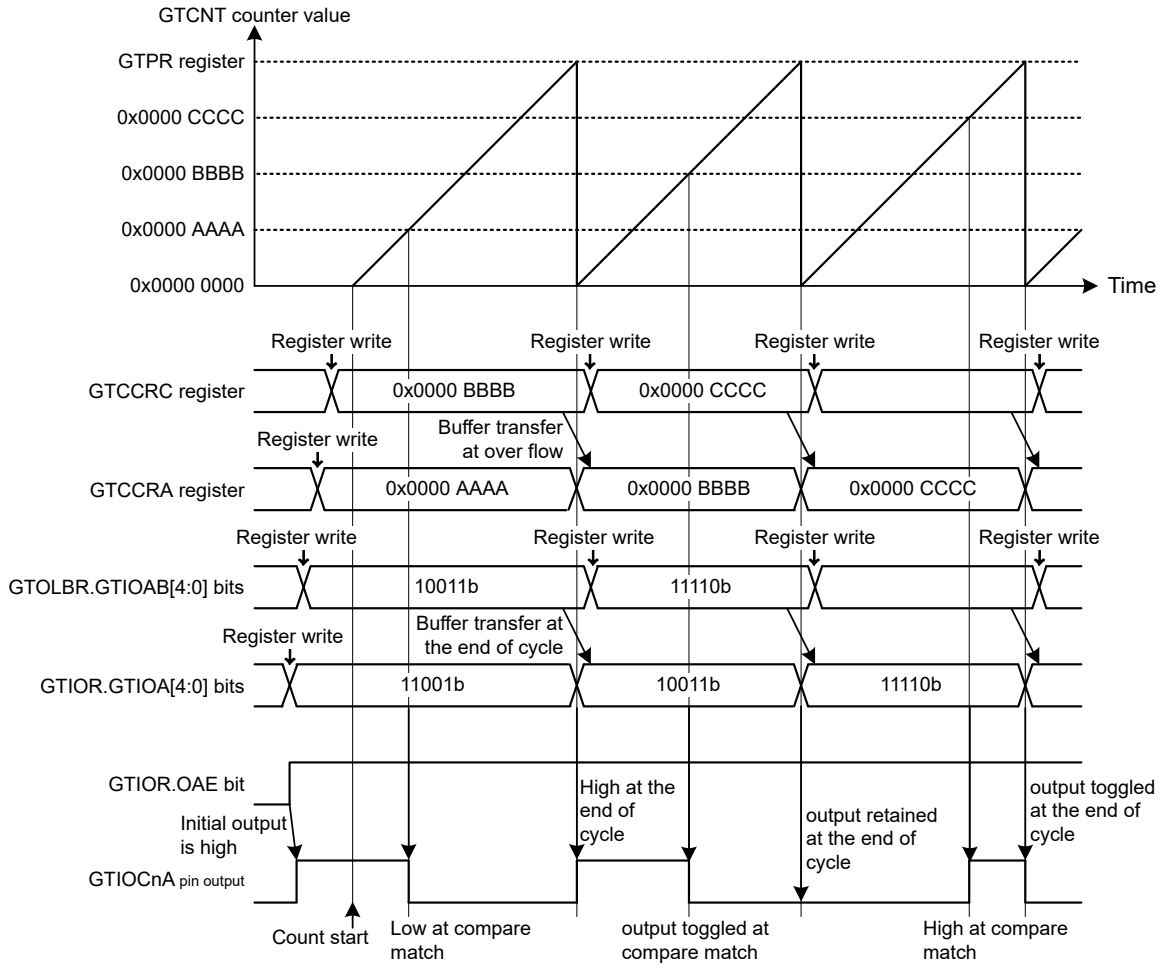
21.3.2.4 Buffer operation for GTIOA and GTIOB

The GTOLBR.GTIOAB[4:0] bits can function as the buffer register of the GTIOR.GTIOA[4:0] bits and the GTOLBR.GTIOBB[4:0] bits can function as the buffer register of the GTIOR.GTIOB[4:0] bits.

Buffer transfer timing can be set with the GTBER2.OLTTm[1:0] bits (m = A, B). It can be selected from the end of cycle or GTCCR register compare match (in saw-wave mode), from crest, trough, or both crest and trough (in triangle-wave mode and complementary PWM mode). In the case that the GTBER2.OLTTm[1:0] bits are 00b, buffer transfer is not performed.

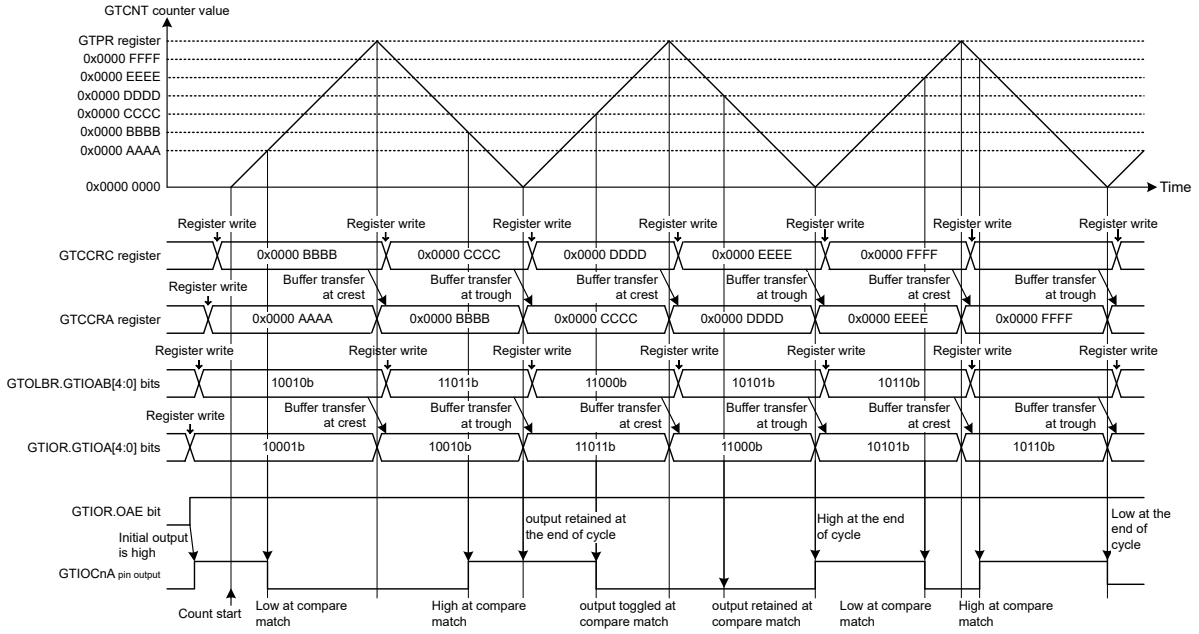
In complementary PWM mode, it is prohibited to set the buffer transfer timing to overlap with the dead time. Therefore, when the buffer transfer timing is crest, set the GTCCRm(m = A,C,E) register to satisfy GTCCRm < GTPR. And when the buffer transfer timing is trough, set the GTCCRm register to satisfy GTDVU < GTCCRm.

Figure 21.36 to Figure 21.38 show examples of buffer operation of the GTIOR.GTIOA[4:0] and GTIOR.GTIOB[4:0] bits. Table 21.23 shows an example for setting buffer operation of the GTIOR.GTIOA[4:0] and GTIOR.GTIOB[4:0] bits.



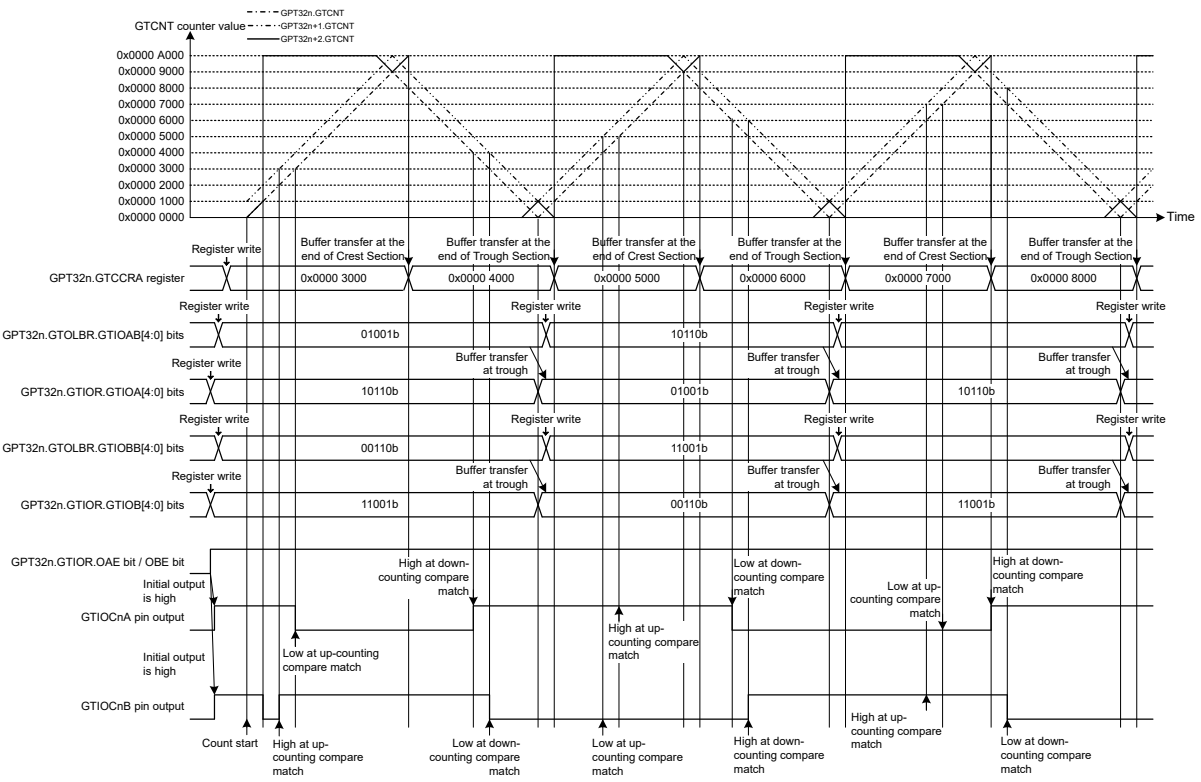
Note: Initial output is synchronized to the count clock.

Figure 21.36 Example of Buffer Operation of the GTIOA and GTIOB Bits (Up-Counting in Saw-Wave PWM Mode 1, Buffer Transfer at the End of Cycle)



Note: Initial output is synchronized to the count clock.

Figure 21.37 Example of Buffer Operation of the GTIOA and GTIOB Bits (Triangle-Wave PWM Mode 2, Buffer Transfer at Crests and Troughs)



Note: Initial output is synchronized to the count clock.

Figure 21.38 Example of Buffer Operation of the GTIOA and GTIOB Bits (Complementary PWM Mode 3, Buffer Transfer at Troughs)

Table 21.23 Example for Setting Buffer Operation of the GTIOA and GTIOB

No.	Step Name	Description
1	Set operating mode	Set the operating mode with GTCR.MD[3:0]. (In Figure 21.36, 0000b (saw-wave PWM mode 1) is set, in Figure 21.37, 0101b (triangle-wave PWM mode 2) is set, in Figure 21.38, 1110b (complementary PWM mode 3) is set.)
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. (In Figure 21.36, after 11b is set in GTUDDTYC[1:0], 01b is set in GTUDDTYC[1:0] (up-counting).)
3	Select count clock	Select the count clock with GTCR.TPCS[3:0] bits.
4	Set cycle	Other than the saw-wave PWM mode 2, Set the cycle in GTPR. In the saw-wave PWM mode 2, Select the counter clear source compare match register GTCCRx (x = A to F) by GTCSR.CSCMSC[2:0] bits and set the cycle in that register.
5	Set initial value for counter	Set the initial value in the GTCNT counter.
6	Set GTIOCnm pin function	Set the GTIOCnm pin function with GTIOA[4:0] and GTIOB[4:0] in GTIOR. (In Figure 21.36, GTIOA[4:0] = 11001b, in Figure 21.37, GTIOA[4:0] = 10001b, in Figure 21.38, GTIOA[4:0] = 10110b and GTIOB[4:0] = 11001b.)
7	Enable GTIOCnm pin output	Set to enable the GTIOCnm pin output with OAE and OBE in GTIOR.
8	Set buffer operation	Set buffer operation with OLTTm[1:0] bits in GTBER2 register. (In Figure 21.36, 01b is set in OLTTA[1:0], in Figure 21.37, 11b is set in OLTTA[1:0], in Figure 21.38, 10b is set in OLTTA[1:0] and 10b is set in OLTTB[1:0].)
9	Set buffer value	For buffer operation, set the GTIOCnm pin function in one cycle after the current cycle (in saw-wave mode and in triangle-wave/complementary PWM mode with buffer transfer at crest or trough) or half cycle after the current cycle (in triangle-wave/complementary PWM mode with buffer transfer at crest and trough) in the GTOLBR register.
10	Start count operation	Set GTCR.CST to 1 to start count operation.
11	Set buffer value of each cycle	For buffer operation, set the GTIOCnm pin function in one cycle after the current cycle (in saw-wave mode and in triangle-wave/complementary PWM mode with buffer transfer at crest or trough) or half cycle after the current cycle (in triangle-wave/complementary PWM mode with buffer transfer at crest and trough) in the GTOLBR register.

Note: n = 4 to 9
m = A, B

21.3.3 PWM Output Operating Mode

The GPT can output PWM waveforms to the GTIOCnA or GTIOCnB pin (n = 0 to 9) by a compare match between the GTCNT counter and GTCCRA or GTCCRB.

By setting GTDTCR, GTDVU, and GTDVD, the compare match value for a negative-phase waveform with dead time can automatically be set to GTCCRB.

In complementary PWM mode, it is possible to output PWM waveforms (positive phase, negative phase) with dead time that guarantees the linearity of the PWM output pulse width near 0% and 100% duty.

In saw-wave mode other than saw-wave PWM mode 2, or triangle-wave mode, or the master channel of complementary PWM mode, the signal synchronized with the PWM cycle can be output from the GTCPPOn output terminal by setting the GTIOR.PSYE bit to 1. The GTCPPOn output is toggled at the end of cycle in saw-wave mode or at the timing of crest / trough / GTCNT counter clearing in triangle-wave mode or complementary PWM mode. The initial output of GTCPPOn is low, and it becomes high when the count starts.

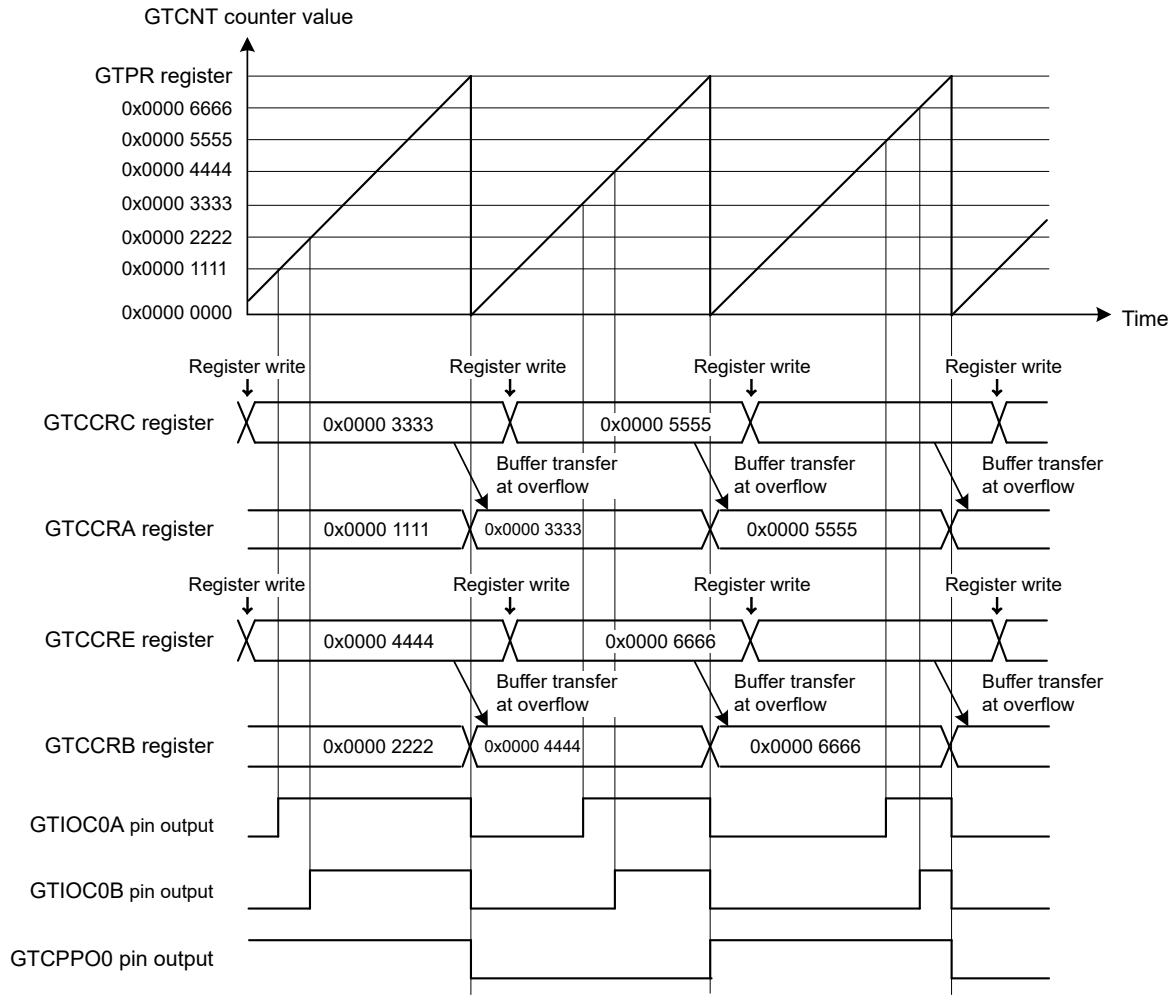
21.3.3.1 Saw-Wave PWM Mode 1

In saw-wave PWM mode 1, GTCNT performs saw-wave (half-wave) operation by setting the cycle in GTPR and a PWM waveform is output to the GTIOCnA or GTIOCnB pin (n = 0 to 9) when a GTCCRA or GTCCRB compare match occurs. The pin output value can be selected from low output, high output, or toggle output separately for a compare match and for the cycle end according to the GTIOR setting.

When 0 is set in the GTIOR.OxEOCD (x = A, B) bit and the timing of the end of cycle and the timing of GTCCRx register compare match are the same time, the output pin performs along the PWM output setting for the end of cycle set by the GTIOR.GTIOx[3:2] bit.

When 1 is set in the GTIOR.OxEOCD bit, GTIOCnx output is retained.

Figure 21.39 shows an example of saw-wave PWM mode 1 operation, and Table 21.24 shows an example for setting saw-wave PWM mode 1.



Note: The base address of registers are the same.

Figure 21.39 Example of saw-wave PWM mode 1 operation with up-counting, buffer operation, high output at GTCCRA/GTCCRB compare match, low output at cycle end, and GTIOR.PSYE = 1

Table 21.24 Example for setting saw-wave PWM mode 1 (1 of 2)

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits or GTCR.MD[3:0] bits. In Figure 21.39, 000b (saw-wave PWM mode 1) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 21.39, after 11b is set in the GTUDDTYC[1:0] bits, 01b is set in the GTUDDTYC[1:0] bits (up-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Set the cycle in the GTPR register.
5	Set initial value for counter	Set the initial value in the GTCNT counter.
6	Set GTIOCnm pin function	Set the GTIOCnm pin function with the GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register. In Figure 21.39, GTIOA[4:0] = 00110b and GTIOB[4:0] = 00110b.
7	Enable GTCPPOn pin output	Set to enable/disable the GTCPPOn pin output with the PSYE bit in the GTIOR register.

Table 21.24 Example for setting saw-wave PWM mode 1 (2 of 2)

No.	Step Name	Description
8	Enable GTIOCnm pin output*1	Set to enable the GTIOCnm pin output with the OAE and OBE bits in the GTIOR register.
9	Set buffer operation	Set buffer operation with the CCRA[1:0] and CCRB[1:0] bits in the GTBER register. In Figure 21.39, CCRA[1:0] = 01b and CCRB[1:0] = 01b.
10	Set compare match value*1	Set the GTIOCnA pin transition in the GTCCRA register and the GTIOCnB pin transition in the GTCCRB register.
11	Set buffer value	For buffer operation, set the GTIOCnA and GTIOCnB pins transitions in 1 cycle after the current cycle in the GTCCRC and GTCCRE registers, respectively. For double buffer operation, also set the GTIOCnA and GTIOCnB pins transitions in 2 cycles after the current cycle in the GTCCRD and GTCCRF registers, respectively.
12	Start count operation	Set the GTCR.CST bit to 1 to start count operation.
13	Set buffer value for each cycle	For buffer operation, set the GTIOCnA and GTIOCnB pins transitions in 1 cycle after the current cycle in the GTCCRC and GTCCRE registers, respectively. For double buffer operation, also set the GTIOCnA and GTIOCnB pins transitions in 2 cycles after the current cycle in the GTCCRD and GTCCRF registers, respectively.

Note: n: 0 to 9
m: A, B

Note 1. When using PWM Delay Generation circuit, change the order of setting for the GTIOCnm pin output enable and setting for a compare match value.

21.3.3.2 Saw-Wave PWM Mode 2

The saw-wave PWM mode 2 is a mode in which the GTCNT counter is operated as a saw wave by up-counting without using the GTPR register, and the PWM waveform is output by the compare match of the GTCCRA and GTCCRB registers. The pin output level can be selected from low output, high output, or toggle output separately for a compare match according to the GTIOR register setting.

The GTIOCnA pin is used as an output pin. Use the GTIOR.GTIOB[1:0] bits for setting the GTIOCnA pin output at a compare match of the GTCCRB register.

When a counter clear occurs due to the GTCNT counter clearing source selected in the GTCSR register, this is handled at the end of cycle and PWM output operation at the end of the cycle selected with the GTIOR.GTIOA[3:2] bits is performed. If a counter clear (at the end of cycle) conflicts with a PWM output change due to a GTCCRx (x = A, B) register compare match, PWM output operation is performed at the end of cycle (in the case of the GTIOR.OxEOCD bit = 0) or the PWM output is retained (in the case of GTIOR.OxEOCD bit = 1).

Figure 21.40 to Figure 21.42 shows an example of saw-wave PWM mode 2 operation. Table 21.25 shows an example for setting saw-wave PWM mode 2.

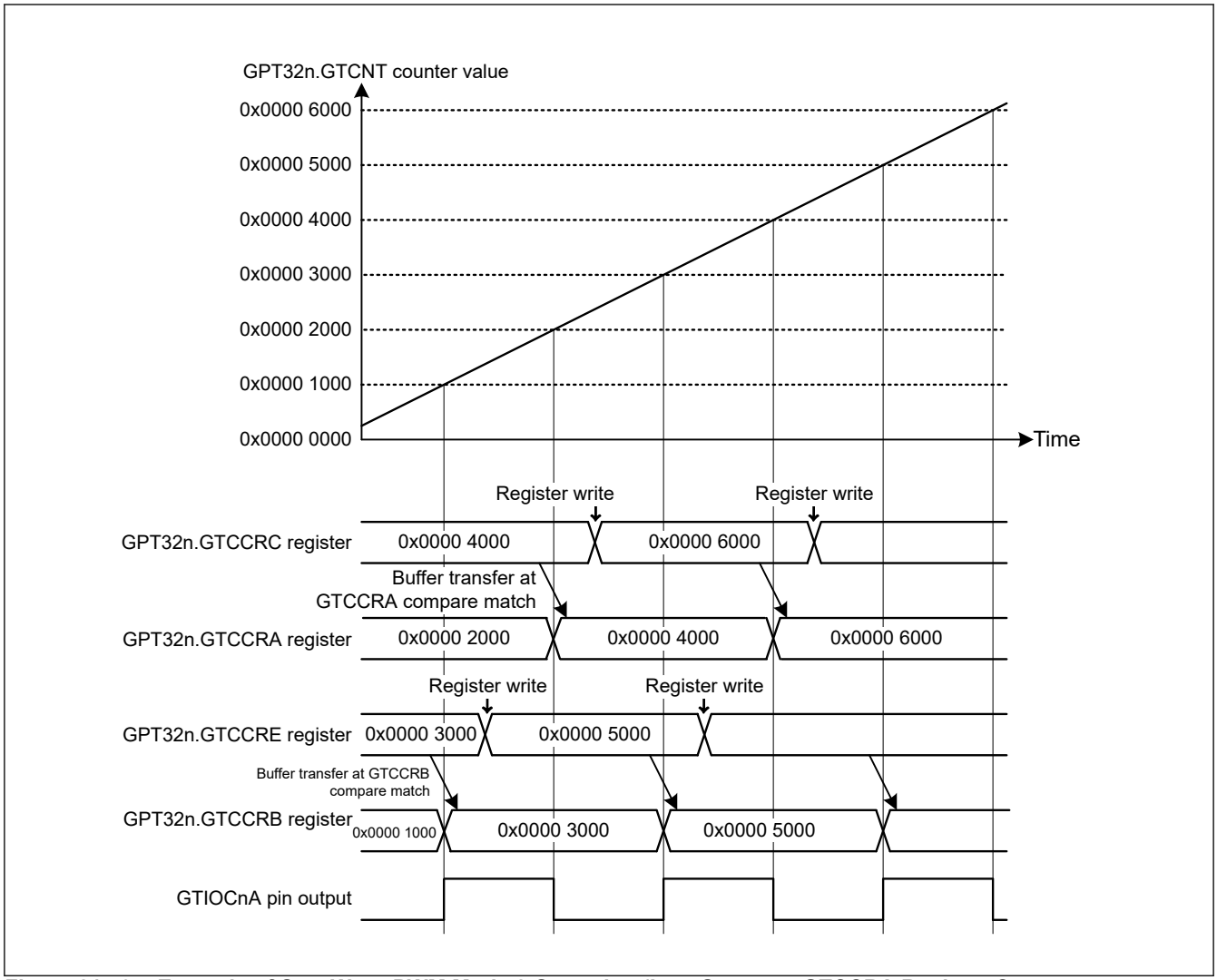


Figure 21.40 Example of Saw-Wave PWM Mode 2 Operation (Low Output at GTCCRA Register Compare Match, High Output at GTCCRB Register Compare Match, Single Buffer Operation, No Clear Setting) (n = 4 to 9)

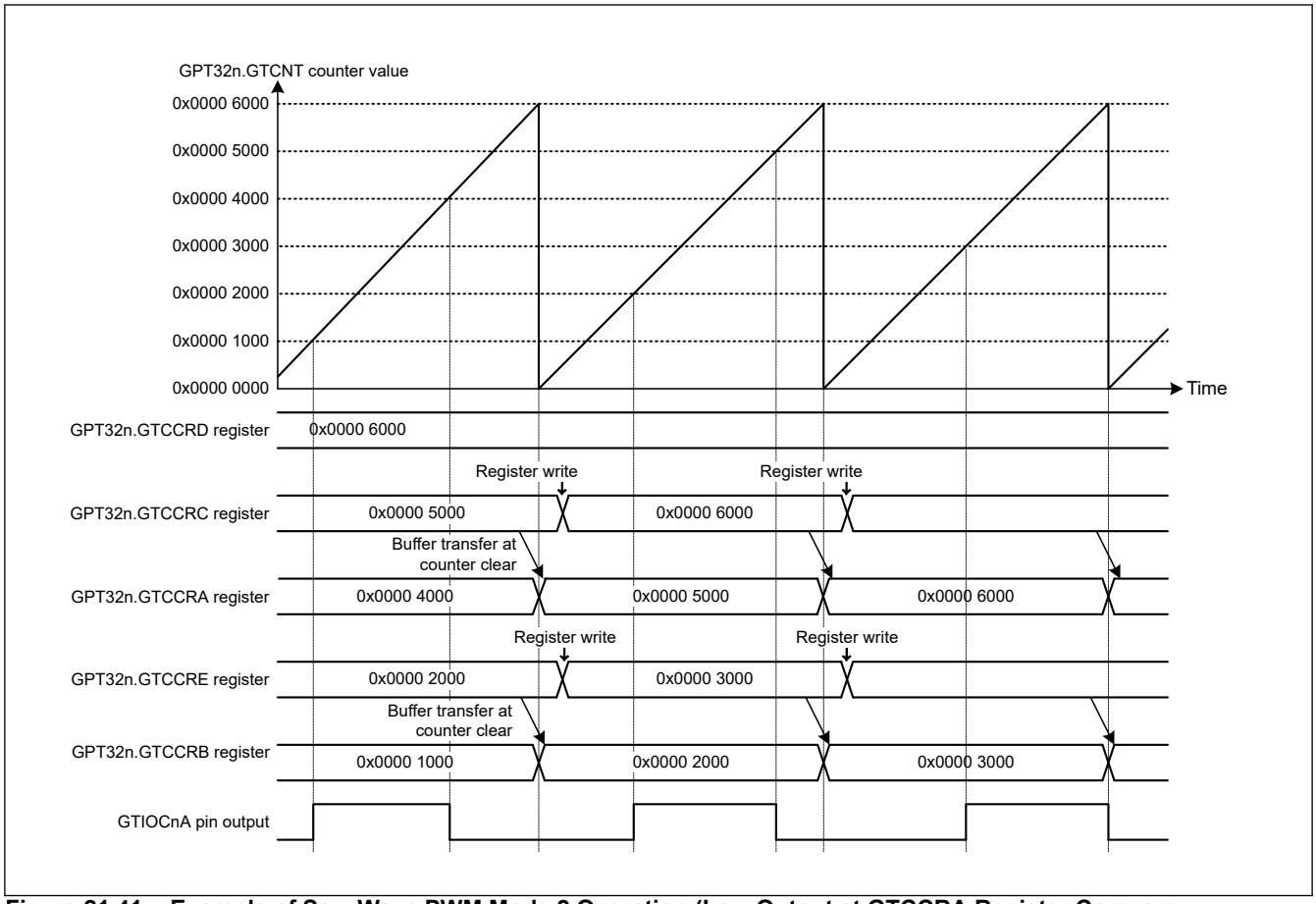


Figure 21.41 Example of Saw-Wave PWM Mode 2 Operation (Low Output at GTCCRA Register Compare Match, High Output at GTCCRB Register Compare Match, Low Output at the End of Cycle, Single Buffer Operation, Cleared at GTCCRD Register Compare Match, GTIOR.OAEOCD Bit = 0) (n = 4 to 9)

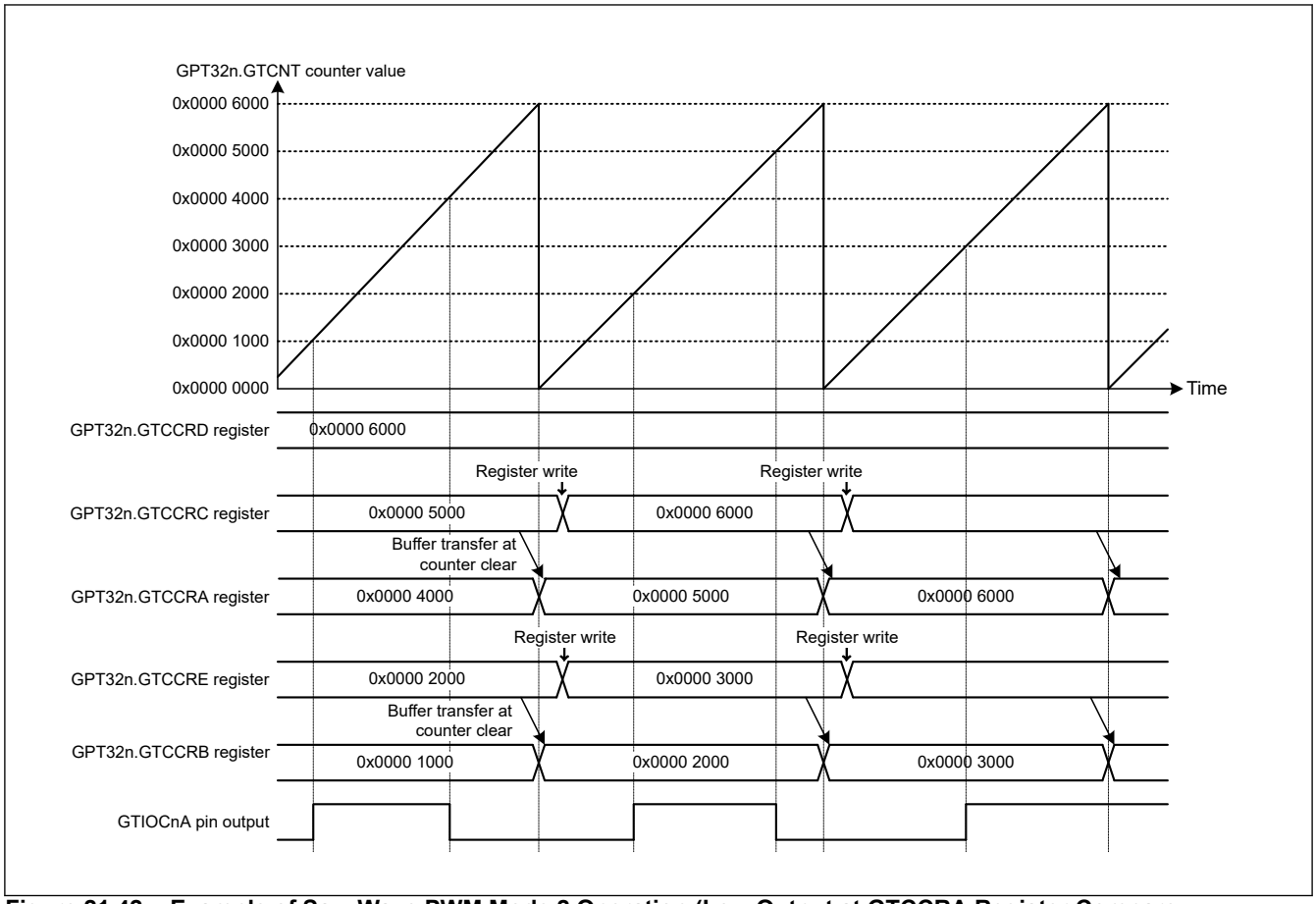


Figure 21.42 Example of Saw-Wave PWM Mode 2 Operation (Low Output at GTCCRA Register Compare Match, High Output at GTCCRB Register Compare Match, Low Output at the End of Cycle, Single Buffer Operation, Cleared at GTCCRD Register Compare Match, GTIOR.OAEOCD Bit =1) (n = 4 to 9)

Table 21.25 Example for Setting Saw-wave PWM mode 2

No.	Step Name	Description
1	Set operating mode	Set the operating mode with GTCR.MD[3:0]. (In Figure 21.40 to Figure 21.42, 0010b (saw-wave PWM mode 2) is set.)
2	Select count clock	Select the count clock with GTCR.TPCS[3:0] bits.
3	Set initial value for counter	Set the initial value in the GTCNT counter.
4	Set GTIOCnA pin function	Set the GTIOCnA pin function with GTIOA[4:0] and GTIOB[4:0] in GTIOR. (In Figure 21.40 to Figure 21.42, GTIOA[4:0] = 00001b and GTIOB[1:0] = 10b.)
5	Enable GTIOCnA pin output*1	Set to enable the GTIOCnA pin output with OAE in GTIOR.
6	Set buffer operation	Set buffer operation with CCRA[1:0] and CCRB[1:0] bits in GTBER (In Figure 21.40 to Figure 21.42, CCRA[1:0] = 01b and CCRB[1:0] = 01b.)
7	Set compare match value*1	Set the GTIOCnA pin changing point in GTCCRA and GTCCRB.
8	Set buffer value	For buffer operation, set the GTCCRA register value and the GTCCRB register value which used in one cycle after the current cycle in the GTCCRC and GTCCRE registers, respectively. For double buffer operation, also set the GTCCRA register value and the GTCCRB register value which used in two cycles after the current cycle in the GTCCRD and GTCCRF registers, respectively.
9	Start count operation	Set GTCR.CST to 1 to start count operation.
10	Set buffer value of each cycle	For buffer operation, set the GTCCRA register value and the GTCCRB register value which used in one cycle after the current cycle in the GTCCRC and GTCCRE registers, respectively. For double buffer operation, also set the GTCCRA register value and the GTCCRB register value which used in two cycles after the current cycle in the GTCCRD and GTCCRF registers, respectively.

Note: n: 4 to 9
Note 1. When using PWM Delay Generation circuit, change the order of setting for the GTIOCnA pin output enable and setting for a compare match value.

21.3.3.3 Saw-Wave One-Shot Pulse Mode

The saw-wave one-shot pulse mode is a mode in which the cycle is set in GTPR, the GTCNT counter performs saw-wave (half-wave) operation and a PWM waveform is output to the GTIOCnA or GTIOCnB pin (n = 0 to 9) at a compare match of GTCCRA or GTCCRB with buffer operation fixed.

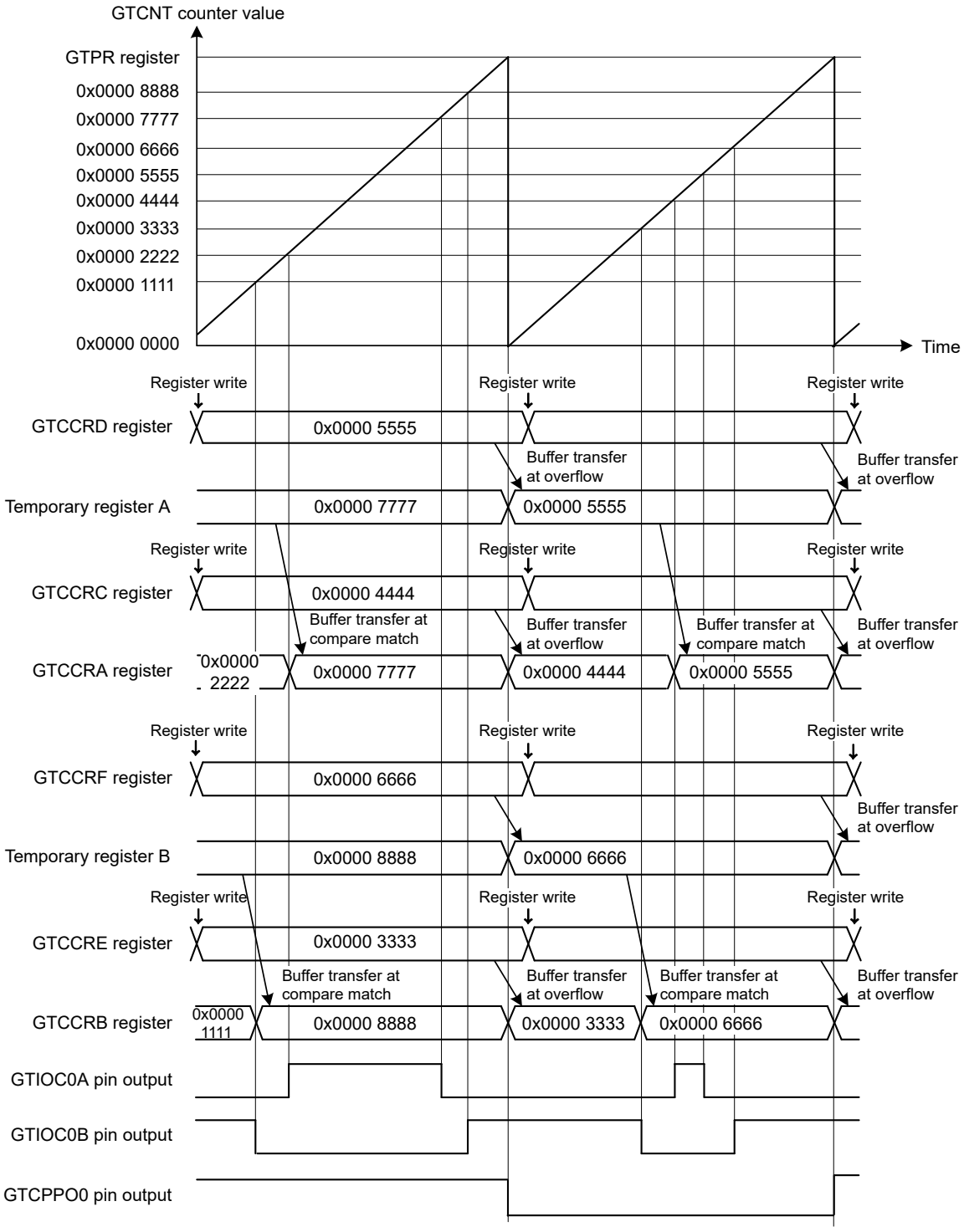
Buffer operation in saw-wave one-shot pulse mode is different from the usual buffer operation. Buffer transfer is performed from:

- GTCCRC to GTCCRA at the cycle end
- GTCCRE to GTCCRB at the cycle end
- GTCCRD to temporary register A at the cycle end
- GTCCRF to temporary register B at the cycle end
- Temporary register A to GTCCRA at a GTCCRA compare match
- Temporary register B to GTCCRB at a GTCCRB compare match.

The pin output value can be selected from low output, high output, or toggled output separately for a compare match and the cycle end according to the GTIOR setting. When the GTBER.CCRSWT bit is set to 1 while count operation is stopped, the buffer is transferred forcibly from the GTCCRD register to temporary register A and from the GTCCRF register to temporary register B. By setting GTDTCR, GTDVU, and GTDVD, a compare match value for a negative-phase waveform with dead time can automatically be set to GTCCRB.

When the GTBER.DBRTECm (m = A, B) bit is set to 1, transfer from an intermediate buffer to the GTCCRm register is repeated on a cyclic basis with using the temporary register x (x = C, E) and temporary register m which operate as intermediate buffers for the GTCCRx and GTCCRm registers, respectively, even while buffer transfer is disabled (repeated double buffer operation function during disabling of buffer transfer). For details, see section 21.8.2.2. Repeated Double-Buffered Operation When Disabling GTCCR Buffer Transfer.

Figure 21.43 shows an example of saw-wave one-shot pulse mode operation, and Table 21.26 shows an example for setting saw-wave one-shot pulse mode.



Note: The base address of registers are the same.

Figure 21.43 Example of saw-wave one-shot pulse mode operation with up-counting, low output from the GTIOCnA pin and high output from the GTIOCnB pin at count start, output toggled at GTCCRA/ GTCCRB compare match, output retained at cycle end, and GTIOR.PSYE = 1

Table 21.26 Example setting for saw-wave one-shot pulse mode

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits or GTCR.MD[3:0] bits. In Figure 21.43, 001b (saw-wave one-shot pulse mode) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 21.43, after 11b is set in the GTUDDTYC[1:0] bits, 01b is set in the GTUDDTYC[1:0] bits (up-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Set the cycle in the GTPR register.
5	Set initial value for counter	Set the initial value in the GTCNT counter.
6	Set GTIOCnm pin function	Set the GTIOCnm pin function with the GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register. In Figure 21.43, GTIOA[4:0] = 00011b and GTIOB[4:0] = 10011b.
7	Enable GTCPPOn pin output	Set to enable/disable the GTCPPOn pin output with the PSYE bit in the GTIOR register.
8	Enable GTIOCnm pin output*1	Set to enable the GTIOCnm pin output with the OAE and OBE bits in the GTIOR register.
9	Set compare match value*1	Set the GTIOCnA pin transition immediately after the count start in the GTCCRC and GTCCRD registers and the GTIOCnB pin transition in the GTCCRE and GTCCRF registers.
10	Set forcible buffer transfer*1	Set the GTBER.CCRSWT bit to 1 to transfer buffer register data forcibly.
11	Set buffer value	For buffer operation, set the GTIOCnA pin transition in one cycle after the current cycle in the GTCCRC and GTCCRD registers and the GTIOCnB pin transition in the GTCCRE and GTCCRF registers.
12	Start count operation	Set the GTCR.CST bit to 1 to start count operation.
13	Set buffer value for each cycle	For buffer operation, set the GTIOCnA pin transition in one cycle after the current cycle in the GTCCRC and GTCCRD registers and the GTIOCnB pin transition in the GTCCRE and GTCCRF registers.

Note: n: 0 to 9
m: A, B

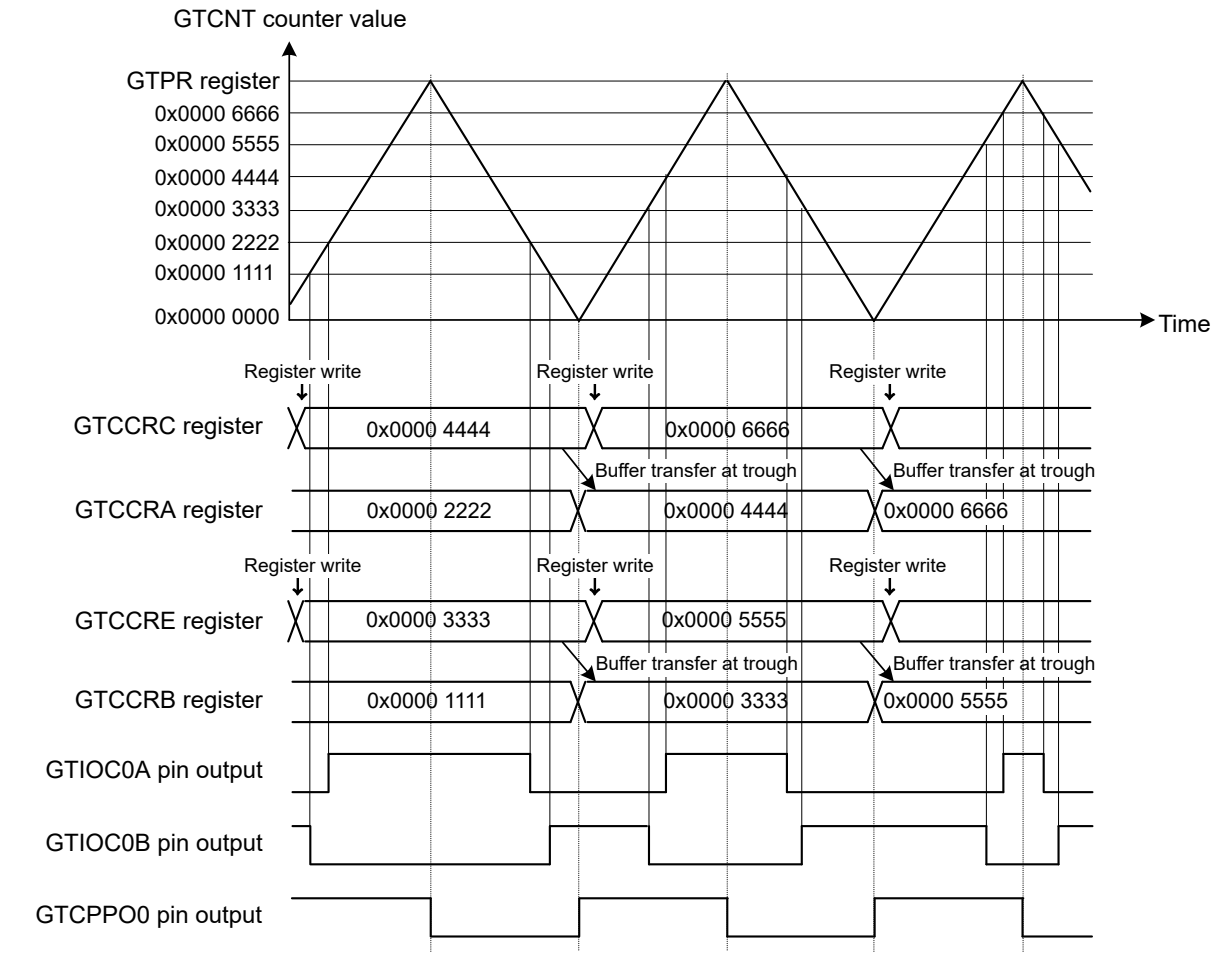
Note 1. When using PWM Delay Generation circuit, change the order of Enable GTIOCnm pin output and Set compare match value + Set forcible buffer transfer.

21.3.3.4 Triangle-Wave PWM Mode 1 (32-Bit Transfer at Trough)

The triangle-wave PWM mode 1 is a mode in which the cycle is set in GTPR. The GTCNT counter performs triangle-wave (full-wave) operation, and a PWM waveform is output to the GTIOCnA or GTIOCnB pin (n = 0 to 9) when a GTCCRA or GTCCRB compare match occurs. Buffer transfer is performed at the trough. The pin output value can be selected from low output, high output, or toggled output separately for a compare match and for the cycle end according to the GTIOR setting.

By setting GTDTCR, GTDVU, and GTDVD, a compare match value for a negative-phase waveform with dead time can automatically be set to GTCCRB.

Figure 21.44 shows an example of a triangle-wave PWM mode 1 operation, and Table 21.27 shows an example for setting a triangle-wave PWM mode 1.



Note: The base address of registers are the same.

Figure 21.44 Example of triangle-wave PWM mode 1 operation with buffer operation, low output from the GTIOCnA pin and high output from the GTIOCnB pin at count start, output toggled at GTCCRA/ GTCCRB register compare match, output retained at cycle end, and GTIOR.PSYE = 1

Table 21.27 Example setting for triangle-wave PWM mode 1 (1 of 2)

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits or GTCR.MD[3:0] bits. In Figure 21.44, 100b (triangle-wave PWM mode 1) is set.
2	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
3	Set cycle	Set the cycle in the GTPR register.
4	Set initial value for counter	Set the initial value in the GTCNT counter.
5	Set GTIOCnm pin function	Set the GTIOCnm pin function with the GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register. In Figure 21.44, GTIOA[4:0] = 00011b and GTIOB[4:0] = 10011b.
6	Enable GTCPPOn pin output	Set to enable/disable the GTCPPOn pin output with the PSYE bit in the GTIOR register.
7	Enable GTIOCnm pin output*1	Set to enable the GTIOCnm pin output with the OAE and OBE bits in the GTIOR register.
8	Set buffer operation	Set buffer operation with the CCRA[1:0] and CCRB[1:0] bits in the GTBER register. In Figure 21.44, CCRA[1:0] = 01b and CCRB[1:0] = 01b.
9	Set compare match value*1	Set the GTIOCnA and GTIOCnB pins transitions in the GTCCRA and GTCCRB registers, respectively.

Table 21.27 Example setting for triangle-wave PWM mode 1 (2 of 2)

No.	Step Name	Description
10	Set buffer value	For buffer operation, set the GTIOCnA and GTIOCnB pins transitions in 1 cycle after the current cycle in the GTCCRC and GTCCRE registers, respectively. For double buffer operation, also set the GTIOCnA and GTIOCnB pins transitions in 2 cycles after the current cycle in the GTCCRD and GTCCRF registers, respectively.
11	Start count operation	Set the GTCR.CST bit to 1 to start count operation.
12	Set buffer value for each cycle	For buffer operation, set the GTIOCnA and GTIOCnB pins transitions in 1 cycle after the current cycle in the GTCCRC and GTCCRE registers, respectively. For double buffer operation, also set the GTIOCnA and GTIOCnB pins transitions in 2 cycles after the current cycle in the GTCCRD and GTCCRF registers, respectively.

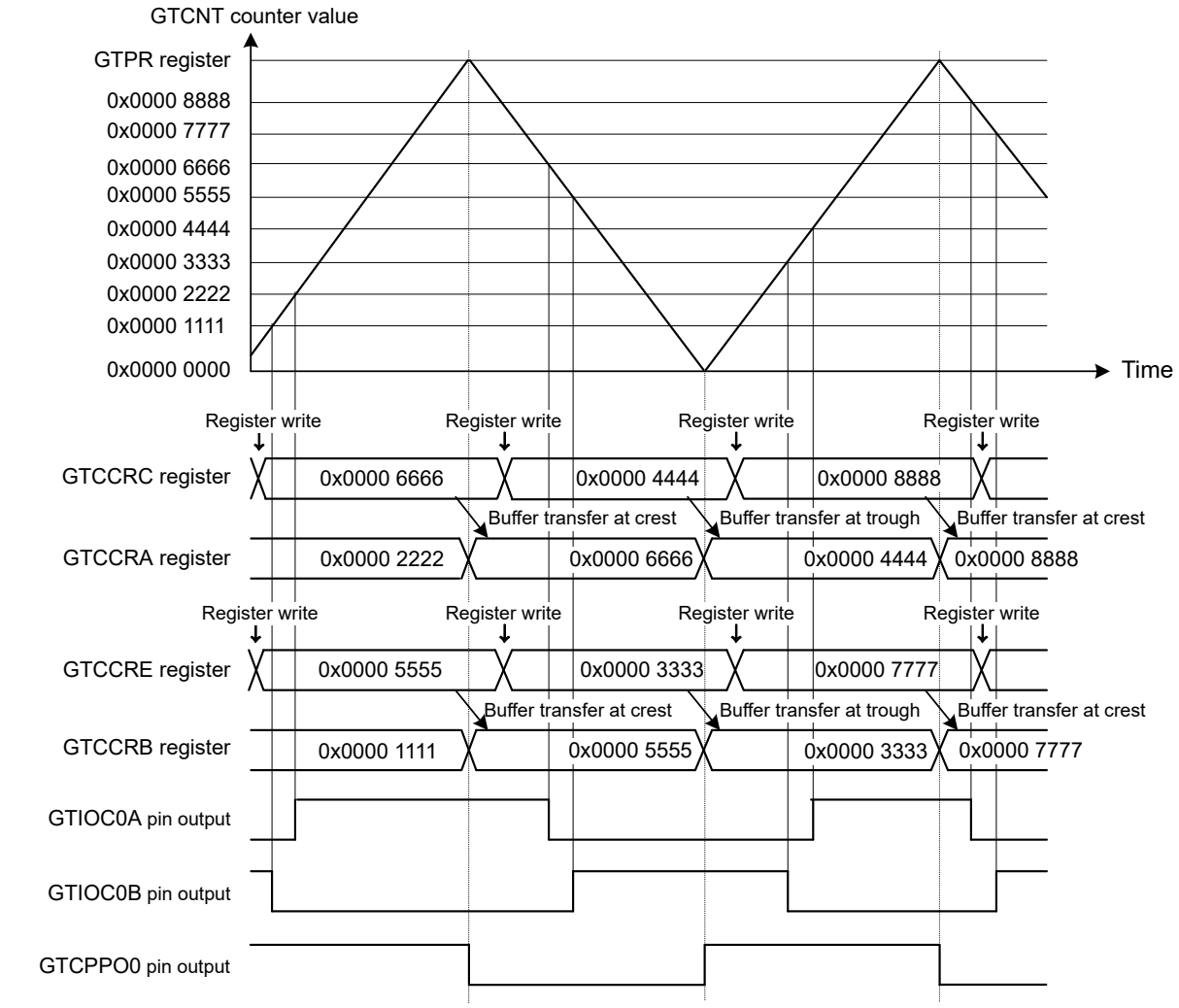
Note: n: 0 to 9
m: A, B
Note 1. When using PWM Delay Generation circuit, change the order of setting for the GTIOCnm pin output enable and setting for a compare match value.

21.3.3.5 Triangle-Wave PWM Mode 2 (32-Bit Transfer at Crest and Trough)

Similarly to triangle-wave PWM mode 1, in triangle-wave PWM mode 2 the cycle is set in GTPR. The GTCNT counter performs triangle-wave (full-wave) operation, and a PWM waveform is output to the GTIOCnA or GTIOCnB pin (n = 0 to 9) when a GTCCRA or GTCCRB compare match occurs. The buffer transfer is performed at both crests and troughs. The pin output value can be selected from low output, high output, or toggle output separately for a compare match and for the cycle end according to the GTIOR setting.

By setting GTDTCR, GTDVU, and GTDVD, a compare match value for a negative-phase waveform with dead time can automatically be set to GTCCRB.

Figure 21.45 shows an example of triangle-wave PWM mode 2 operation, and Table 21.28 shows an example for setting triangle-wave PWM mode 2.



Note: The base address of registers are the same.

Figure 21.45 Example of triangle-wave PWM mode 2 operation with buffer operation, low output from the GTIOCnA pin and high output from the GTIOCnB pin at count start, output toggled at GTCCRA/ GTCCRB compare match, output retained at cycle end, and GTIOR.PSYE = 1

Table 21.28 Example for setting triangle-wave PWM mode 2 (1 of 2)

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits or GTCR.MD[3:0] bits. In Figure 21.45, 101b (triangle-wave PWM mode 2) is set.
2	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
3	Set cycle	Set the cycle in the GTPR register.
4	Set initial value for counter	Set the initial value in the GTCNT counter.
5	Set GTIOCnm pin function	Set the GTIOCnm pin function with the GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register. In Figure 21.45, GTIOA[4:0] = 00011b and GTIOB[4:0] = 10011b.
6	Enable GTCPPOn pin output	Set to enable/disable the GTCPPOn pin output with the PSYE bit in the GTIOR register.
7	Enable GTIOCnm pin output*1	Set to enable the GTIOCnm pin output with the OAE and OBE bits in the GTIOR register.
8	Set buffer operation	Set buffer operation with the CCRA[1:0] and CCRB[1:0] bits in the GTBER register. In Figure 21.45, CCRA[1:0] = 01b and CCRB[1:0] = 01b.
9	Set compare match value*1	Set the GTIOCnA and GTIOCnB pins transitions in the GTCCRA and GTCCRB registers, respectively.

Table 21.28 Example for setting triangle-wave PWM mode 2 (2 of 2)

No.	Step Name	Description
10	Set buffer value	For buffer operation, set the GTIOCnA and GTIOCnB pins transitions in half cycle after the current cycle in the GTCCRC and GTCCRE registers, respectively. For double buffer operation, also set the GTIOCnA and GTIOCnB pins transitions in 1 cycle after the current cycle in the GTCCRD and GTCCRF registers, respectively.
11	Start count operation	Set the GTCR.CST bit to 1 to start count operation.
12	Set buffer value for each half cycle	For buffer operation, set the GTIOCnA and GTIOCnB pins transitions in half cycle after the current cycle in the GTCCRC and GTCCRE registers, respectively. For double buffer operation, also set the GTIOCnA and GTIOCnB pins transitions in 1 cycle after the current cycle in GTCCRD and GTCCRF registers, respectively.

Note: n: 0 to 9
m: A, B
Note 1. When using PWM Delay Generation circuit, change the order of setting for the GTIOCnm pin output enable and setting for a compare match value.

21.3.3.6 Triangle-Wave PWM Mode 3 (64-Bit Transfer at Trough)

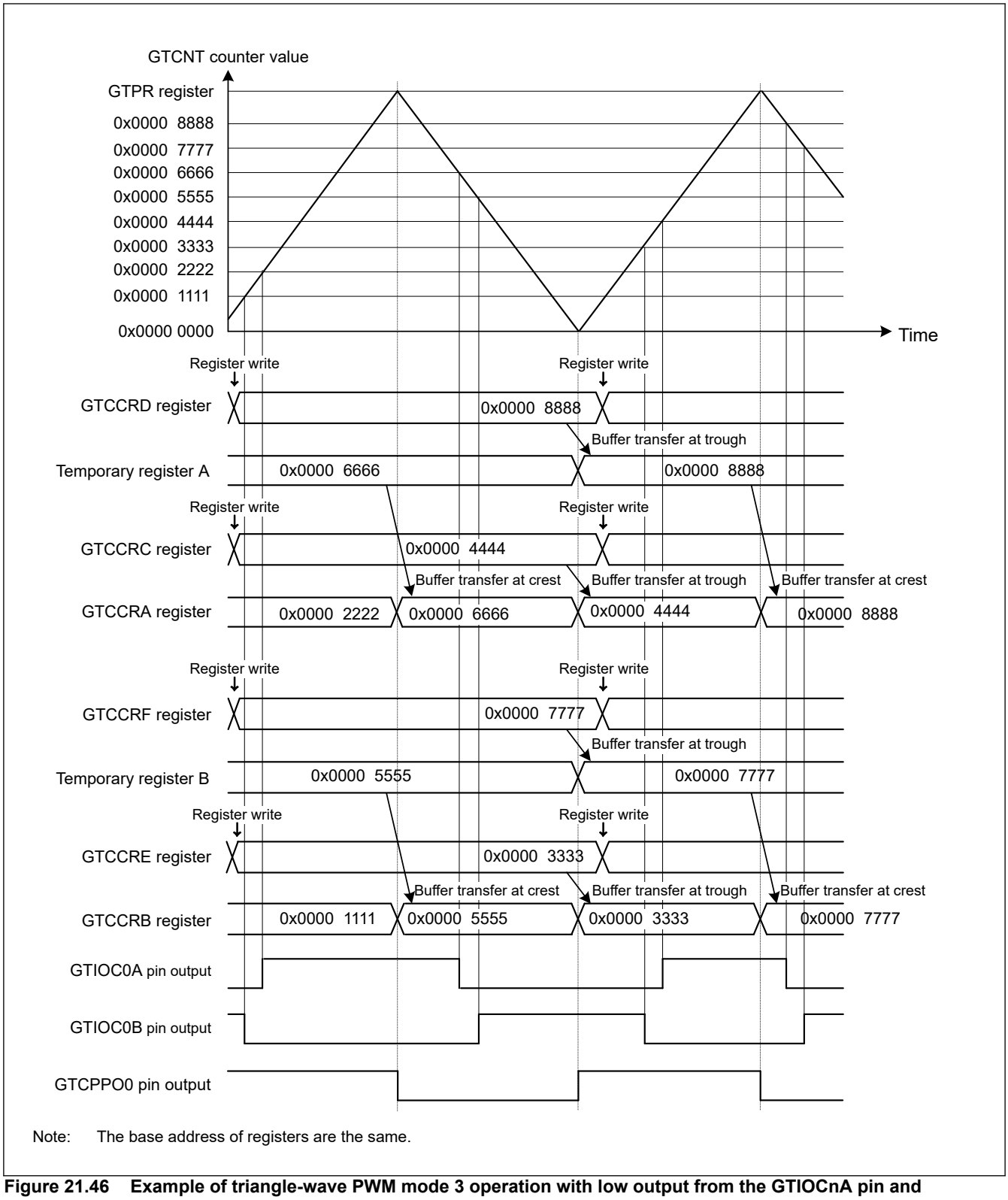
The triangle-wave PWM mode 3 is a mode in which the cycle is set in GTPR. The GTCNT counter performs triangle-wave (full-wave) operation and a PWM waveform is output to the GTIOCnA or GTIOCnB pin (n = 0 to 9) at a compare match of GTCCRA or GTCCRB with buffer operation fixed. Buffer operation in triangle-wave PWM mode 3 is different from the usual buffer operation. Buffer transfer is performed from the following:

- GTCCRC to GTCCRA at the trough
- GTCCRE to GTCCRB at the trough
- GTCCRD to temporary register A at the trough
- GTCCRF to temporary register B at the trough
- Temporary register A to GTCCRA at the crest
- Temporary register B to GTCCRB at the crest.

The pin output value can be selected from low output, high output, or toggled output separately for a compare match and for the cycle end according to the GTIOR setting.

By setting GTDTCR, GTDVU, and GTDVD, a compare match value for a negative-phase waveform with dead time can automatically be set to GTCCRB.

Figure 21.46 shows an example of triangle-wave PWM mode 3 operation, and Table 21.29 shows an example for setting triangle-wave PWM mode 3.



Note: The base address of registers are the same.

Figure 21.46 Example of triangle-wave PWM mode 3 operation with low output from the GTIOCnA pin and high output from the GTIOCnB pin at count start, output toggled at GTCCRA/GTCCRB compare match, output retained at cycle end, and GTIOR.PSYE = 1

Table 21.29 Example setting for triangle-wave PWM mode 3 (1 of 2)

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits or GTCR.MD[3:0] bits. In Figure 21.46, 110b (triangle-wave PWM mode 3) is set.

Table 21.29 Example setting for triangle-wave PWM mode 3 (2 of 2)

No.	Step Name	Description
2	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
3	Set cycle	Set the cycle in the GTPR register.
4	Set initial value for counter	Set the initial value in the GTCNT counter.
5	Set GTIOCnm pin function	Set the GTIOCnm pin function with the GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register. In Figure 21.46, GTIOA[4:0] = 00011b and GTIOB[4:0] = 10011b.
6	Enable GTCPPOn pin output	Set to enable/disable the GTCPPOn pin output with the PSYE bit in the GTIOR register.
7	Enable GTIOCnm pin output*1	Set to enable the GTIOCnm pin output with the OAE and OBE bits in the GTIOR register.
8	Set compare match value*1	Set the GTIOCnA pin transition immediately after the count start in the GTCCRC and GTCCRD registers and the GTIOCnB pin transition in the GTCCRE and GTCCRF registers.
9	Set forcible buffer transfer*1	Set the GTBER.CCRSWT bit to 1 to transfer buffer register data forcibly.
10	Set buffer value	Set the GTIOCnA pin transition in 1 cycle after the current cycle in the GTCCRC and GTCCRD registers and the GTIOCnB pin transition in the GTCCRE and GTCCRF registers.
11	Start count operation	Set the GTCR.CST bit to 1 to start count operation.
12	Set buffer value for each cycle	Set the GTIOCnA pin transition in 1 cycle after the current cycle in the GTCCRC and GTCCRD registers and the GTIOCnB pin transition in the GTCCRE and GTCCRF registers.

Note: n: 0 to 9
m: A, B

Note 1. When using PWM Delay Generation circuit, change the order of Enable GTIOCnm pin output and Set compare match value + Set forcible buffer transfer.

21.3.3.7 Complementary PWM mode 1,2,3

In complementary PWM mode, a three-phase PWM waveform with dead time that ensures the linearity in the vicinity of duty 0% and 100% can be output using the GTCNT counter of consecutive three channels. There are four modes depending on differences in buffer operation: (1) complementary PWM mode 1 (transfer at crests), (2) complementary PWM mode 2 (transfer at troughs), (3) complementary PWM mode 3 (transfer at crests and troughs), and (4) complementary PWM mode 4 (immediate transfer).

Figure 21.47 shows the block diagram in complementary PWM modes 1 to 3.

Among consecutive three channels, the lowest channel is referred to as master channel, and the adjacent upper two channels are referred to as slave channel 1 (lower) and slave channel 2 (upper).

The GTCNT counter of each channel performs individual count operation under the cycle operation by the master channel. In each channel, compare match with the GTCCRA register is performed selecting one of the three GTCNT counters in each operation section, and a positive-phase waveform and a negative-phase waveform are output from the GTIOCn+iA pin (i = 0, 1, 2) and the GTIOCn+iB pin respectively with a non-overlapping section of the dead time value set in the GTDVU register of the master channel.

The GTCCRA register performs buffer operation by the GTCCRC register, temporary register A, and GTCCRD register. In complementary PMW mode 3, setting the GTBER2.CP3DB bit to 1 also enables buffer operation of the GTCCRA register by the GTCCRE register, temporary register B, and GTCCRF register, allowing double buffer operation.

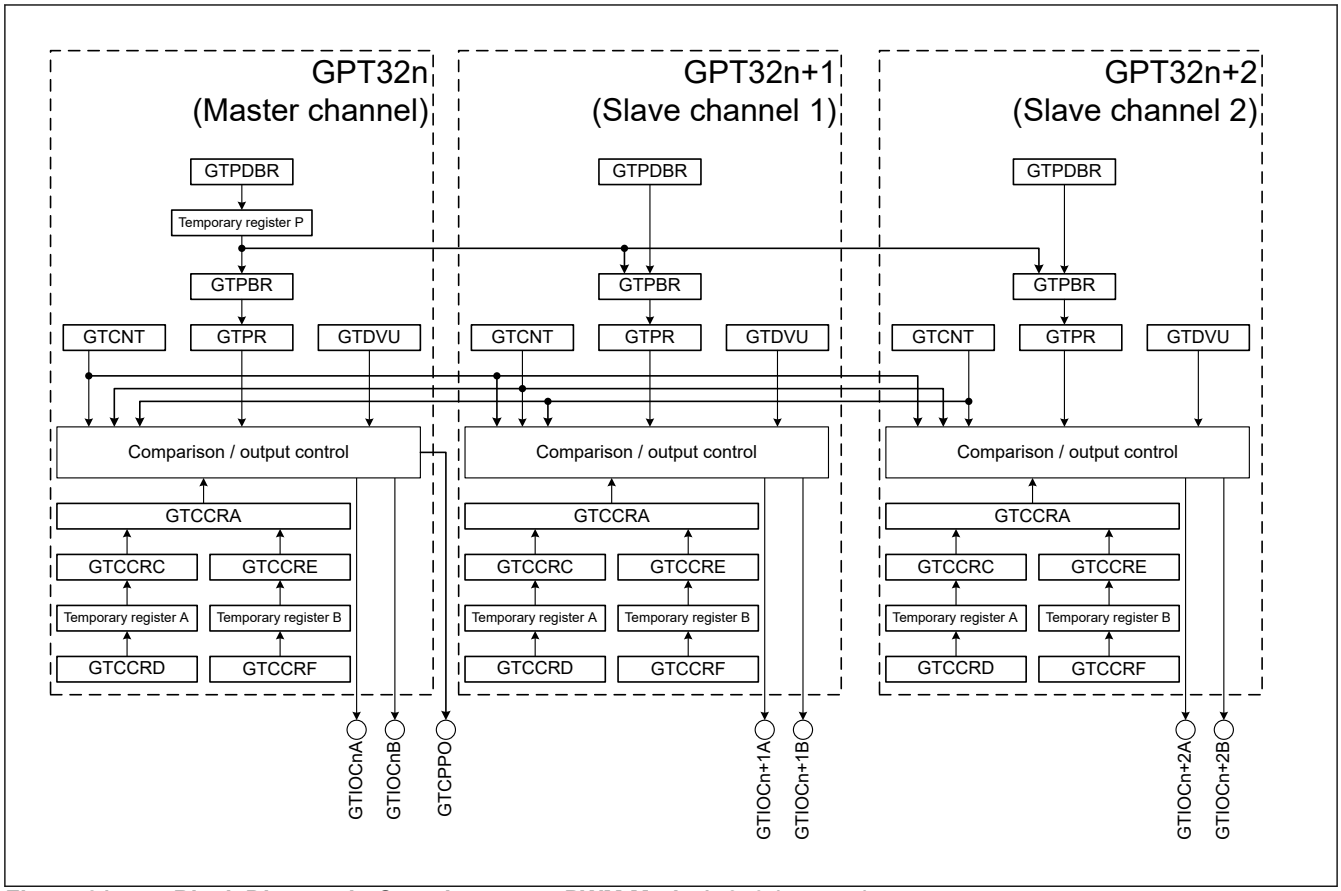


Figure 21.47 Block Diagram in Complementary PWM Mode 1, 2, 3 (n = 4, 7)

The GPT32n.GTCNT counter performs count operation for triangle waves using the GPT32n.GTPR register as a cycle register. A section where the GPT32n.GTCNT counter value is not larger than the dead time value is referred to as trough section.

The GPT32n+1.GTCNT counter performs count operation with the value (GPT32n.GTCNT counter value + dead time value set in the GPT32n.GTDVU register). A section where the GPT32n+1.GTCNT counter value is larger than the GPT32n.GTPR register value is referred to as crest section.

Crest section and trough section are classified into up-counting crest section, down-counting crest section, up-counting trough section, and down-counting trough section according to counting direction. A section between trough section and crest section is referred to as up-counting middle section or down-counting middle section according to counting direction. A section equivalent to the up-counting trough section after starting count operation is referred to as initial output section where operation differs partially from other up-counting trough sections.

The GPT32n+2.GTCNT counter functions as a counter to ensure the linearity in the vicinity of duty 0% and 100%. In a crest section, this counter performs count operation for a triangle wave (up-counting after down-counting) with the value (GPT32n.GTPR register value + dead time value) as an initial value and the GPT32n.GTPR register value as a trough. This counter is cleared to 0 at the end of the crest section, and then stops counting until the next trough section. In a trough section, this counter performs count operation for a triangle wave with an initial value of 0 and dead time value as a crest. This counter becomes the value (GPT32n.GTPR register value + dead time value) at the end of the trough section, and then stops counting until the next crest section. In the initial output section, however, this counter counts up with an initial value of 0 until the dead time value, and then becomes the value (GPT32n.GTPR register value + dead time value).

If the counter stops and then restarts in complementary PWM mode, the counter of each channel returns to the initial value after starting count operation, and then starts counting from the initial output section.

Table 21.30 and Table 21.31 shows count operation (counting direction/counting range) in each section. In these tables, registers with no channel identification indicate that the same value is stored in them of the master channel, slave channel 1, and slave channel 2.

Table 21.30 Count Operation in Complementary PWM Mode

Counter	Initial Value	Initial Output Section (After Start)	Up-Counting Middle Section	Up-Counting Crest Section	Down-Counting Crest Section
GPT32n.GTCNT	0	Up-counting 0 to GTDVU	Up-counting GTDVU +1 to GTPR-GTDVU	Up-counting GTPR-GTDVU+1 to GTPR	Down-counting GTPR-1 to GTPR-GTDVU
GPT32n+1.GTCNT	GTDVU	Up-counting GTDVU to GTDVU×2	Up-counting GTDVU×2+1 to GTPR	Up-counting GTPR+1 to GTPR+GTDVU	Down-counting GTPR+GTDVU-1 to GTPR
GPT32n+2.GTCNT	0	Up-counting 0 to GTDVU	Stop GTPR+GTDVU	Down-counting GTPR+GTDVU-1 to GTPR	Up-counting GTPR+1 to GTPR+GTDVU

Table 21.31 Count Operation in Complementary PWM Mode

Counter	Down-Counting Middle Section	Down-Counting Trough Section	Up-Counting Trough Section
GPT32n.GTCNT	Down-counting GTPR-GTDVU-1 to GTDVU	Down-counting GTDVU-1 to 0	Up-counting 1 to GTDVU
GPT32n+1.GTCNT	Down-counting GTPR-1 to GTDVU×2	Down-counting GTDVU×2-1 to GTDVU	Up-counting GTDVU+1 to GTDVU×2
GPT32n+2.GTCNT	Stop 0	Up-counting 1 to GTDVU	Down-counting GTDVU-1 to 0

In complementary PWM mode, the GTCCRA register performs unusual buffer operation.

Data transfers from the GTCCRD register to the temporary register A and from the GTCCRF register to the temporary register B are performed at the same time in three channels by writing a value to the GTCCRD register of the GPT32n+2 channel.

Data transfers from the temporary register A and temporary register B to the GTCCRC and GTCCRE registers vary depending on the transfer timing to the temporary register A and temporary register B. Data transfers from the GTCCRC and GTCCRE registers to the GTCCRA register are performed according to each complementary PWM mode name (crest transfer, trough transfer, and crest/trough transfer).

Buffer operation of the GTPR register in complementary PWM mode is described in section 21.3.2.1. GTPR Register Buffer Operation. Do not perform buffer operation for the GTDVU register in complementary PWM mode.

Table 21.32 shows buffer transfer timing during single buffer operation in complementary PWM modes 1 to 3. Table 21.33 shows buffer transfer timing during double buffer operation in complementary PWM mode 3.

Table 21.32 Single Buffer Transfer Timing in Complementary PWM Mode 1, 2, 3

Buffer Transfer	Complementary PWM Mode 1	Complementary PWM Mode 2	Complementary PWM Mode 3 (Single Buffer)
GTCCRD ↓ Temporary register A	After one GTCLK cycle from GTCCRD register write of slave channel 2 (GPT32n+2)	After one GTCLK cycle from GTCCRD register write of slave channel 2 (GPT32n+2)	After one GTCLK cycle from GTCCRD register write of slave channel 2 (GPT32n+2)
Temporary register A ↓ GTCCRC	(1) When data is transferred to temporary register A in up-counting middle section After one GTCLK cycle from transfer to temporary register A (2) When data is transferred to temporary register A in a section other than up-counting middle section At the end of trough section	(1) When data is transferred to temporary register A in down-counting middle section After one GTCLK cycle from transfer to temporary register A (2) When data is transferred to temporary register A in a section other than down-counting middle section At the end of crest section	(1) When data is transferred to temporary register A in middle section After one GTCLK cycle from transfer to temporary register A (2) When data is transferred to temporary register A in a section other than middle section At the end of crest and trough sections
GTCCRC ↓ GTCCRA	At the end of crest section Counter clear in up-counting middle section and crest section	At the end of trough section (excluding initial output section) Counter clear in down-counting middle section and trough section	At the end of crest section At the end of trough section (excluding initial output section) Counter clear

Table 21.33 Double Buffer Transfer Timing in Complementary PWM Mode 3

Transfer from GTCCRD to GTCCRA		Transfer from GTCCRF to GTCCRA	
Buffer Transfer	**Transfer Timing**	**Buffer Transfer**	**Transfer Timing**
GTCCRD ↓ Temporary register A	After one GTCLK cycle from GTCCRD register write of slave channel 2 (GPT32n+2)	GTCCRF ↓ Temporary register B	After one GTCLK cycle from GTCCRD register write of slave channel 2 (GPT32n+2)
Temporary register A ↓ GTCCRC	(1) When data is transferred to temporary register A in middle section After one GTCLK cycle from transfer to temporary register A (2) When data is transferred to temporary register A in a section other than middle section At the end of crest and trough sections	Temporary register B ↓ GTCCRE	(1) When data is transferred to temporary register B in middle section After one GTCLK cycle from transfer to temporary register B (2) When data is transferred to temporary register B in a section other than middle section At the end of crest and trough sections
GTCCRC ↓ GTCCRA	At the end of crest section Counter clear	GTCCRE ↓ GTCCRA	At the end of trough section (excluding initial output section)

An output level change in the positive-phase waveform from the GTIOCn+iA pin (i = 0, 1, 2) and the negative-phase waveform from the GTIOCn+iB pin occurs at a compare match in combination of counters and registers determined for each operation section. In middle sections, the positive-phase waveform output level changes at a compare match of the GPT32n.GTCNT counter and the GTCCRA register, and the negative-phase waveform output level changes at a compare match between the GPT32n+1.GTCNT counter and the GTCCRA register. In crest and trough sections, compare match operation is performed using the GPT32n+2.GTCNT counter, GTCCRC register, and GTCCRE register to ensure the linearity in the vicinity of duty 0% and 100%.

In the case that the compare match value equals to or larger than the GPT32n.GTPR register value, the duty becomes 0% (positive-phase waveform OFF, negative-phase waveform ON). In the case that the compare match value is 0, the duty becomes 100% (positive-phase waveform ON, negative-phase waveform OFF).

Table 21.34 lists combinations of counters and registers used for compare match operation to generate a positive-phase waveform and a negative-phase waveform in each operation section.

Table 21.34 Combinations of Counters and Registers for Compare Match Operation in Complementary PWM Mode

	Up-Counting Middle Section	Up-Counting Crest Section	Down-Counting Crest Section	Down-Counting Middle Section	Down-Counting Trough Section	Up-Counting Trough Section
Negative-phase OFF	GPT32n +1.GTCNT	GPT32n +1.GTCNT	—	GPT32n +2.GTCNT*1	GPT32n +2.GTCNT	GPT32n +1.GTCNT
	GTCCRA	GTCCRA	—	GTCCRC (GTCCRE for double buffer operation)	GTCCRC (GTCCRE for double buffer operation)	GTCCRC (GTCCRE for double buffer operation)
Positive-phase ON	GPT32n.GTCNT	GPT32n.GTCNT	GPT32n +2.GTCNT	—	GPT32n.GTCNT *1	GPT32n.GTCNT
	GTCCRA	GTCCRA	GTCCRA	—	GTCCRC (GTCCRE for double buffer operation)	GTCCRC (GTCCRE for double buffer operation)
Positive-phase OFF	GPT32n +2.GTCNT*1	GPT32n +2.GTCNT	GPT32n.GTCNT	GPT32n.GTCNT	GPT32n.GTCNT	—
	GTCCRC	GTCCRC	GTCCRC	GTCCRA	GTCCRA	—
Negative-phase ON	—	GPT32n +1.GTCNT*1	GPT32n +1.GTCNT	GPT32n +1.GTCNT	GPT32n +1.GTCNT	GPT32n +2.GTCNT
	—	GTCCRC	GTCCRC	GTCCRA	GTCCRA	GTCCRA

Note 1. Compare match is performed only at the time of final count in the target section, but is not performed at count values other than the final count.

In the case of normal complementary PWM mode waveform, a PWM waveform change occurs in the order of negative-phase OFF → positive-phase ON → positive-phase OFF → negative-phase ON. However, this order may vary depending on operation section and register values. In this case, OFF takes precedence in trough sections and ON takes precedence in crest sections (for negative-phase waveforms), and ON takes precedence in trough sections and OFF takes precedence in crest sections (for positive-phase waveforms). A lower-priority compare match that occurs at the same time or after a higher-priority compare match is ignored.

In the initial output section, the initial output set in the GTIOR register is retained. In the case that the GTCCRA register value is larger than the GTDVU register value at the end of the initial output section, negative phase is enabled. In the case that the GTCCRA register value is not larger than the GTDVU register value, positive phase is enabled.

As operation examples of normal complementary PWM mode waveform where compare match operation occurs in the middle section, Figure 21.48 and Figure 21.49 show complementary PWM mode 1, Figure 21.50 and Figure 21.51 show complementary PWM mode 2, Figure 21.52 and Figure 21.53 show single buffer complementary PWM mode 3, and Figure 21.54 and Figure 21.55 show double buffer complementary PWM mode 3.

Figure 21.56 to Figure 21.67 show complementary PWM mode waveforms where compare match operation occurs in crest sections and trough sections and differences due to compare match occurrence order.

Figure 21.68 and Figure 21.69 show examples of initial output operation according to the GTCCRA register value.

Table 21.35 shows an example for setting complementary PWM modes 1 to 3.

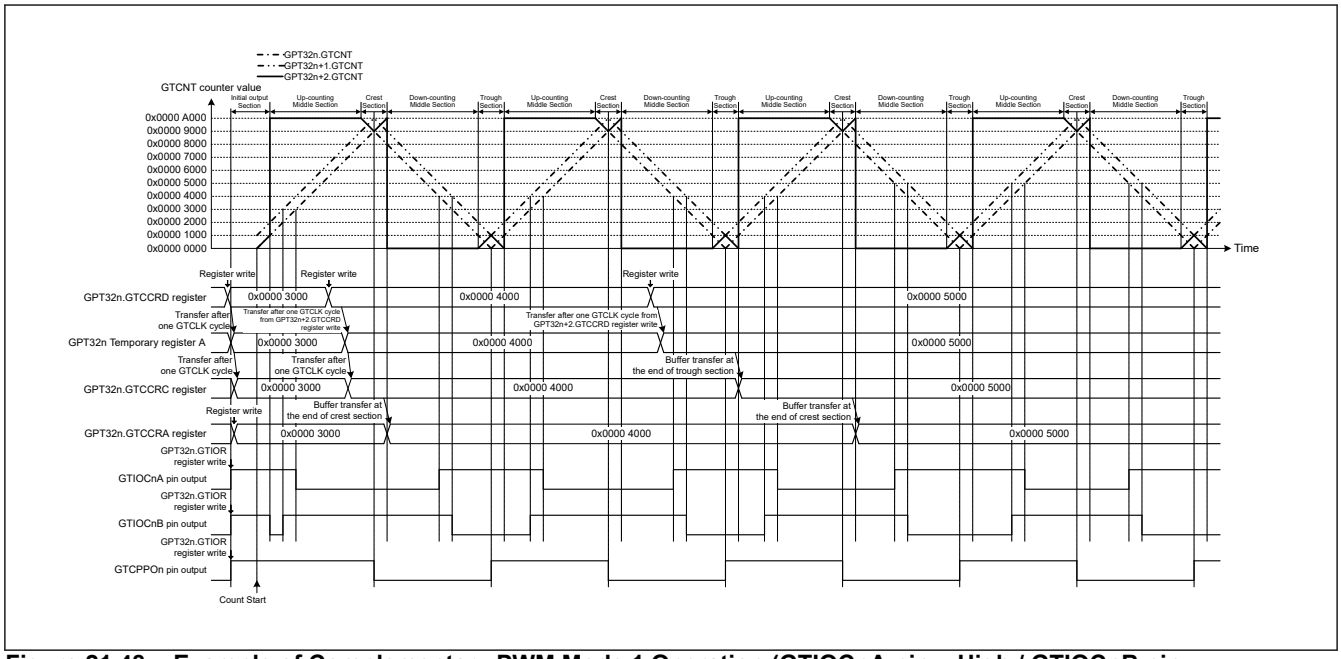


Figure 21.48 Example of Complementary PWM Mode 1 Operation (GTIOCnA pin = High / GTIOCnB pin = High as initial output, GTIOCnA pin = Low / GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High / GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 1000 and updating GTCCRD register in middle section) (n = 4, 7)

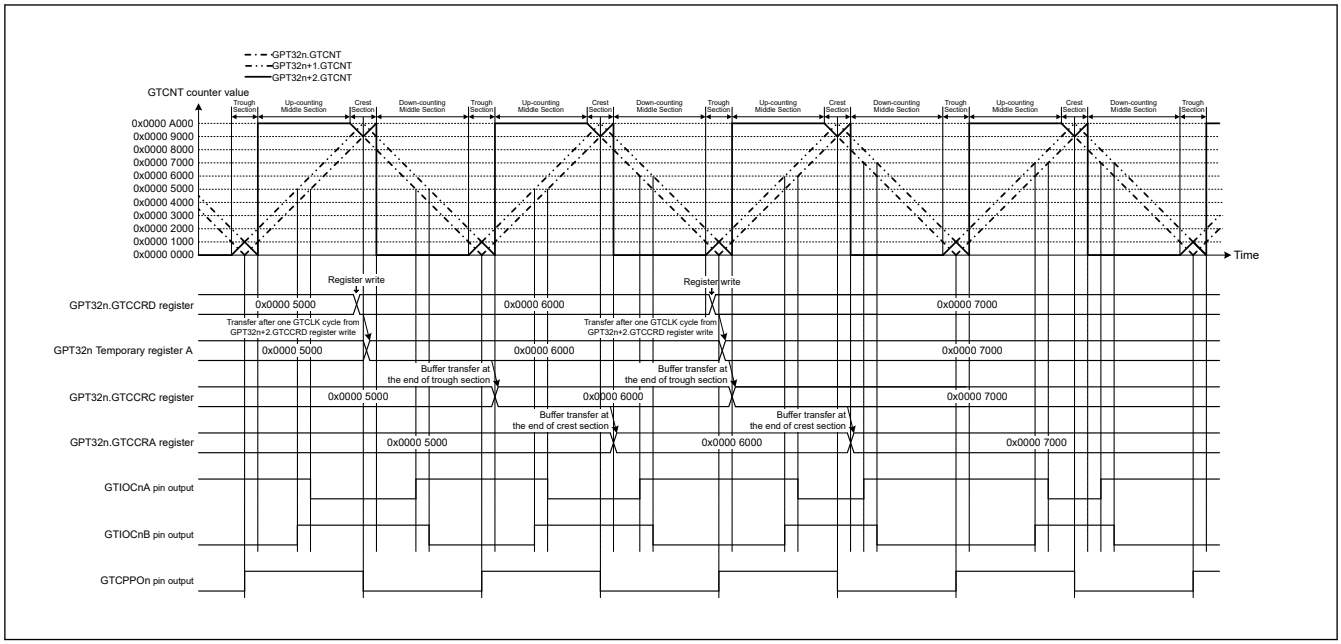


Figure 21.49 Example of Complementary PWM Mode 1 Operation (GTIOCnA pin = Low / GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High / GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 1000 and updating GTCCRD register in crest and trough sections) (n = 4, 7)

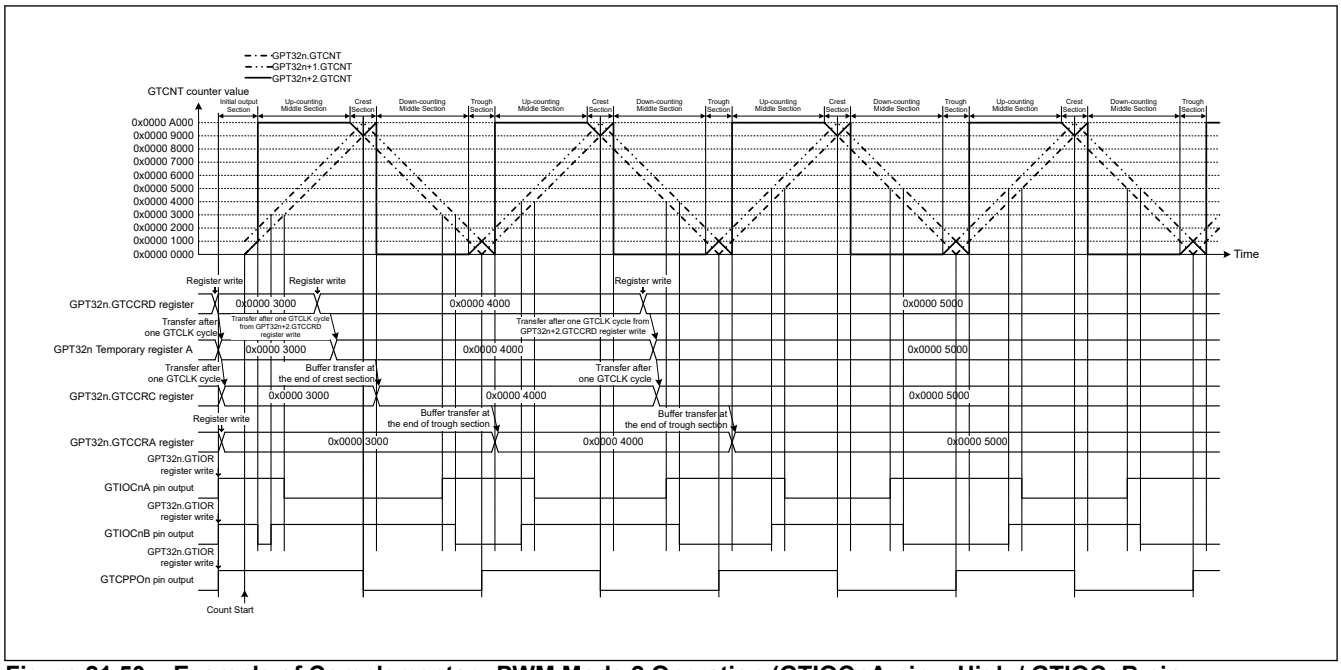


Figure 21.50 Example of Complementary PWM Mode 2 Operation (GTIOCnA pin = High / GTIOCnB pin = High as initial output, GTIOCnA pin = Low / GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High / GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 1000 and updating GTCCRD register in middle section) (n = 4, 7)

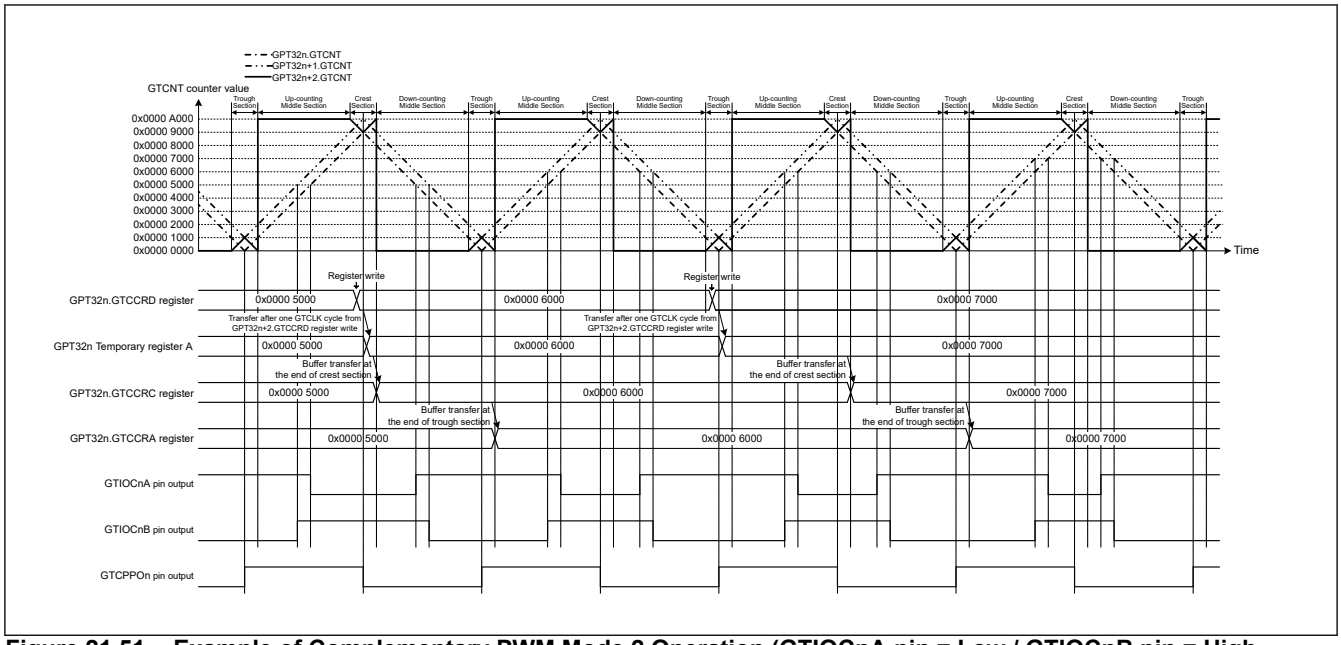


Figure 21.51 Example of Complementary PWM Mode 2 Operation (GTIOCnA pin = Low / GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High / GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 1000 and updating GTCCRD register in crest and trough sections) (n = 4, 7)

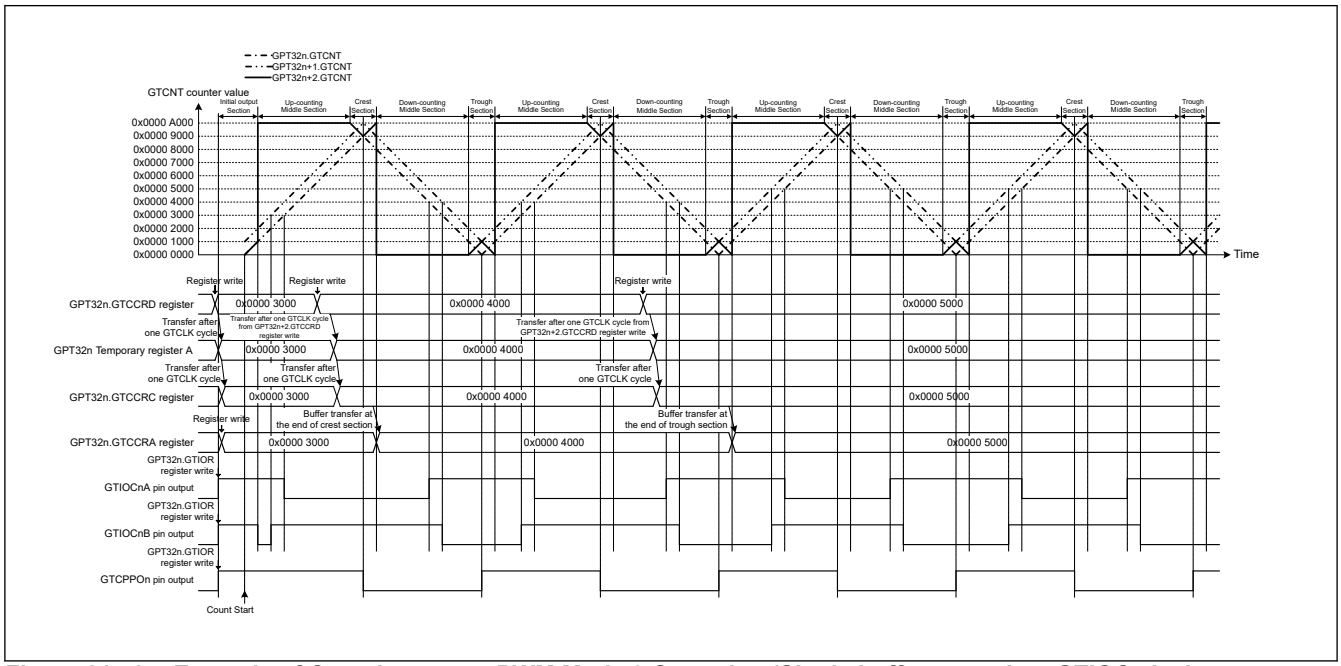


Figure 21.52 Example of Complementary PWM Mode 3 Operation (Single buffer operation, GTIOCnA pin = High / GTIOCnB pin = High as initial output, GTIOCnA pin = Low / GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High / GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 1000 and updating GTCCRD register in middle section) (n = 4, 7)

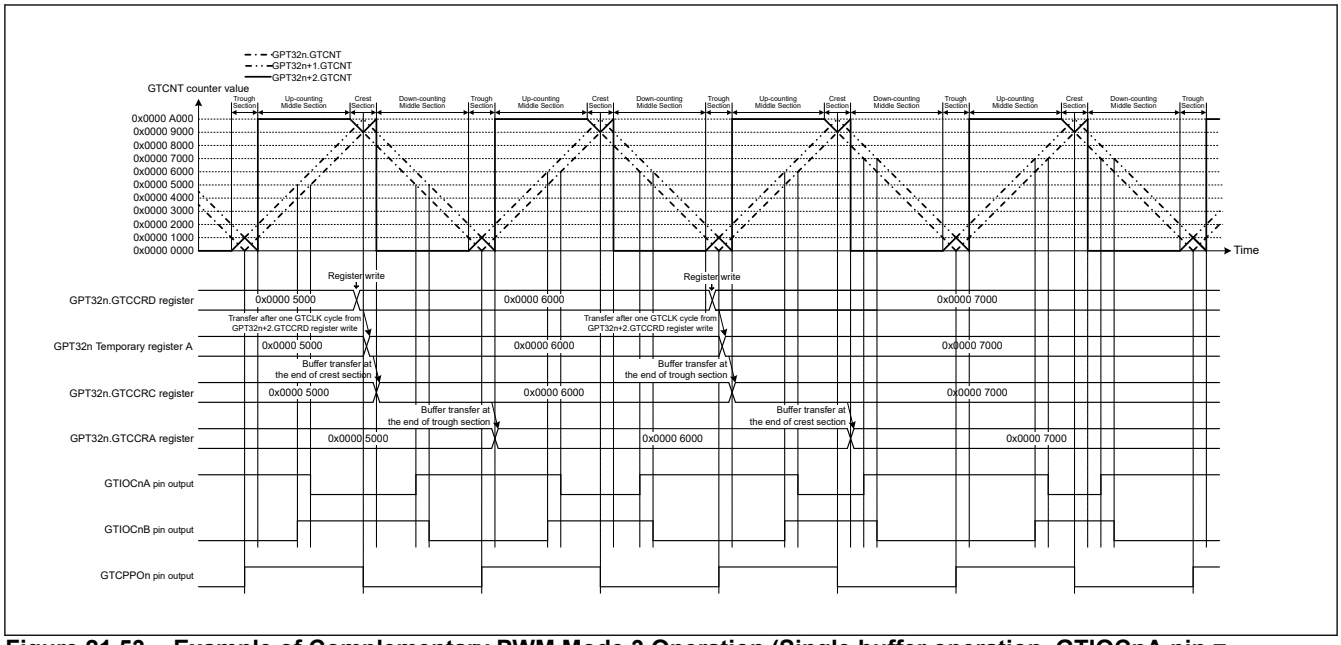


Figure 21.53 Example of Complementary PWM Mode 3 Operation (Single buffer operation, GTIOCnA pin = Low / GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High / GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 1000 and updating GTCCRD register in crest and trough sections) (n = 4, 7)

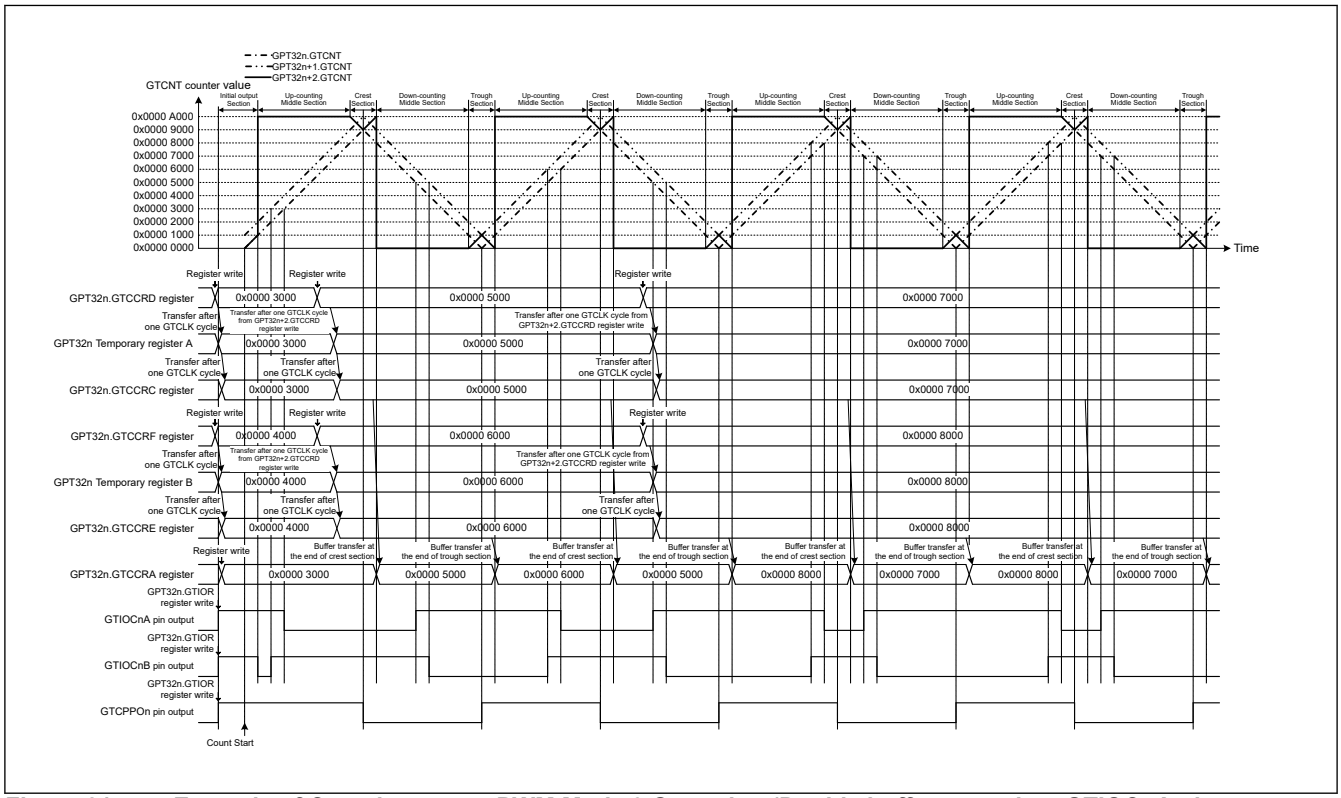


Figure 21.54 Example of Complementary PWM Mode 3 Operation (Double buffer operation, GTIOCnA pin = High / GTIOCnB pin = High as initial output, GTIOCnA pin = Low / GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High / GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 1000 and updating GTCCRD register in middle section) (n = 4, 7)

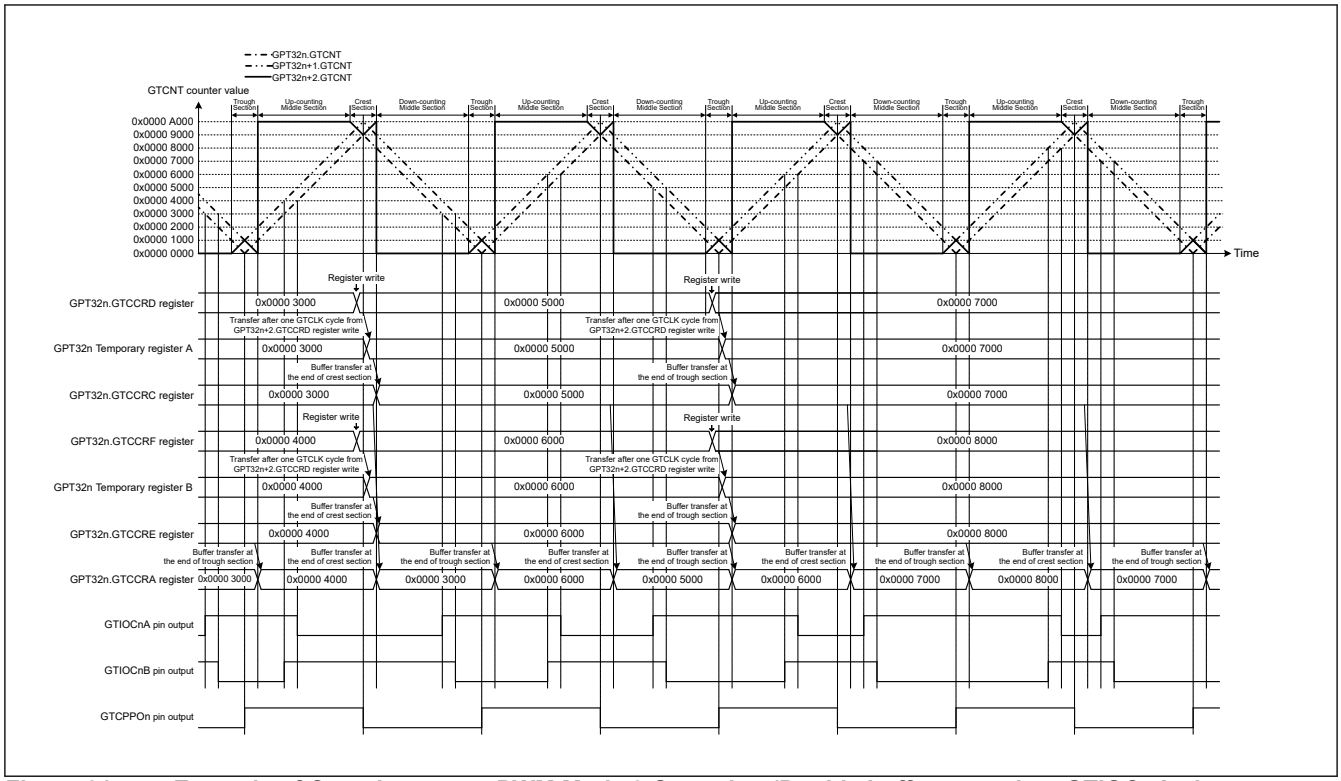


Figure 21.55 Example of Complementary PWM Mode 3 Operation (Double buffer operation, GTIOCnA pin = Low / GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High / GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 1000 and updating GTCCRD register in crest and trough sections) (n = 4, 7)

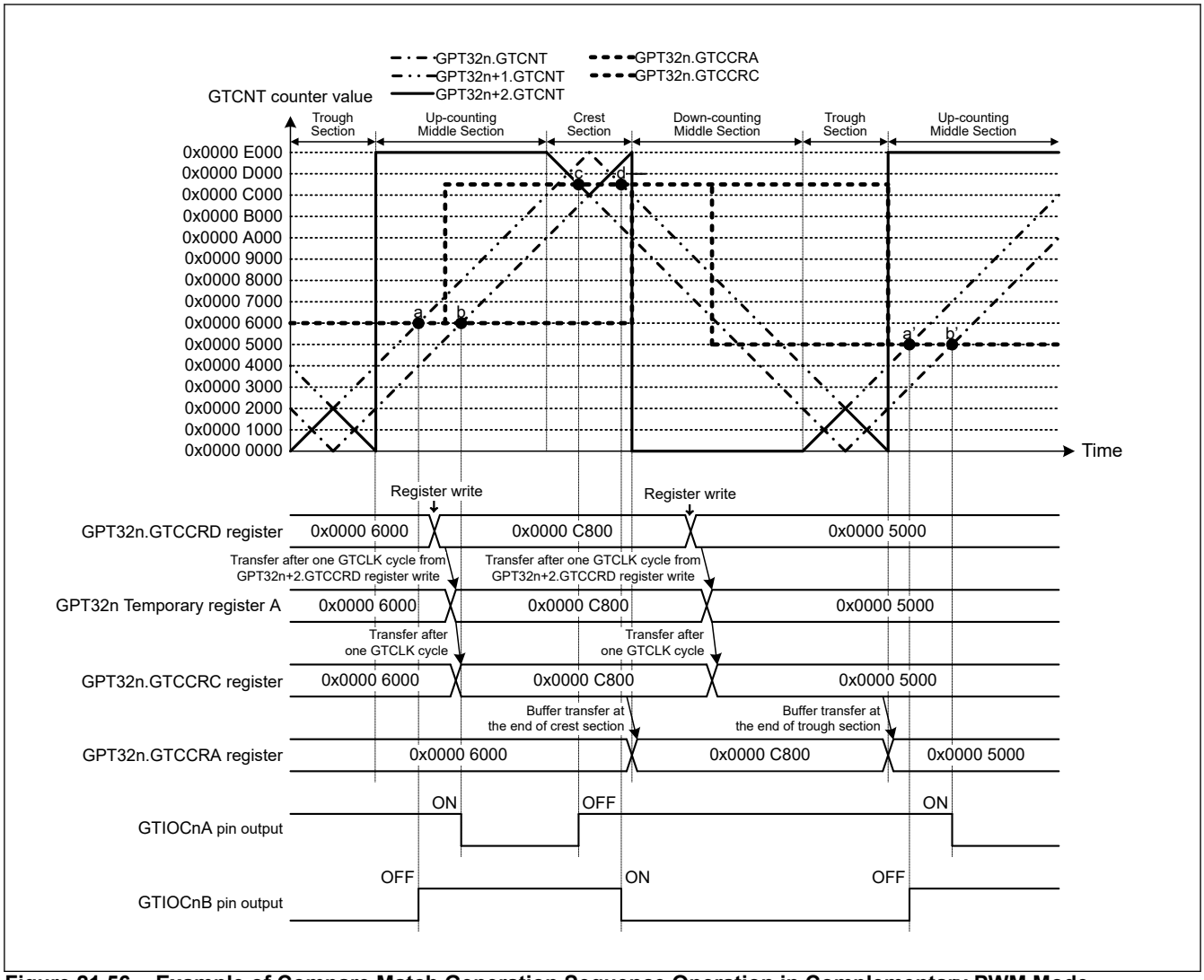


Figure 21.56 Example of Compare Match Generation Sequence Operation in Complementary PWM Mode (Complementary PWM mode 3 single buffer operation, GTIOCnA pin = Low / GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High / GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 2000, compare match generation order: a → b → c → d) (n = 4, 7)

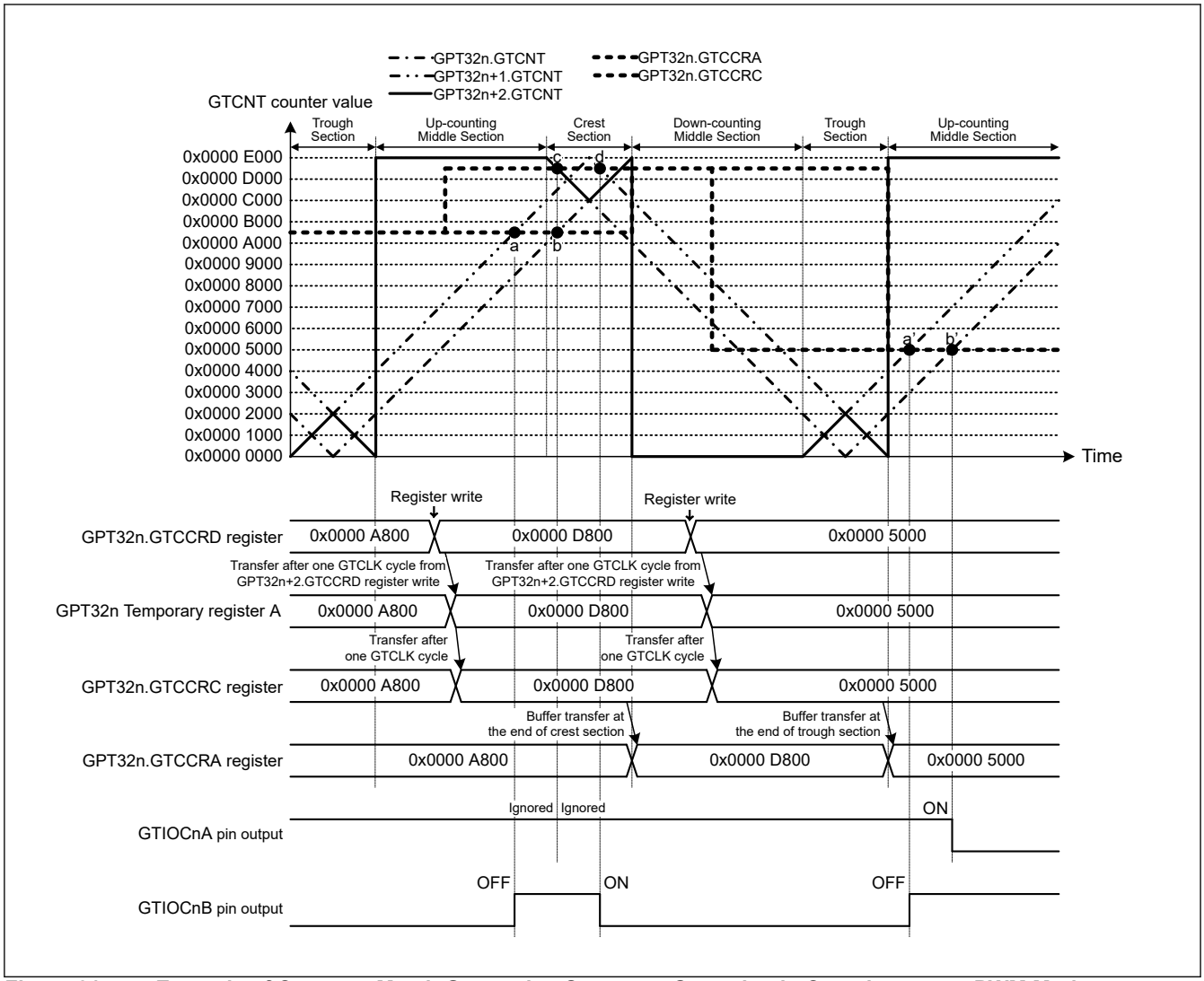


Figure 21.57 Example of Compare Match Generation Sequence Operation in Complementary PWM Mode (Complementary PWM mode 3 single buffer operation, GTIOCnA pin = Low / GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High / GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 2000, compare match generation order: a → (b, c) → d) (n = 4, 7)

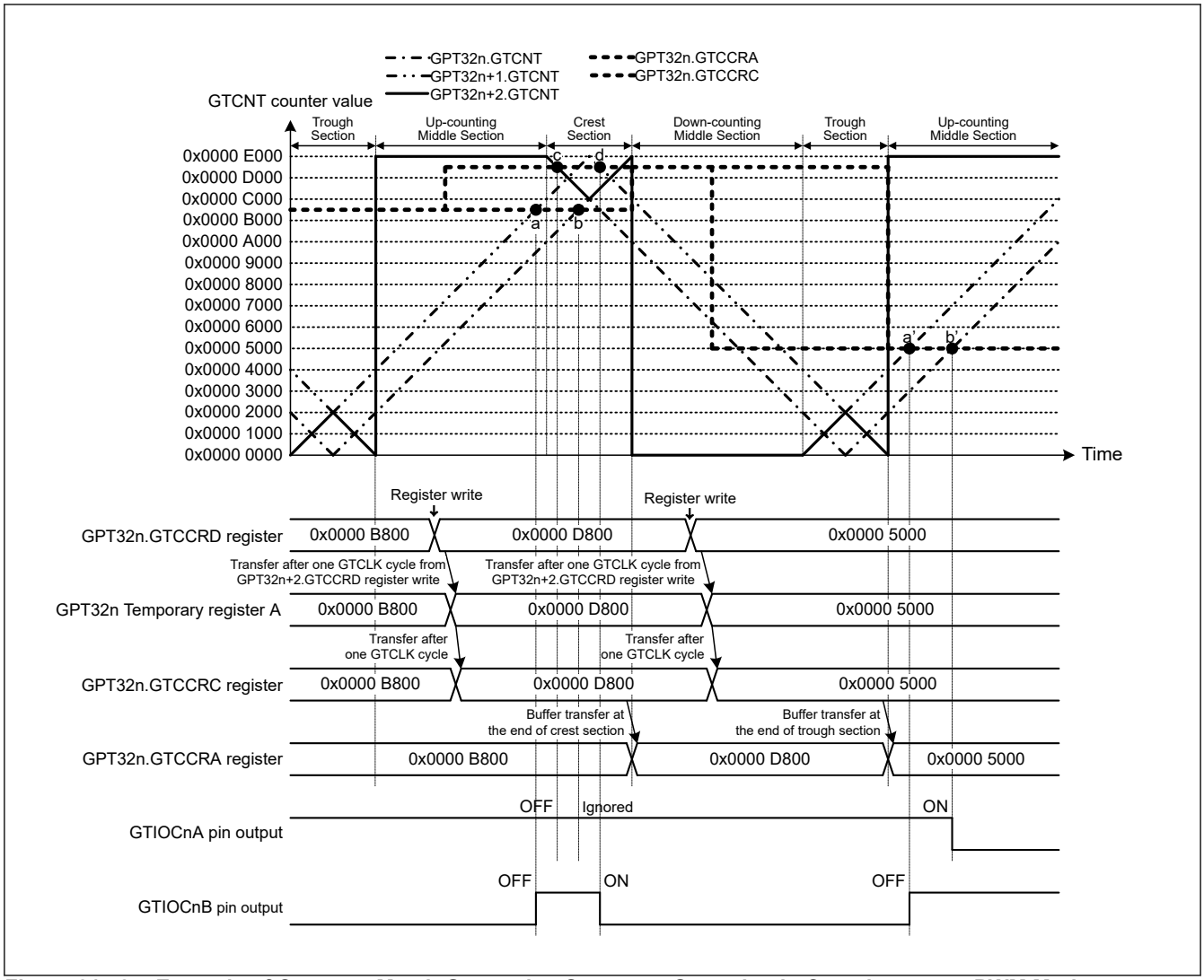


Figure 21.58 Example of Compare Match Generation Sequence Operation in Complementary PWM Mode (Complementary PWM mode 3 single buffer operation, GTIOCnA pin = Low / GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High / GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 2000, compare match generation order: a → c → b → d) (n = 4, 7)

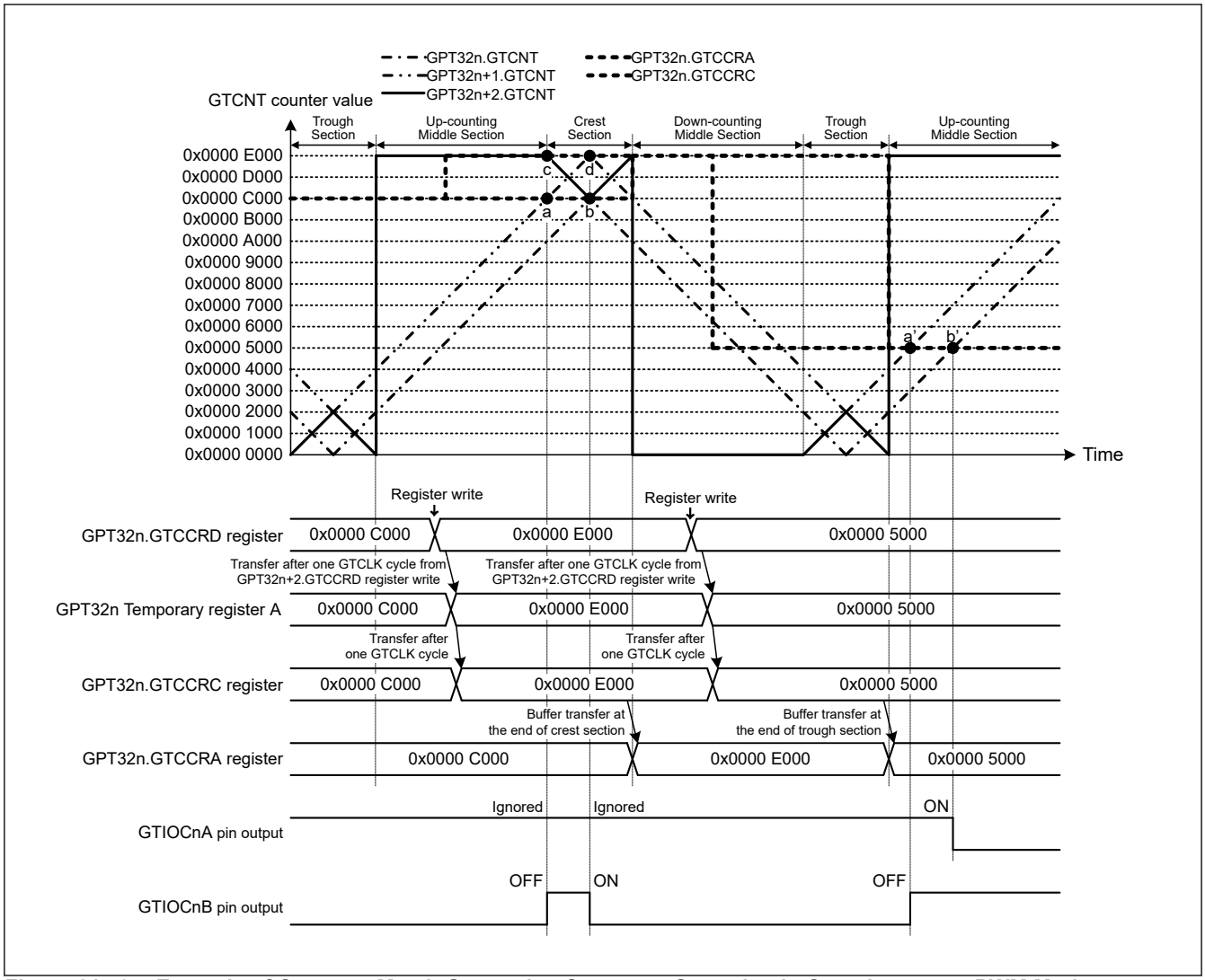


Figure 21.59 Example of Compare Match Generation Sequence Operation in Complementary PWM Mode (Complementary PWM mode 3 single buffer operation, GTIOCnA pin = Low / GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High / GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 2000, compare match generation order: (a, c) → (b, d)) (n = 4, 7)

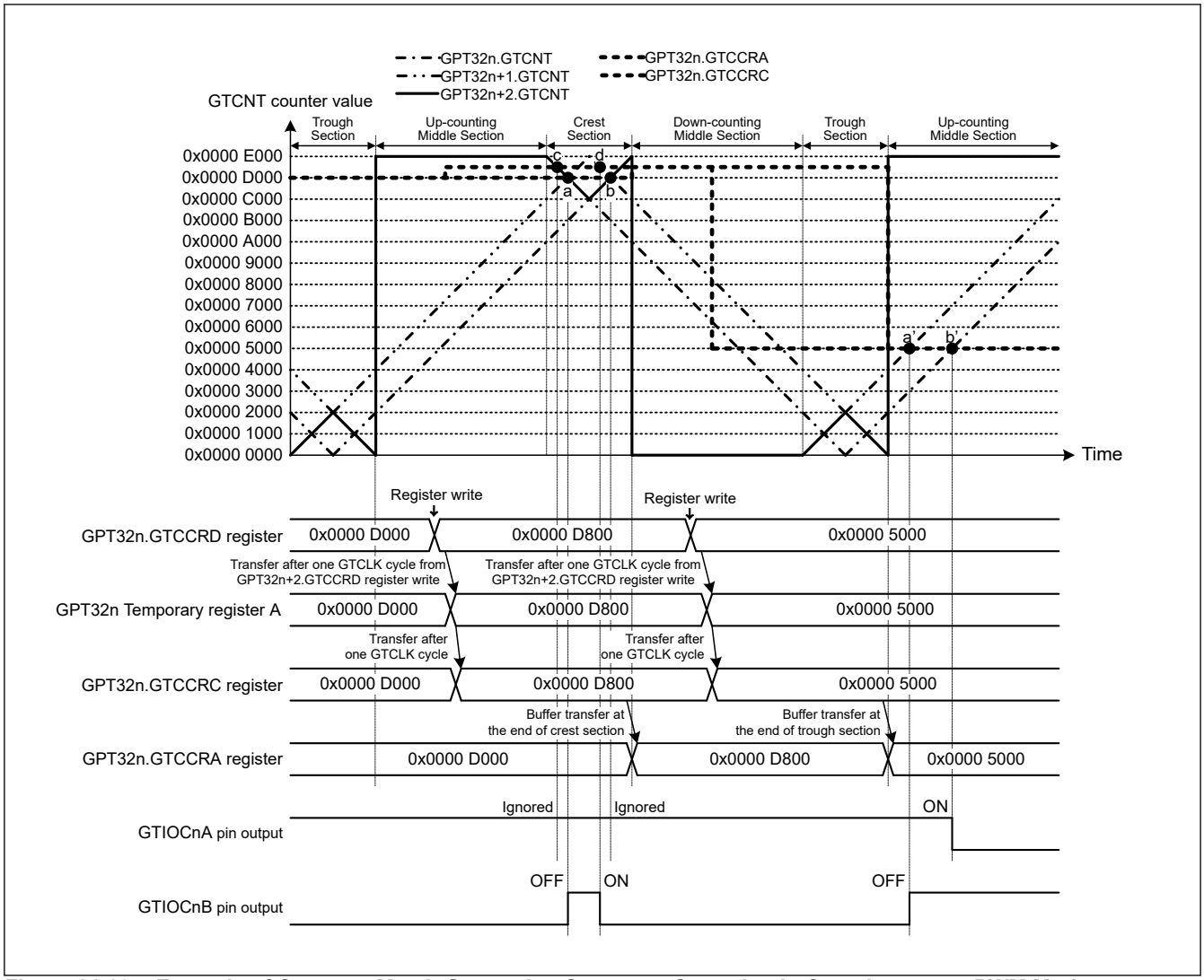


Figure 21.60 Example of Compare Match Generation Sequence Operation in Complementary PWM Mode (Complementary PWM mode 3 single buffer operation, GTIOCnA pin = Low / GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High / GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 2000, compare match generation order: c → a → d → b) (n = 4, 7)

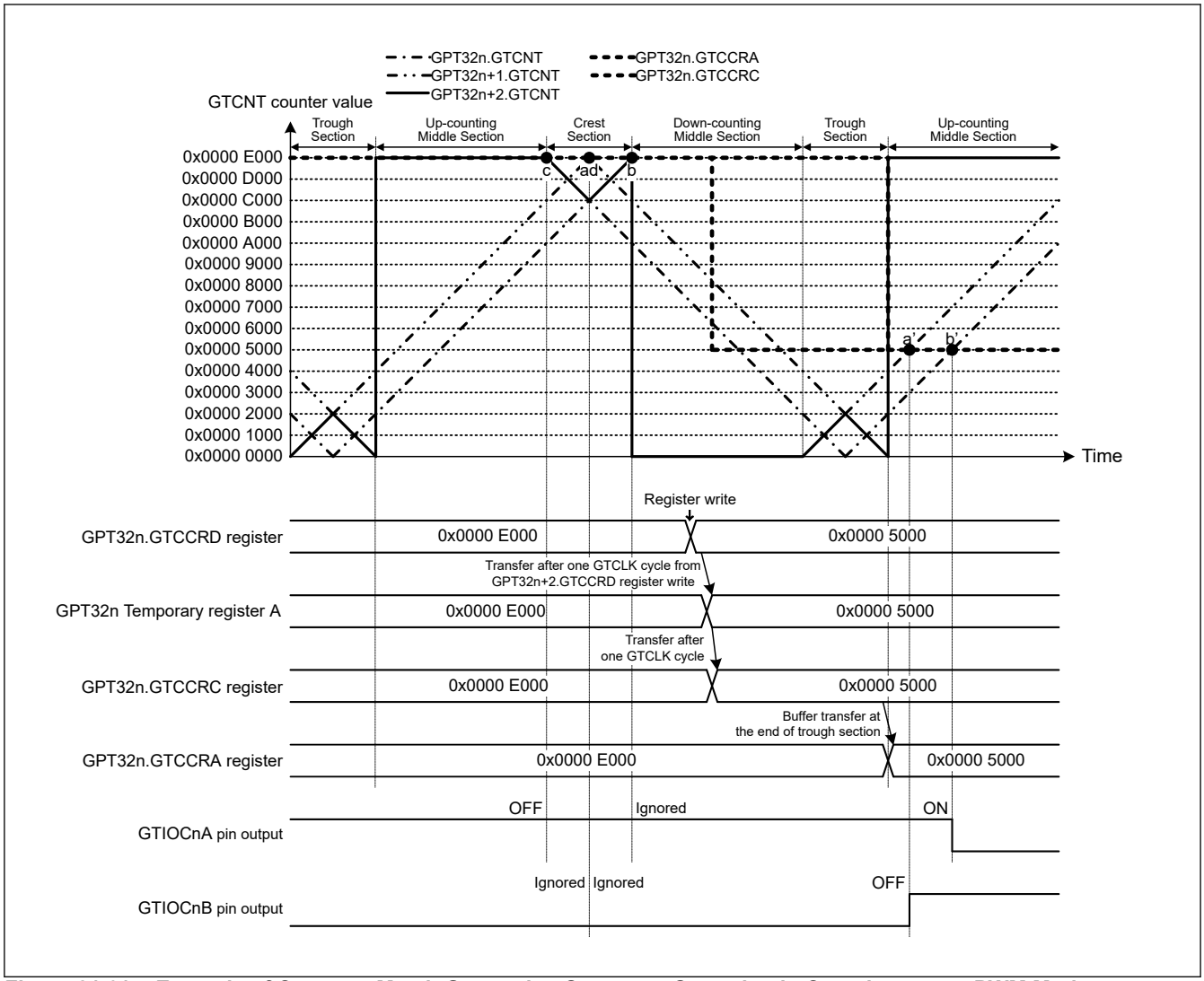


Figure 21.61 Example of Compare Match Generation Sequence Operation in Complementary PWM Mode (Complementary PWM mode 3 single buffer operation, GTIOCnA pin = Low / GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High / GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 2000, compare match generation order: c → (a, d) → b) (n = 4, 7)

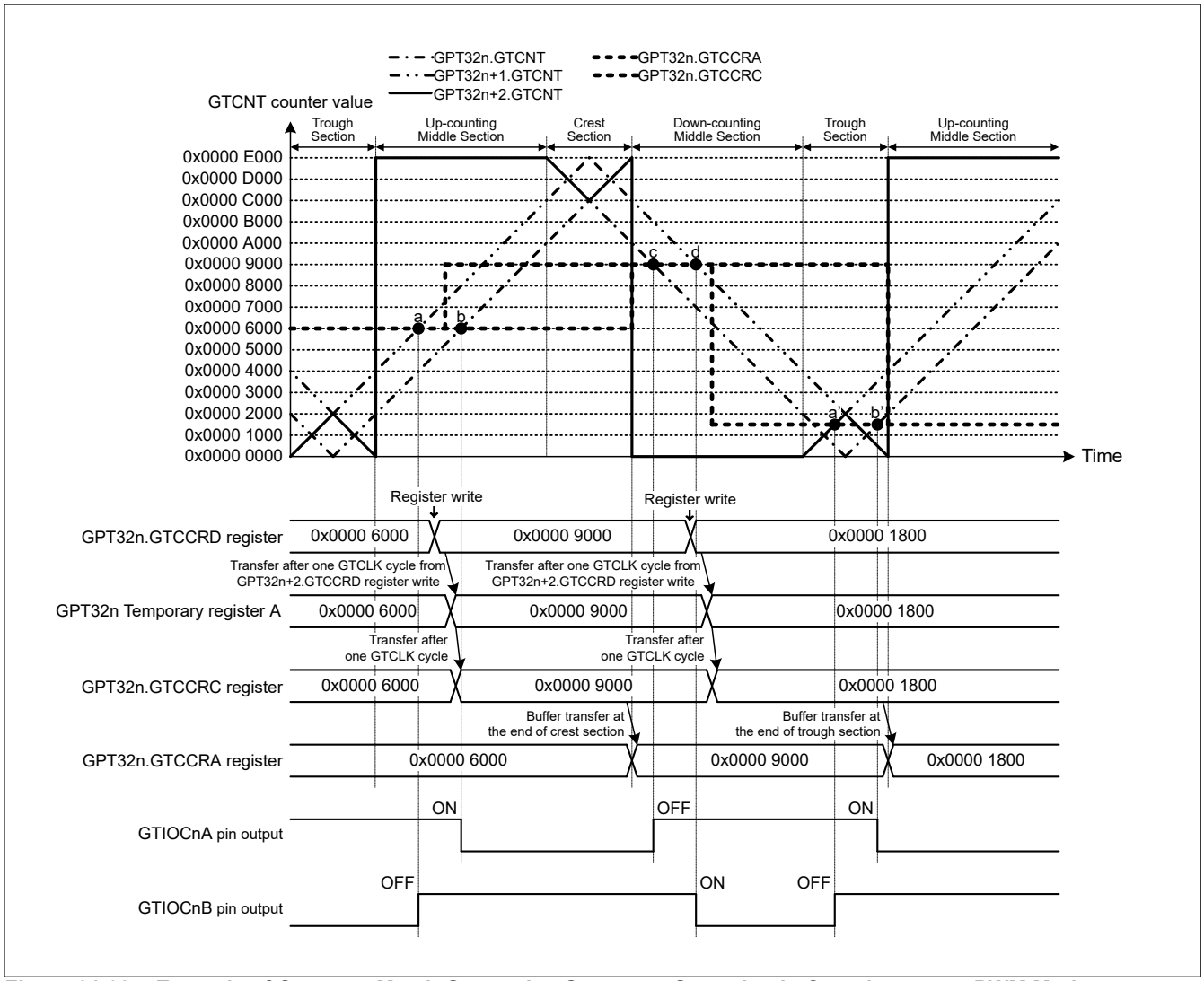


Figure 21.62 Example of Compare Match Generation Sequence Operation in Complementary PWM Mode (Complementary PWM mode 3 single buffer operation, GTIOCnA pin = Low / GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High / GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 2000, compare match generation order: c → d → a' → b') (n = 4, 7)

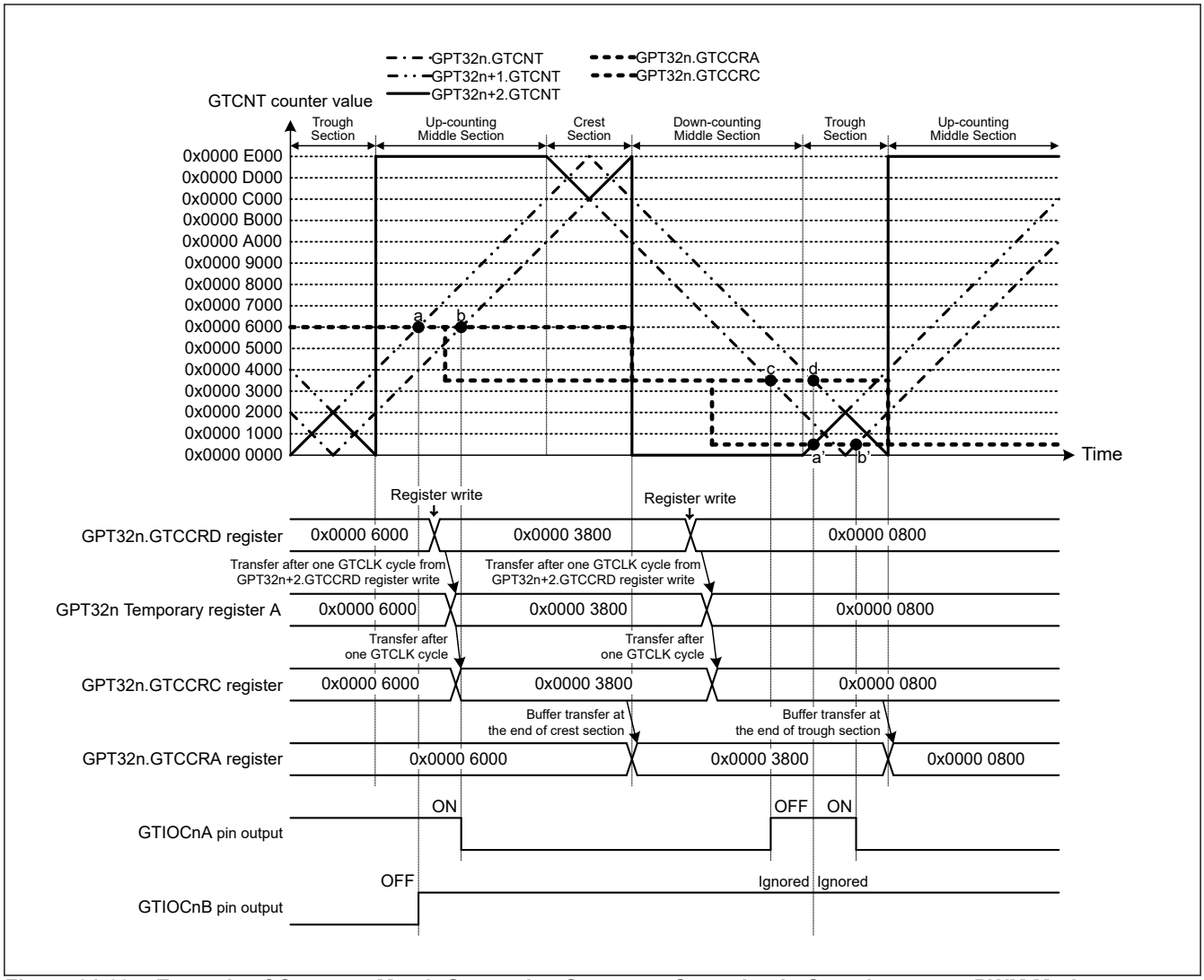


Figure 21.63 Example of Compare Match Generation Sequence Operation in Complementary PWM Mode (Complementary PWM mode 3 single buffer operation, GTIOCnA pin = Low / GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High / GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 2000, compare match generation order: c → (d, a') → b') (n = 4, 7)

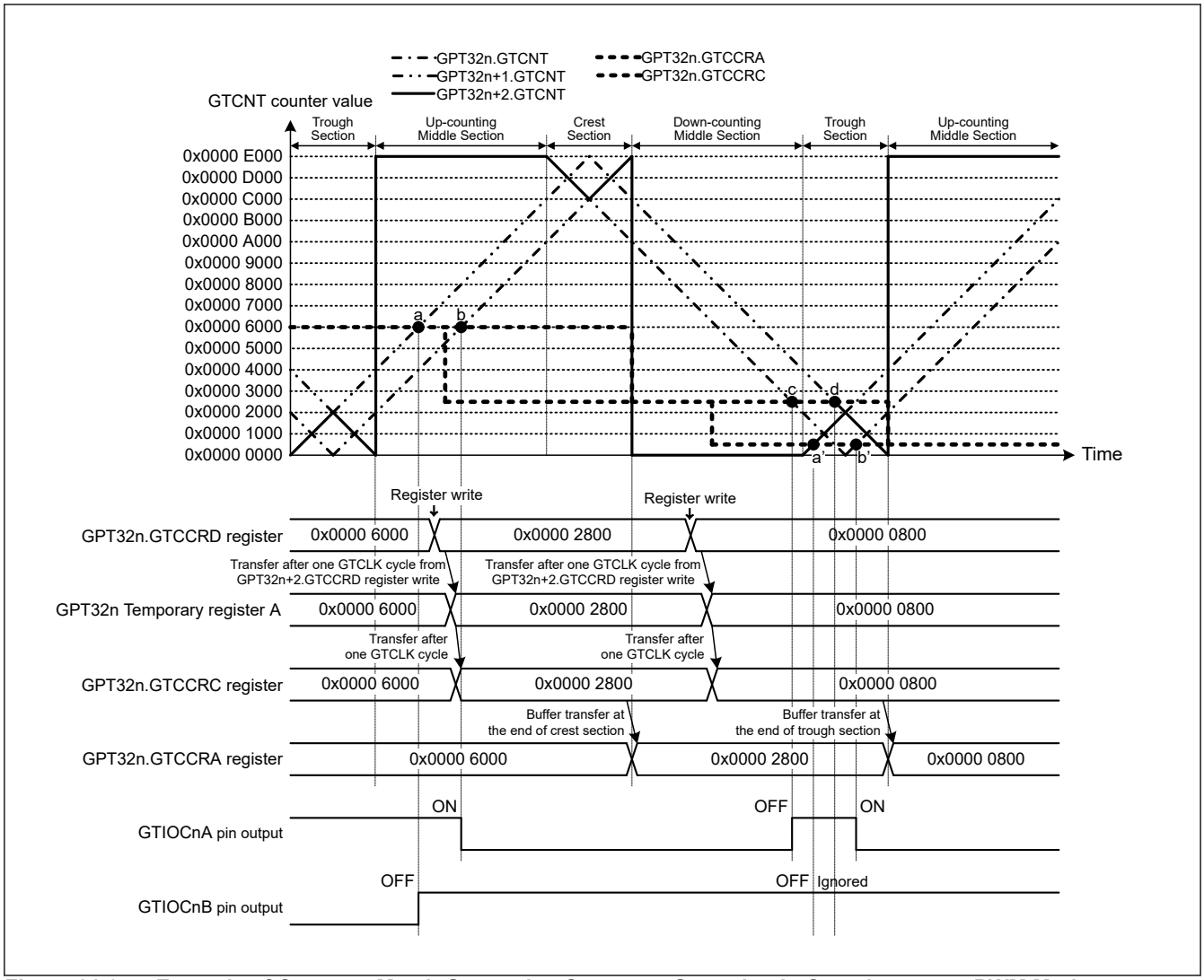


Figure 21.64 Example of Compare Match Generation Sequence Operation in Complementary PWM Mode (Complementary PWM mode 3 single buffer operation, GTIOCnA pin = Low / GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High / GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 2000, compare match generation order: c → a' → d → b') (n = 4, 7)

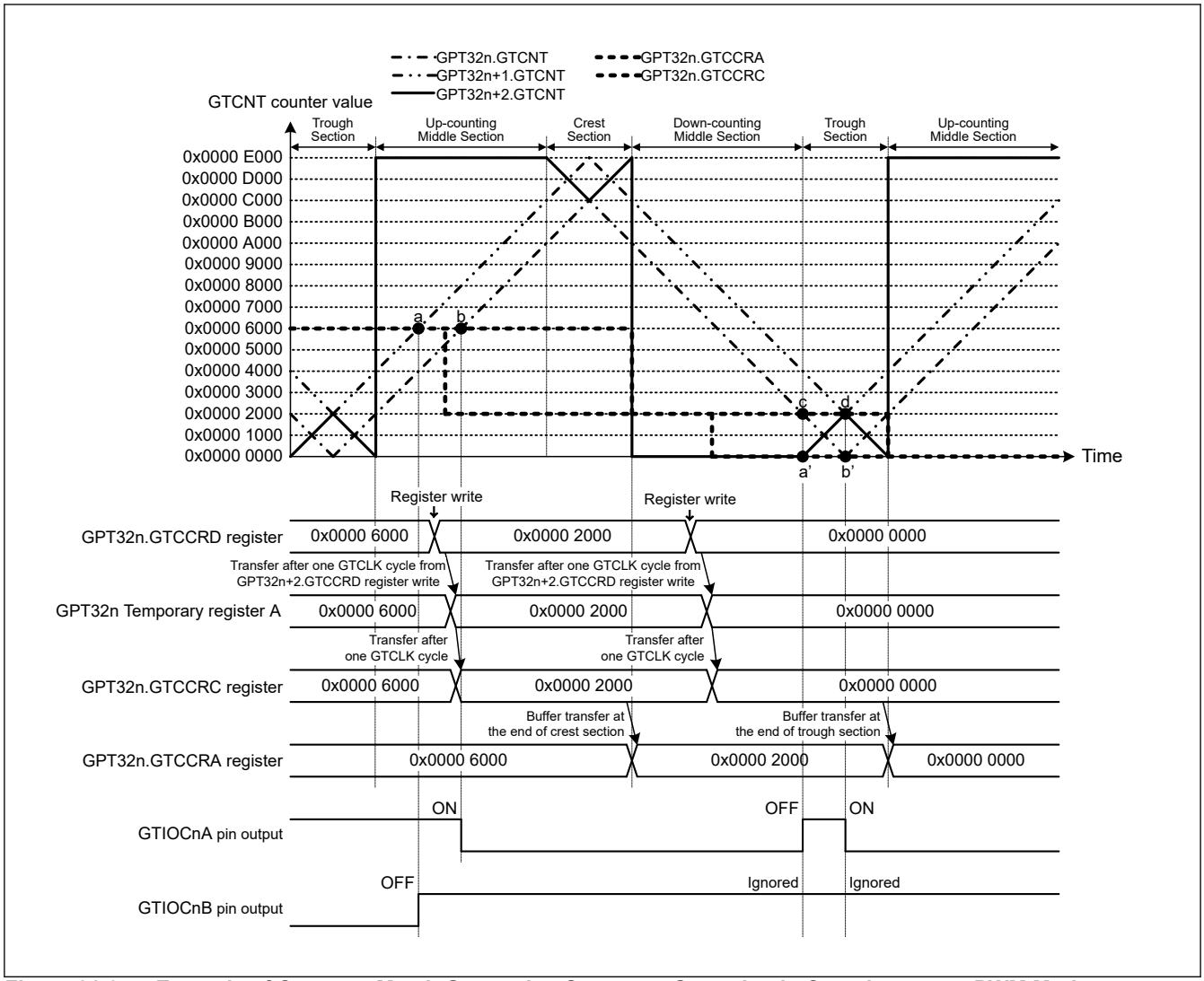


Figure 21.65 Example of Compare Match Generation Sequence Operation in Complementary PWM Mode (Complementary PWM mode 3 single buffer operation, GTIOCnA pin = Low / GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High / GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 2000, compare match generation order: (c, a') → (d, b')) (n = 4, 7)

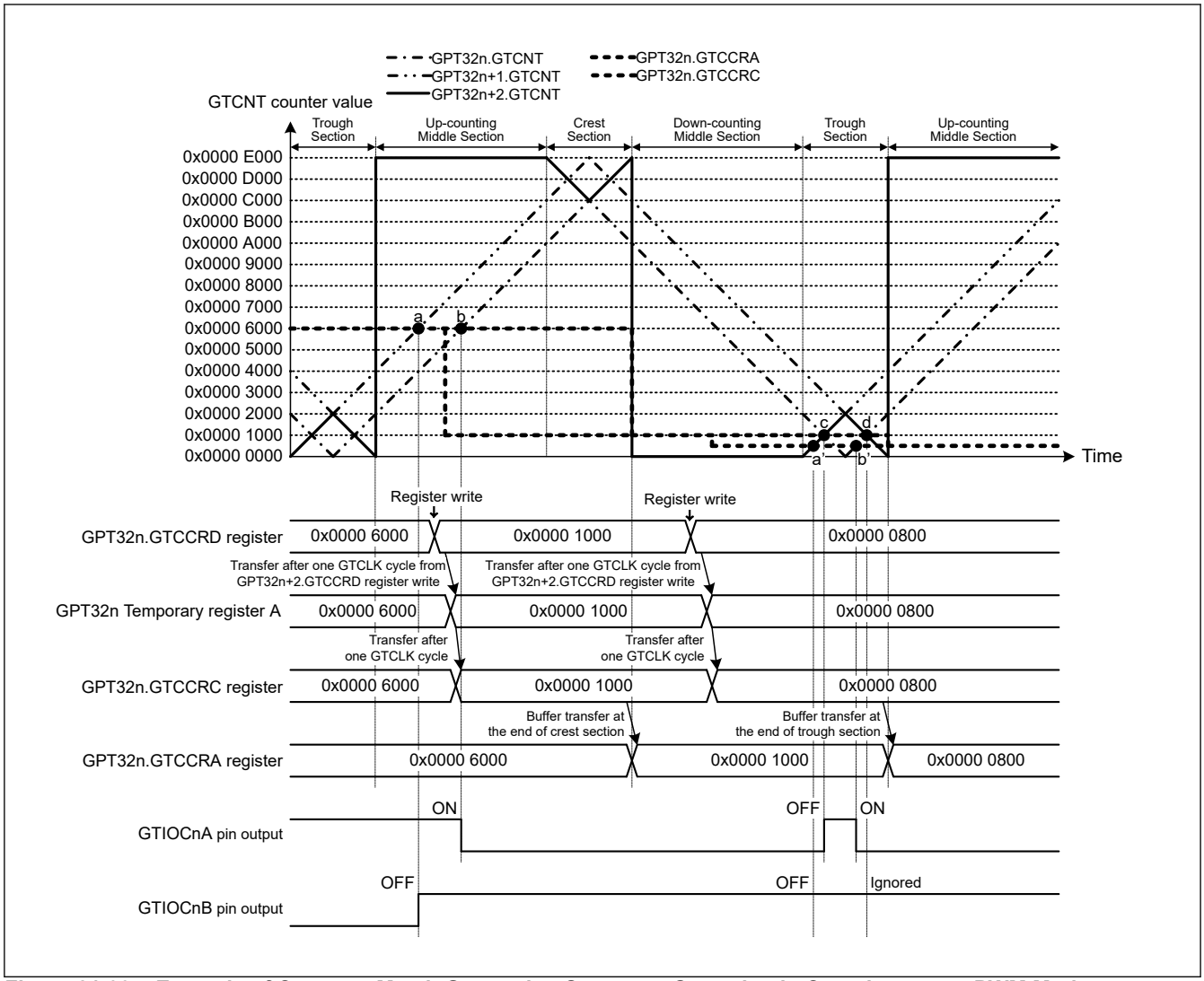


Figure 21.66 Example of Compare Match Generation Sequence Operation in Complementary PWM Mode (Complementary PWM mode 3 single buffer operation, GTIOCnA pin = Low / GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High / GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 2000, compare match generation order: a' → c → b' → d) (n = 4, 7)

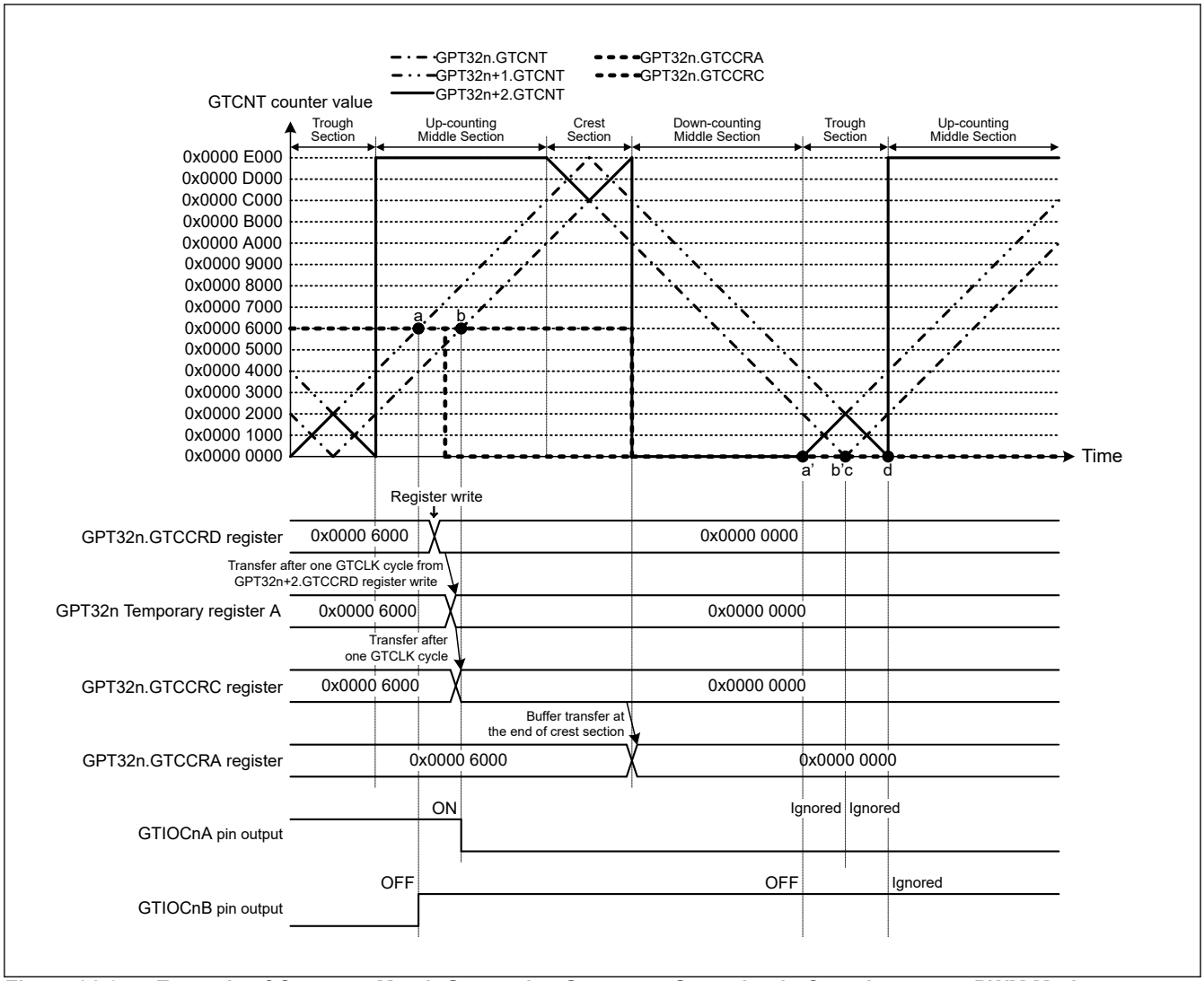


Figure 21.67 Example of Compare Match Generation Sequence Operation in Complementary PWM Mode (Complementary PWM mode 3 single buffer operation, GTIOCnA pin = Low / GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High / GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 2000, compare match generation order: a' → (b', c) → d) (n = 4, 7)

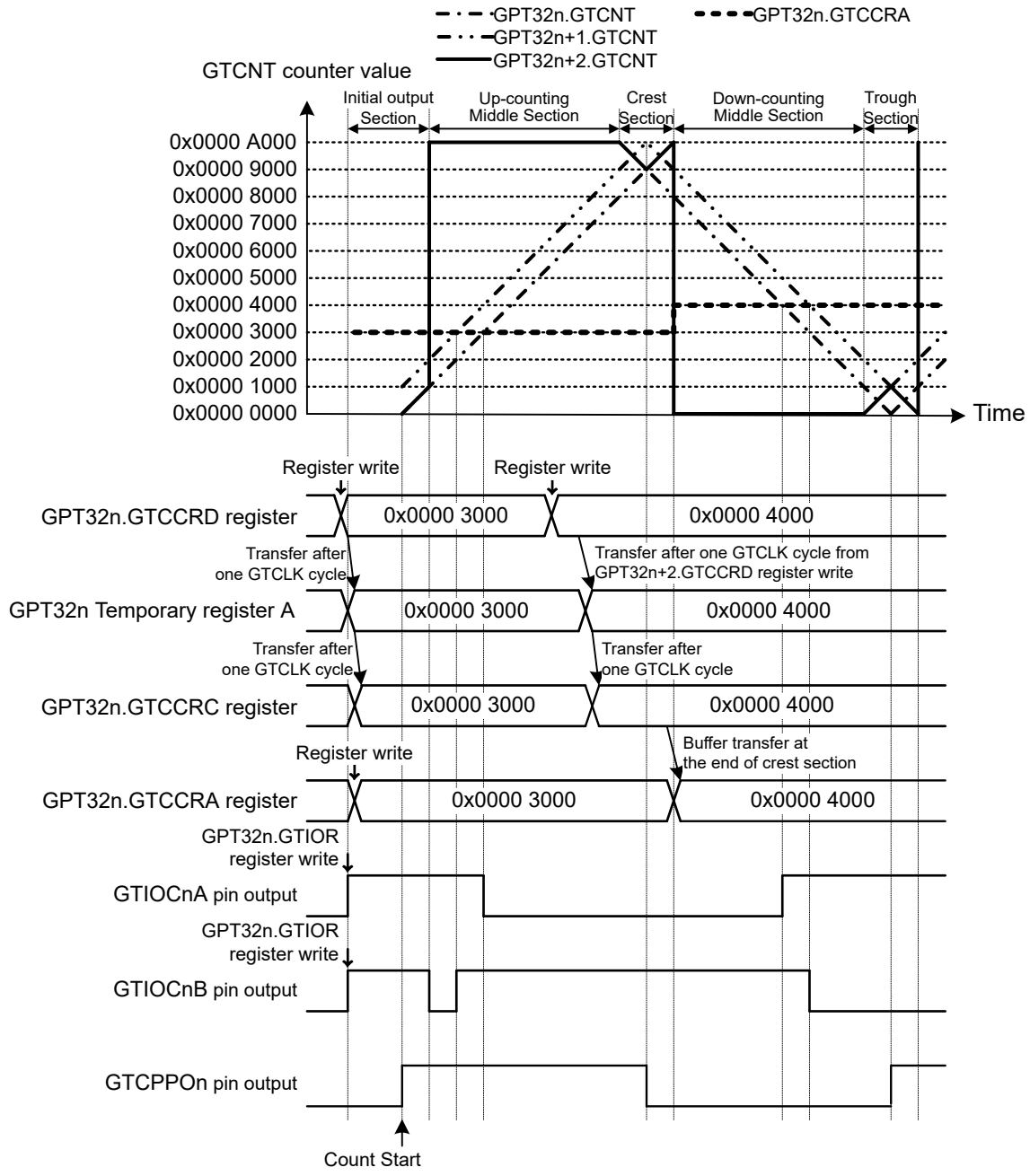


Figure 21.68 Example of Complementary PWM Mode Initial Output Operation (Complementary PWM mode 1 operation, GTIOCnA pin = Low / GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High / GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 1000, in the case that the initial GTCCRA register value is larger than the dead time value) (n = 4, 7)

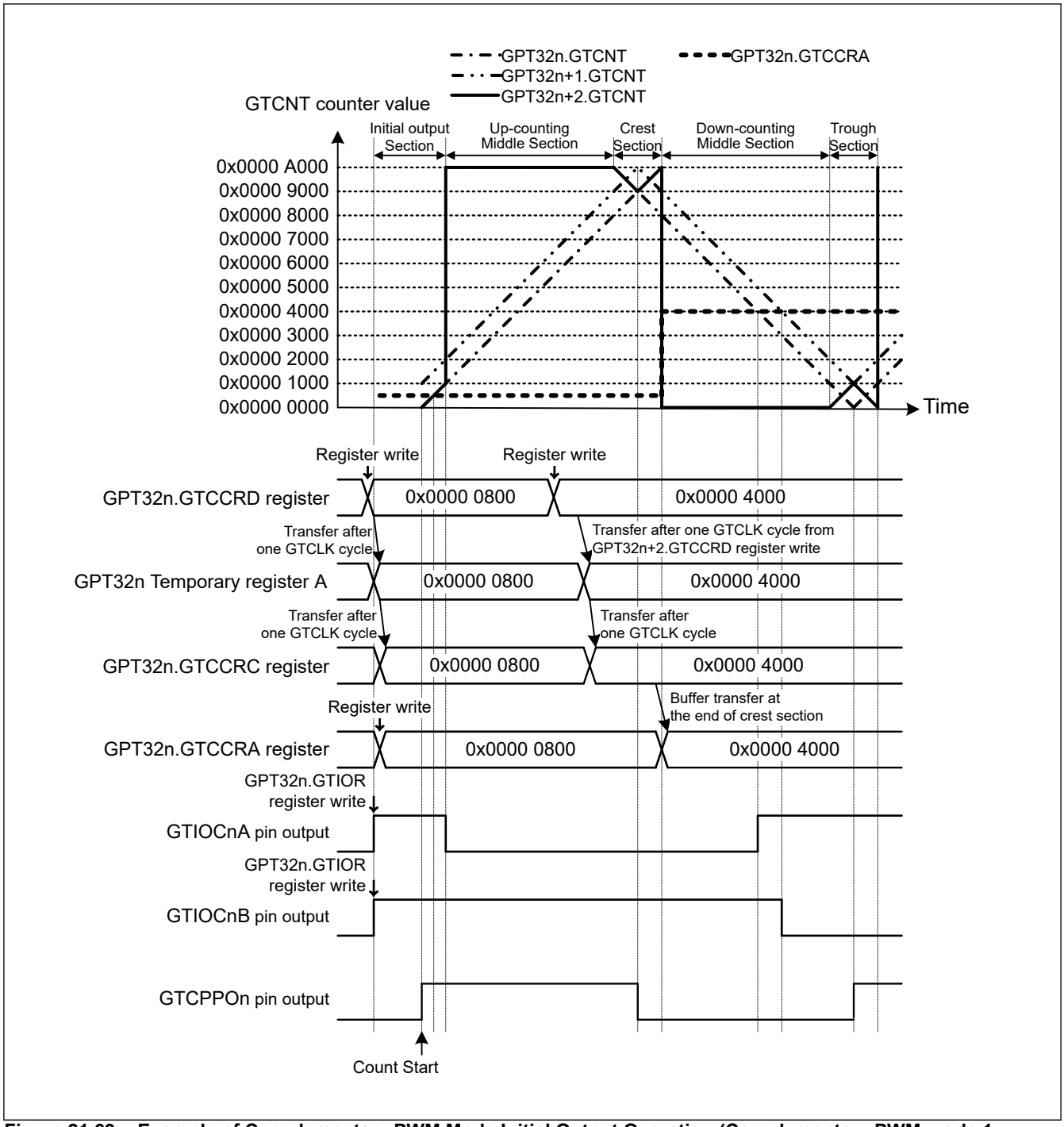


Figure 21.69 Example of Complementary PWM Mode Initial Output Operation (Complementary PWM mode 1 operation, GTIOCnA pin = Low / GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High / GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 1000, in the case that the initial GTCCRA register value is equal to or less than the dead time value) (n = 4, 7)

Table 21.35 Example for Setting Complementary PWM Mode 1,2,3 (1 of 2)

No.	Step Name	Description
1	Set operating mode	Set the operating mode with GTCR.MD[3:0] bits of GTP32n channel.
2	Select count clock	Select the count clock with GTCR.TPCS[3:0] bits of GPT32n channel.
3	Set cycle	Set the cycle in GTPR of GPT32n channel.
4	Set GTIOCnm /GTIOCn+1m / GTIOCn+2m pin function	Set the function of the GTIOCnm, GTIOCn+1m, and GTIOCn+2m pins with GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register of the GPT32n, GPT32n+1, and GPT32n+2 channels.

Table 21.35 Example for Setting Complementary PWM Mode 1,2,3 (2 of 2)

No.	Step Name	Description
5	Enable GTCPPOn pin output	Set to enable or disable PWM synchronous output from the GTCPPOn pin with the PSYE bit in the GTIOR register of the GPT32n channel.
6	Enable GTIOCnm /GTIOCn +1m /GTIOCn+2m pin output	Set to enable the output from the GTIOCnm, GTIOCn+1m, and GTIOCn+2m pins with the OAE and OBE bits in the GTIOR register of the GPT32n, GPT32n+1, and GPT32n+2 channels.
7	Set buffer operation	In complementary PWM mode 3, set buffer operation with the GTBER2.CP3DB bit of the GPT32n, GPT32n+1, and GPT32n+2 channels.
8	Set compare match value	Set the output pin changing point during up-counting after count start in the GTCCRA register of the GPT32n, GPT32n+1, and GPT32n+2 channels.
9	Set buffer value	For single buffer operation, set data (to be transferred for the first buffer transfer to the GTCCRA register after count start) in the GTCCRD register. For buffer operation in complementary PWM mode 2, set the same value as the GTCCRA register in the GTCCRD register because the first buffer transfer does not take place. For double buffer operation in complementary PWM mode 3, set data to be transferred at the first end of crest in the GTCCRD register and set data to be transferred at the first end of trough in the GTCCRF register.
10	Set dead time value	Set the dead time value in GTDVU of GTP32n channel.
11	Start count operation	Set GTCR.CST of GPT32n channel to 1 to start count operation.
12	Set buffer value for each cycle	For single buffer operation, set data (to be transferred for the next buffer transfer to the GTCCRA register) in the GTCCRD register. For double buffer operation in complementary PWM mode 3, set data to be transferred for the next buffer transfer in the GTCCRF register. Make settings for the GPT32n+2.GTCCRD register finally. (Data is transferred to the temporary register.)

Note: n = 4, 7
m = A, B

21.3.3.8 Complementary PWM mode 4

In complementary PWM mode 4, the value written to the GTCCRD and GTCCRF registers is immediately applied to compare match operation by transferring data also to the GTCCRA register during buffer transfer to a temporary register before crest or trough transfer timing.

Figure 21.70 shows the block diagram in complementary PWM mode 4.

In the configuration of complementary PWM mode 4, a buffer transfer path from the GTCCRD register to the GTCCRC and GTCCRA registers and a buffer transfer path from the GTCCRF register to the GTCCRE and GTCCRA registers are added to other complementary PWM modes shown in Table 21.35.

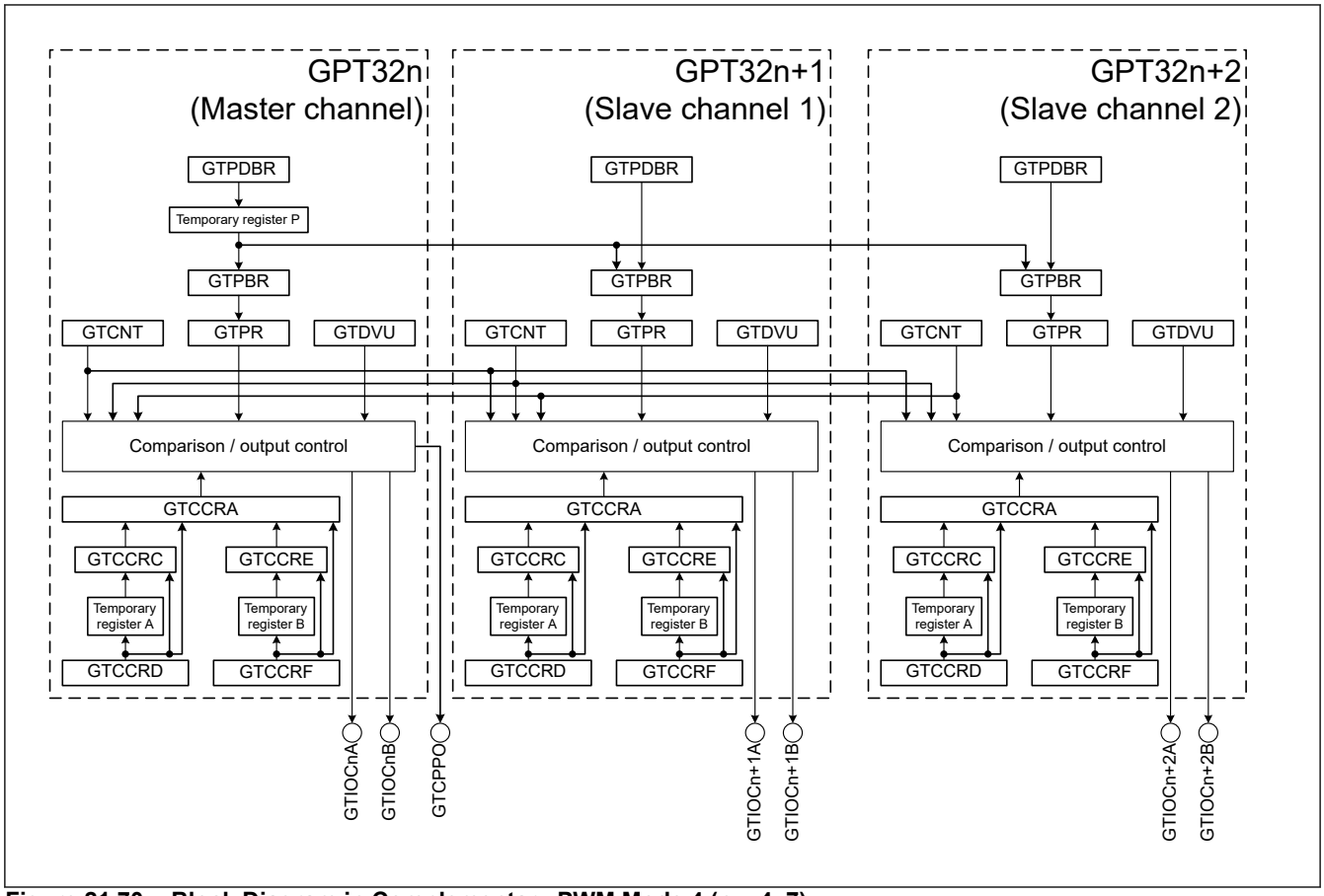


Figure 21.70 Block Diagram in Complementary PWM Mode 4 (n = 4, 7)

Count operation is performed in the same way as complementary PWM modes 1 to 3. See Table 21.30 and Table 21.31.

With respect to buffer operation and PWM waveform changes in complementary PWM mode 4, buffer transfer (shown in Figure 21.70) from the GTCCRD and GTCCRF registers is added based on the complementary PWM mode 3 operation. Buffer transfer and PWM waveform are controlled according to operation section, state of comparison with the GTCCRA register, and write value.

The double buffer function by writing to the GTCCRF register can be enabled or disabled with the GTBER2.CP3DB bit. In double buffer operation, the value written to the GTCCRD register is used as a compare match value for positive-phase OFF (negative-phase ON) during down-counting, and the value written to the GTCCRF register is used as a compare match value for negative-phase OFF (positive-phase ON) during up-counting. Transfer register, transfer value, and PWM output changes are controlled by operation section (where the value is written), state of comparison with the GTCCRA register, and write value.

Double buffer operation is guaranteed only in the up-counting middle section and down-counting middle section. Setting a value not larger than the dead time value and not less than the count cycle is prohibited.

In single buffer operation, a compare match value is written only to the GTCCRD register, and transfer register, transfer value, and PWM output changes are controlled by operation section (where the value is written), state of comparison with the GTCCRA register, and write value.

Transfer from the GTCCRD register to the temporary register A and transfer from the GTCCRF register to the temporary register B are performed in the same way as other complementary PWM modes. Transfers are concurrently performed in three channels by writing a value to the GPT32n+2.GTCCRD register. Transfer from the GTCCRD register to the GTCCRC register, GTCCRA register, temporary register B, and GTCCRE register and transfer from the GTCCRF register to the GTCCRE register and the GTCCRA register are performed at the same time as the above-mentioned transfer to the temporary register.

Table 21.36 and Table 21.37 show immediate buffer transfer (for transfer to the temporary register by writing a value to the GPT32n+2.GTCCRD register) to the GTCCRC and GTCCRA registers by writing a value to the GTCCRD register during single buffer operation in complementary PWM mode 4 for each compare match state in each operation section. Transfers (from the GTCCRD register to the temporary register A, from the temporary register A to the GTCCRC register, and from

the GTCCRC register to the GTCCRA register) other than those shown in Table 21.36 and Table 21.37 are the same as single buffer transfer in complementary PWM mode 3 shown in Table 21.32.

Table 21.38 and Table 21.39 show immediate buffer transfer (for transfer to the temporary register by writing a value to the GPT32n+2.GTCCRD register) to each register by writing a value to the GTCCRD and GTCCRF registers during double buffer operation in complementary PWM mode 4 for each compare match state in each operation section. Transfers (from the GTCCRD register to the temporary register A, from the GTCCRF register to the temporary register B, from the temporary register A to the GTCCRC register, from the temporary register B to the GTCCRE register, and from the GTCCRC and GTCCRE registers to the GTCCRA register) other than those shown in Table 21.38 and Table 21.39 are the same as double buffer transfer in complementary PWM mode 3 shown in Table 21.33.

Table 21.36 Immediate Single Buffer Transfer from GTCCRD Register in Complementary PWM Mode 4 (1 of 2)

Operation Section	Compare Match State	Immediate Transfer Destination Register	
		GTCCRC	GTCCRA
Up- counting middle section	Before up-counting compare match	GTCCRD	(i) In the case of GTCCRD > GPT32n+1.GTCNT GTCCRD (ii) In the case of GTCCRD < = GPT32+1.GTCNT GPT32n+1.GTCNT Negative-phase OFF
	Up-counting dead time period	GTCCRD	No transfer
	After up-counting compare match	GTCCRD	No transfer
Up-counting crest section	Before up-counting compare match	Before down-counting compare match GTCCRD After down-counting dead time start No transfer	Before down-counting compare match (i) In the case of GTCCRD > GPT32n+1.GTCNT GTCCRD (ii) In the case of GTCCRD < = GPT32n+1.GTCNT GPT32n+1.GTCNT Negative-phase OFF After down-counting dead time start No transfer
	Up-counting dead time period	Before down-counting compare match GTCCRD After down-counting dead time start No transfer	No transfer
	After up-counting compare match	Before down-counting compare match (i) In the case of GTCCRD < GPT32n+2.GTCNT GTCCRD (ii) In the case of GTCCRD = > GPT32n+2.GTCNT GPT32n+2.GTCNT Positive-phase OFF After down-counting dead time start No transfer	No transfer

Table 21.36 Immediate Single Buffer Transfer from GTCCRD Register in Complementary PWM Mode 4 (2 of 2)

Operation Section	Compare Match State	Immediate Transfer Destination Register	
		GTCCRC	GTCCRA
Down-counting crest section	Before down-counting compare match	Up-counting dead time period (i) In the case of GTCCRD < GPT32n+1.GTCNT GTCCRD (ii) In the case of GTCCRD = > GPT32n+1.GTCNT GPT32n+1.GTCNT Negative-phase ON After up-counting compare match (i) In the case of GTCCRD < GPT32n.GTCNT GTCCRD (ii) In the case of GTCCRD = > GPT32n.GTCNT GPT32n.GTCNT Positive-phase OFF	No transfer
	Down-counting dead time period	No transfer	No transfer
	After down-counting compare match	No transfer	No transfer

Table 21.37 Immediate Single Buffer Transfer from GTCCRD Register in Complementary PWM Mode 4 (1 of 2)

Operation Section	Compare Match State	Immediate Transfer Destination Register	
		GTCCRC	GTCCRA
Down-counting middle section	Before down-counting compare match	GTCCRD	(i) In the case of GTCCRD < GPT32n.GTCNT GTCCRD (ii) In the case of GTCCRD = > GPT32n.GTCNT GPT32n.GTCNT Positive-phase OFF
	Down-counting dead time period	GTCCRD	No transfer
	After down-counting compare match	GTCCRD	No transfer

Table 21.37 Immediate Single Buffer Transfer from GTCCRD Register in Complementary PWM Mode 4 (2 of 2)

Operation Section	Compare Match State	Immediate Transfer Destination Register	
		GTCCRC	GTCCRA
Down-counting trough section	Before down-counting compare match	Before up-counting compare match GTCCRD After up-counting dead time start No transfer	Before up-counting compare match (i) In the case of GTCCRD < GPT32n.GTCNT GTCCRD (ii) In the case of GTCCRD = > GPT32n.GTCNT GPT32n.GTCNT Positive-phase OFF After up-counting dead time start No transfer
	Down-counting dead time period	Before up-counting compare match GTCCRD After up-counting dead time start No transfer	No transfer
	After down-counting compare match	Before up-counting compare match (i) In the case of GTCCRD > GPT32n+2.GTCNT GTCCRD (ii) In the case of GTCCRD < = GPT32n+2.GTCNT GPT32n+2.GTCNT Negative-phase OFF After up-counting dead time start No transfer	No transfer
Up- counting trough section	Before up-counting compare match	Down-counting dead time period (i) In the case of GTCCRD > GPT32n.GTCNT GTCCRD (ii) In the case of GTCCRD < = GPT32n.GTCNT GPT32n.GTCNT Positive-phase ON After down-counting compare match (i) In the case of GTCCRD > GPT32n+1.GTCNT GTCCRD (ii) In the case of GTCCRD < = GPT32n+1.GTCNT GPT32n+1.GTCNT Negative-phase OFF	No transfer
	Up-counting dead time period	No transfer	No transfer
	After up-counting compare match	No transfer	No transfer

Table 21.38 Immediate Double Buffer Transfer from GTCCRD and GTCCRF Registers in Complementary PWM Mode 4

Operation Section	Compare Match State	Immediate Transfer Destination Register		
		GTCCRC	GTCCRE	GTCCRA
Up-counting middle section	Before up-counting compare match	GTCCRD	GTCCRF	(i) In the case of GTCCRF > GPT32n+1.GTCNT GTCCRF (ii) In the case of GTCCRF < = GPT32+1.GTCNT GPT32n+1.GTCNT Negative-phase OFF
	Up-counting dead time period	GTCCRD	GTCCRF	No transfer
	After up-counting compare match	GTCCRD	GTCCRF	No transfer

Table 21.39 Immediate Double Buffer Transfer from GTCCRD and GTCCRF Registers in Complementary PWM Mode 4

Operation Section	Compare Match State	Immediate Transfer Destination Register		
		GTCCRC	GTCCRE	GTCCRA
Down-counting middle section	Before down-counting compare match	GTCCRD	GTCCRF	(i) In the case of GTCCRD < GPT32n.GTCNT GTCCRD (ii) In the case of GTCCRD => GPT32n.GTCNT GPT32n.GTCNT Positive-phase OFF
	Down-counting dead time period	GTCCRD	GTCCRF	No transfer
	After down-counting compare match	GTCCRD	GTCCRF	No transfer

Figure 21.71 to Figure 21.75 show examples of single buffer operation for each operation section in complementary PWM mode 4.

Figure 21.76 to Figure 21.79 show examples of double buffer operation for each operation section in complementary PWM mode 4.

Table 21.40 shows an example for setting complementary PWM mode 4.

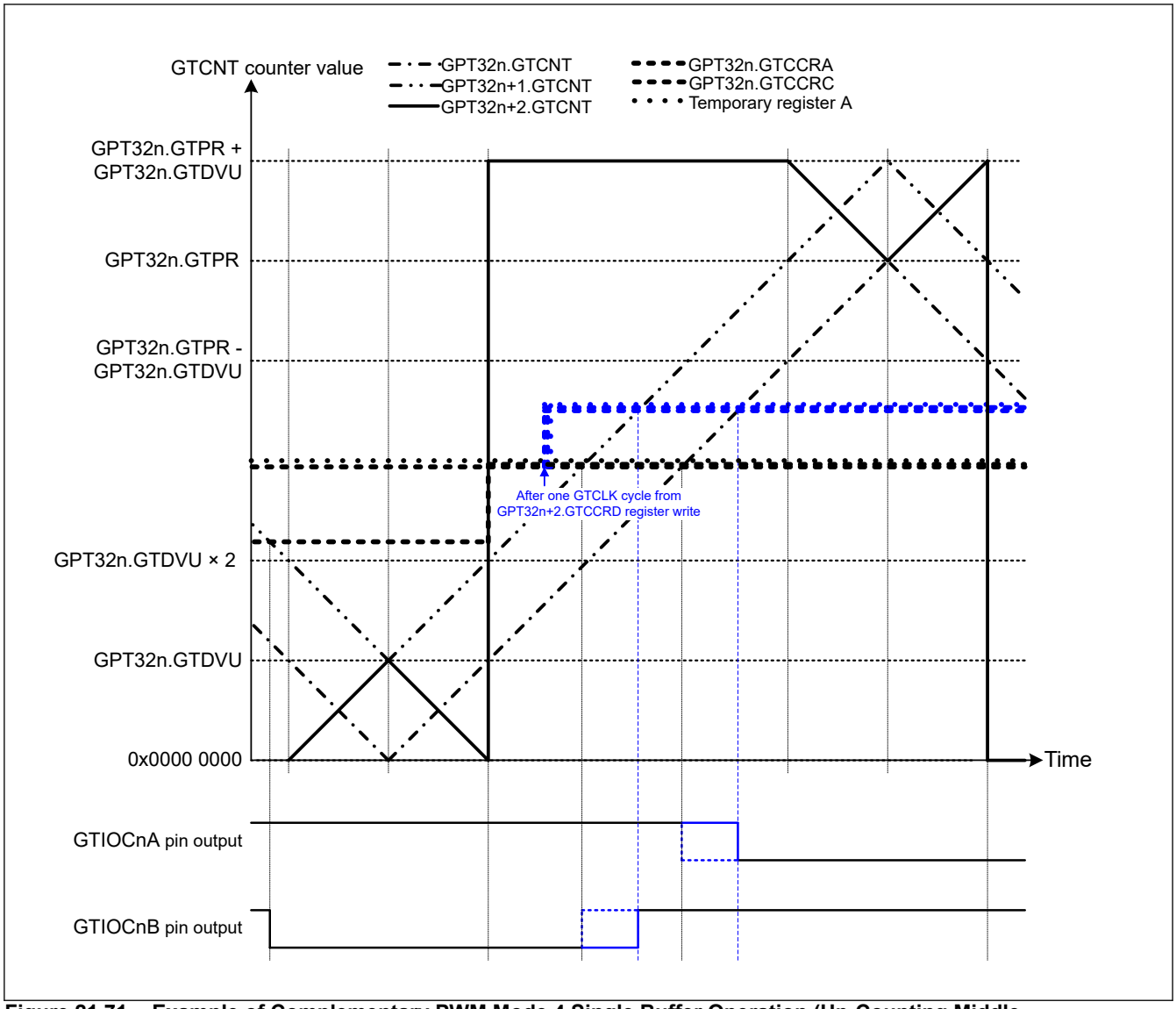


Figure 21.71 Example of Complementary PWM Mode 4 Single Buffer Operation (Up-Counting Middle Section) (Complementary PWM mode 4 single buffer operation, GTIOCnA pin = Low / GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High / GTIOCnB pin = Low at GTCCRA register compare match during down-counting, when a value larger than the GPT32n+1.GTCNT value is written to GTCCRD register before up-counting compare match) (n = 4, 7)

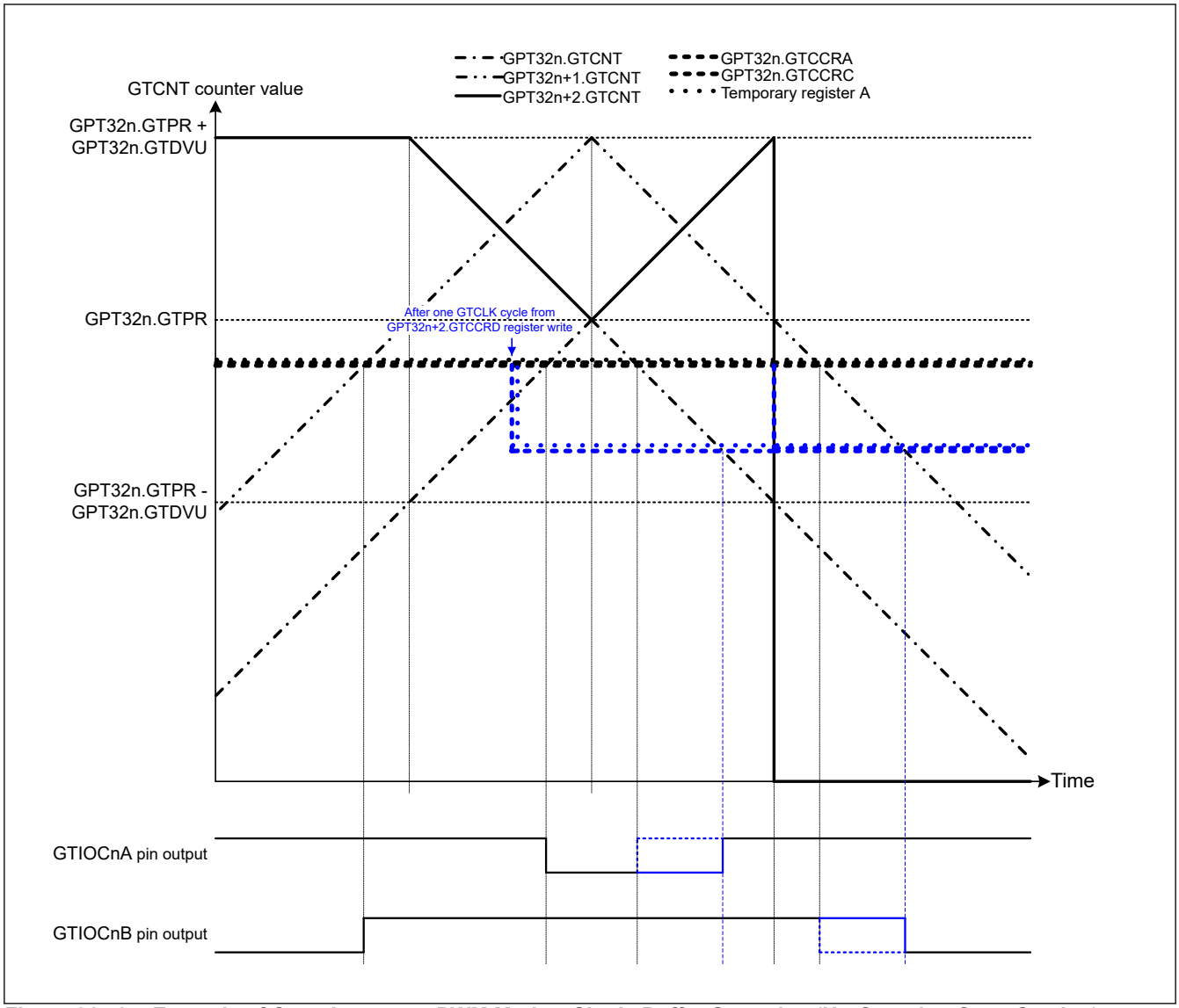


Figure 21.72 Example of Complementary PWM Mode 4 Single Buffer Operation (Up-Counting Crest Section) (Complementary PWM mode 4 single buffer operation, GTIOCnA pin = Low / GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High / GTIOCnB pin = Low at GTCCRA register compare match during down-counting, when a value is written to GTCCRD register during the up-counting dead time) (n = 4, 7)

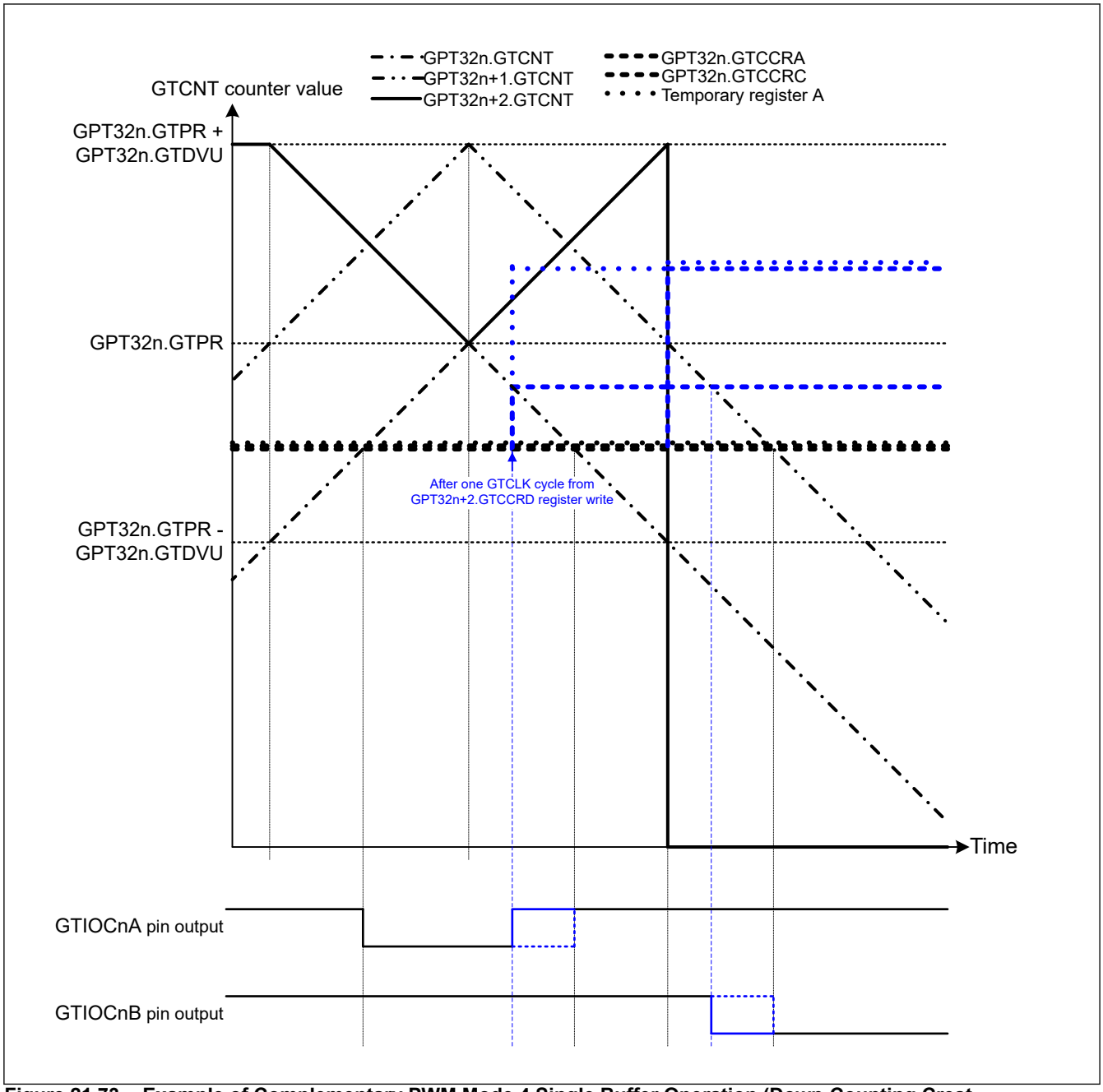


Figure 21.73 Example of Complementary PWM Mode 4 Single Buffer Operation (Down-Counting Crest Section) (Complementary PWM mode 4 single buffer operation, GTIOCnA pin = Low / GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High / GTIOCnB pin = Low at GTCCRA register compare match during down-counting, when a value not less than the GPT32n.GTCNT value is written to GTCCRD register after up-counting compare match before down-counting compare match) (n = 4, 7)

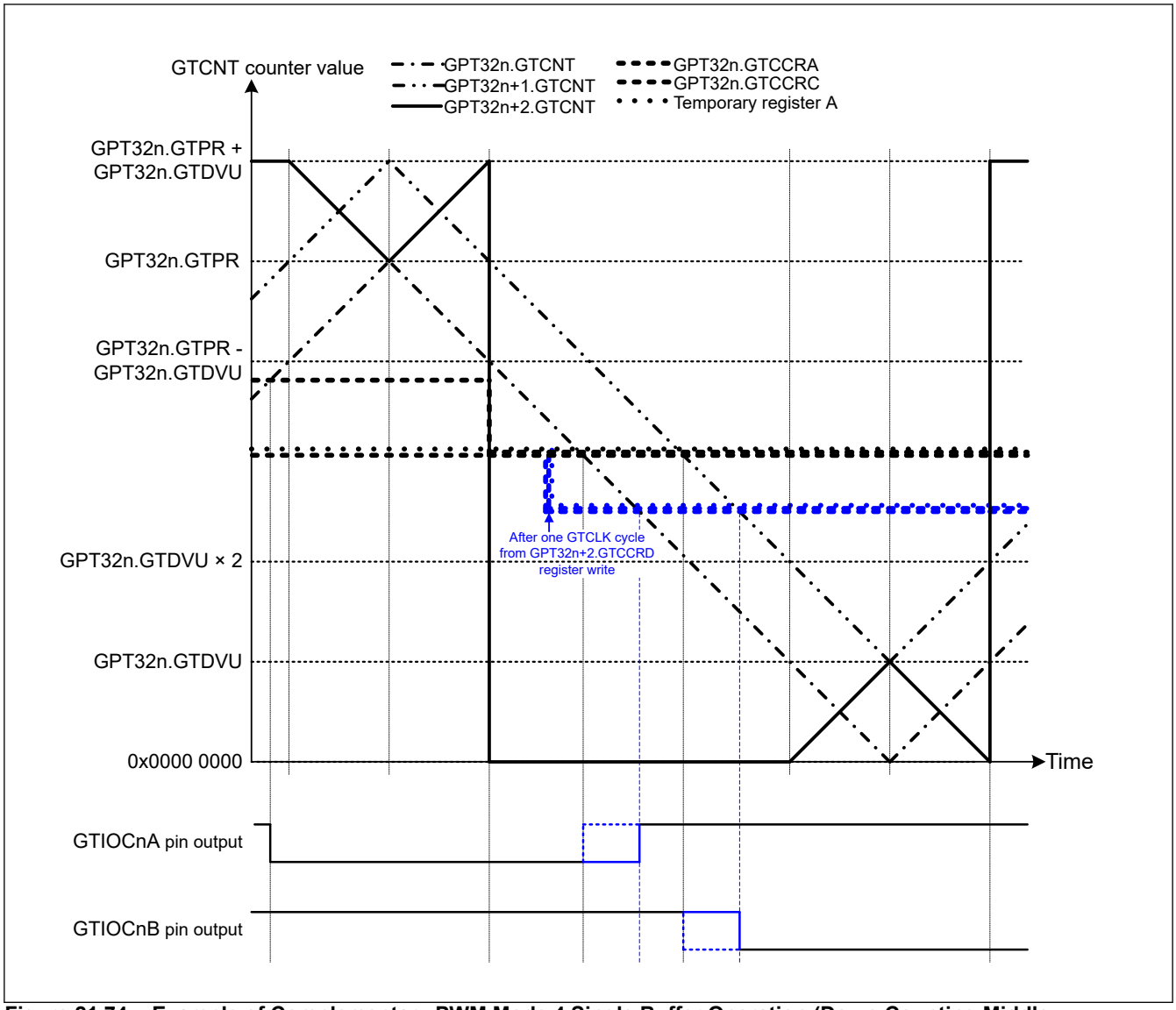


Figure 21.74 Example of Complementary PWM Mode 4 Single Buffer Operation (Down-Counting Middle Section) (Complementary PWM mode 4 single buffer operation, GTIOCnA pin = Low / GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High / GTIOCnB pin = Low at GTCCRA register compare match during down-counting, when a value smaller than the GPT32n.GTCNT value is written to GTCCRD register before down-counting compare match) (n = 4, 7)

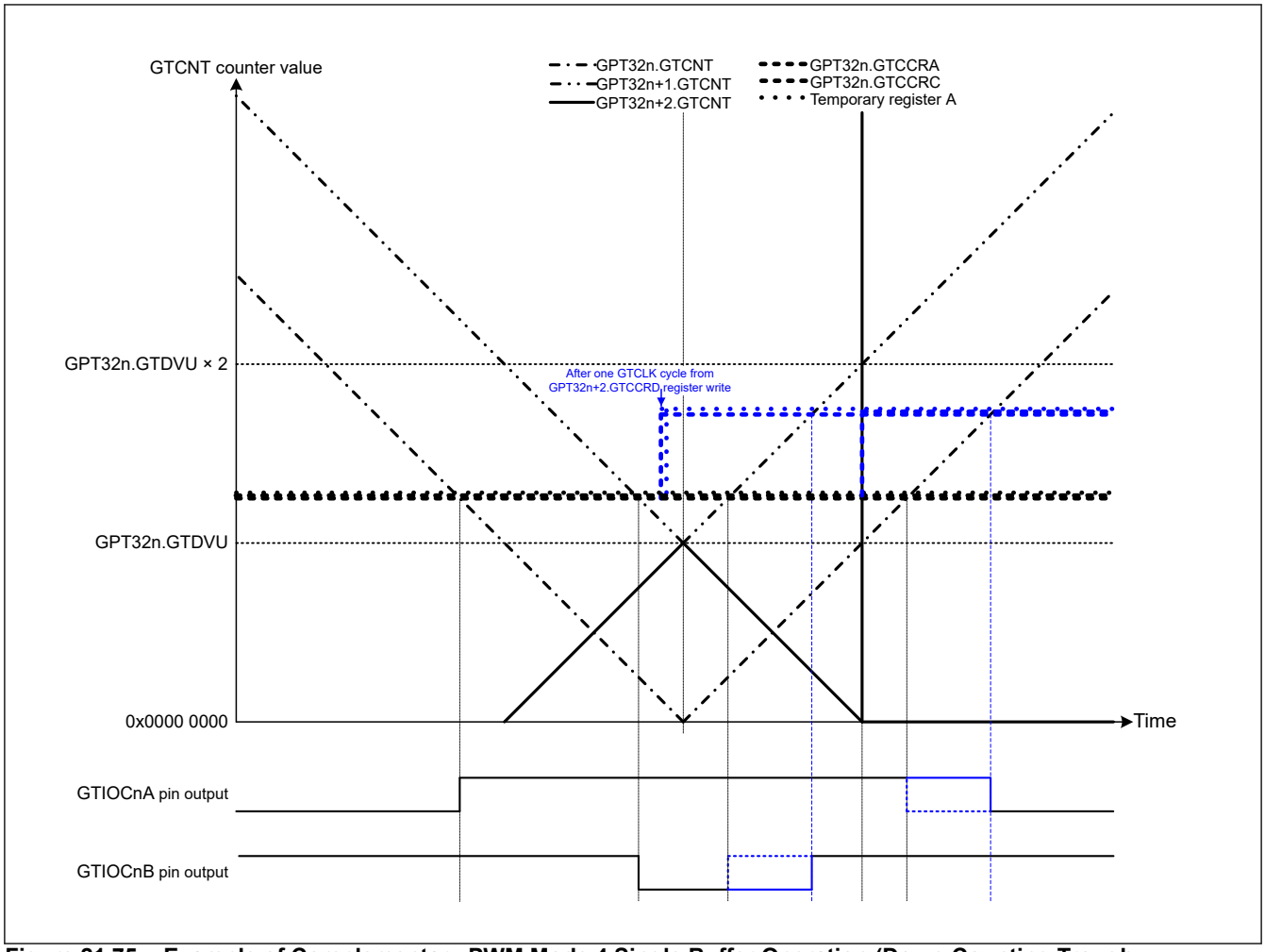


Figure 21.75 Example of Complementary PWM Mode 4 Single Buffer Operation (Down-Counting Trough Section) (Complementary PWM mode 4 single buffer operation, GTIOCnA pin = Low / GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High / GTIOCnB pin = Low at GTCCRA register compare match during down-counting, when a value larger than the GPT32n+2.GTCNT value is written to GTCCRD register after down-counting compare match) (n = 4, 7)

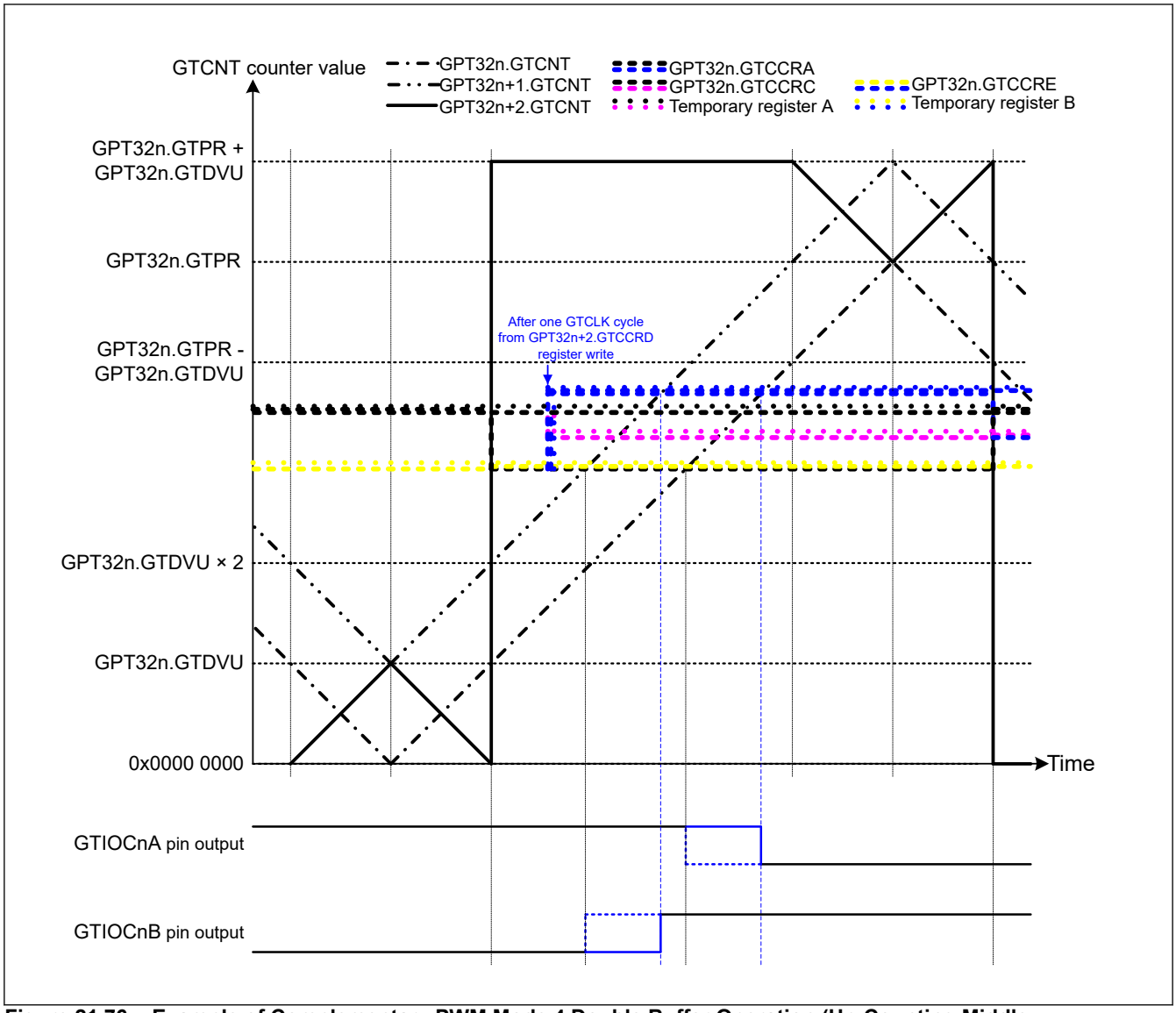


Figure 21.76 Example of Complementary PWM Mode 4 Double Buffer Operation (Up-Counting Middle Section) (Complementary PWM mode 4 double buffer operation, GTIOCnA pin = Low / GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High / GTIOCnB pin = Low at GTCCRA register compare match during down-counting, when a value larger than the GPT32n+1.GTCNT value is written to GTCCRF register before up-counting compare match) (n = 4, 7)

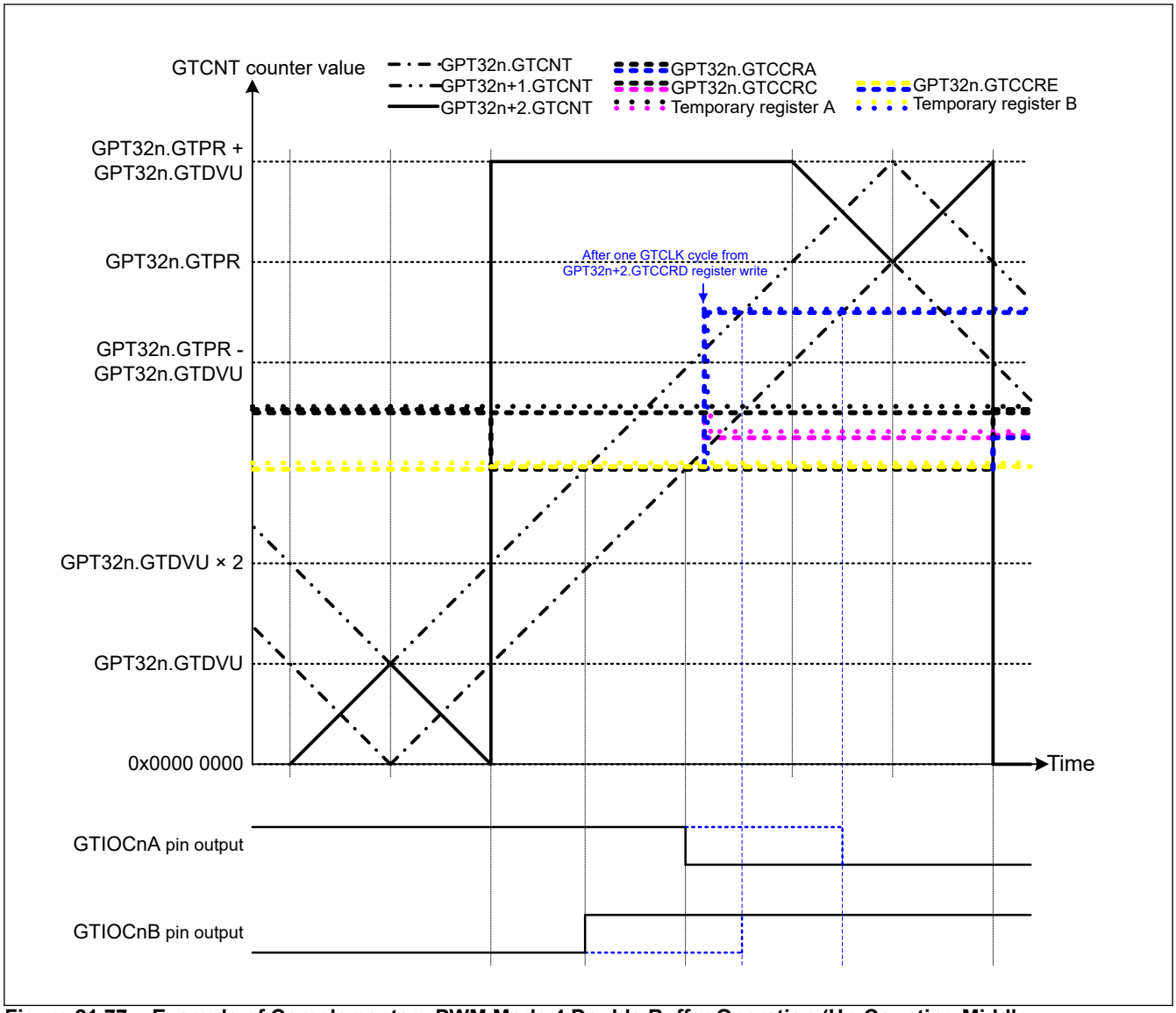


Figure 21.77 Example of Complementary PWM Mode 4 Double Buffer Operation (Up-Counting Middle Section) (Complementary PWM mode 4 double buffer operation, GTIOCnA pin = Low / GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High / GTIOCnB pin = Low at GTCCRA register compare match during down-counting, when a value is written to GTCCRF register after up-counting compare match) (n = 4, 7)

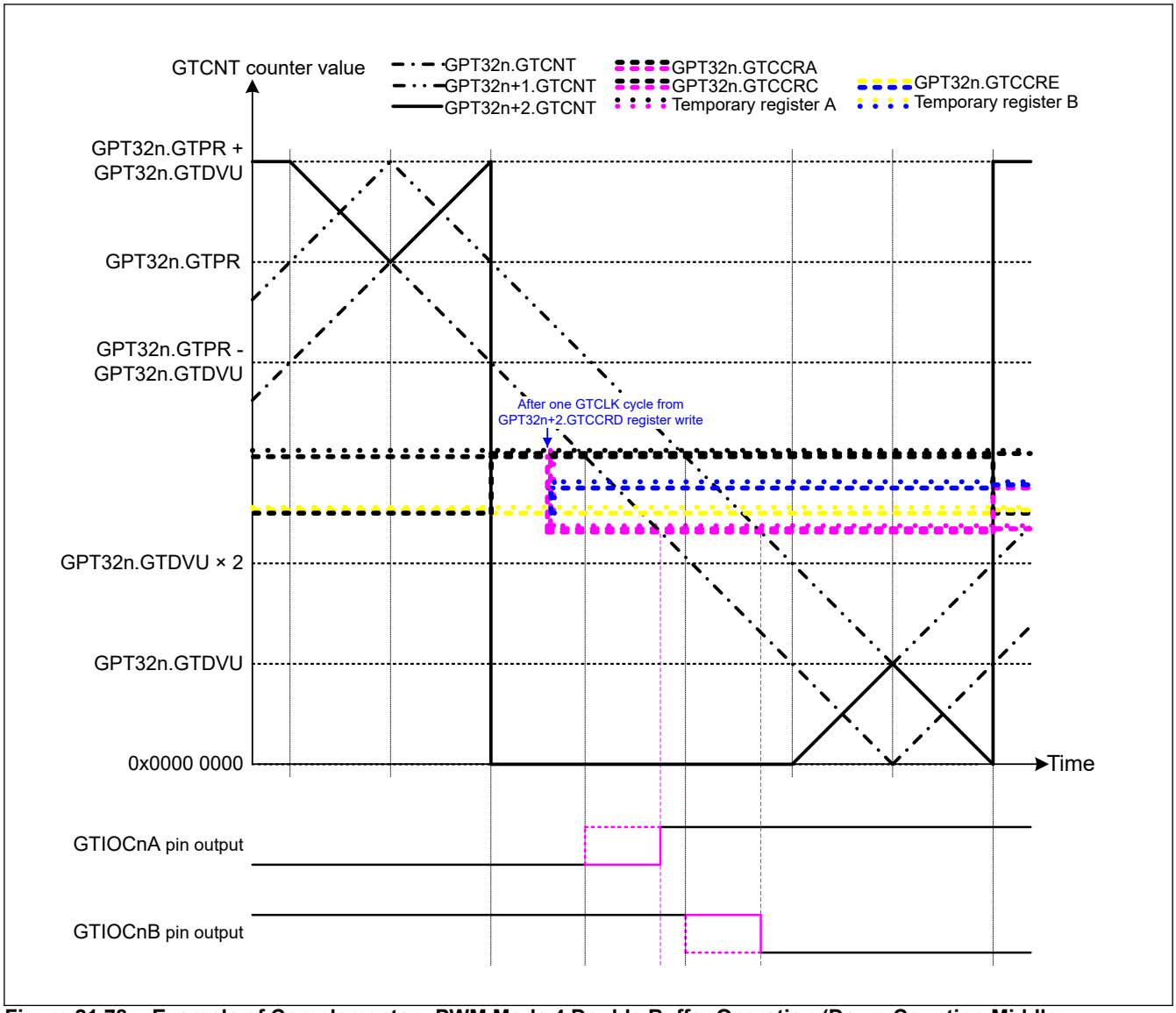


Figure 21.78 Example of Complementary PWM Mode 4 Double Buffer Operation (Down-Counting Middle Section) (Complementary PWM mode 4 double buffer operation, GTIOCnA pin = Low / GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High / GTIOCnB pin = Low at GTCCRA register compare match during down-counting, when a value smaller than the GPT32n.GTCNT value is written to GTCCRD register before down-counting compare match) (n = 4, 7)

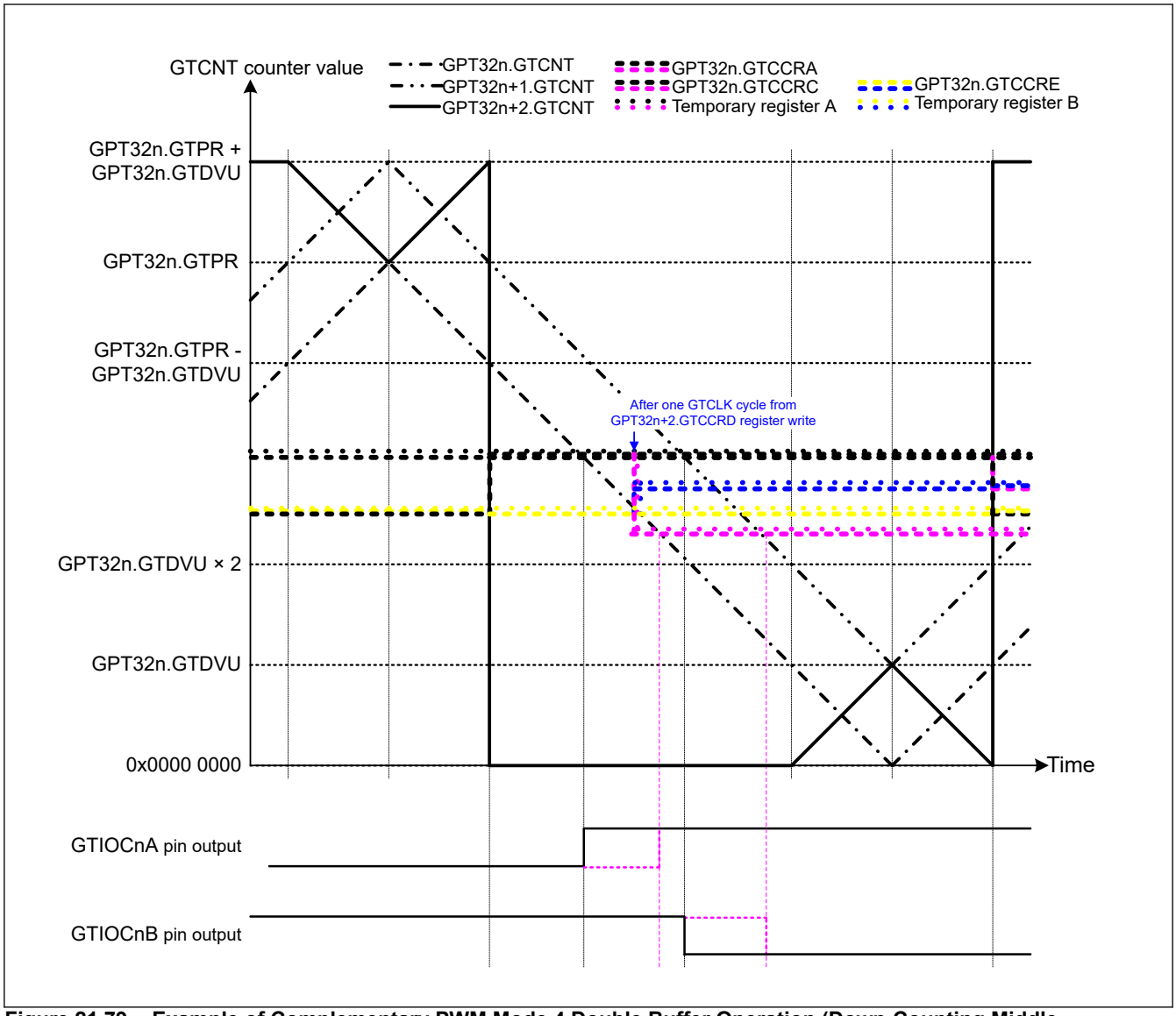


Figure 21.79 Example of Complementary PWM Mode 4 Double Buffer Operation (Down-Counting Middle Section) (Complementary PWM mode 4 double buffer operation, GTIOCnA pin = Low / GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High / GTIOCnB pin = Low at GTCCRA register compare match during down-counting, when a value is written to GTCCRD register during the down-counting dead time) (n = 4, 7)

Table 21.40 Example for Setting Complementary PWM Mode 4 (1 of 2)

No.	Step Name	Description
1	Set operating mode	Set the operating mode (1111b) with GTCR.MD[3:0] of GTP32n channel.
2	Select count clock	Select the count clock with GTCR.TPCS[3:0] bits of GPT32n channel.
3	Set cycle	Set the cycle in GTPR of GPT32n channel.
4	Set GTIOCnm /GTIOCn+1m / GTIOCn+2m pin function	Set the function of the GTIOCnm, GTIOCn+1m, and GTIOCn+2m pins with GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register of the GPT32n, GPT32n+1, and GPT32n+2 channels.
5	Enable GTIOCnm /GTIOCn +1m /GTIOCn+2m pin output	Set to enable the output from the GTIOCnm, GTIOCn+1m, and GTIOCn+2m pins with the OAE and OBE bits in the GTIOR register of the GPT32n, GPT32n+1, and GPT32n+2 channels.
6	Set buffer operation	Set buffer operation with the GTBER2.CP3DB bit of the GPT32n, GPT32n+1, and GPT32n+2 channels.
7	Set compare match value	Set the output pin changing point during up-counting after count start in the GTCCRA register of the GPT32n, GPT32n+1, and GPT32n+2 channels.

Table 21.40 Example for Setting Complementary PWM Mode 4 (2 of 2)

No.	Step Name	Description
8	Set buffer value	For single buffer operation, set data (to be transferred for the first buffer transfer to the GTCCRA register after count start) in the GTCCRD register. For double buffer operation, set data to be transferred at the first end of crest in the GTCCRD register and set data to be transferred at the first end of trough in the GTCCRF register.
9	Set dead t ime value	Set the dead time value in GTDVU of GTP32n channel.
10	Start count operation	Set GTCR.CST of GPT32n channel to 1 to start count operation.
11	Set compare match value to be transferred immediately	For single buffer operation, set data (to be transferred immediately to the GTCCRA register) in the GTCCRD register. For double buffer operation, set the compare match value for down-counting to be transferred immediately in the GTCCRD register and the compare match value for up-counting in the GTCCRF register. Make settings for the GPT32n+2.GTCCRD register finally. (Data is transferred to the temporary register.)

Note: n = 4, 7
m = A, B

21.3.4 Automatic Dead Time Setting Function

By setting GTDTCR, a compare match value for a negative waveform with dead time obtained by a compare match value for a positive waveform (GTCCRA value) and specified dead time value (GTDVU and GTDVD value) can automatically be set to GTCCRB.

The automatic dead time setting function can be used in saw-wave one-shot pulse mode and all the triangle PWM modes.

Dead time can be separately set for the first half and second half of a waveform. Dead time for the changing point in the first half of a negative waveform is set in the GTDVU register and that in the second half is set in GTDVD register. The same dead time can also be set for the first and second halves by setting the GTDTCR.TDFER bit to 1.

The GTDBU register can be used as a buffer register of the GTDVU register, and the GTDBD register can be used as a buffer register of the GTDVD registers. Buffer transfer is performed at the end of the cycle (in saw-wave mode: either of an overflow of the GTCNT counter (up-counting), an underflow (down-counting), or the GTCNT counter clearing; in triangle-wave mode: a trough).

The change point of the negative-phase waveform, which is automatically calculated, is obtained by reading the GTCCRB register. Writing to GTCCRB is prohibited when the automatic dead time setting function is used.

Do not set the dead time that makes the change point of the waveform exceeding the count period. When any dead-time setting which would generate a dead-time error is made, adjust the change points of the positive- and negative-phase waveforms to generate waveforms with secured dead-time as shown in Table 21.41. The adjusted change point of the negative-phase waveform is automatically set in the GTCCRB register. An internal signal is used to judge the change point of the positive-phase waveform, thus the value of the GTCCRA register is not updated by the adjusted value.

In saw-wave one-shot pulse mode, if the order of the change point becomes inconsistent by adjustment of the waveform change point due to occurrence of dead time errors, or if the change point exceeds the count period even after the adjustment, the complementary relation between the positive- and negative-phases cannot be guaranteed.

In triangle-wave PWM mode, if dead time exceeds the count period by setting 0x00000000 or a value greater than or equal to the setting value of the GTPR register is set in the GTCCRA register, output change is controlled by the output protection function (see section 21.8.4. Output Protection Function for GTIOCnm Pin Output (n = 0 to 9; m = A, B)). When the GTCCRA register is greater than or equal to [GTPR register + GTDVm (m = U, D) register], [GTPR register - 1] is set in the GTCCRB register as the upper limit.

Automatic setting for a dead time value to the GTCCRB register is performed at the next count clock after the register value for calculating the automatic setting value is updated. In triangle-wave mode, it also can be done at the next count clock from the current crest.

Table 21.41 Adjustment of the Waveform Change Point When a Dead-Time Error Occurs

PWM Output Operating Mode	Count Direction	Period	Condition for Dead Time Error	Change Point of the Positive-Phase Waveform after Adjustment	Change Point of the Negative-Phase Waveform after Adjustment
Saw-wave one-shot pulse mode	Up-counting	First half	GTCCRA - GTDVU < 0	GTDVU	0
		Second half	GTCCRA + GTDVD > GTPR (GTCCRA + GTDVU > GTPR)*1	GTPR - GTDVD (GTPR - GTDVU)*1	GTPR
	Down-counting	First half	GTCCRA + GTDVU > GTPR	GTPR - GTDVU	GTPR
		Second half	GTCCRA - GTDVD < 0 (GTCCRA - GTDVU < 0)*1	GTDVD (GTDVU)*1	0
Triangle-wave PWM mode 1/2/3	Up-counting	(First half)	GTCCRA - GTDVU ≤ 0	GTDVU + 1	1
	Down-counting	(Second half)	GTCCRA - GTDVD < 0 (GTCCRA - GTDVU < 0)*1	GTDVD (GTDVU)*1	0

Note 1. In the case of GTDTCR.TDFER = 1.

Figure 21.80 to Figure 21.83 show examples of automatic dead time setting function operation. Table 21.42 and Table 21.43 show the setting examples.

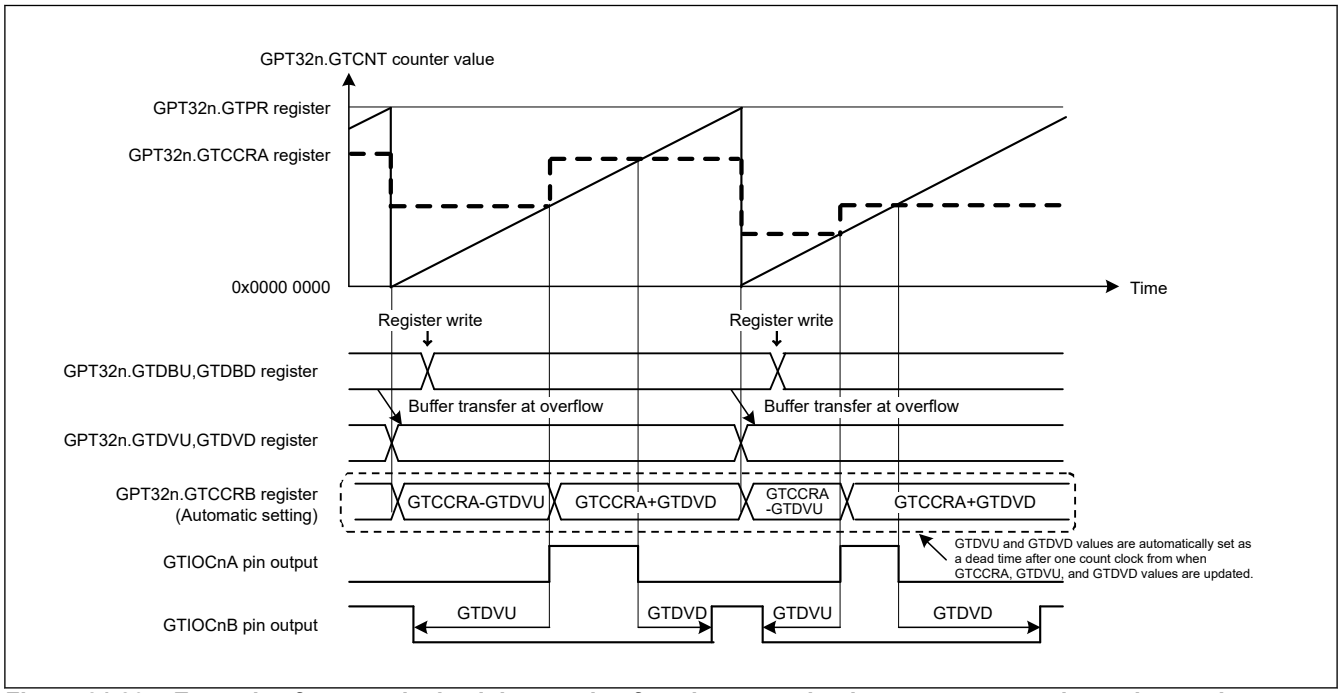


Figure 21.80 Example of automatic dead time setting function operation in saw-wave one-shot pulse mode, up-counting, GTDVU and GTDVD set to buffer operation, and active-high

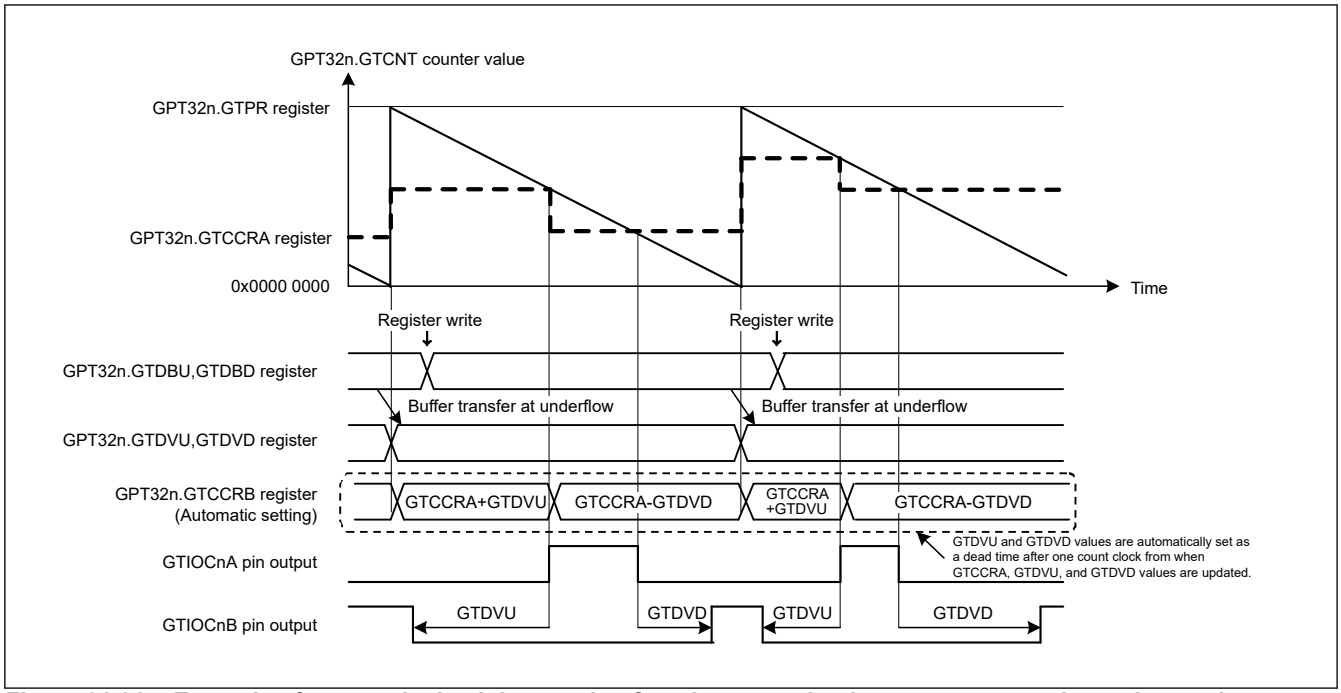


Figure 21.81 Example of automatic dead time setting function operation in saw-wave one-shot pulse mode, down-counting, GTDVU and GTDVD set to buffer operation, and active-high

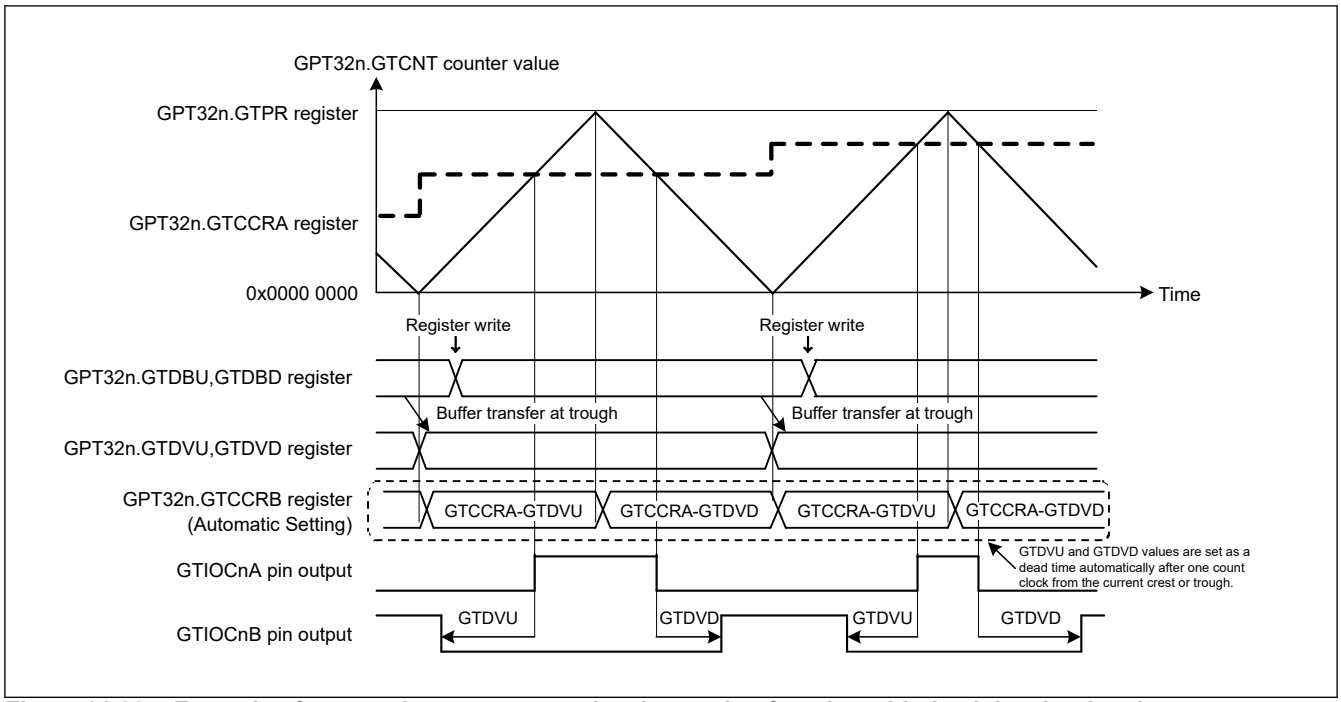


Figure 21.82 Example of automatic compare-match value setting function with dead time in triangle-wave PWM mode 1, GTDVU and GTDVD set to buffer operation, and active-high

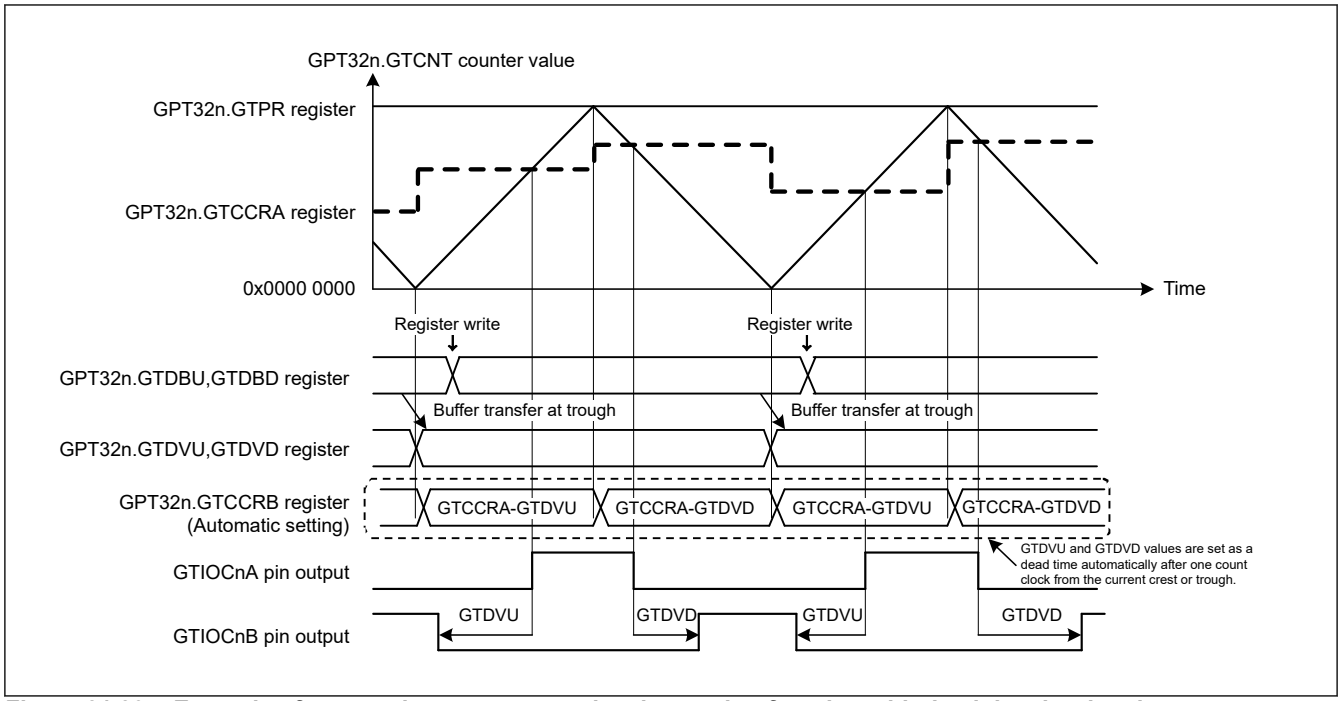


Figure 21.83 Example of automatic compare-match value setting function with dead time in triangle-wave PWM mode 2 or 3, GTDVU and GTDVD set to buffer operation, and active-high

Table 21.42 Example setting for automatic dead time setting function in saw-wave one-shot pulse mode, and triangle-wave PWM mode 3 (1 of 2)

No.	Step Name	Description
1	Set operating mode	Set the operating mode with GTCR.MD[2:0] or GTCR.MD[3:0]. In Figure 21.80 and Figure 21.81, 001b or 0001b (saw-wave one-shot pulse mode) is set. In Figure 21.83, 110b or 0110b (triangle-wave PWM mode 3) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 21.80, 01b is set after 11b is set in GTUDDTYC[1:0] (up count). In Figure 21.81, 00b is set after 10b is set in GTUDDTYC[1:0] (down count).
3	Select count clock	Select the count clock with GTCR.TPCS[3:0].
4	Set cycle	Set the cycle in GTPR.
5	Set initial value for counter	Set the initial value in the GTCNT counter.
6	Set GTIOCnm pin function	Set the GTIOCnm pin function with GTIOA[4:0] and GTIOB[4:0] in GTIOR. In Figure 21.80, Figure 21.81, and Figure 21.83, GTIOA[4:0] = 00011b and GTIOB[4:0] = 10011b.
7	Enable GTIOCnm pin output*1	Set to enable the GTIOCnm pin output with OAE and OBE in GTIOR.
8	Set buffer value for compare match*1	Set the GTIOCnA pin transition immediately after the count start in GTCCRC and GTCCRD.
9	Set forcible buffer transfer for compare match*1	Set GTBER.CCRSWT to 1 to transfer buffer register data forcibly to GTCCRA.
10	Set buffer value for compare match	Set the GTIOCnA pin transition in 1 cycle after the current cycle in GTCCRC and GTCCRD.
11	Set automatic dead time setting function	Set GTDTCR.TDE to 1 to enable the automatic dead time setting function.
12	Set buffer operation for dead time setting	Set buffer operation with TDBUE and TDBDE bits in GTDTCR.
13	Set dead time value	Set the first half dead time value in GTDVU and the second half dead time in GTDVD. When GTDVU is set with GTDTCR.TDFER bit set to 1, the same value is also set to GTDVD, the same dead time value can be set for the first and second halves.
14	Set buffer value for dead time	For buffer operation, set the first half dead time in one cycle after the current cycle in GTDBU and the second half dead time in GTDBD.

Table 21.42 Example setting for automatic dead time setting function in saw-wave one-shot pulse mode, and triangle-wave PWM mode 3 (2 of 2)

No.	Step Name	Description
15	Start count operation	Set GTCR.CST to 1 to start count operation.
16	Set buffer value for each cycle	Set the GTIOCnA pin transition in 1 cycle after the current cycle in GTCCRC and GTCCRD. When the dead time register is used for buffer operation, set the dead time value in the first half of the next cycle from the current cycle to GTDBU and the dead time value in the second half to GTDBD.

Note: n: 0 to 9
m: A, B
Note 1. When using PWM Delay Generation circuit, change the order of setting for Enable GTIOCnm pin output and setting for Set buffer value for compare match + Set forcible buffer transfer for compare match.

Table 21.43 Example setting for automatic dead time setting function in triangle-wave PWM mode 1 or 2

No.	Step Name	Description
1	Set operating mode	Set the operating mode with GTCR.MD[2:0] or GTCR.MD[3:0]. In Figure 21.82, 100b or 0100b (triangle-wave PWM mode 1) is set. In Figure 21.83, 101b or 0101b (triangle-wave PWM mode 2) is set.
2	Select count clock	Select the count clock with GTCR.TPCS[3:0].
3	Set cycle	Set the cycle in GTPR.
4	Set initial value for counter	Set the initial value in the GTCNT counter.
5	Set GTIOCnm pin function	Set the GTIOCnm pin function with GTIOA[4:0] and GTIOB[4:0] in GTIOR. In Figure 21.82 and Figure 21.83, GTIOA[4:0] = 00011b and GTIOB[4:0] = 10011b.
6	Enable GTIOCnm pin output*1	Set to enable the GTIOCnm pin output with OAE and OBE in GTIOR.
7	Set buffer operation for compare match	Set buffer operation with CCRA in GTBER.
8	Set compare match value*1	Set the GTIOCnA pin transition in GTCCRA.
9	Set buffer value for compare match	For buffer operation, set the GTIOCnA pin transition in 1 cycle after the current cycle (in triangle-wave PWM mode 1) or half cycle after the current cycle (in triangle-wave PWM mode 2) in GTCCRC. For double buffer operation, also set the GTIOCnA pin transition in 2 cycles after the current cycle (in triangle-wave PWM mode 1) or 1 cycle after the current cycle (in triangle-wave PWM mode 2) in GTCCRD.
10	Set automatic dead time setting function	Set GTDTCR.TDE to 1 to enable the automatic dead time setting function.
11	Set buffer operation for dead time setting	Set buffer operation with TDBUE and TDBDE bits in GTDTCR.
12	Set dead time value	Set the first half dead time value in GTDVU and the second half dead time in GTDVD. When GTDVU is set with GTDTCR.TDFER bit set to 1, the same value is also set to GTDVD, the same dead time value can be set for the first and second halves.
13	Set buffer value for dead time	For buffer operation, set the first half dead time in one cycle after the current cycle in GTDBU and the second half dead time in GTDBD.
14	Start count operation	Set GTCR.CST to 1 to start count operation.
15	Set buffer value for each cycle	When the compare match register is used for buffer operation, set the GTIOCnA pin transition in 1 cycle after the current cycle (in triangle-wave PWM mode 1) or half cycle after the current cycle (in triangle-wave PWM mode 2) in GTCCRC. When the compare match register is used for double-buffered operation, set the GTIOCnA pin changing point in two cycles after the current cycle (in triangle-wave PWM mode 1) or one cycle after the current cycle (in triangle-wave PWM mode 2) in GTCCRD. In the same way, set the dead time value in the first half of the cycle after current cycle in GTDBU and the dead time in the second half in GTDBD.

Note: n: 0 to 9
m: A, B
Note 1. When using PWM Delay Generation circuit, change the order of setting for the GTIOCnm pin output enable and setting for a compare match value.

21.3.5 Count Direction Changing Function

The count direction of the GTCNT counter can be changed by modifying the UD bit in GTUDDTYC.

In saw-wave mode, if the UD bit in GTUDDTYC is modified during count operation, the count direction is changed at an overflow (when modified during up-counting) or an underflow (when modified during down-counting). If the GTUDDTYC.UD bit is modified while the count operation stops and the GTUDDTYC.UDF bit is 0, the GTUDDTYC.UD bit modification is not reflected at the start of counting and the count direction is changed at an overflow or an underflow. If the UDF bit is set to 1 while the count operation stops, the GTUDDTYC.UD bit value at that time is reflected at the start of counting.

In triangle-wave mode, the count direction does not change even though the UD bit in GTUDDTYC is modified during the count operation. Similarly, even though the GTUDDTYC.UD bit is modified while the count operation stops and GTUDDTYC.UDF bit is 0, the GTUDDTYC.UD bit value is not reflected to the count operation. If the GTUDDTYC.UDF bit is set to 1 while the count operation is stopped, the GTUDDTYC.UD bit value at that time is reflected at the start of counting.

If the count direction changes during a saw-wave count operation, the GTPR value after the start of up-counting is reflected in the count cycle during up-counting and the GTPR value after the start of down-counting is reflected in the count cycle during down-counting.

Figure 21.84 shows an example of count direction changing function operation.

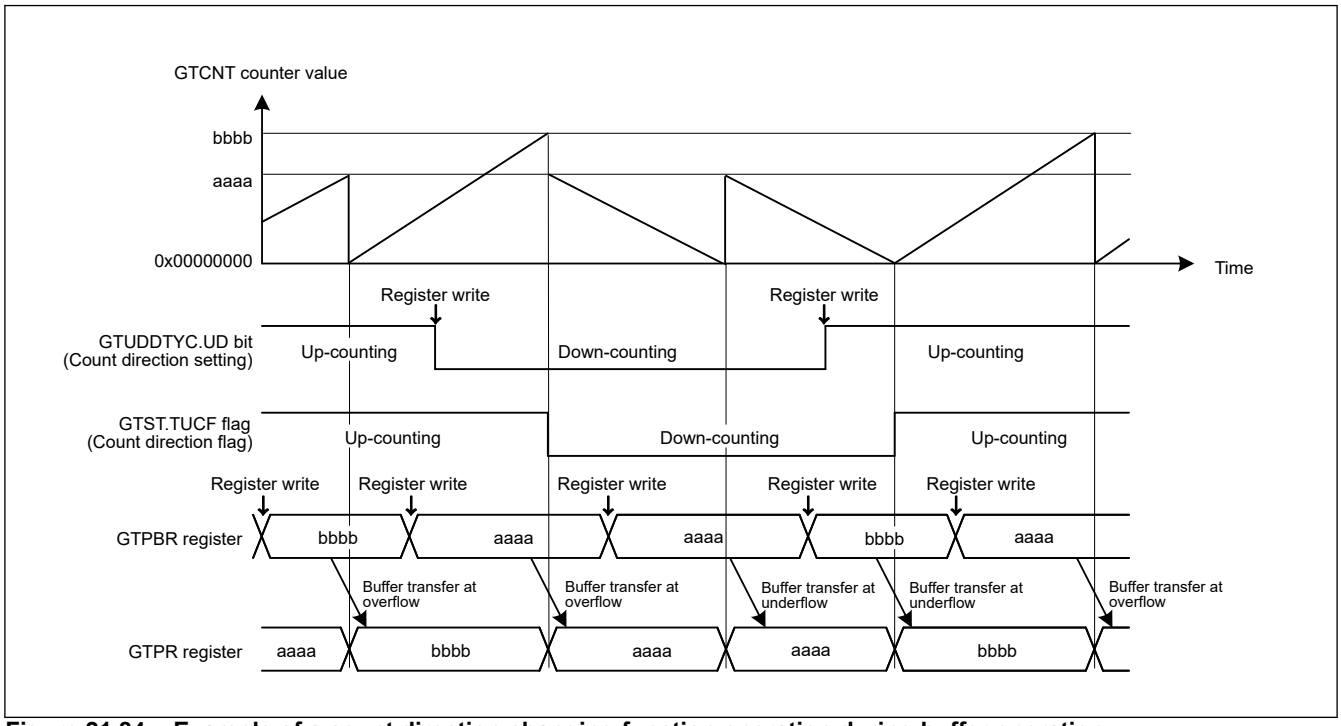


Figure 21.84 Example of a count direction changing function operation during buffer operation

21.3.6 Function of Output Duty 0% and 100%

The output duty of the GTIOCnA pin and the GTIOCnB pin (n = 0 to 9) are set to 0% or 100% by changing the GTUDDTYC.OADTY bit or GTUDDTYC.OBDTY bit.

This function is invalid in saw-wave PWM mode 2 or complementary PWM mode.

In saw-wave mode, if the GTUDDTYC.OADTY bit or the GTUDDTYC.OBDTY bit is modified during the count operation, the output duty setting is reflected at an overflow (when modified during up-counting) or an underflow (when modified during down-counting). If the GTUDDTYC.OADTY bit or the GTUDDTYC.OBDTY bit is modified while the count operation is stopped and the GTUDDTYC.OADTYF or the GTUDDTYC.OBDTYF bit is 0, the output duty modification is not reflected at the start of counting. The output duty changes at an overflow or an underflow. If the GTUDDTYC.OADTY bit or the GTUDDTYC.OBDTY bit is modified while the count operation is stopped and the GTUDDTYC.OADTYF or the GTUDDTYC.OBDTYF bit is 1, the GTUDDTYC.OADTY bit or the GTUDDTYC.OBDTY bit value at that time is reflected at the start of counting.

In triangle-wave mode, if the GTUDDTYC.OADTY bit or the GTUDDTYC.OBDTY bit is modified during the count operation, the output duty setting is reflected an underflow.

If the GTUDDTYC.OADTY bit or the GTUDDTYC.OBDTY bit is modified while the count operation is stopped and the GTUDDTYC.OADTYF or the GTUDDTYC.OBDTYF bit is 0, the output duty modification is not reflected at the start of counting. The output duty changes at an underflow. If the GTUDDTYC.OADTY bit or the GTUDDTYC.OBDTY bit is modified while the count operation stops and the GTUDDTYC.OADTYF or the GTUDDTYC.OBDTYF bit is 1, the output duty modification is reflected at the start of counting.

In performing 0% or 100% duty operation, GPT internally continues to:

- Perform compare match operation
- Set compare match flag
- Output interrupt
- Perform buffer operation.

When the control is changed from 0% or 100% duty setting to compare match, the output value of GTIOCnA pin at cycle end is decided by GTIOR.GTIOA[3:2] and GTUDDTYC.OADTYR. The output value of GTIOCnB pin at cycle end is decided by GTIOR.GTIOB[3:2] and GTUDDTYC.OBDTYR.

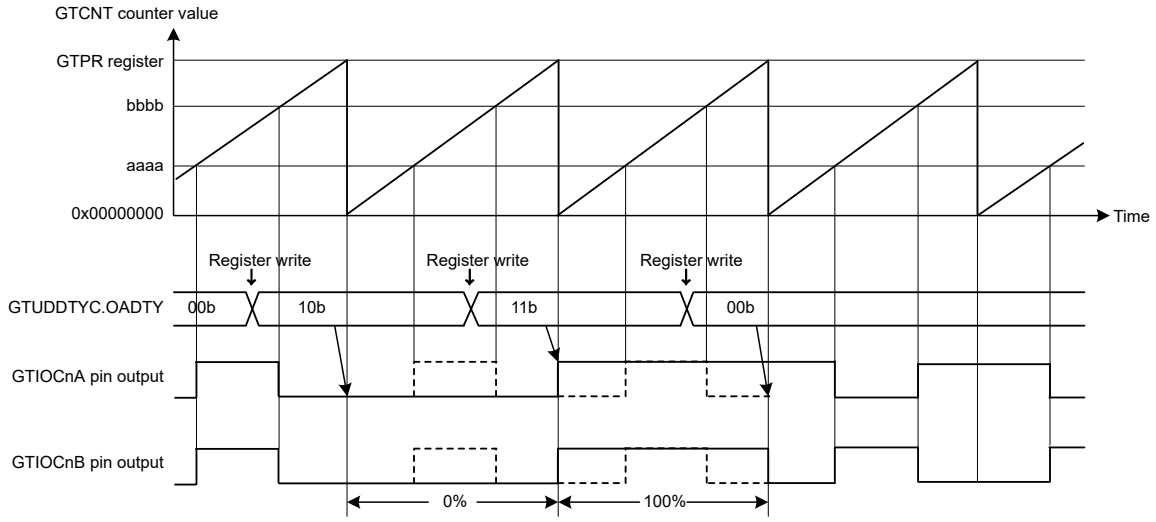
When GTIOR.GTIOA[3:2] and GTIOR.GTIOB[3:2] are set to 01b, the output pins output low at cycle end. When GTIOR.GTIOA[3:2] and GTIOR.GTIOB[3:2] are set to 10b, the output pins output high at cycle end.

GTUDDTYC.OADTYR selects the value that is the object of output retained/toggled at cycle end, when GTIOR.GTIOm[3:2] are set to 00b (output retained at cycle end) or when GTIOR.GTIOm[3:2] are set to 11b (output toggled at cycle end). Table 21.44 shows the values of GTIOCnA and GTIOCnB pin output at cycle end.

Table 21.44 Output values after releasing 0% or 100% duty setting (m = A, B)

GTIOR.GTIOm[3:2]	Compare match value at cycle end masked by 0% or 100% duty setting	GTUDDTYC.OmDTYR in duty 0% setting		GTUDDTYC.OmDTYR in duty 100% setting	
		0	1	0	1
00 (output retained at cycle end)	0	0	0	1	0
	1	0	1	1	1
01 (low output at cycle end)	—	0	0	0	0
10 (high output at cycle end)	—	1	1	1	1
11 (output toggled at cycle end)	0	1	1	0	1
	1	1	0	0	0

Figure 21.85 shows an example of output duty 0% and 100% function.



Note: The base address of registers are the same.

Note: [Setting examples]

GTIOR.GTIOA[4:0] bits:	00011b initial low output, output toggled at compare match, output retained at cycle end
GTUDDTYC.OADTYR bit:	0b The function selected by GTIOA[3:2] is applied to the output value when the duty cycle is set after release from 0 or 100% duty-cycle setting.
GTIOR.GTIOB[4:0] bits:	00011b initial low output, output toggled at compare match, output retained at cycle end
GTUDDTYC.OBDTYR bit:	1 The function of GTIOB[3:2] is applied to the compare match output value which is masked after 0 or 100% duty setting release.

Figure 21.85 Example of output duty 0% and 100% function

21.3.7 Hardware Count Start/Count Stop and Clear Operation

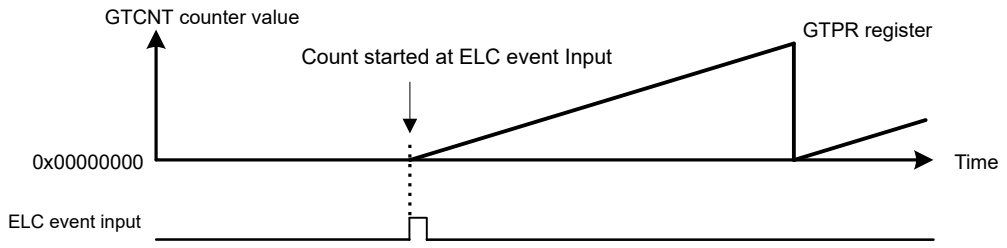
The GTCNT counter can be started, stopped, or cleared by the following hardware sources:

- External trigger input
- ELC event input
- GTIOCnA and GTIOCnB pin input (n = 0 to 9).

21.3.7.1 Hardware Start Operation

The GTCNT counter can be started by selecting a hardware source using GTSSR.

Figure 21.86 shows an example of a count start operation by a hardware source. Table 21.45 shows the setting example.



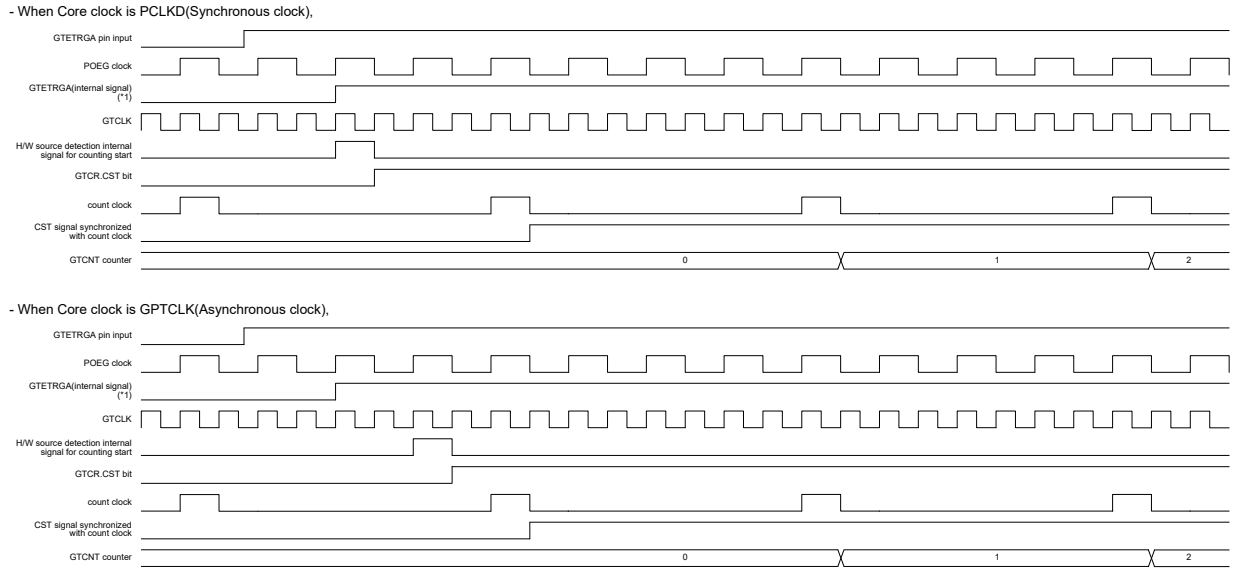
Note: ELC event input: ELC_GPTA event input

Figure 21.86 Example of count start operation by a hardware source started at the input of the signal from the ELC_GPTA event

Table 21.45 Example setting for count start operation by a hardware source

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits or GTCR.MD[3:0] bits. In Figure 21.86, 000b or 0000b (saw-wave PWM mode 1) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 21.86, after 11b is set in the GTUDDTYC[1:0] bits, 01b is set in the GTUDDTYC[1:0] bits (up-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Set the cycle in the GTPR register.
5	Set initial value for counter	Set the initial value in the GTCNT counter. In Figure 21.86, 0x00000000 is set.
6	Set hardware count start	Select a hardware source for starting count operation in the GTSSR register. In Figure 21.86, GTSSR.SSELCA = 1
7	Set hardware source operation	Set operation of the hardware source selected by the GTSSR register and start counting. In Figure 21.86, the ELC_GPTA event input operation is set.

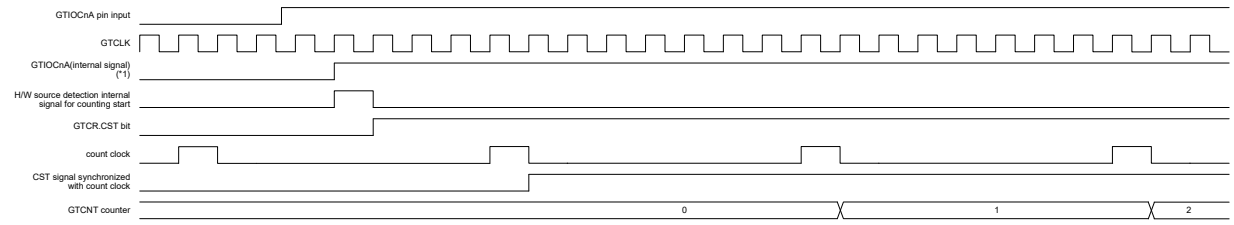
Figure 21.87 shows an example of timing of operations to start counting in response to a rising edge of the input on the GTETRGA pin.



Note: POEG clock: PCLKB
Note 1. The GTETRGA internal signal reflects the signal being input to GPT when the digital noise filter for the GTETRGA pin in the POEG is not in use.

Figure 21.87 Example of Timing of Operations to Start Counting in Response to a Rising Edge of the Input on the GTETRGA Pin

Figure 21.88 shows an example of timing of operations to start counting in response to a rising edge of the input on the GTIOCnA pin.



Note 1. This is the case when the digital noise filter is not in use.

Figure 21.88 Example of Timing of Operations to Start Counting in Response to a Rising Edge of the Input on the GTIOCnA Pin

Figure 21.89 shows an example of the timing of operations to start counting in response to ELC_GPTA event input.

This is an example of operations to start counting by the GPT321.GTCNT counter in response to a signal. An event signal is output to the ELC after matches in comparison with the GPT320.GTCCRA register. This is selected as a trigger for output to the GPT321 by the ELC as ELC_GPTA.

ELC passes the event signal output from GPT320 without delay to GPT321.

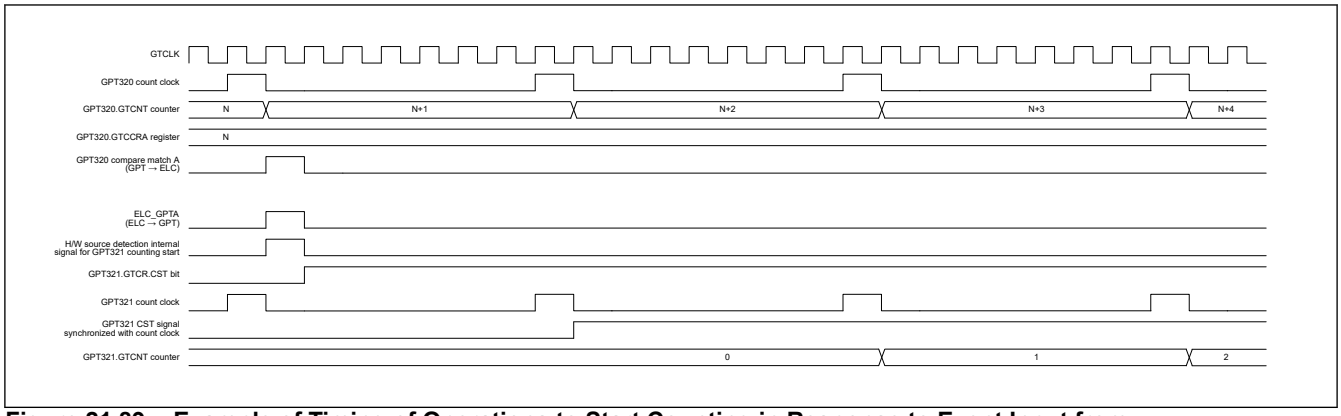
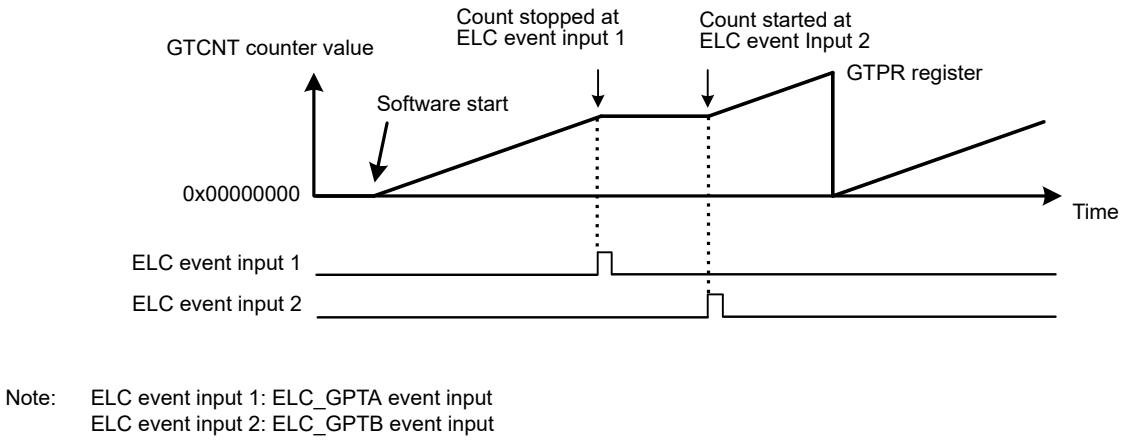


Figure 21.89 Example of Timing of Operations to Start Counting in Response to Event Input from ELC_GPTA

21.3.7.2 Hardware Stop Operation

The GTCNT counter can be stopped by selecting a hardware source using GTPSR.

Figure 21.90 shows an example of a count stop operation by a hardware source. Table 21.46 shows the setting example. In this example, the count operation stops at the ELC_GPTA event input and restarts at the ELC_GPTB event input.



Note: ELC event input 1: ELC_GPTA event input
ELC event input 2: ELC_GPTB event input

Figure 21.90 Example of count stop operation by hardware source started by software, stopped at ELC_GPTA input, and restarted at ELC_GPTB input

Table 21.46 Example setting for count stop operation by a hardware source (1 of 2)

No.	Step Name	Description
1	Set operating mode	Set the operating mode with GTCR.MD[2:0] or GTCR.MD[3:0] bits. In Figure 21.90, 000b or 0000b (saw-wave PWM mode 1) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 21.90, after 11b is set in GTUDDTYC[1:0], 01b is set in GTUDDTYC[1:0] (up-counting).
3	Select count clock	Select the count clock with GTCR.TPCS[3:0].
4	Set cycle	Set the cycle in GTPR.
5	Set initial value for counter	Set the initial value in the GTCNT counter. In Figure 21.90, 0x00000000 is set.
6	Set hardware count start	Select a hardware source for starting count operation in GTSSR register, and wait for count start by the hardware source. In Figure 21.90, GTSSR.SSELCB = 1.
7	Set hardware count stop	Select a hardware source for stopping count operation in GTPSR register and wait for count stop by the hardware source. In Figure 21.90, GTPSR.PSELCA = 1.

Table 21.46 Example setting for count stop operation by a hardware source (2 of 2)

No.	Step Name	Description
8	Set hardware source operation	Set operation of the hardware source selected in GTSSR register or GTPSR register, and start or stop counting. In Figure 21.90, ELC_GPTA input operation and ELC_GPTB input operation are set.

Figure 21.91 shows an example of a count start/stop operation by a hardware source. Table 21.47 shows the setting example. In this example, the counter operates during the high-level periods of the external trigger input GTETRGA.

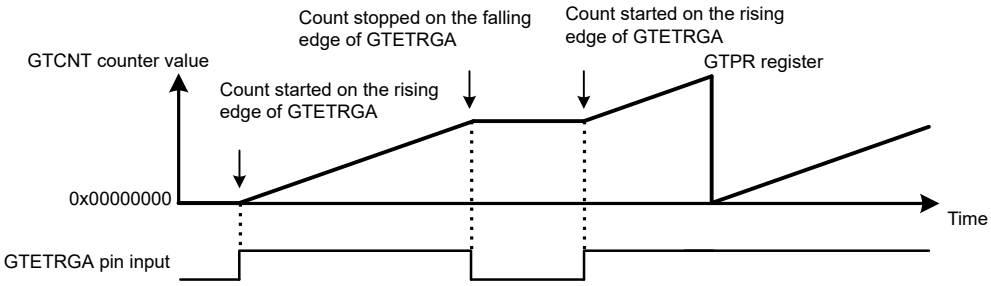
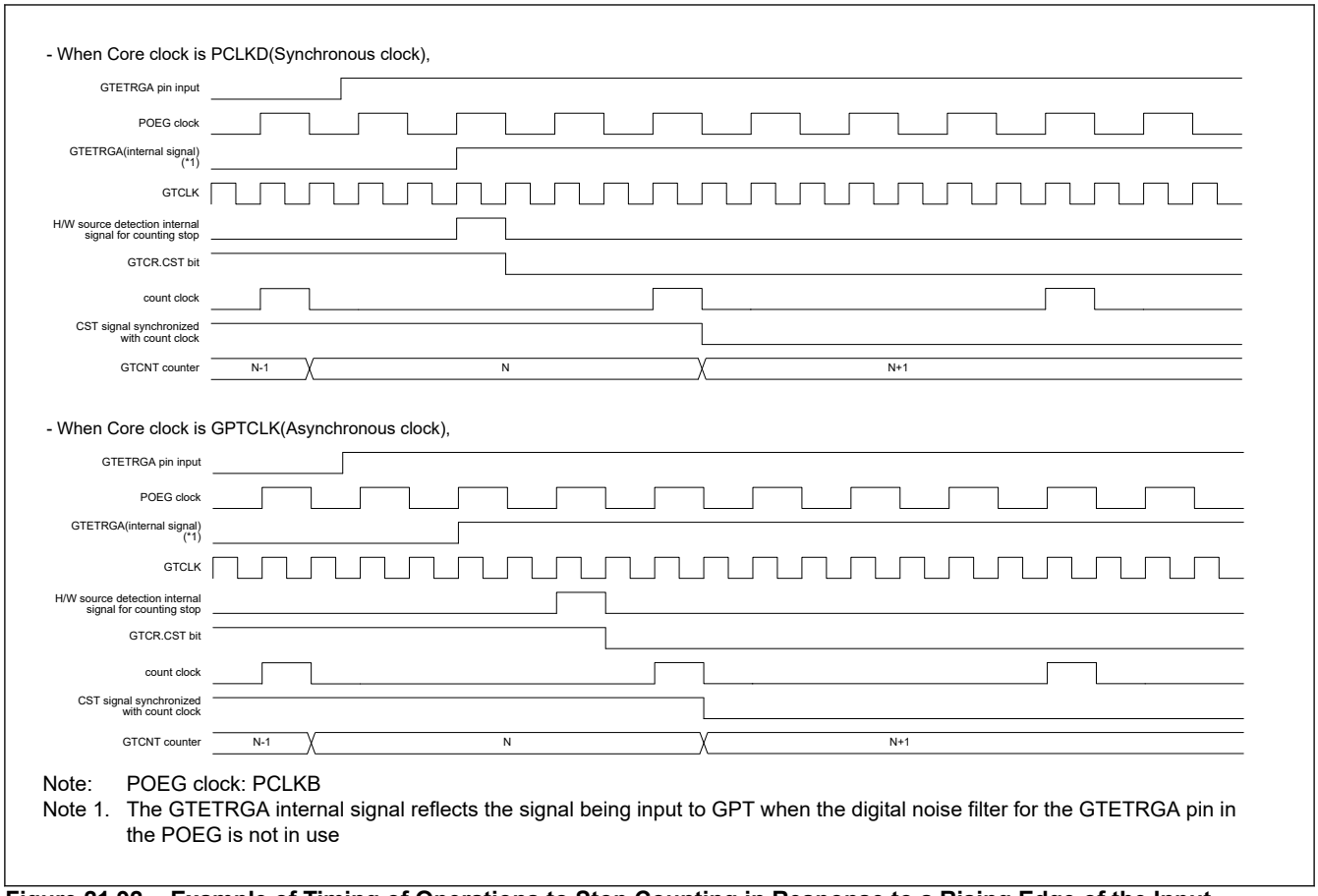


Figure 21.91 Example of count start/stop operation by a hardware source started on the rising edge of GTETRGA pin input, and stopped on the falling edge of GTETRGA pin input

Table 21.47 Example setting for count start/stop operation by a hardware source

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits or GTCR.MD[3:0] bits. In Figure 21.91, 000b or 0000b (saw-wave PWM mode 1) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 21.91, after 11b is set in the GTUDDTYC[1:0] bits, 01b is set in the GTUDDTYC[1:0] bits (up-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Set the cycle in the GTPR register.
5	Set initial value for counter	Set the initial value in the GTCNT counter. In Figure 21.91, 0x00000000 is set.
6	Set hardware count start	Select a hardware source for starting count operation with the GTSSR register, and wait for count start by the hardware source. In Figure 21.91, GTSSR.SSGTRGAR = 1.
7	Set hardware count stop	Select a hardware source for stopping count operation with the GTPSR register, and wait for count stop by the hardware source. In Figure 21.91, GTPSR.PSGTRGAF = 1.
8	Set hardware source operation	Set operation of the hardware source selected in the GTSSR register or GTPSR register and start or stop counting. In Figure 21.91, the GTETRGA pin operation is set.

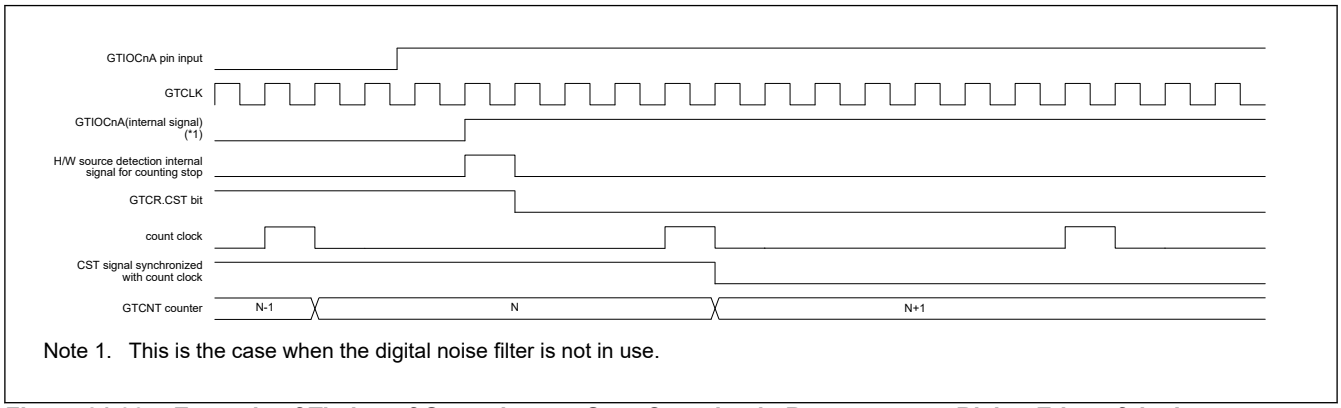
Figure 21.92 shows an example of timing of operations to stop counting in response to a rising edge of the input on the GTETRGA pin.



Note: POEG clock: PCLKB
Note 1. The GTETRGA internal signal reflects the signal being input to GPT when the digital noise filter for the GTETRGA pin in the POEG is not in use

Figure 21.92 Example of Timing of Operations to Stop Counting in Response to a Rising Edge of the Input on the GTETRGA Pin

Figure 21.93 shows an example of timing of operations to stop counting in response to a rising edge of the input on the GTIOCnA pin.



Note 1. This is the case when the digital noise filter is not in use.

Figure 21.93 Example of Timing of Operations to Stop Counting in Response to a Rising Edge of the Input on the GTIOCnA Pin

Figure 21.94 shows an example of timing of operations to stop counting in response to event input from ELC_GPTA.

This is an example of operations to stop counting by the GPT321.GTCNT counter in response to a signal. An event signal is output to the ELC after matches in comparison with the GPT320.GTCCRA register. This is selected as a trigger for output to the GPT321 by the ELC as ELC_GPTA.

ELC passes the event signal output from GPT320 without delay to GPT321.

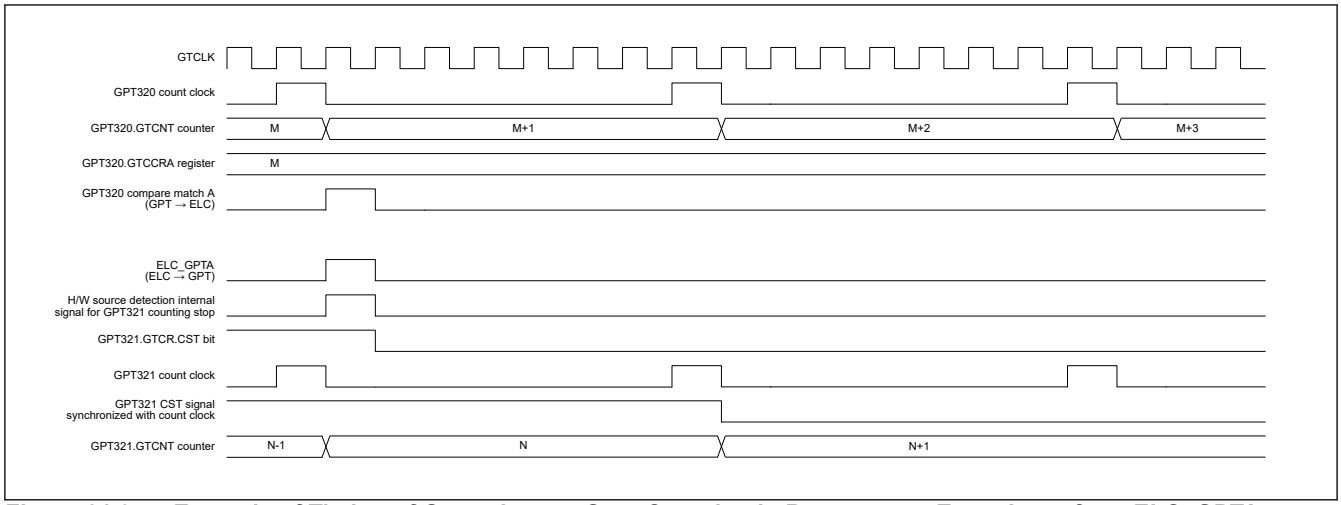


Figure 21.94 Example of Timing of Operations to Stop Counting in Response to Event Input from ELC_GPTA

21.3.7.3 Hardware Clear Operation

The GTCNT counter can be cleared by selecting a hardware source using GTCSR. The GPTn_OVF/GPTn_UDF (n = 0 to 9) interrupt (overflow/underflow interrupt) is not generated when the GTCNT counter is cleared by a hardware source or by software.

Figure 21.95 and Figure 21.96 show examples of the GTCNT counter clearing operation by a hardware source. Table 21.48 shows the setting example. In this example, the GTCNT counter starts at the ELC_GPTA input, and the counter stops and clears at the ELC_GPTB input.

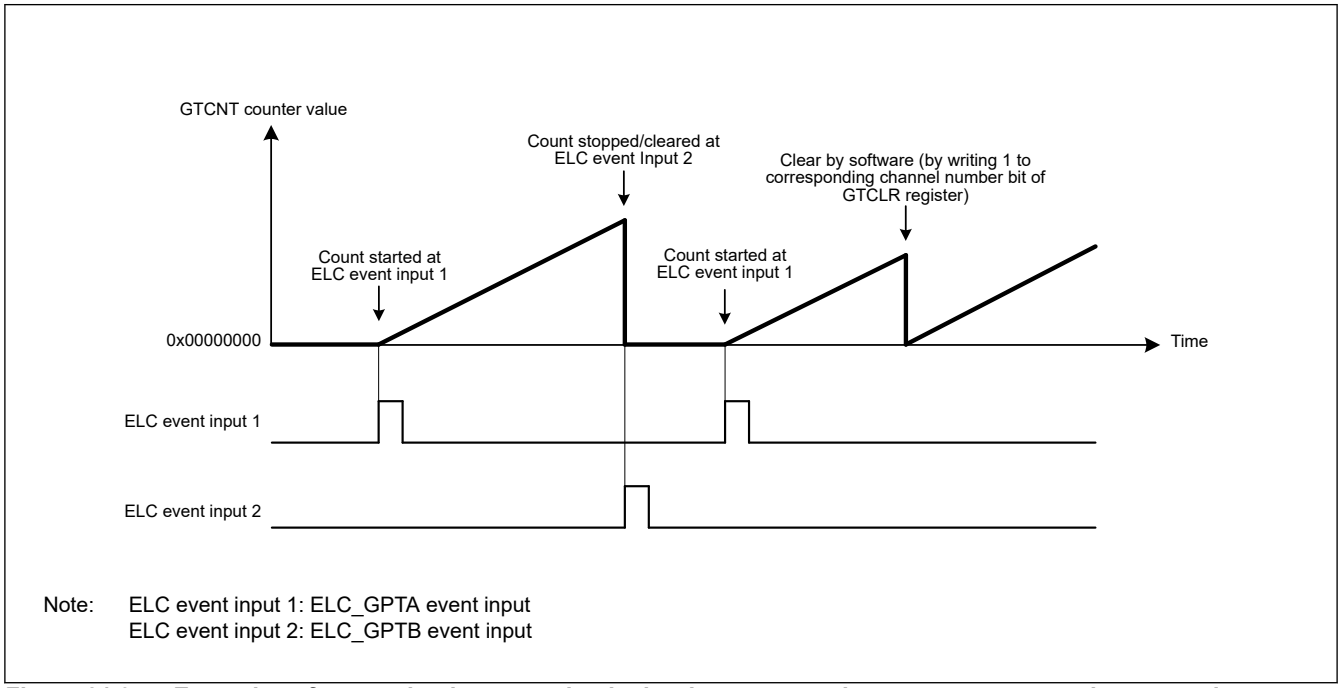
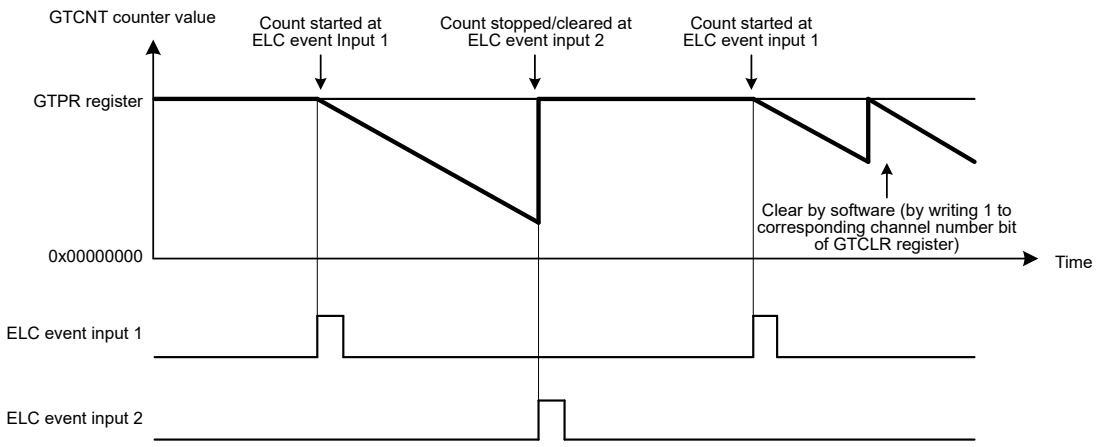


Figure 21.95 Examples of count clearing operation by hardware source in saw wave up-counting, started at ELC_GPTA input, and stopped/cleared at ELC_GPTB input



Note: ELC event input 1: ELC_GPTA event input
ELC event input 2: ELC_GPTB event input

Figure 21.96 Examples of count clearing operation by hardware source in saw wave down-counting, started at ELC_GPTA input, and stopped/cleared at ELC_GPTB input

Table 21.48 Example setting for count clearing operation by a hardware source

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits or GTCR.MD[3:0] bits. In Figure 21.95 and Figure 21.96, 000b or 0000b (saw-wave PWM mode 1) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 21.95, after 11b is set in the GTUDDTYC[1:0] bits, 01b is set in the GTUDDTYC[1:0] bits (up-counting). In Figure 21.96, after 10b is set in the GTUDDTYC[1:0] bits, 00b is set in the GTUDDTYC[1:0] bits (down-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Set the cycle in the GTPR register.
5	Set initial value for counter	Set the initial value in the GTCNT counter. In Figure 21.95, 0x00000000 is set. In Figure 21.96, the GTPR register value is set.
6	Set hardware count start	Select a hardware source for starting count operation in the GTSSR register, and wait for count start by the hardware source. In Figure 21.95 and Figure 21.96, GTSSR.SSELCA = 1.
7	Set hardware count stop	Select a hardware source for stopping count operation in the GTPSR register, and wait for count stop by the hardware source. In Figure 21.95 and Figure 21.96, GTPSR.PSELCB = 1.
8	Set hardware count clear	Select a hardware source for clearing count operation in the GTCSR register, and wait for count clear by the hardware source. In Figure 21.95 and Figure 21.96, GTCSR.CSELCB = 1.
9	Set hardware source operation	Set operation of the hardware source selected in the GTSSR register, GTPSR register or GTCSR register and start, stop or clear counting. In Figure 21.95 and Figure 21.96, the ELC_GPTA input and ELC_GPTB input are set.

The GPTn_OVF/GPTn_UDF (n = 0 to 9) interrupt (overflow/underflow interrupt) is not generated when the counter is cleared by a hardware source or by software.

Figure 21.97 shows the relationship between the counter clearing by a hardware source and the GPTn_OVF (n = 0 to 9) interrupt.

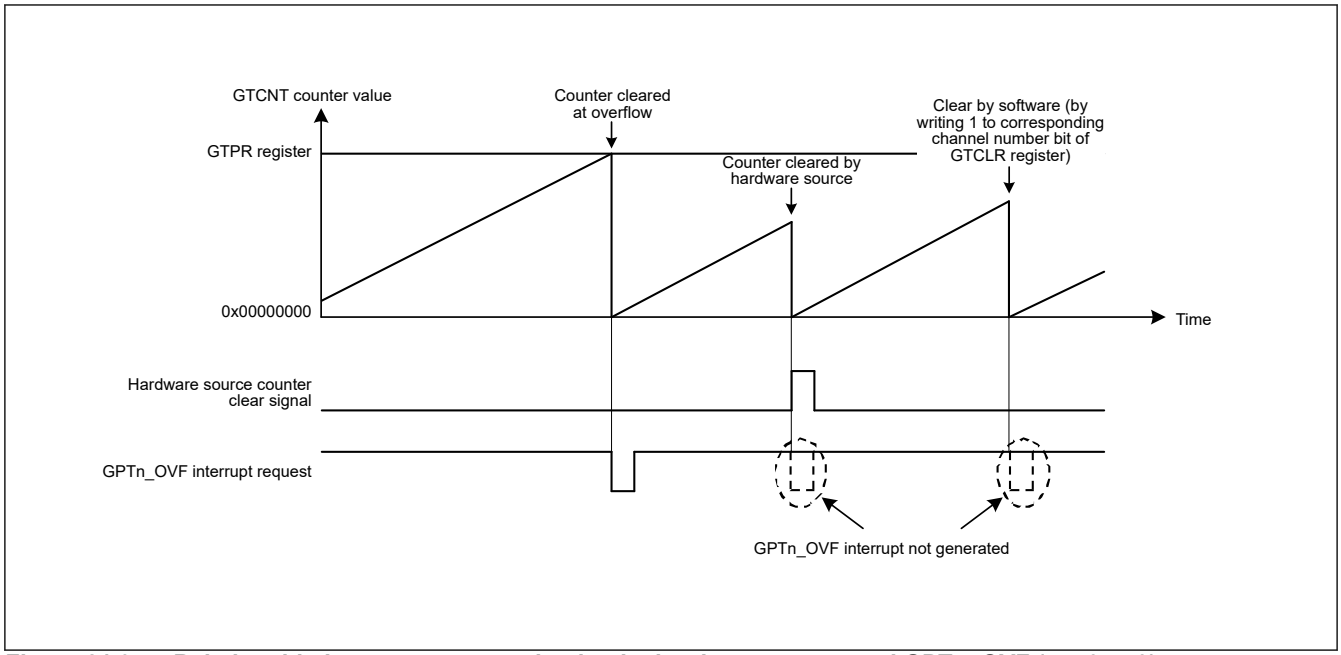
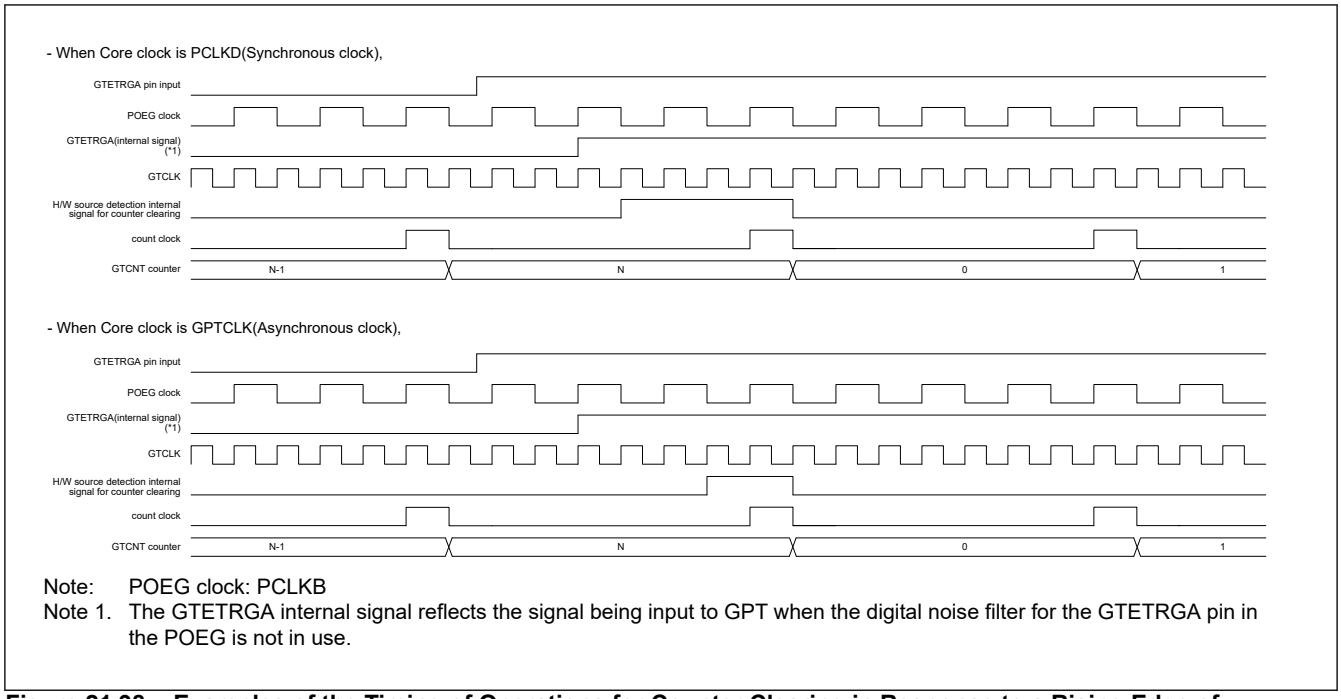


Figure 21.97 Relationship between counter clearing by hardware source and GPTn_OVF (n = 0 to 9) interrupt

Figure 21.98 shows an example of the timing of operations to clear the counter in response to a rising edge of the input on the GTETRGA pin when a clock signal produced by frequency-dividing the GTCLK signal is used as the counter clock for the GTCNT counter.

The GTCNT counter is cleared when counting is in progress after the GPT32 has detected the internal clearing signal.

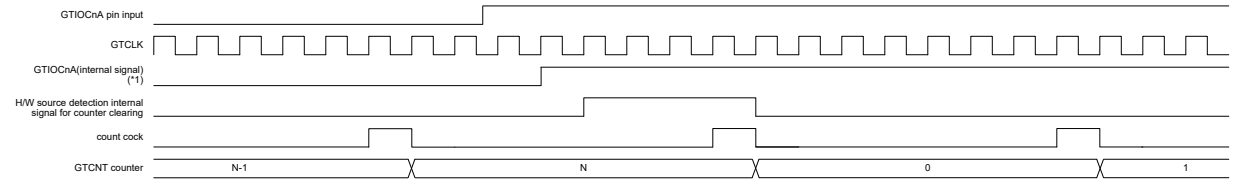


Note: POEG clock: PCLKB
Note 1. The GTETRGA internal signal reflects the signal being input to GPT when the digital noise filter for the GTETRGA pin in the POEG is not in use.

Figure 21.98 Examples of the Timing of Operations for Counter Clearing in Response to a Rising Edge of the Input on the GTETRGA Pin (During the Counting of Cycles of Clock Signal Produced by Dividing the GTCLK Frequency)

Figure 21.99 shows an example of the timing of operations to clear the counter in response to a rising edge of the input on the GTIOCnA pin when a clock signal produced by frequency-dividing the GTCLK signal is used as the counter clock for the GTCNT counter.

The GTCNT counter is cleared when counting is in progress after the GPT32 has detected the internal clearing signal.



Note 1. This is the case when the digital noise filter is not in use.

Figure 21.99 Examples of Timing of Operations for Counter Clearing in Response to a Rising Edge of the Input on the GTIOCnA Pin (During the Counting of Cycles of Clock Signal Produced by Dividing the GTCLK Frequency)

Figure 21.100 shows an example of the timing of operations to clear the counter in response to event input from ELC_GPTA when a clock signal produced by frequency-dividing the GTCLK signal is used as the counter clock for the GTCNT counter.

This is an example of operations to clear counting by the GPT321.GTCNT counter in response to a signal. An event signal is output to the ELC after matches in comparison with the GPT320.GTCCRA register. This is selected as a trigger for output to the GPT321 by the ELC as ELC_GPTA.

ELC passes the event signal output from GPT320 without delay to GPT321.

The GTCNT counter is cleared when counting is in progress after the GPT32 has detected the internal clearing signal.

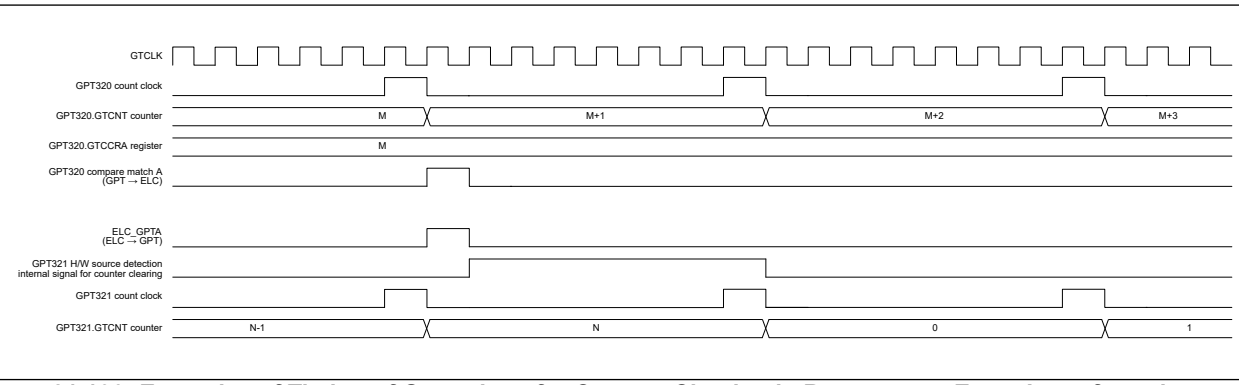
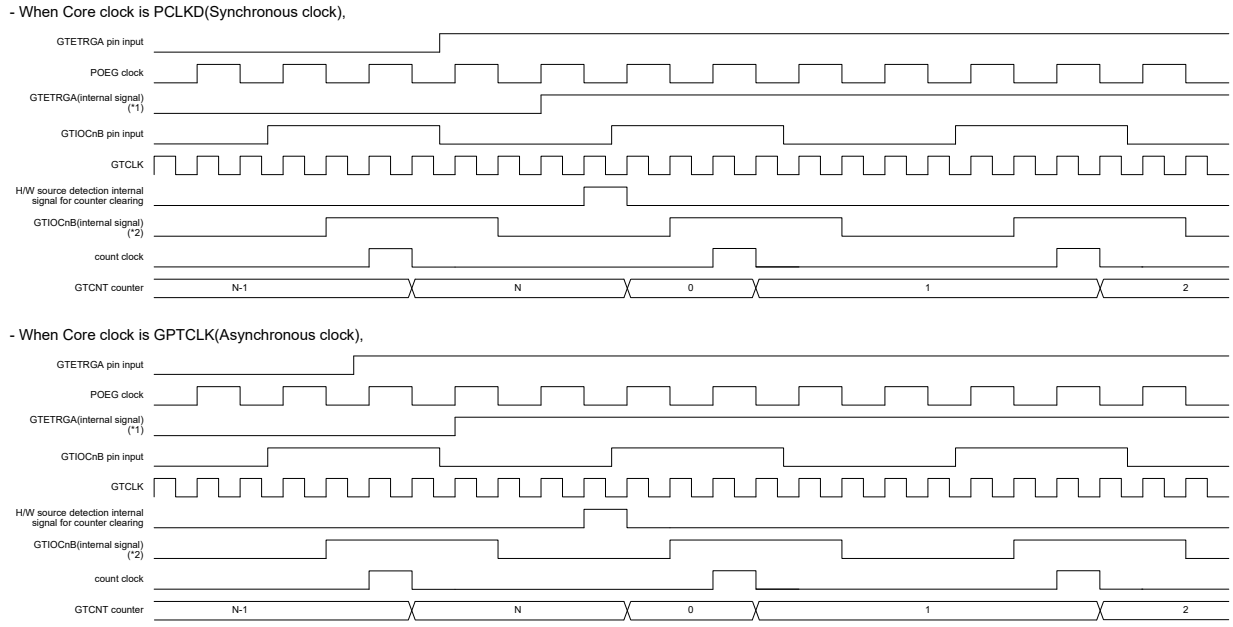


Figure 21.100 Examples of Timing of Operations for Counter Clearing in Response to Event Input from the ELC_GPTA (During the Counting of Cycles of Clock Signal Produced by Dividing the GTCLK Frequency)

Figure 21.101 shows an example of the timing of operations to clear the counter in response to a rising edge of the input on the GTETRGA pin input when counting is triggered by a hardware source.

The GTCNT counter is cleared in synchronization with GTCLK after the GPT32 has detected the internal clearing signal.

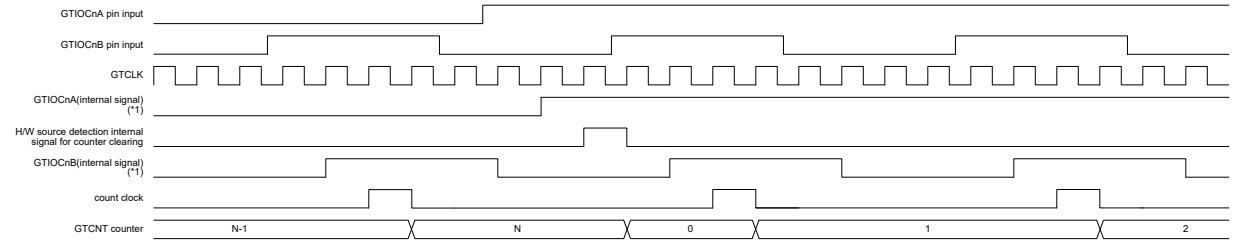


Note: POEG clock: PCLKB
Note 1. The GTETRGA internal signal reflects the signal being input to GPT when the digital noise filter for the GTETRGA pin in the POEG is not in use.
Note 2. This is the case when the digital noise filter is not in use.

Figure 21.101 Examples of the Timing of Operations for Counter Clearing in Response to a Rising Edge of the Input on the GTETRGA Pin (During Counting Triggered by Hardware Source)

Figure 21.102 shows an example of the timing of operations to clear the counter in response to a rising edge of the input on the GTIOCnA pin input when counting is triggered by a hardware source.

The GTCNT counter is cleared in synchronization with GTCLK after the GPT32 has detected the internal clearing signal.



Note 1. This is the case when the digital noise filter is not in use.

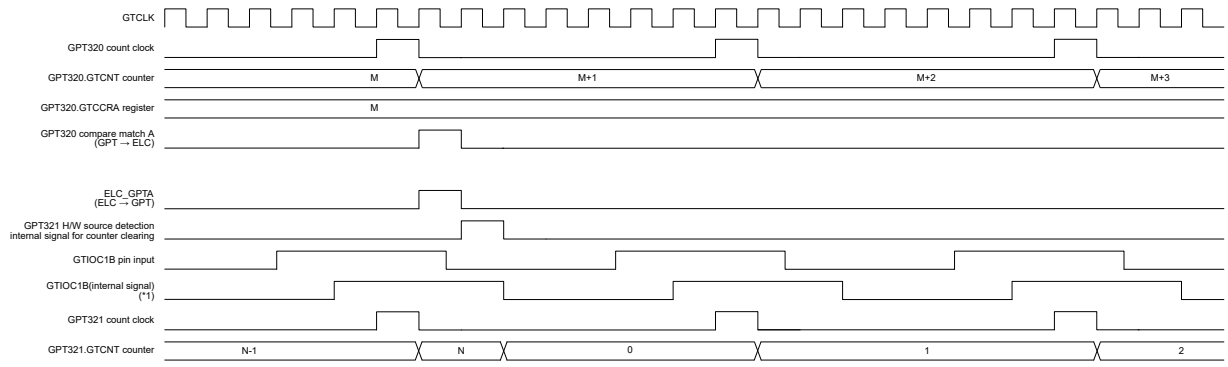
Figure 21.102 Examples of the Timing of Operations for Counter Clearing in Response to a Rising Edge of the Input on the GTIOCnA Pin (During Counting Triggered by Hardware Source)

Figure 21.103 shows an example of the timing of operations for clearing the counter in response to the input of an event signal from the ELC_GPTA when counting is triggered by a hardware source.

This is an example of operations to clear counting by the GPT321.GTCNT counter in response to a signal. An event signal is output to the ELC after matches in comparison with the GPT320.GTCCRA register. This is selected as a trigger for output to the GPT321 by the ELC as ELC_GPTA.

ELC passes the event signal output from GPT320 without delay to GPT321.

The GTCNT counter is cleared in synchronization with GTCLK after the GPT32 has detected the internal clearing signal.



Note 1. This is the case when the digital noise filter is not in use.

Figure 21.103 Examples of the Timing of Operations for Counter Clearing in Response to Event Input from the ELC_GPTA (During Counting Triggered by Hardware Source)

21.3.8 Synchronized Operation

Synchronized operation on channels such as a synchronized start, stop, and clear operation can be performed.

21.3.8.1 Synchronized Operation by Software

The GTCNT counters can be started, stopped, and cleared on multiple channels by setting the associated GTSTR, GTSTP, or GTCLR bits simultaneously to 1.

Count start with a phase difference is possible by setting the initial value in the GTCNT counter and setting the associated GTSTR bits simultaneously to 1.

When the GTCNT synchronous set/clear function is enabled on the GTCR.SSCEN bit, the GTCNT registers of the channel set in same group on the GTCR.SSCGRP [1:0] bits can be written at the same time.

The synchronous GTCNT write is invalid in complementary PWM mode.

When either the SSCE bit or the SSCD bit of the GTSECR register is set to 1, the GTCR.SSCEN bits of the channels that selected on the GTSECSR register are set to 0 or 1 and the GTCNT synchronous set/clear function of multiple channels are enabled or disabled at the same time.

Because the clock of count operation is selected by GTCR.TPCS[3:0] bits in respective channels, if the clock period of each channel that performs synchronous operation (count start/stop/clear) is different from others, the synchronous operation timings of every channels are not exact same.

Figure 21.104 shows an example of a simultaneous start, stop, and clear by software. Figure 21.105 shows an example of phase start operation by software. Figure 21.106, Figure 21.107, Figure 21.108 show an example of simultaneous start/stop/clearing with different count period.

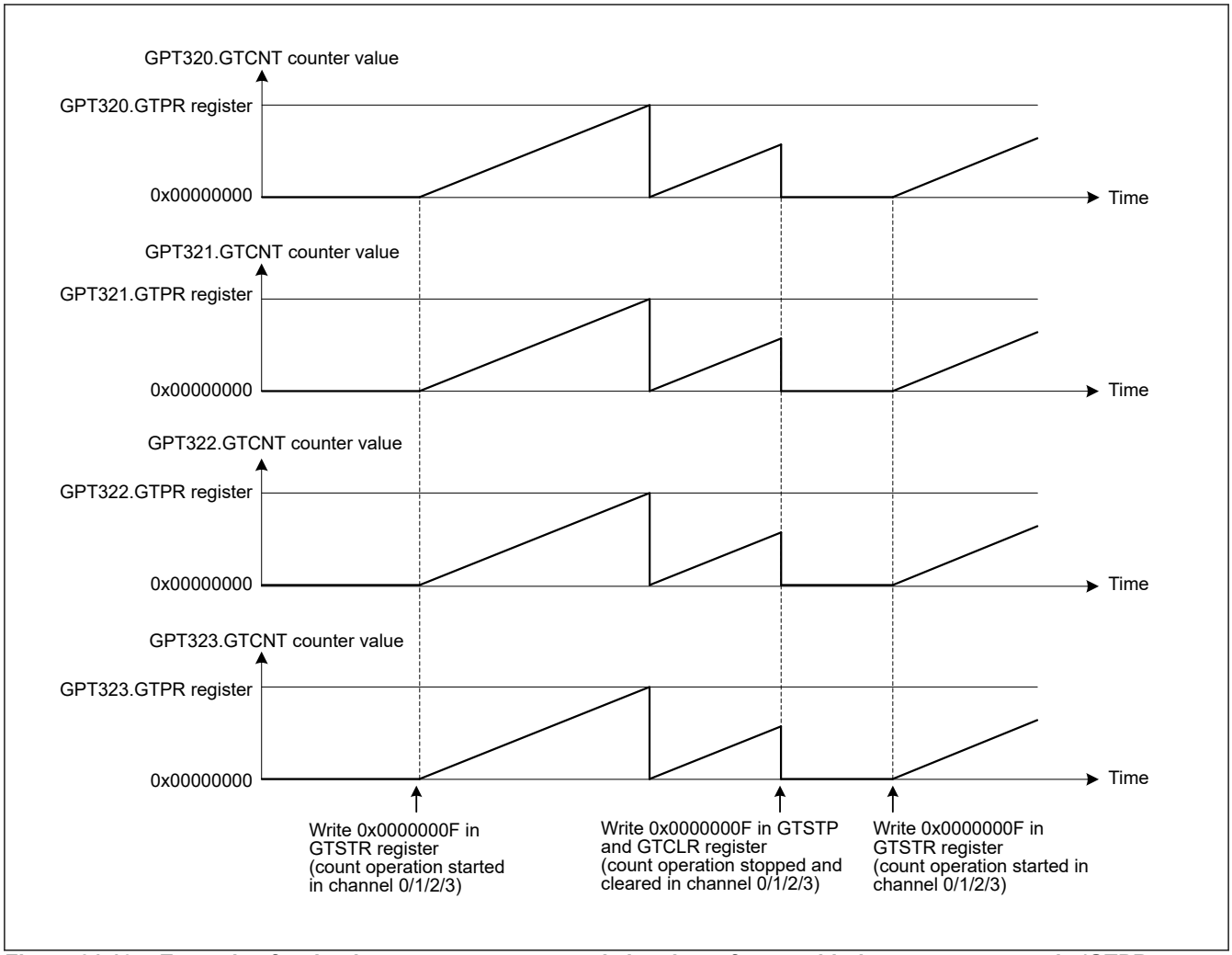


Figure 21.104 Example of a simultaneous start, stop, and clear by software with the same count cycle (GTPR register value)

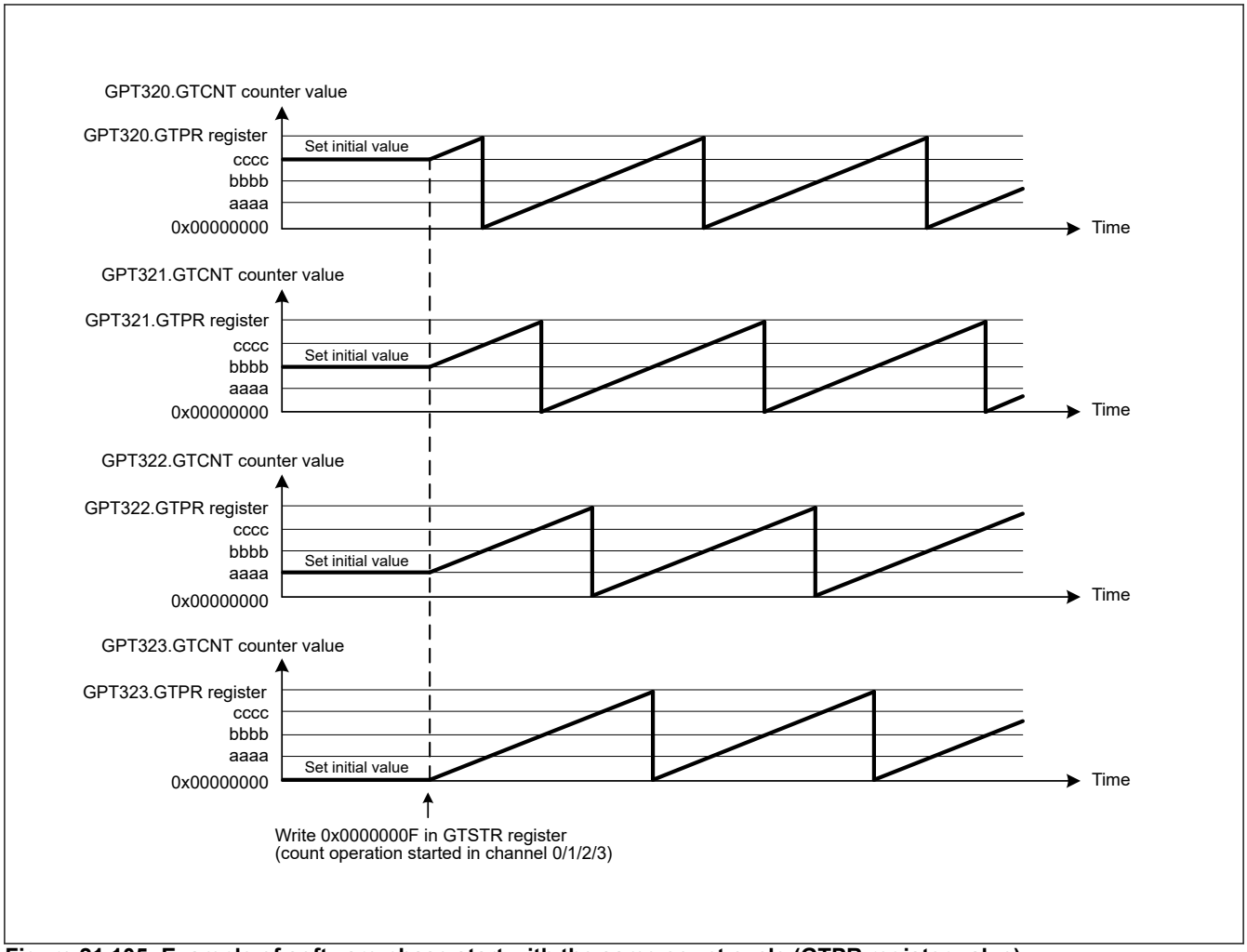


Figure 21.105 Example of software phase start with the same count cycle (GTPR register value)

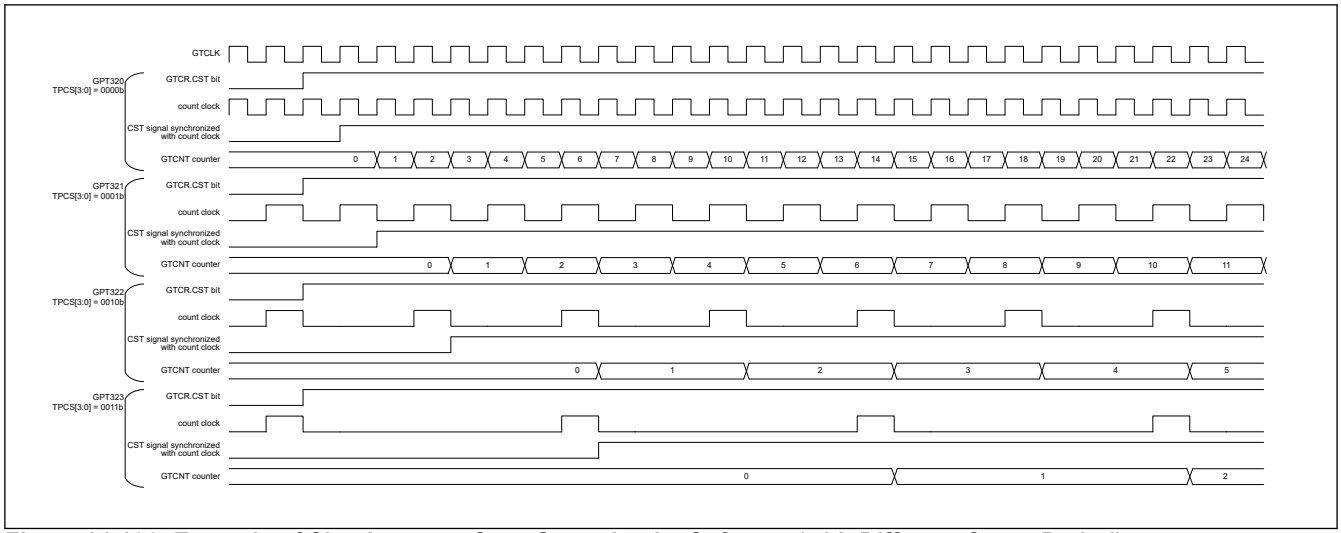


Figure 21.106 Example of Simultaneous Start Operation by Software (with Different Count Period)

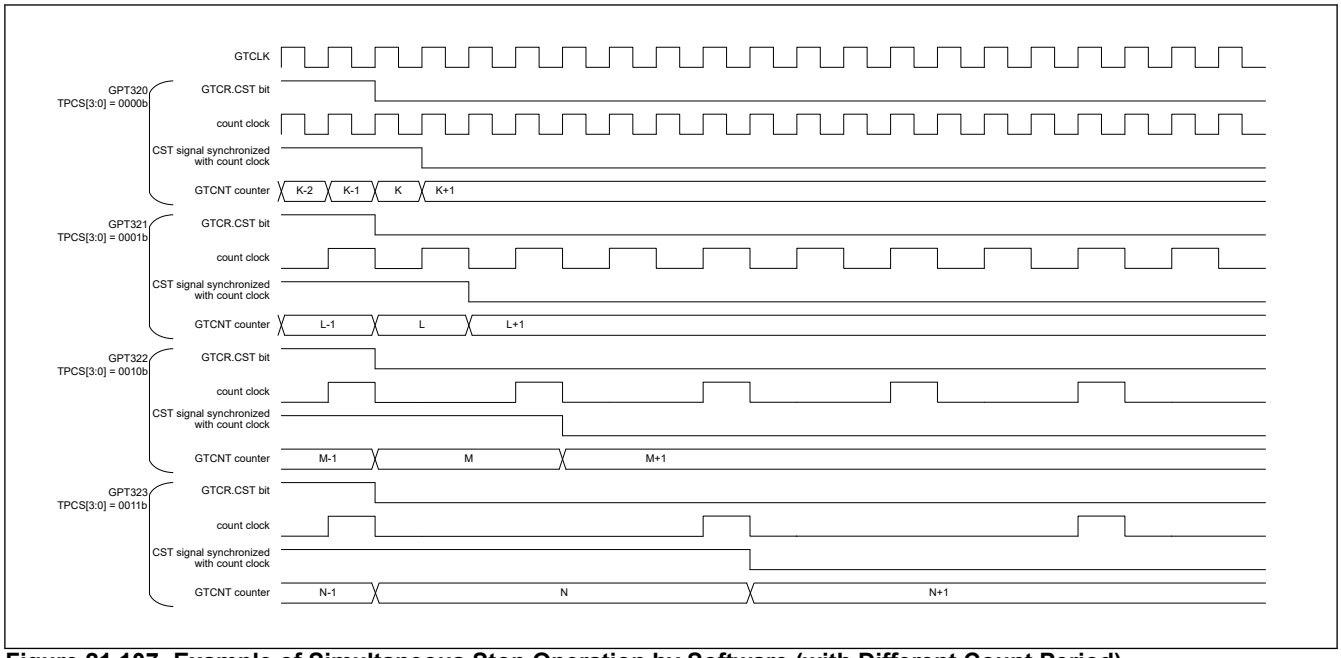


Figure 21.107 Example of Simultaneous Stop Operation by Software (with Different Count Period)

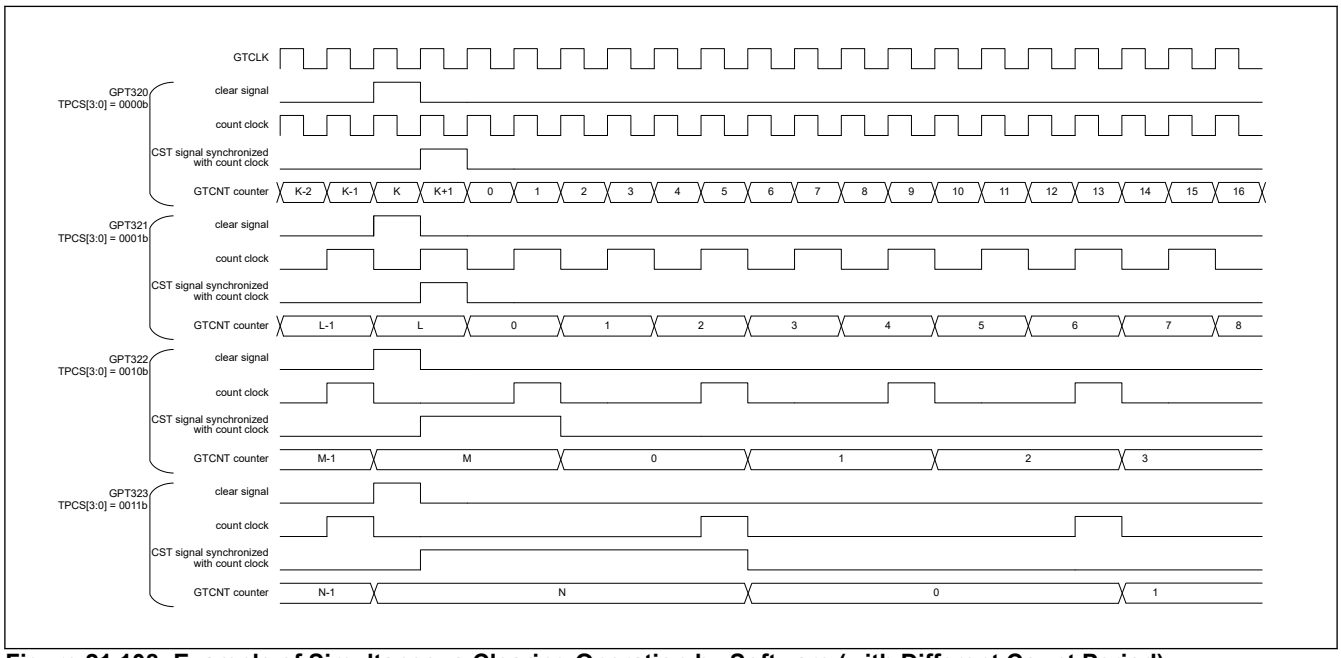
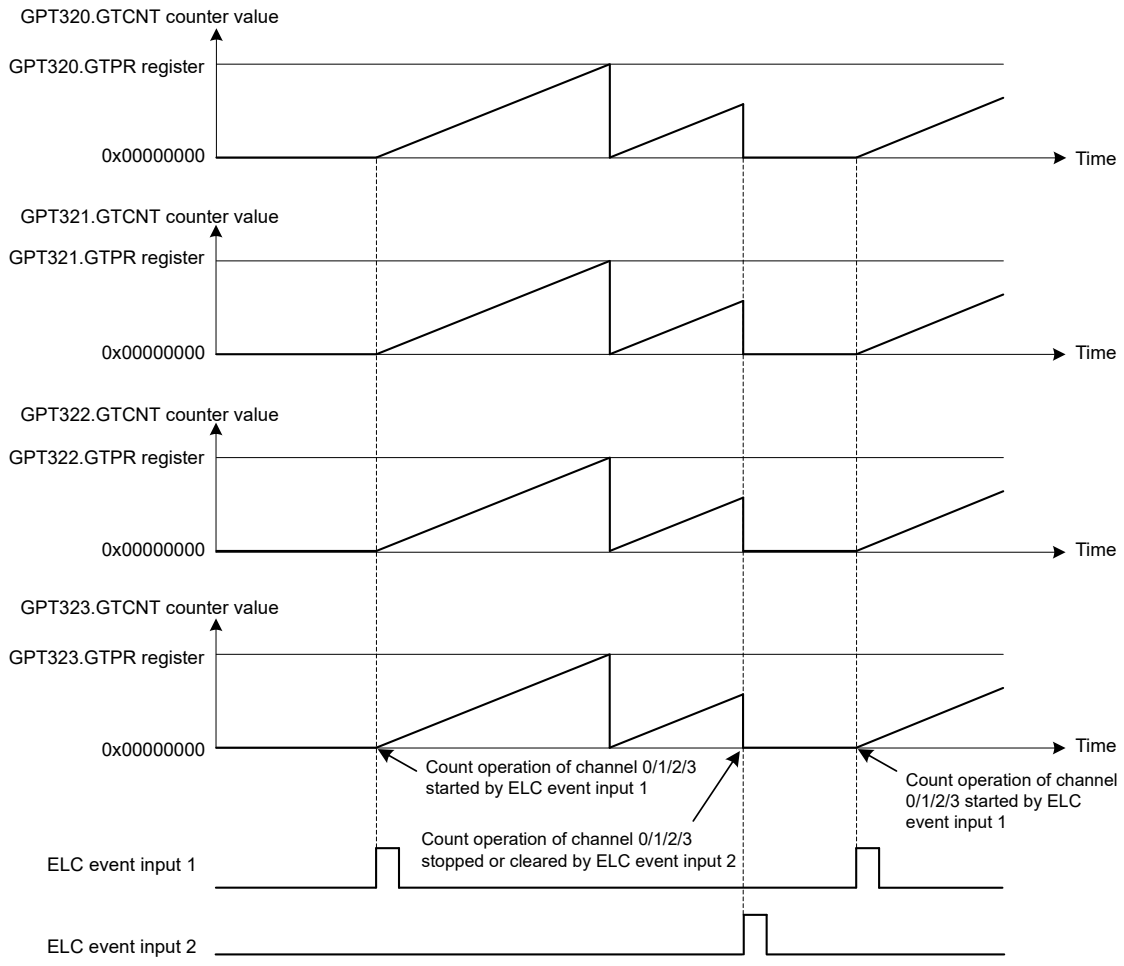


Figure 21.108 Example of Simultaneous Clearing Operation by Software (with Different Count Period)

21.3.8.2 Synchronized Operation by Hardware

The counters for multiple channels can be started, stopped, and cleared simultaneously by the following hardware sources. Hardware sources that can cause a synchronized operation are external trigger input and ELC event input. Synchronized operation through the GTIOCnA and GTIOCnB pin inputs is possible by setting an ELC event due to input capture as a hardware source (n = 0 to 9).

Figure 21.109 shows an example of a simultaneous start, stop, and clear operation by a hardware source. Table 21.49 shows the setting example.



Note: ELC event input 1: ELC_GPTA event input
ELC event input 2: ELC_GPTB event input

Figure 21.109 Example of a simultaneous start, stop, and clear by a hardware source with the same count cycle (GTPR register value)

Table 21.49 Example setting for simultaneous start by a hardware source (1 of 2)

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits or GTCR.MD[3:0] bits. In Figure 21.109, 000b or 0000b (saw-wave PWM mode 1) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 21.109, after 11b is set in the GTUDDTYC[1:0] bits, 01b is set in the GTUDDTYC[1:0] bits (up-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Set the cycle in the GTPR register.
5	Set initial value for counter	Set the initial value in the GTCNT counter. In Figure 21.109, 0x00000000 is set.
6	Set hardware count start	Select a hardware source for starting count operation with the GTSSR register, and wait for count start by the hardware source. In Figure 21.109, GTSSR.SSELCA = 1.

Table 21.49 Example setting for simultaneous start by a hardware source (2 of 2)

No.	Step Name	Description
7	Set hardware count stop	Select a hardware source for stopping count operation with the GTPSR register, and wait for count stop by the hardware source. In Figure 21.109, GTPSR.PSELCB = 1.
8	Set hardware count clear	Select a hardware source for clearing count operation with the GTCSR register, and wait for count clear by the hardware source. In Figure 21.109, GTCSR.CSELCB = 1.
9	Set hardware source operation	Set operation of the hardware source selected in the GTSSR, GTPSR, or GTCSR registers, and start, stop, or clear counting. In Figure 21.109, ELC_GPTA input and ELC_GPTB input are set.

21.3.8.3 Synchronous Clear Operation by Inter Channel Cooperation

The counters of other channels can be cleared at the same time with the counter clear of some channel caused by the compare match, input capture, saw wave up-count overflow, saw wave down-count underflow, and the GTIOCnA / GTIOCnB pin input selected in the GTCSR register.

The counter clear caused by GTCLR register, external trigger input and the ELC input can perform as synchronous clear when the same counter clear factor is set to target channels of synchronous clear. Therefore, these factors are not prepared for the synchronous clear operation by inter channel cooperation.

Set the channel that generates the synchronization clear source and the channel that is synchronized clear in the same synchronous set/clear group with the GTCR.SSCGRP[1:0] bits.

Similar to the synchronous operation in section 21.3.8.1. Synchronized Operation by Software, if the count clock selected by the GTCR.TPCS [3:0] bits is different for each channel, the synchronous clear operation cannot be performed at exact same timing. Similar to the example of simultaneous clearing operation by software in Figure 21.108, if the count clock is different for each channel, the synchronous clear factor is synchronized with the count clock of each channel before counter clearing.

When either the SSCE bit or the SSCD bit of the GTSECR register is set to 1, the GTCR.SSCEN bits of the channels that selected on the GTSECSR register are set to 0 or 1 and the GTCNT synchronous set/clear function of multiple channels are enabled or disabled at the same time.

Figure 21.110 shows an example of synchronous clear operation by inter channel cooperation and Table 21.50 shows an example for setting synchronous clear operation by inter channel cooperation.

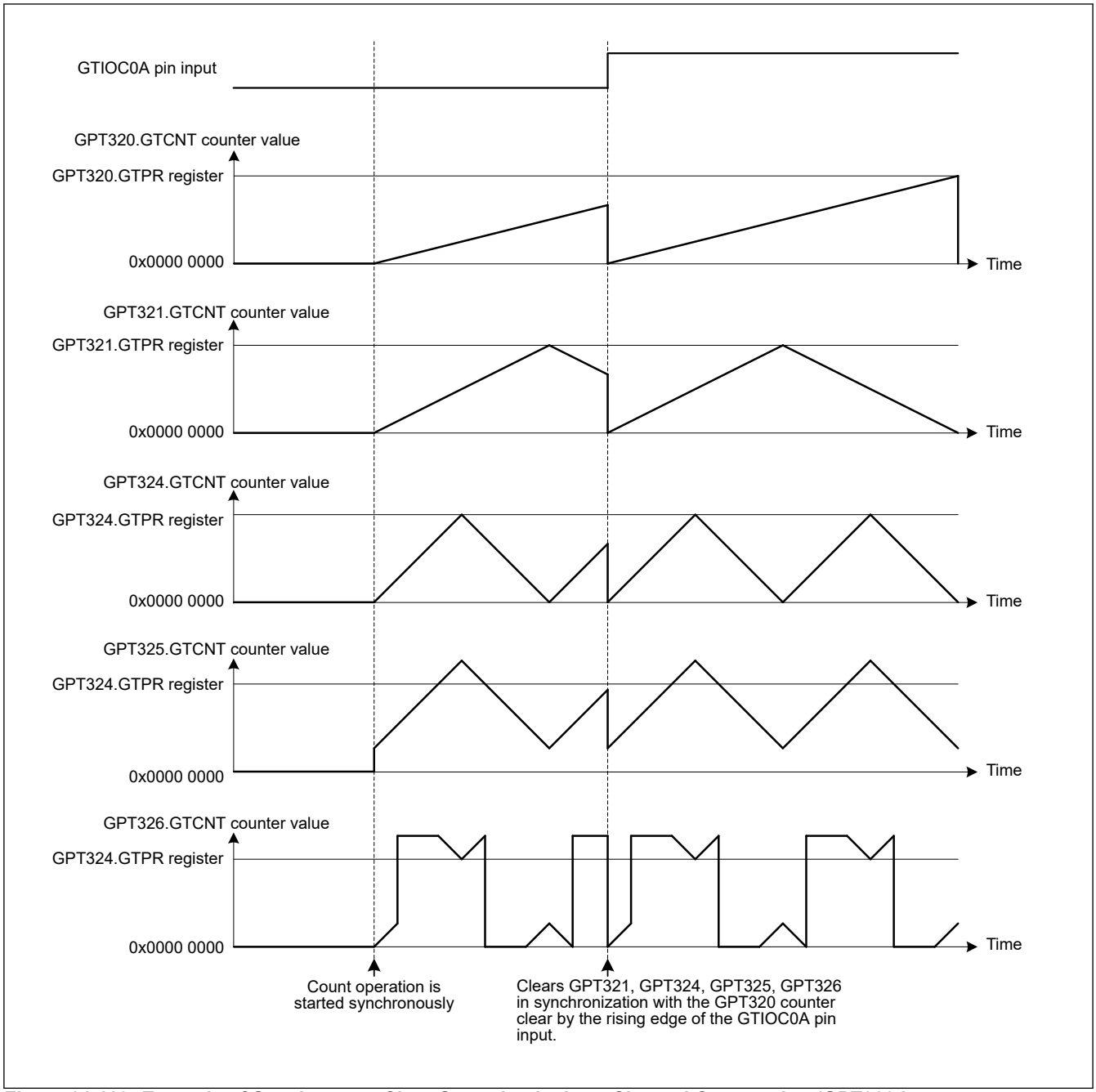


Figure 21.110 Example of Synchronous Clear Operation by Inter Channel Cooperation (GPT320 is saw-wave and counter is cleared by the rising edge of the GTIOC0A, GPT321 is triangle wave, GPT324,5,6 are complementary PWM mode. GPT320,1,4,5,6 are the same synchronous set/clear group)

Table 21.50 Example for Setting Synchronous Clear Operation by Inter Channel Cooperation (1 of 2)

No.	Step Name	Description
1	Set operating mode	Set the operating mode with GTCR.MD[2:0] or GTCR.MD[3:0]. In Figure 21.110, 000b (saw-wave PWM mode 1) is set to GPT320. 100b (triangle-wave PWM mode 1) is set to GPT321. 1100b (complementary PWM mode 1) is set to GPT324(GPT325,6).
2	Set count direction	Select the count direction (up or down) with GTUDDTYC. In Figure 21.110, for GPT320 and GPT321, lower 2 bits of GTUDDTYC is set to 11b, and then lower 2 bits of GTUDDTYC is set to 01b (up-counting).
3	Select count clock	Select the count clock with GTCR.TPCS[3:0] bits of the corresponding channel.
4	Set cycle	Set the cycle in GTPR of the corresponding channel. When complementary PWM mode, set the cycle in GTPR of the master channel.

Table 21.50 Example for Setting Synchronous Clear Operation by Inter Channel Cooperation (2 of 2)

No.	Step Name	Description
5	Set initial value for counter	Set the initial value in the GTCNT counter of the corresponding channel.
6	Inter channel cooperation synchronous clear setting (Source channel)	Set the GTINTAD register and GTCR.SCGTIOC bit in the source channel of inter channel cooperation synchronous clear to enable synchronous clear. When complementary PWM mode, set GTINTAD of the master channel. In Figure 21.110, GPT320.GTCR.SCGTIOC bit is 1.
7	Inter channel cooperation synchronous clear setting (Cleared channels)	Set GTCSR.CSCMSC[2:0] bits in the cleared channels of inter channel cooperation synchronous clear to select the counter clear by synchronous counter clearing group. When complementary PWM mode, set GTCSR of the master channel. In Figure 21.110, GTCSR.CSCMSC[2:0] bits of GPT321 and GPT324 is 111b.
8	Set group of inter channel cooperation synchronous clear	Set the same value to GTCR.SSCGRP[1:0] bits in the source channel and cleared channels of inter channel cooperation synchronous clear and set them in the same synchronous set/clear group. When complementary PWM mode, set GTCR of the master channel.
9	Enable inter channel cooperation synchronous clear	Set GTCR.SSCEN bits in the source channel and cleared channels of inter channel cooperation synchronous clear to enable synchronous clear.

21.3.9 PWM Output Operation Examples

(1) Synchronized PWM output

The GPT outputs 10 × 2 phases of linked PWM waveforms for a maximum of GPT × 10 channels.

Figure 21.111 shows an example in which four channels perform synchronized operation in saw-wave PWM mode 1 and eight phases of PWM waveforms are output. The GTIOCnA is set so that it outputs low as the initial value, high at a GTCCRA compare match, and low at the cycle end. The GTIOCnB is set so that it outputs low as the initial value, high at a GTCCRB compare match, and low at the cycle end.

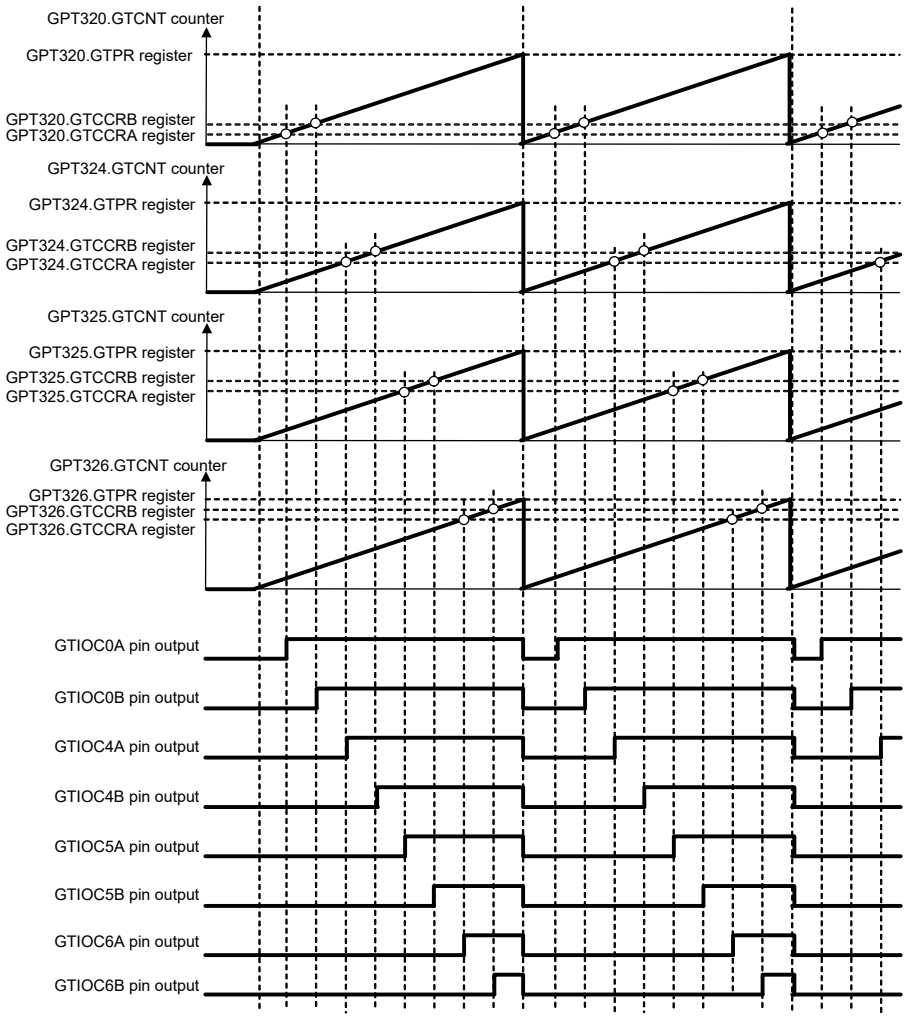


Figure 21.111 Example of synchronized PWM output

(2) 3-phase saw-wave complementary PWM output

Figure 21.112 shows an example in which three channels perform synchronized operation in saw-wave PWM mode 1 and 3-phase complementary PWM waveforms are output. The GTIOCnA pin is set so that it outputs low as the initial value, high at a GTCCRA compare match, and low at the cycle end. The GTIOCnB pin is set so that it outputs high as the initial value, low at a GTCCRB compare match, and high at the cycle end.

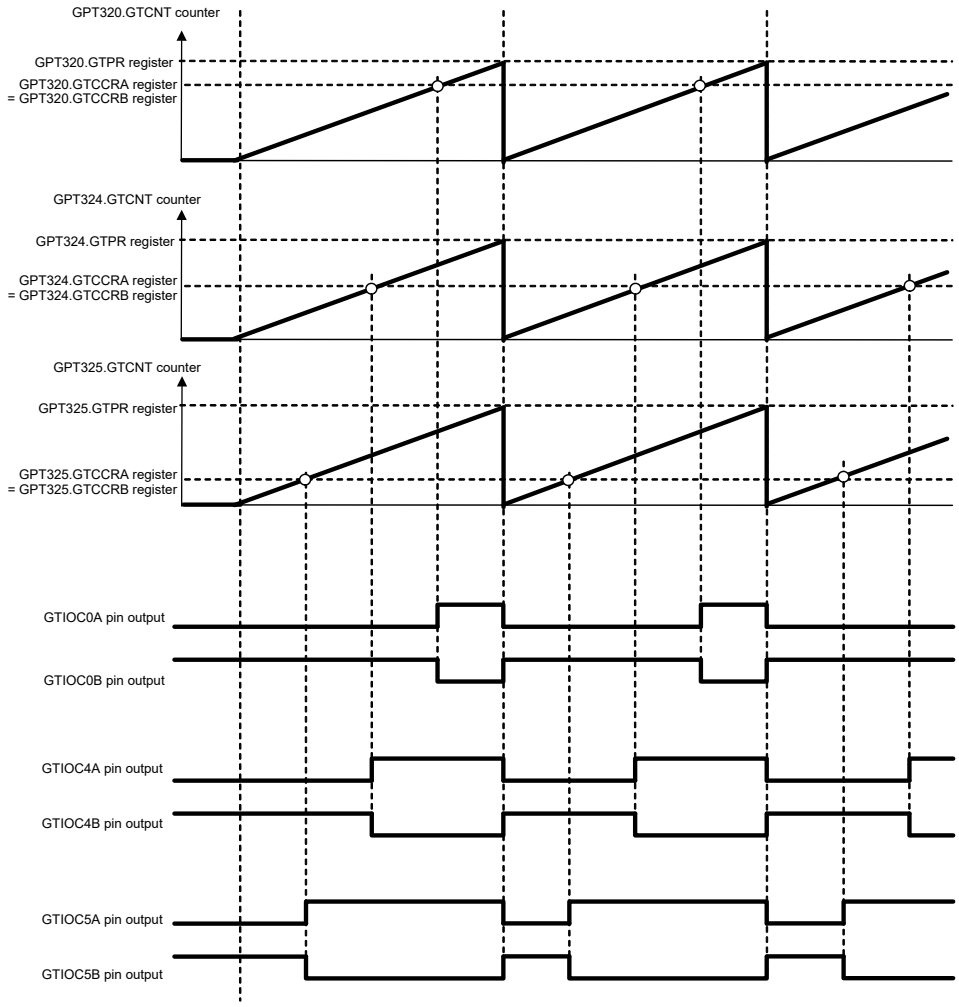


Figure 21.112 Example of 3-phase saw-wave complementary PWM output

(3) 3-phase saw-wave complementary PWM output with automatic dead time setting

Figure 21.113 shows an example in which three channels perform synchronized operation in saw-wave one-shot pulse mode with automatic dead time setting and 3-phase complementary PWM waveforms are output. The GTIOCnA pin is set so that it outputs low as the initial value, toggles the output at a GTCCRA compare match, and retains the output at the cycle end. The GTIOCnB pin is set so that it outputs high as the initial value, toggles the output at a GTIOCnB compare match, and retains the output at the cycle end.

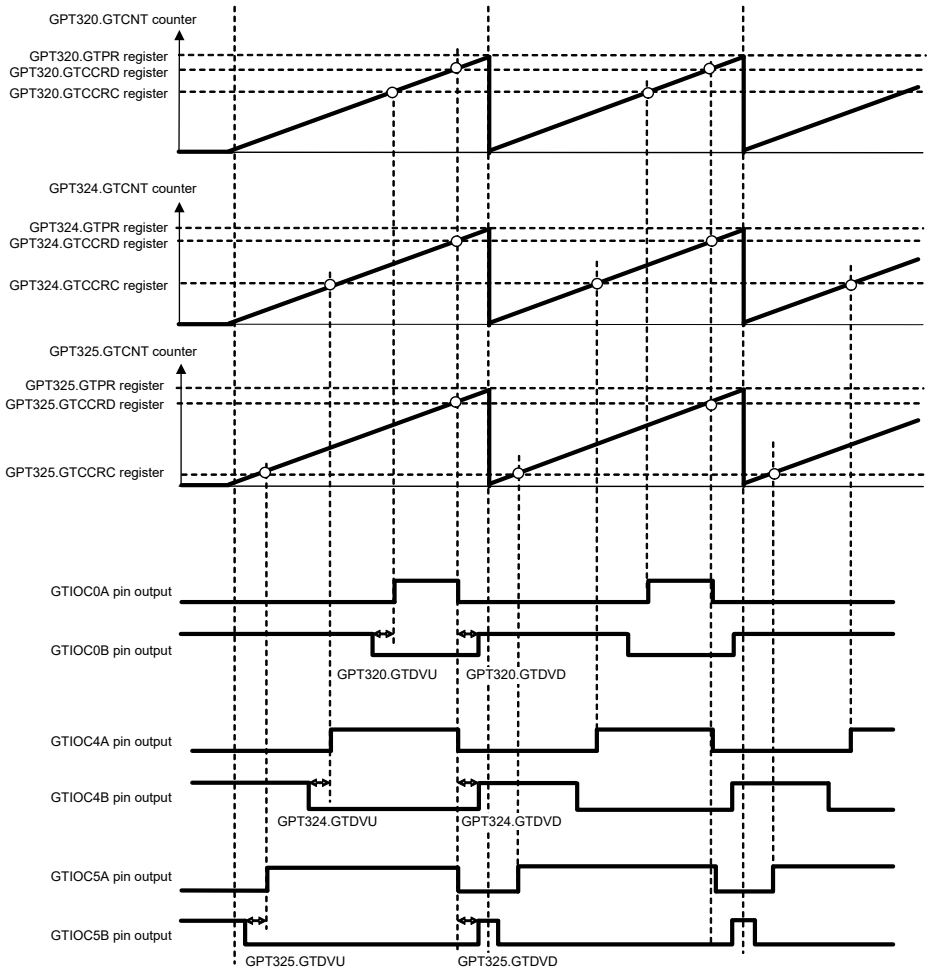


Figure 21.113 Example of 3-phase saw-wave complementary PWM output with automatic dead time setting

(4) 3-phase triangle-wave complementary PWM output

Figure 21.114 shows an example in which three channels perform synchronized operation in triangle-wave PWM mode 1 and 3-phase complementary PWM waveforms are output. The GTIOCnA pin is set so that it outputs low as the initial value, toggles the output at a GTCCRA compare match, and retains the output at the cycle end. The GTIOCnB pin is set so that it outputs high as the initial value, toggles the output at a GTCCRB compare match, and retains the output at the cycle end.

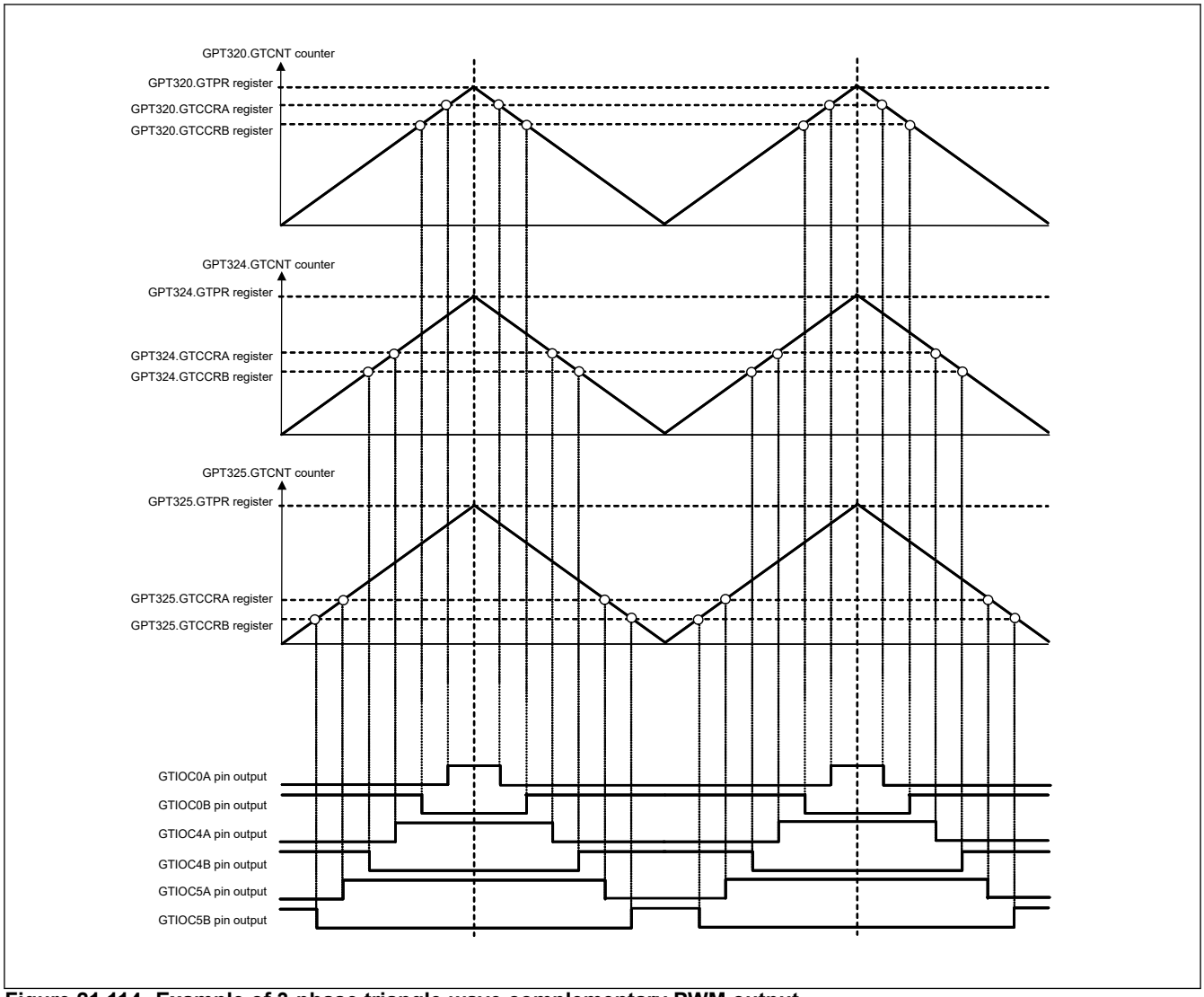


Figure 21.114 Example of 3-phase triangle-wave complementary PWM output

(5) 3-phase triangle-wave complementary PWM output with automatic dead time setting

Figure 21.115 shows an example in which three channels perform synchronized operation in triangle-wave PWM mode 1 with automatic dead time setting and 3-phase complementary PWM waveforms are output. The GTIOCnA pin is set so that it outputs low as the initial value, toggles the output at a GTCCRA compare match, and retains the output at the cycle end. The GTIOCnB pin is set so that it outputs high as the initial value, toggles the output at a GTCCRB compare match, and retains the output at the cycle end.

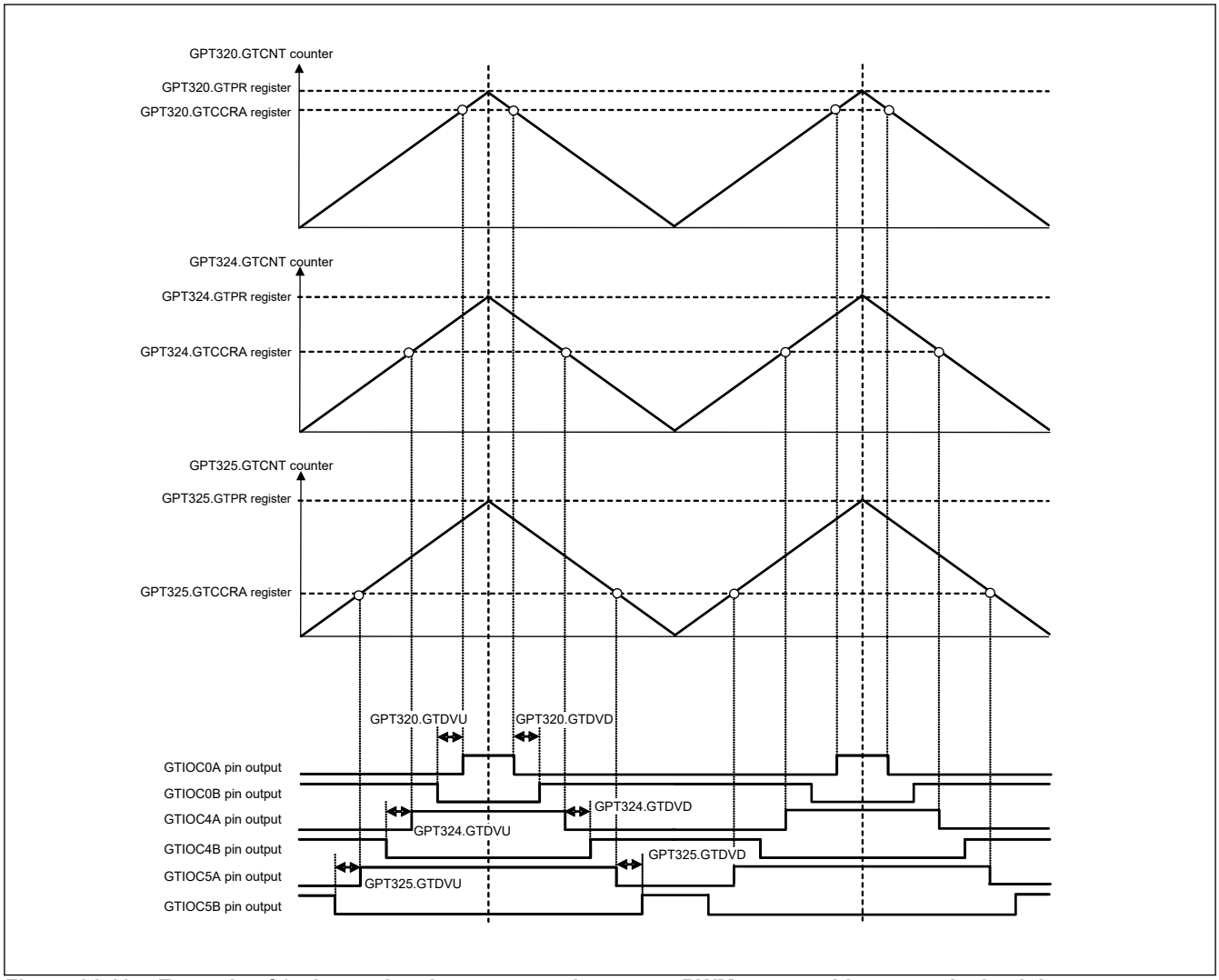


Figure 21.115 Example of 3-phase triangle-wave complementary PWM output with automatic dead time setting

(6) 3-phase asymmetric triangle-wave complementary PWM output with automatic dead time setting

Figure 21.116 shows an example in which three channels perform synchronized operation in triangle-wave PWM mode 3 with automatic dead time setting and 3-phase complementary PWM waveforms are output. The GTIOCnA is set so that it outputs low as the initial value, toggles the output at a GTCCRA compare match, and retains the output at the cycle end. The GTIOCnB is set so that it outputs high as the initial value, toggles the output at a GTCCRB compare match, and retains the output at the cycle end.

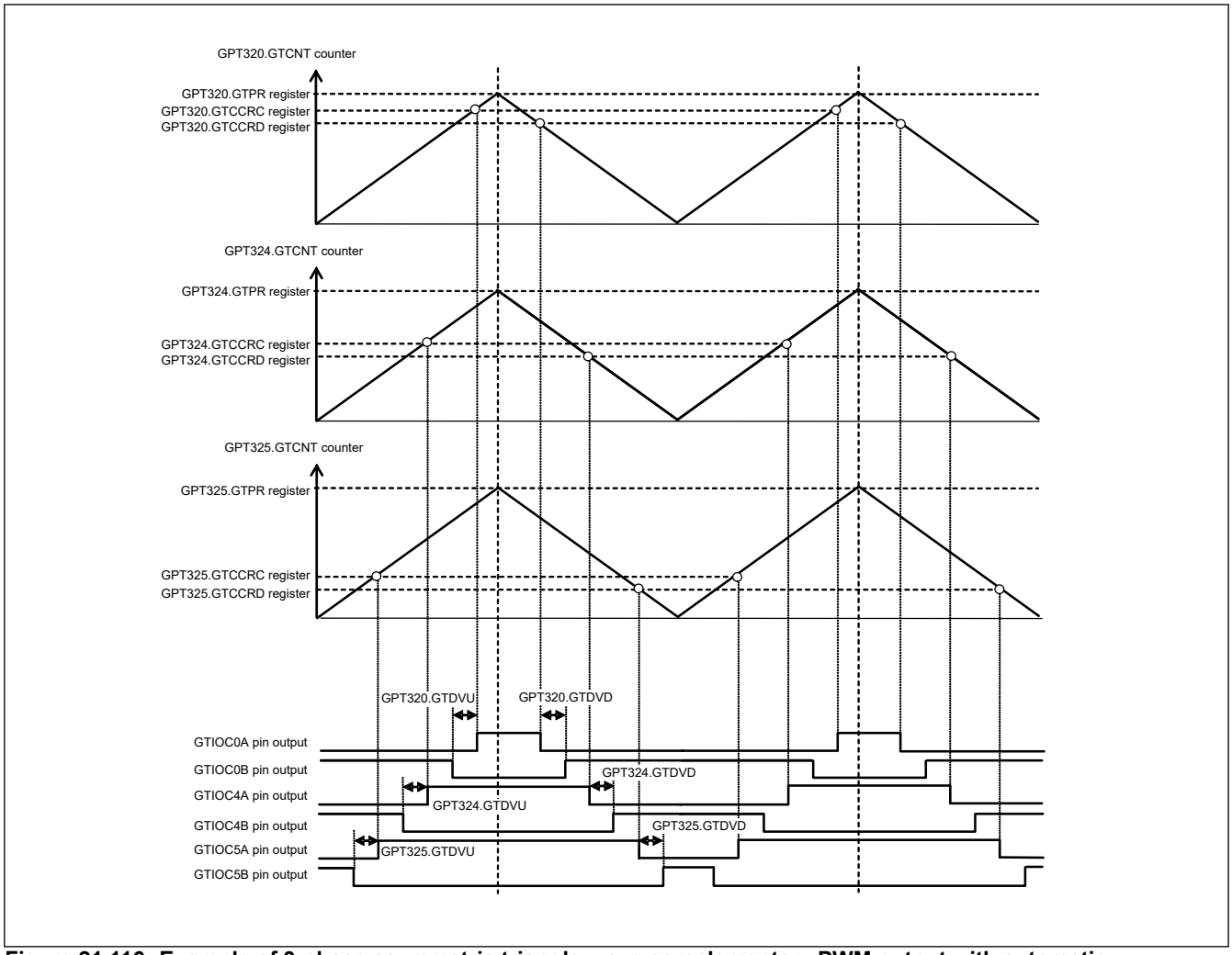


Figure 21.116 Example of 3-phase asymmetric triangle-wave complementary PWM output with automatic dead time setting

21.3.10 Period Count Function

By setting the GTPC register, the end of period can be counted.

The number of period to be counted should be set into the GTPC.PCNT counter when the GTPC.PCEN bit is 0. When the PCEN bit is 1, the PCNT counter can be read, but writing is disabled. When the PCEN bit is 1, down-counting is performed at the end of period. When the PCNT counter is 1 at the end of period, it becomes 0 and counting is stopped to finish the period count function. At that time, the GTST.PCF flag is set, and the period count function finish interrupt request GPTn_PC is generated. When the GTPC.ASTP bit is 1, the GTCNT counter is also stopped at the same time that the period count function is finished.

When the GTCNT counter is stopped while period count function is enabled, the PCNT counter keeps its value. When the GTCNT counter restarts counting and the PCEN bit is 1, the PCNT counter restarts down-counting from the hold value.

When the PCEN bit is changed from 0 to 1 while the PCNT counter is 0 and the ASTP bit is 1, the GTCNT counter is stopped at the count clock immediately after that.

When either GTSECR.SPCE bit or GTSECR.SPCD bit is set to 1, the PCEN bit in the channels set to 1 by the GTSECSR register is simultaneously set the value to enable or disable the period count function for multiple channels.

Figure 21.117 and Figure 21.118 show examples of PWM cycle count function.

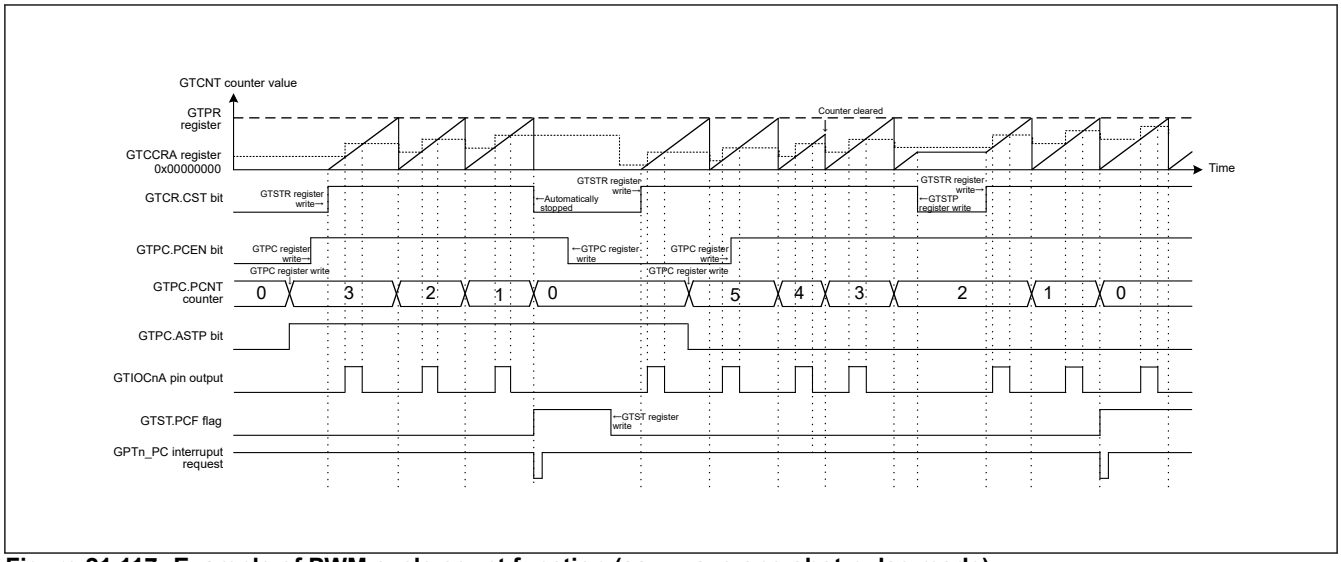
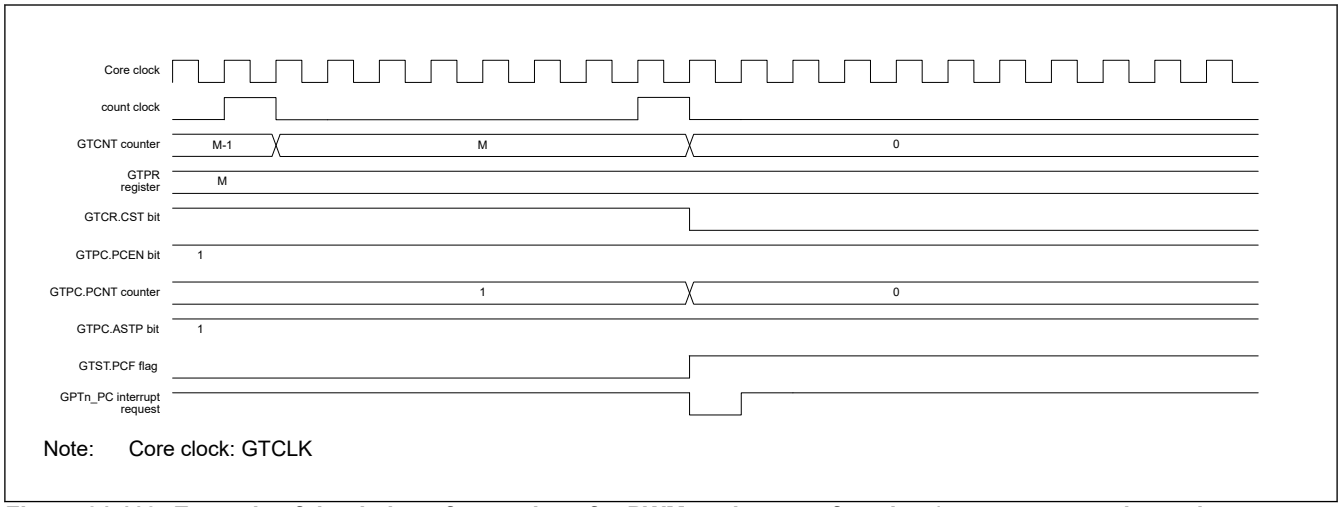


Figure 21.117 Example of PWM cycle count function (saw-wave one-shot pulse mode)



Note: Core clock: GTCLK

Figure 21.118 Example of the timing of operations for PWM cycle count function (saw-wave one-shot pulse mode, up-counting)

21.3.11 Phase Counting Function

The phase difference between the GTIOCnA and GTIOCnB pin (n = 0 to 3) inputs is detected and the associated GTCNT counts up or counts down. The detectable phase difference is available in any combination with the relationship between the edge and the level of GTIOCnA and GTIOCnB pin inputs being set in the GTUPSR and GTDNSR registers. For details on count operation, see section 21.3.1.1. Counter operation.

Figure 21.119 to Figure 21.128 show an example of phase counting modes 1 to 5 operation when the GTIOCnA, GTIOCnB pins are used. Table 21.51 to Table 21.60 show conditions of up-counting or down-counting and list settings for the GTUPSR and GTDNSR registers which is corresponding to Figure 21.119 to Figure 21.128.

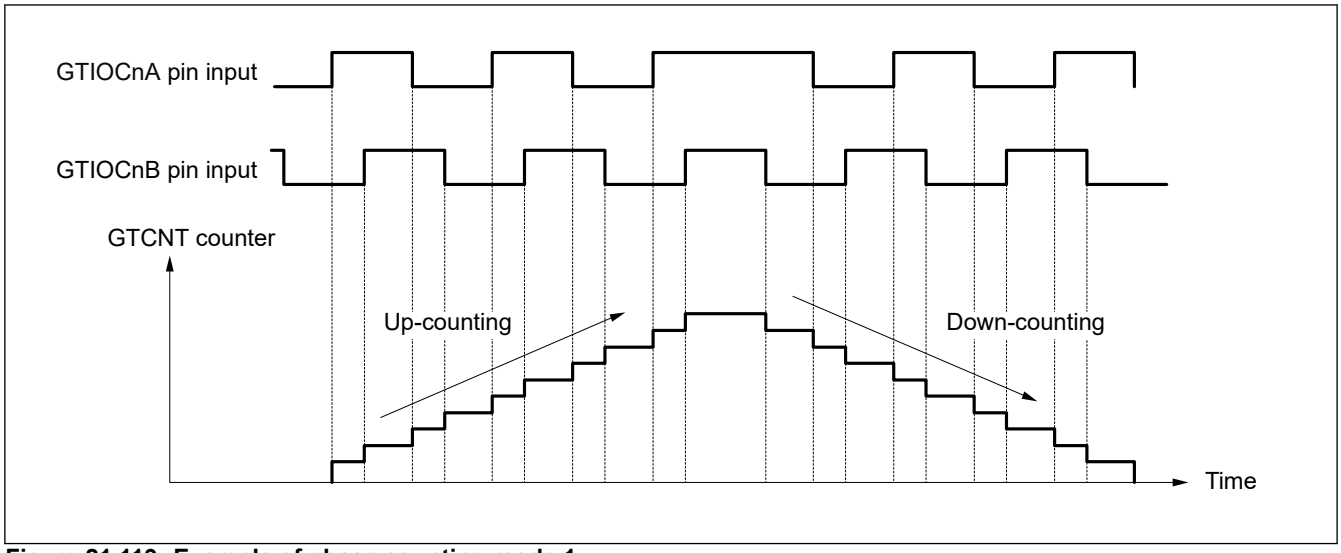


Figure 21.119 Example of phase counting mode 1

Table 21.51 Conditions of up-counting/down-counting in phase counting mode 1

: Rising edge

: Falling edge

GTIOCnA pin input	GTIOCnB pin input	Operation	Register setting
High	↑	Up-counting	GTUPSR = 0x00006900 GTDNSR = 0x00009600
Low	↓		
↑	Low		
↓	High		
High	↓	Down-counting	
Low	↑		
↑	High		
↓	Low		

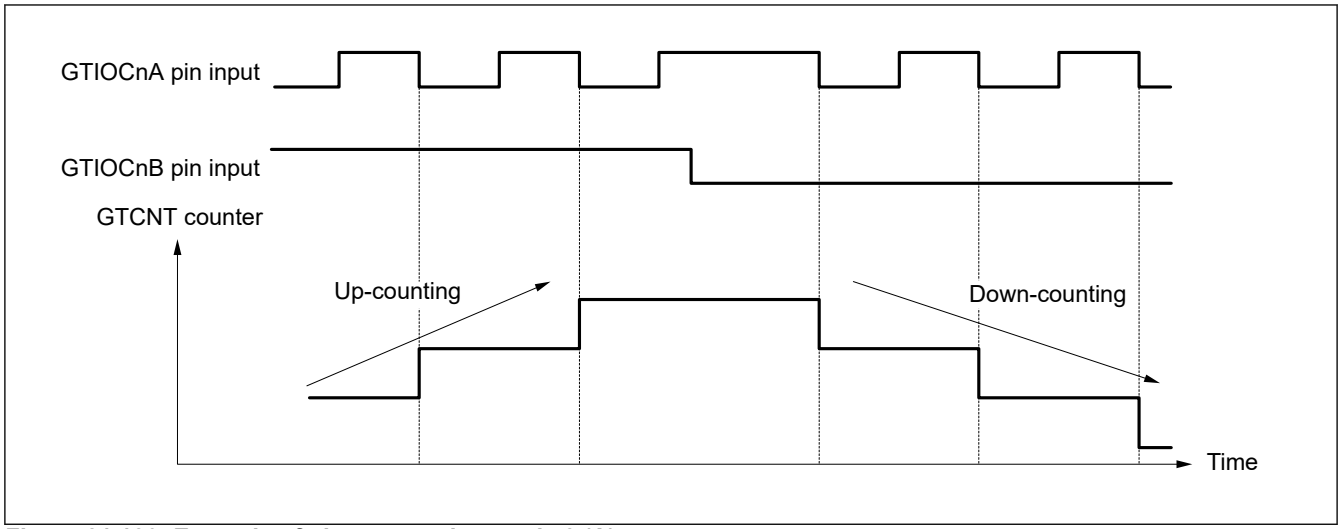


Figure 21.120 Example of phase counting mode 2 (A)

Table 21.52 Conditions of up-counting/down-counting in phase counting mode 2 (A)

: Rising edge

: Falling edge

GTIOCnA pin input	GTIOCnB pin input	Operation	Register setting
High	↑	Not counting	GTUPSR = 0x00000800 GTDNSR = 0x00000400
Low	↓		
↑	Low		
↓	High	Up-counting	
High	↓	Not counting	
Low	↑		
↑	High		
↓	Low	Down-counting	

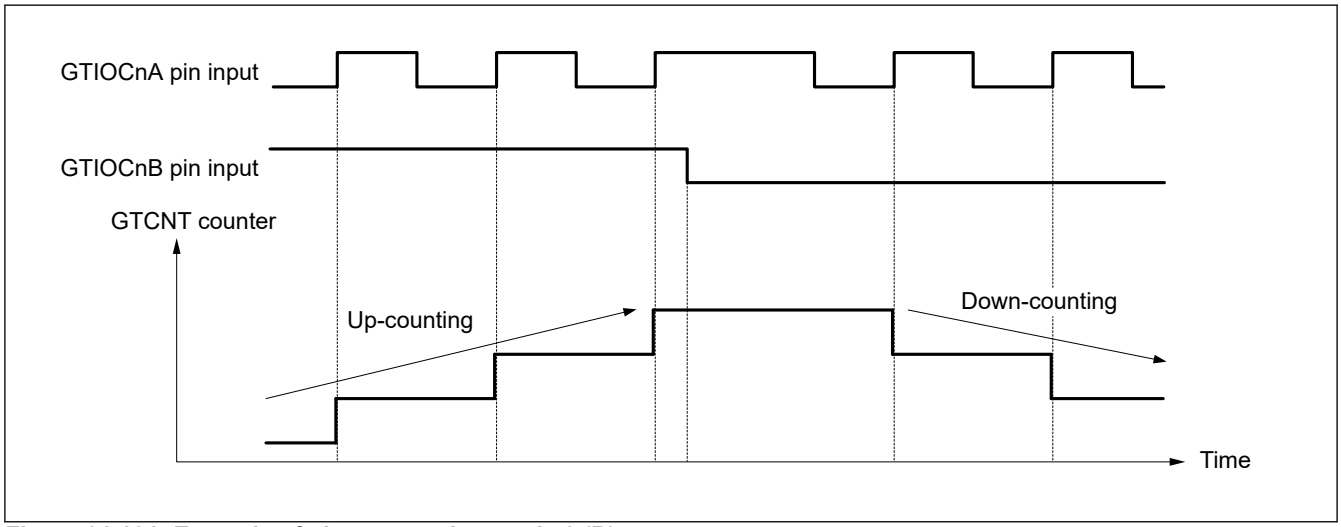


Figure 21.121 Example of phase counting mode 2 (B)

Table 21.53 Conditions of up-counting/down-counting in phase counting mode 2 (B)

: Rising edge

: Falling edge

GTIOCnA pin input	GTIOCnB pin input	Operation	Register setting
High		Not counting	GTUPSR = 0x00000200 GTDNSR = 0x00000100
Low			
	Low	Down-counting	
	High	Not counting	
High			
Low			
	High	Up-counting	
	Low	Not counting	

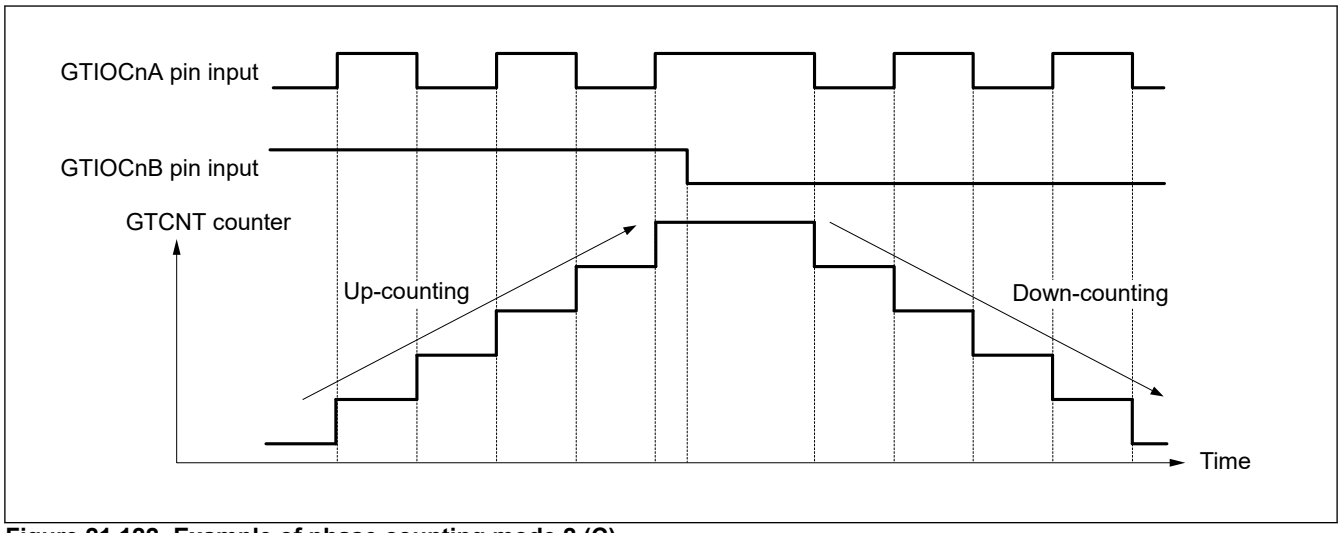


Figure 21.122 Example of phase counting mode 2 (C)

Table 21.54 Conditions of up-counting/down-counting in phase counting mode 2 (C)

: Rising edge

: Falling edge

GTIOCnA pin input	GTIOCnB pin input	Operation	Register setting
High		Not counting	GTUPSR = 0x00000A00 GTDNSR = 0x00000500
Low			
	Low	Down-counting	
	High	Up-counting	
High		Not counting	
Low			
	High	Up-counting	
	Low	Down-counting	

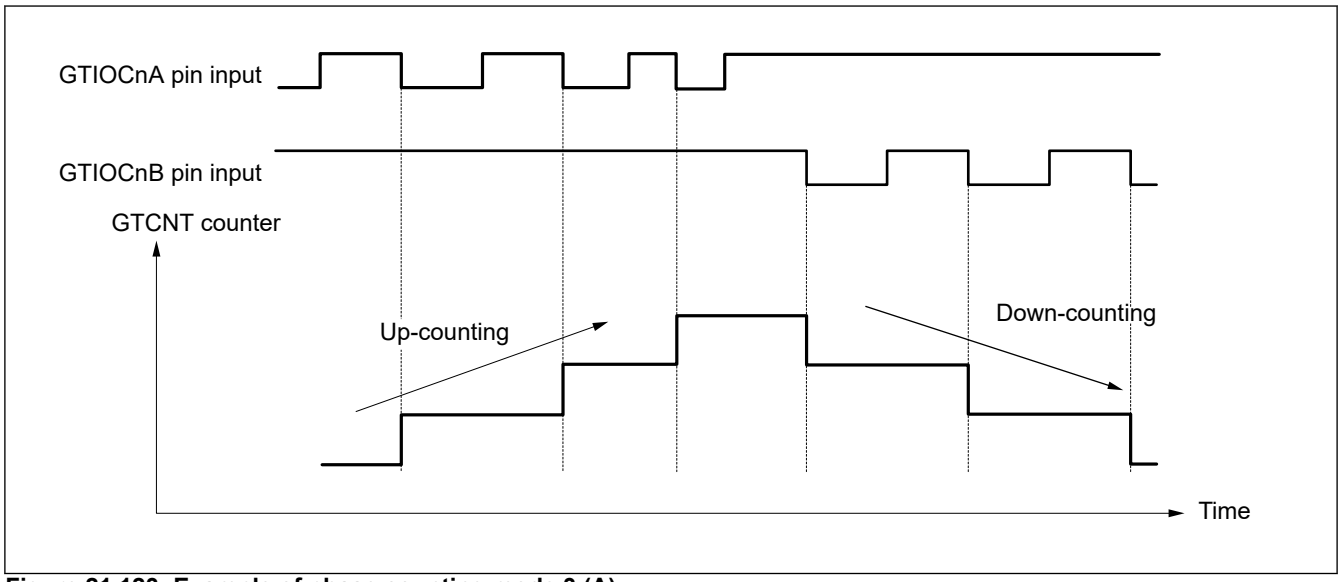


Figure 21.123 Example of phase counting mode 3 (A)

Table 21.55 Conditions of up-counting/down-counting in phase counting mode 3 (A)

: Rising edge

: Falling edge

GTIOCnA pin input	GTIOCnB pin input	Operation	Register setting
High		Not counting	GTUPSR = 0x00000800 GTDNSR = 0x00008000
Low			
	Low		
	High	Up-counting	
High		Down-counting	
Low		Not counting	
	High		
	Low		

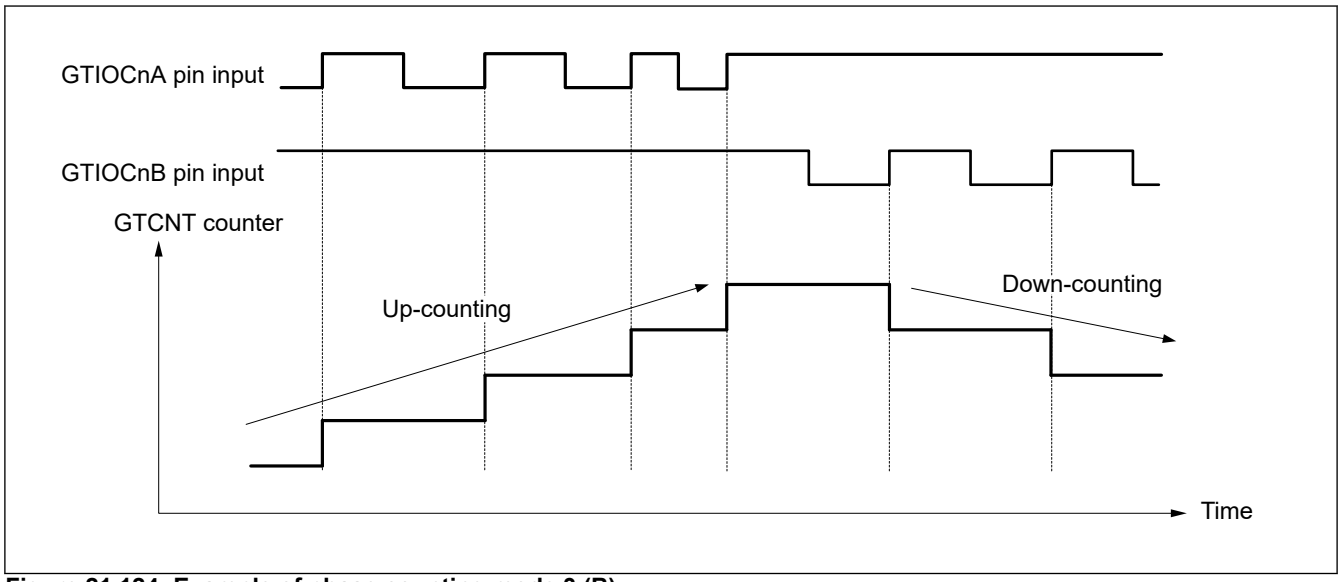


Figure 21.124 Example of phase counting mode 3 (B)

Table 21.56 Conditions of up-counting/down-counting in phase counting mode 3 (B)

: Rising edge

: Falling edge

GTIOCnA pin input	GTIOCnB pin input	Operation	Register setting
High	↑	Down-counting	GTUPSR = 0x00000200 GTDNSR = 0x00002000
Low	↓	Not counting	
↑	Low		
↓	High		
High	↓		
Low	↑		
↑	High	Up-counting	
↓	Low	Not counting	

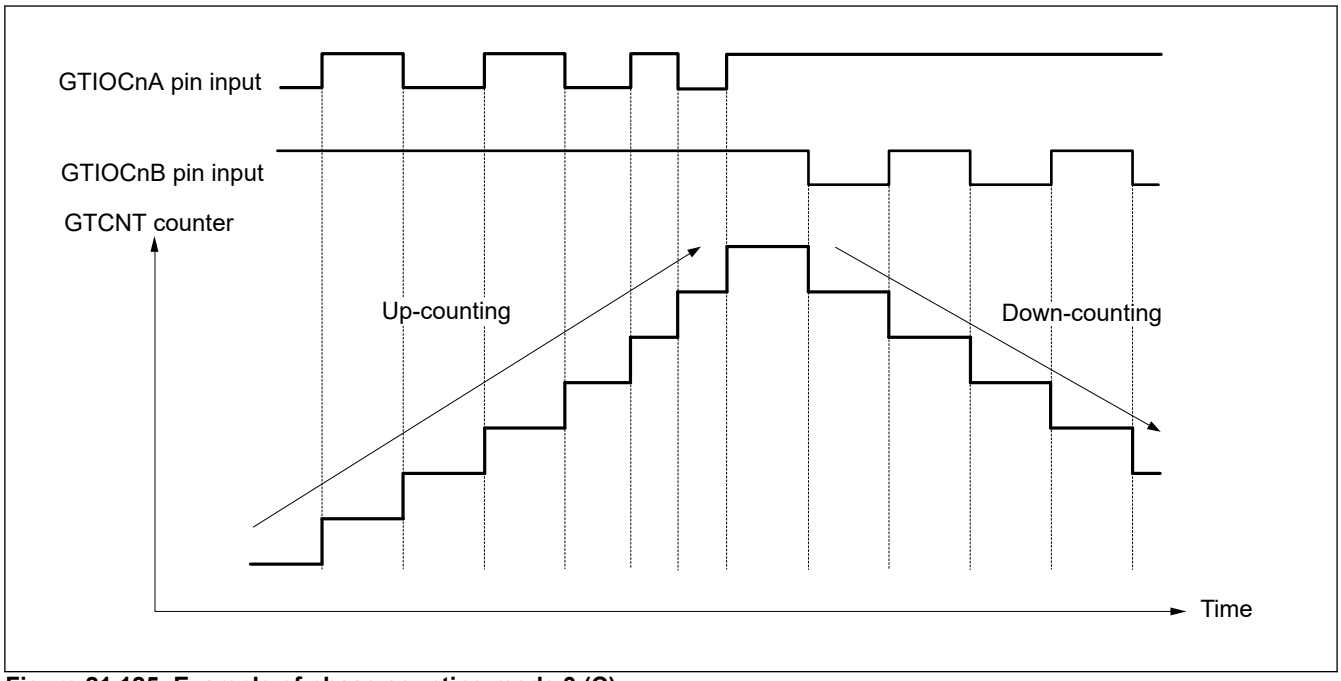


Figure 21.125 Example of phase counting mode 3 (C)

Table 21.57 Conditions of up-counting/down-counting in phase counting mode 3 (C)

: Rising edge

: Falling edge

GTIOCnA pin input	GTIOCnB pin input	Operation	Register setting
High		Down-counting	GTUPSR = 0x00000A00 GTDNSR = 0x0000A000
Low		Not counting	
	Low		
	High	Up-counting	
High		Down-counting	
Low		Not counting	
	High	Up-counting	
	Low	Not counting	

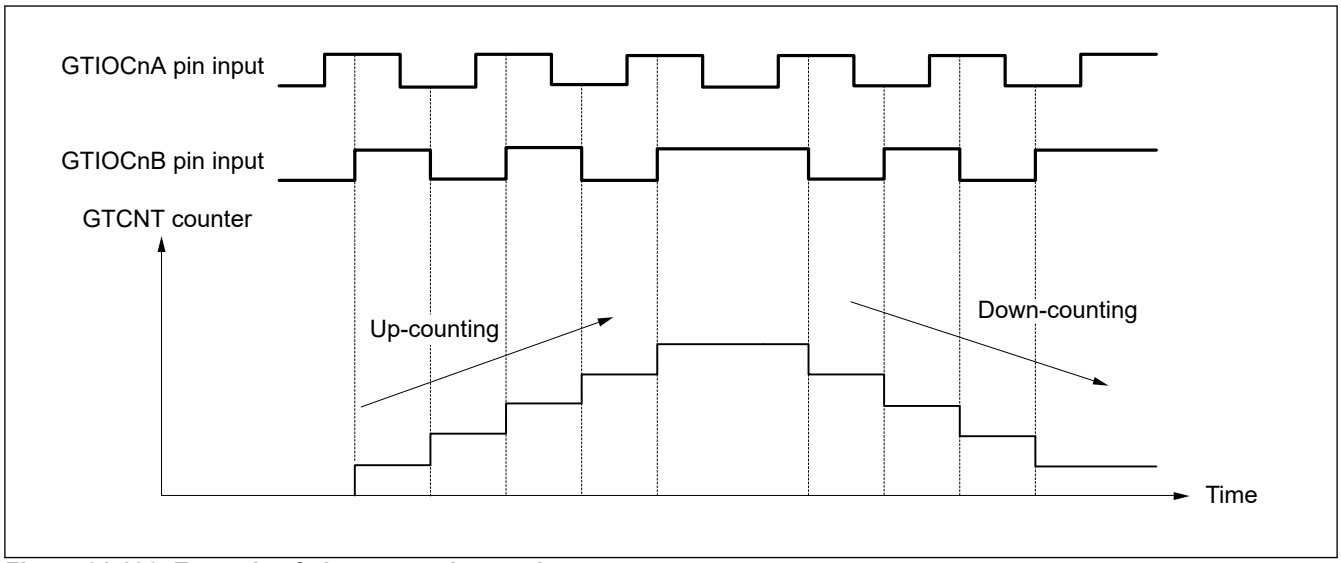


Figure 21.126 Example of phase counting mode 4

Table 21.58 Conditions of up-counting/down-counting in phase counting mode 4

: Rising edge

: Falling edge

GTIOCnA pin input	GTIOCnB pin input	Operation	Register setting
High	↑	Up-counting	GTUPSR = 0x00006000 GTDNSR = 0x00009000
Low	↓		
↑	Low	Not counting	
↓	High		
High	↓	Down-counting	
Low	↑		
↑	High	Not counting	
↓	Low		

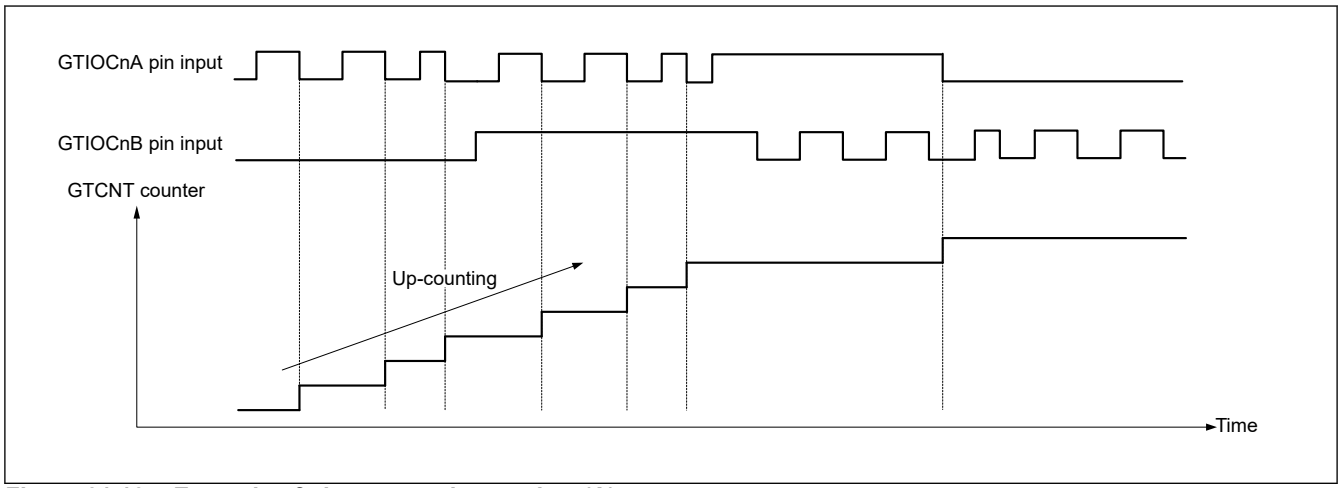


Figure 21.127 Example of phase counting mode 5 (A)

Table 21.59 Conditions of up-counting/down-counting in phase counting mode 5 (A)

: Rising edge

: Falling edge

GTIOCnA pin input	GTIOCnB pin input	Operation	Register setting
High		Not counting	GTUPSR = 0x00000C00 GTDNSR = 0x00000000
Low			
	Low		
	High	Up-counting	
High		Not counting	
Low			
	High		
	Low	Up-counting	

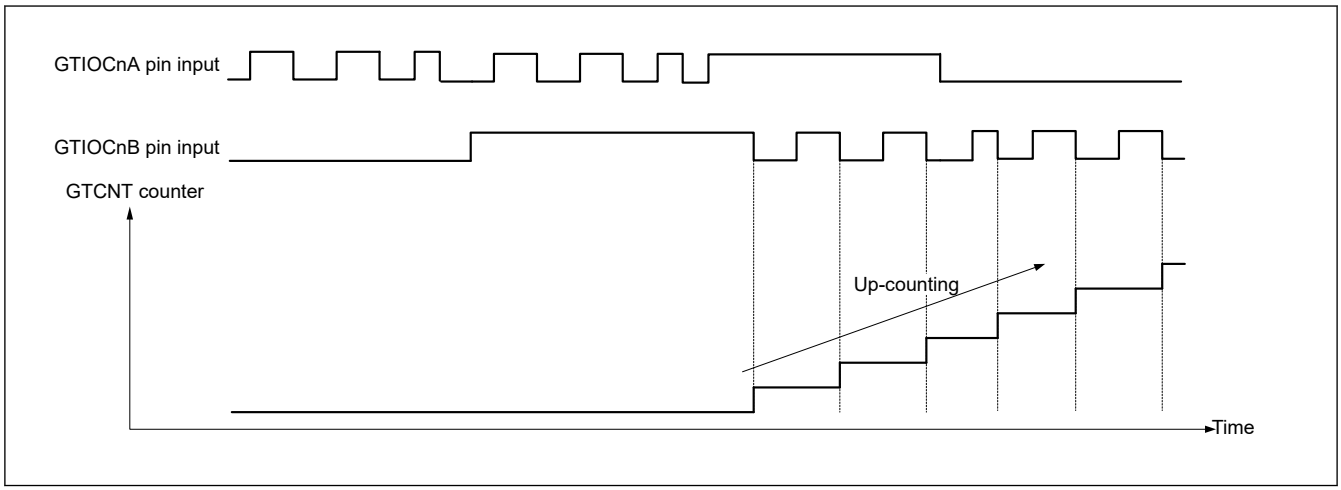


Figure 21.128 Example of phase counting mode 5 (B)

Table 21.60 Conditions of up-counting/down-counting in phase counting mode 5 (B)

: Rising edge

: Falling edge

GTIOCnA pin input	GTIOCnB pin input	Operation	Register setting
High		Not counting	GTUPSR = 0x0000C000 GTDNSR = 0x00000000
Low		Up-counting	
	Low	Not counting	
	High		
High		Up-counting	
Low		Not counting	
	High		
	Low		

21.3.12 External pulse width measuring function

The pulse width of GTIOCnA pin input (n = 0 to 3), GTIOCnB pin input, and GTETRGA / GTETRGB / GTETRGC / GTETRGD pin inputs can be measured.

The setting to enable or disable count-up of the GTCNT counter and the input pin and level which measured pulse width are selected by the USILVL[3:0] bits of the GTUPSR register.

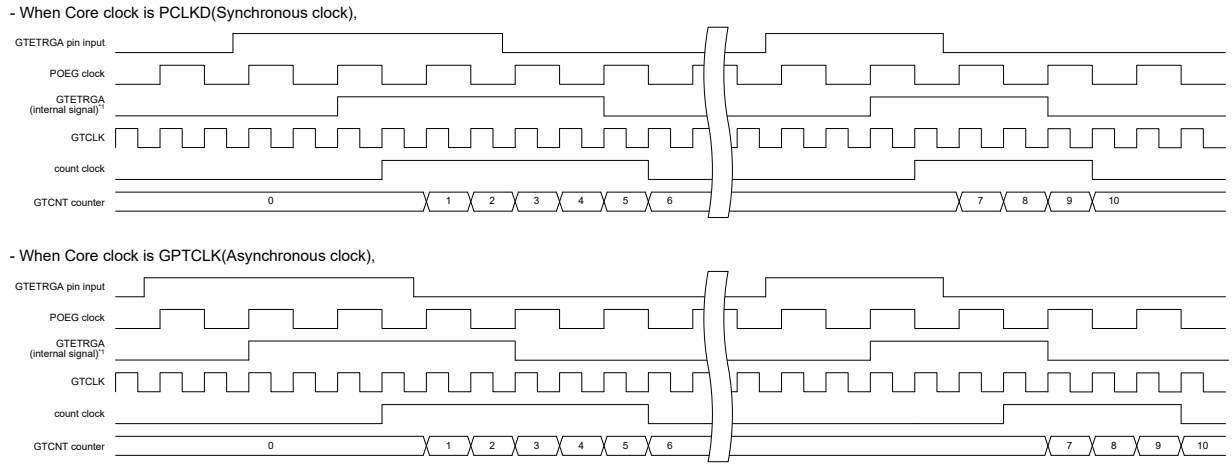
The setting to enable or disable count-down of the GTCNT counter and the input pin and level which measured pulse width are selected by the DSILVL[3:0] bits of the GTDNSR register.

The setting to enable both count-up and count-down of the GTCNT counter at the same time is prohibited.

The counting operation performs periodic counting with the period of the GTPR register.

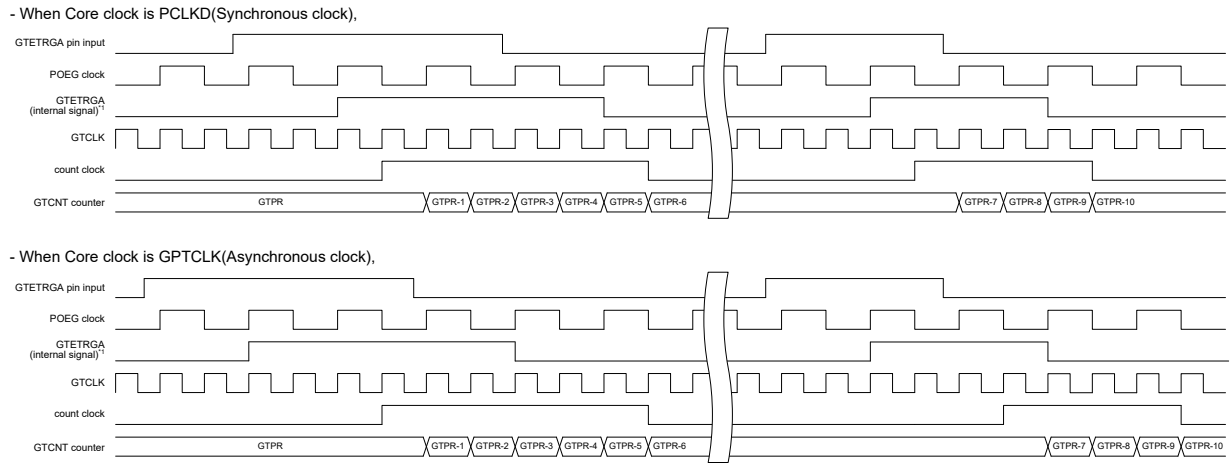
If the phase counting function and the pulse width measuring function are enabled at the same time, the pulse width measuring function does not work and the phase counting function works.

Figure 21.129, Figure 21.130 show examples of external pulse width measuring function and Table 21.61 shows example for setting external pulse width measuring function.



Note: POEG clock: PCLKB
Note 1. The input signal to GPT when the digital noise filter of GTETRGA pin is not used in POEG is shown.

Figure 21.129 Example of External pulse width measuring function(Up-counting)



Note: POEG clock: PCLKB
Note 1. The input signal to GPT when the digital noise filter of GTETRGA pin is not used in POEG is shown.

Figure 21.130 Example of External pulse width measuring function(Down-counting)

Table 21.61 Example for Setting External pulse width measuring function

No.	Step Name	Description
1	Set external pulse width measuring function	Enable external pulse width measuring function with the GTUPSR.USILVL[3:0] bits for up-counting operation and the GTDNSR.DSILVL[3:0] bits for down-counting operation and select the input pin and level to measure. In Figure 21.129, GTUPSR.USILVL[3:0] = 1001b (count up when GTETRGA pin is 1), In Figure 21.130, GTDNSR.DSILVL[3:0] = 1001b (count down when GTETRGA pin is 1)
2	Select count clock	Select the count clock with GTCR.TPCS[3:0] bits of the corresponding channel.
3	Set cycle	Set the cycle in GTPR of the corresponding channel.
4	Set initial value for counter	Set the initial value in the GTCNT counter of the corresponding channel. In Figure 21.129, 0000 0000h is set. In Figure 21.130, the GTPR register value is set.
5	Start count operation	Input a pulse to the input pin to be measured to start count operation.

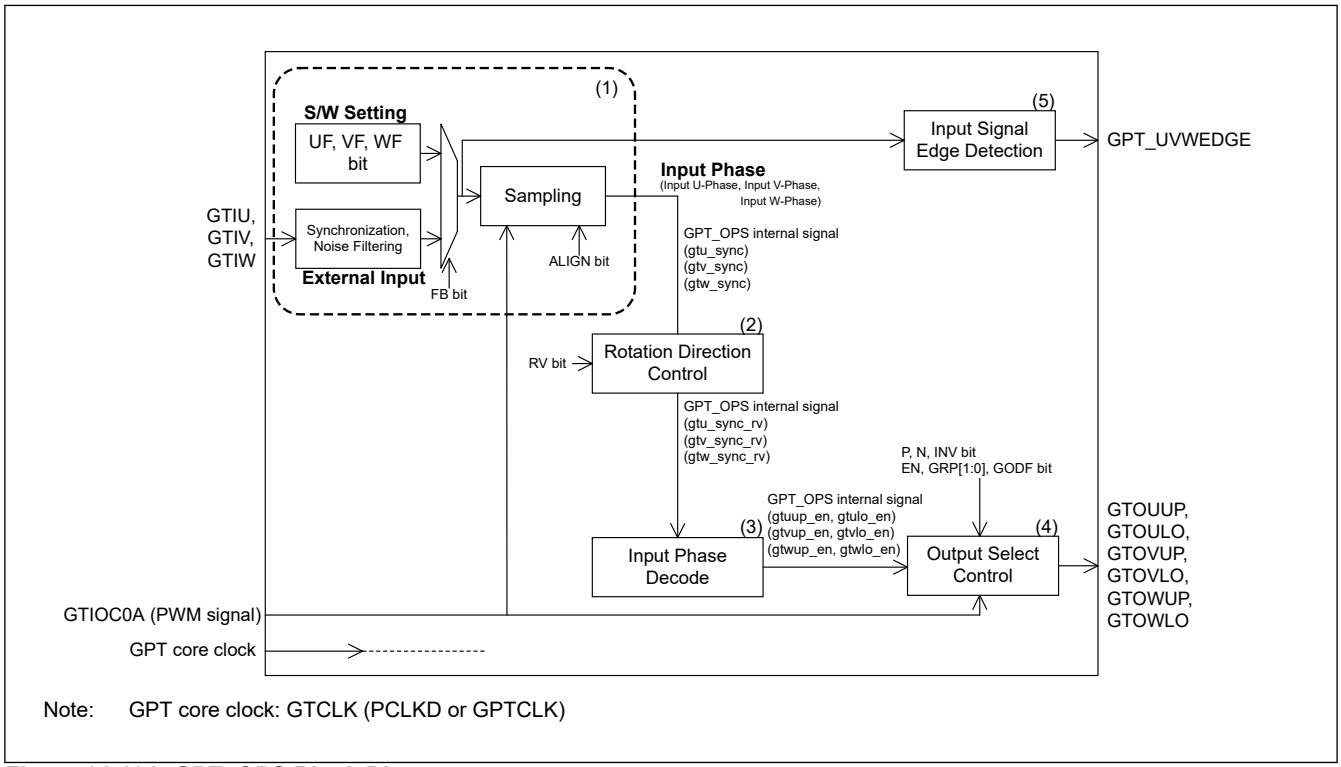
21.3.13 Output Phase Switching (GPT_OPS)

GPT_OPS can easily control Brushless DC motor using Output Phase Switching Control Register (OPSCR).

GPT_OPS uses S/W setting value (OPSCR.UF, VF, WF bits) or external signals detected by the Hall element as input signals. GPT_OPS outputs either level signals or chopped signals by GPT320's PWM as the 6-phase (U-positive phase / negative phase, V-positive phase / negative phase, W-positive phase / negative phase) signals to control motor.

Figure 21.131 shows the block diagram of GPT_OPS.

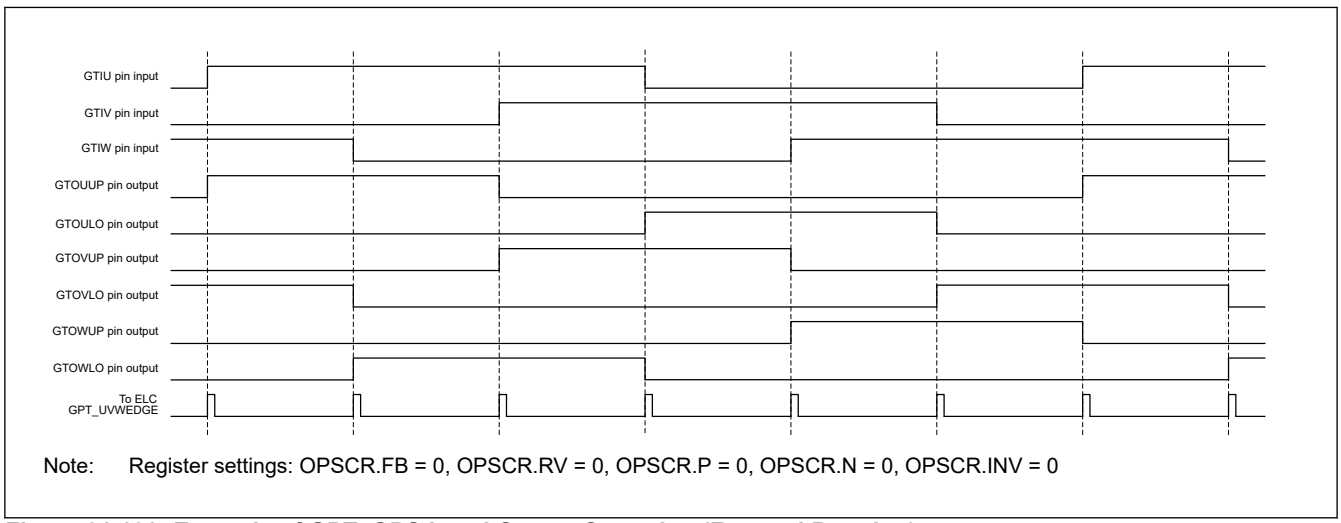
The GPT_UVWEDGE signal is output signal to ELC generated by detecting the edge of input signal.



Note: GPT core clock: GTCLK (PCLKD or GPTCLK)

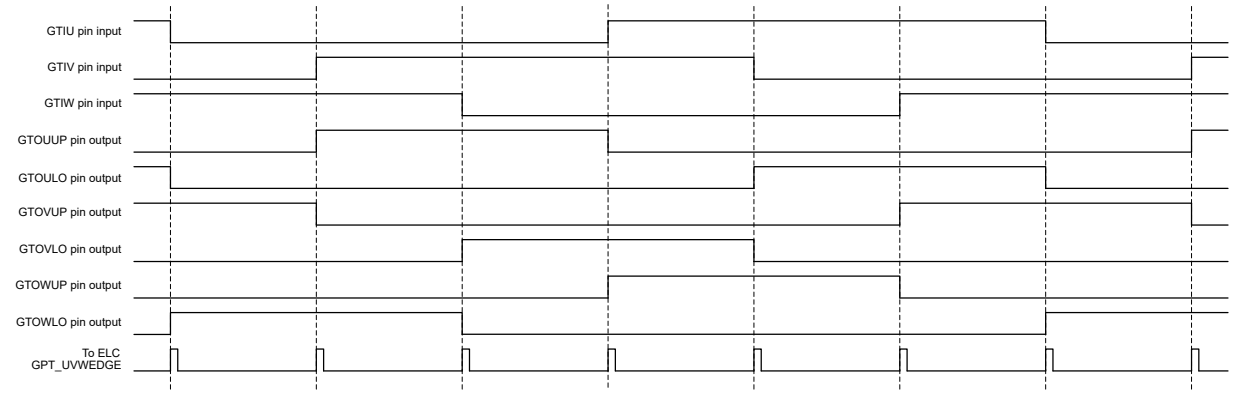
Figure 21.131 GPT_OPS Block Diagram

Figure 21.132 and Figure 21.133 show examples of GPT_OPS level output operation.



Note: Register settings: OPSCR.FB = 0, OPSCR.RV = 0, OPSCR.P = 0, OPSCR.N = 0, OPSCR.INV = 0

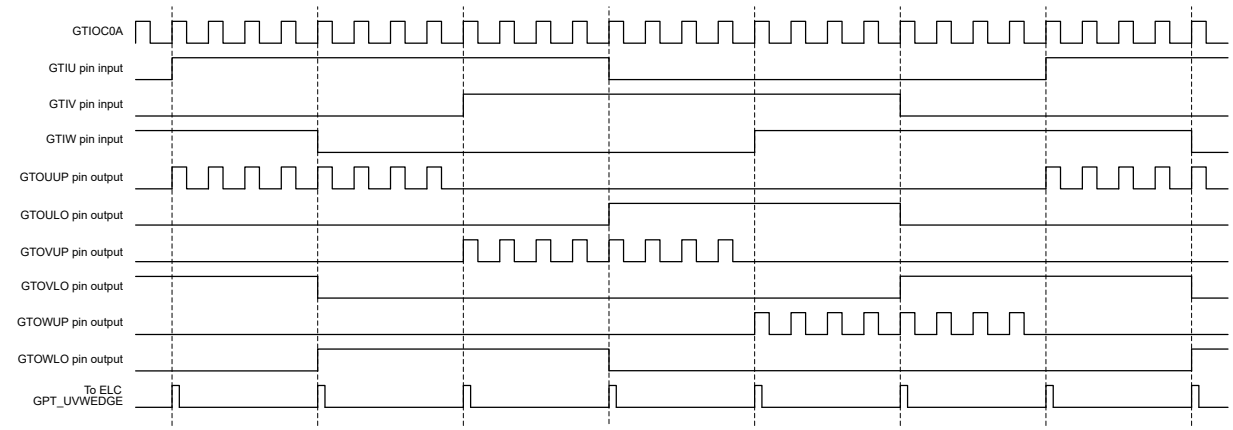
Figure 21.132 Example of GPT_OPS Level Output Operation (Forward Rotation)



Note: Register settings: OPSCR.FB = 0, OPSCR.RV = 1, OPSCR.P = 0, OPSCR.N = 0, OPSCR.INV = 0

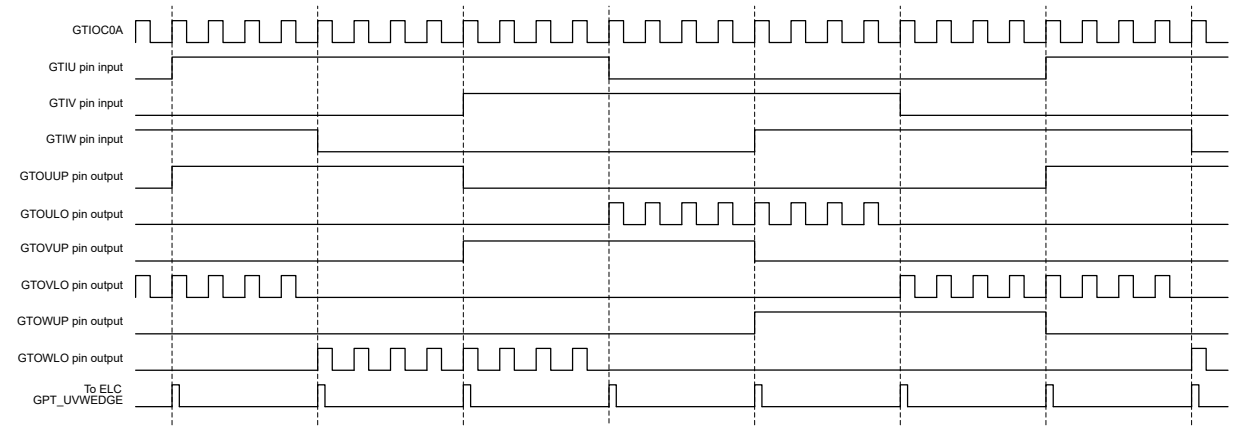
Figure 21.133 Example of GPT_OPS Level Output Operation (Reverse Rotation)

Figure 21.134 and Figure 21.135 show examples of GPT_OPS chopped output operation.



Note: Register settings: OPSCR.FB = 0, OPSCR.RV = 0, OPSCR.P = 1, OPSCR.N = 0, OPSCR.INV = 0

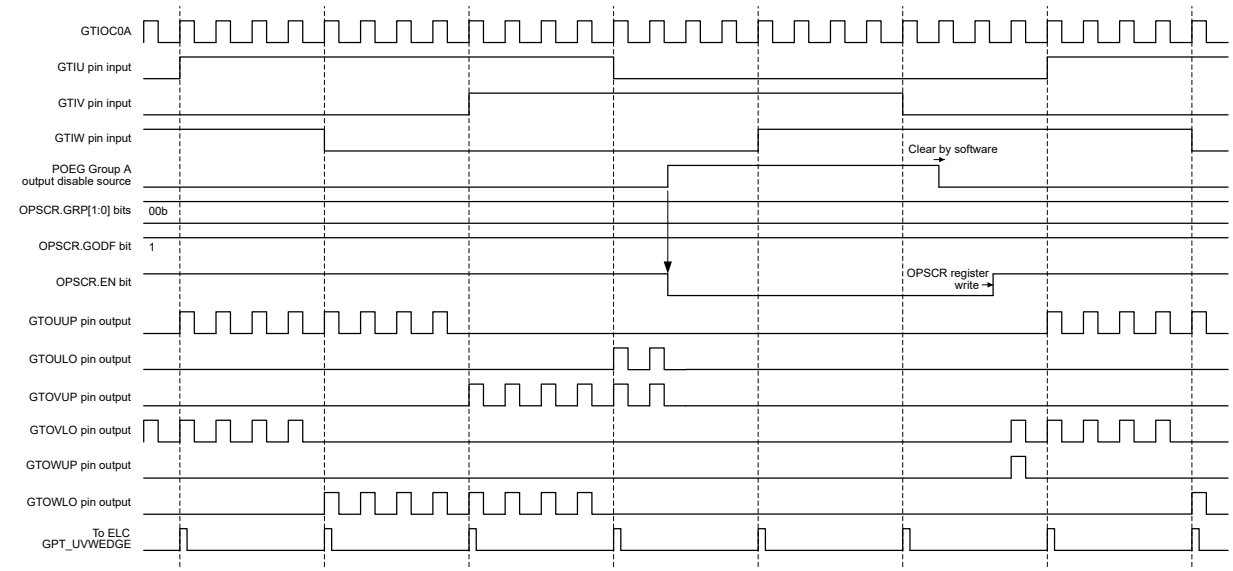
Figure 21.134 Example of GPT_OPS Chopped Output Operation (Positive Phase 120-degree)



Note: Register settings: OPSCR.FB = 0, OPSCR.RV = 0, OPSCR.P = 0, OPSCR.N = 1, OPSCR.INV = 0

Figure 21.135 Example of GPT_OPS Chopped Output Operation (Negative Phase 120-degree)

Figure 21.136 shows an example of GPT_OPS output disable control operation.



Note: Register settings: OPSCR.FB = 0, OPSCR.RV = 0, OPSCR.P = 1, OPSCR.N = 1, OPSCR.INV = 0

Figure 21.136 Example of GPT_OPS Output Disable Control Operation

21.3.13.1 Input Selection and Sampling

The FB bit selects either the software setting value or external input for the input signal.

When the FB bit is 0, the GTIU, GTIV, GTIW external input are selected for the input signal to GPT_OPS after synchronization with the GPT core clock (GTCLK) and noise filtering.

When the FB bit is 1, the software setting value (UF,VF,WF bits) are selected for the input signal to GPT_OPS.

The selected input signals are sampled by the method selected by the ALIGN bit, and they are treated as input phase of GPT_OPS.

When the ALIGN bit is 0, the input signals are sampled by GTCLK.

When the ALIGN bit is 1, the input signals are sampled by the falling edge of GTIOC0A pin output.

The signals after sampling can be read by the U, V, W bits.

Table 21.62 shows the input selection by the FB bit and sampling method by the ALIGN bit.

Table 21.62 Input Selection and Sampling Method

OPSCR Register		Input Selection Sampling Method	Input Phase (GPT_OPS internal signal)
FB bit	ALIGN bit		
0	0	GTIU, GTIV, GTIW external input GTCLK sampling	Input U-phase (gtu_sync) Input V-phase (gtv_sync) Input W-phase (gtw_sync)
	1	GTIU, GTIV, GTIW external input GTIOC0A falling edge sampling	
1	0	Software setting value UF, VF, WF bits GTCLK sampling	
	1	Software setting value UF, VF, WF bits GTIOC0A falling edge sampling	

21.3.13.2 Rotation Direction Control

When the rotation direction is reverse (RV bit = 1), the input phase is inverted.

21.3.13.3 Input phase decode

The 6-phase signals by decoding input phase after rotation direction control are generated.

Table 21.63 and Table 21.64 show the decode tables of input phase to rotate motor in forward (RV = 0) and reverse (RV = 1).

Table 21.63 The Decode Table of Input Phase (Forward Rotation)

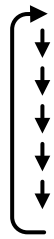
	Input Phase			Input Phase after Rotation Direction Control			6-Phase Signals					
	U Phase	V Phase	W Phase	U Phase	V Phase	W Phase	U Positive Phase	U Negative Phase	V Positive Phase	V Negative Phase	W Positive Phase	W Negative Phase
	gtu_sync	gtv_sync	gtw_sync	gtu_sync_rv	gtv_sync_rv	gtw_sync_rv	gtuup_en	gtulo_en	gtvup_en	gtvlo_en	gtwup_en	gtwlo_en
	1	0	1	1	0	1	1	0	0	1	0	0
	1	0	0	1	0	0	1	0	0	0	0	1
	1	1	0	1	1	0	0	0	1	0	0	1
	0	1	0	0	1	0	0	1	1	0	0	0
	0	1	1	0	1	1	0	1	0	0	1	0
	0	0	1	0	0	1	0	0	0	1	1	0
	0	0	0	0	0	0	0	0	0	0	0	0
	1	1	1	1	1	1	0	0	0	0	0	0

Table 21.64 The Decode Table of Input Phase (Reverse Rotation)

	Input Phase			Input Phase after Rotation Direction Control			6-Phase Signals					
	U Phase	V Phase	W Phase	U Phase	V Phase	W Phase	U Positive Phase	U Negative Phase	V Positive Phase	V Negative Phase	W Positive Phase	W Negative Phase
	gtu_sync	gtv_sync	gtw_sync	gtu_sync_rv	gtv_sync_rv	gtw_sync_rv	gtuup_en	gtulo_en	gtvup_en	gtvlo_en	gtwup_en	gtwlo_en
	1	0	1	0	1	0	0	1	1	0	0	0
	1	0	0	0	1	1	0	1	0	0	1	0
	1	1	0	0	0	1	0	0	0	1	1	0
	0	1	0	1	0	1	1	0	0	1	0	0
	0	1	1	1	0	0	1	0	0	0	0	1
	0	0	1	1	1	0	0	0	1	0	0	1
	0	0	0	0	0	0	0	0	0	0	0	0
	1	1	1	1	1	1	0	0	0	0	0	0

21.3.13.4 Output selection control

The EN, P, N, INV bits select the output wave.

The EN bit enables output of the 6-phase output. When the EN bit is 1, output of the 6-phase output is enabled. When the EN bit is 0, the external pin output is Hi-Z.

The P, N bits select whether chopping positive and negative phase are performed or not. When P, N bits are 1, chopping is performed by GTIOC0A pin output.

When the chopping is performed, there are cases where the PWM width of output is shorter than the width of the PWM used to chop just before or after switching of output phase, depending on the phase difference between the phase output switch timing and the phase of PWM.

The INV bit selects the polarity (either positive logic or negative logic) of phase output.

Table 21.65 and Table 21.66, show the output selection control method for positive and negative phase output.

Table 21.65 Output Selection Control Method (Positive Phase)

EN bit	P bit	INV bit	GTOUUP / GTOVUP / GTOWUP
0	x	x	0 (External pin output is Hi-Z)
1	0	0	Positive logic level output (gtuup_en) (gtvup_en) (gtwup_en)
1	0	1	Negative logic level output (~gtuup_en) (~gtvup_en) (~gtwup_en)
1	1	0	Positive logic chopped output (GTIOC0A & gtuup_en) (GTIOC0A & gtvup_en) (GTIOC0A & gtwup_en)
1	1	1	Negative logic chopped output (~(GTIOC0A & gtuup_en)) (~(GTIOC0A & gtvup_en)) (~(GTIOC0A & gtwup_en))

Table 21.66 Output Selection Control Method (Negative Phase)

EN bit	N bit	INV bit	GTOULO / GTOVLO / GTOWLO
0	x	x	0 (External pin output is Hi-Z)
1	0	0	Positive logic level output (gtulo_en) (gtvlo_en) (gtwlo_en)
1	0	1	Negative logic level output (~gtulo_en) (~gtvlo_en) (~gtwlo_en)
1	1	0	Positive logic chopped output (GTIOC0A & gtulo_en) (GTIOC0A & gtvlo_en) (GTIOC0A & gtwlo_en)
1	1	1	Negative logic chopped output (~(GTIOC0A & gtulo_en)) (~(GTIOC0A & gtvlo_en)) (~(GTIOC0A & gtwlo_en))

21.3.13.5 Output Selection Control (Group Output Disable Function)

When GODF = 1 and signal value selected by the GRP bit is Hi (Output Disable Request), GPT_OPS's output pins are changed to Hi-Z asynchronously and the OPSCR.EN bit is cleared to 0 by the output disable request signal synchronized with GTCLK.

For the return, after clearing the Output Disable Request by software, set the EN bit to 1.

The timing of EN bit cleared to 0 is 3 GTCLK cycles after generating the output disable request. In order to perform the output disable control surely, the output disable request flag in POEG should be cleared in the timing that terminating the output disable request is at least 4 GTCLK cycles after generating the output disable request.

The example of the operation of the group output disable control, see the above-mentioned Figure 21.136.

21.3.13.6 Event Link Controller (ELC) Output

The logical sum of the pulse detected by rising and falling edge of U, V, W phase input is output to the event link controller (ELC). When the high level period of input phase is short, there are cases that the detected edge is not transmitted to the ELC correctly because of the logical sum.

21.3.13.7 GPT_OPS Start Operation Setting Flow

Table 21.67 Example for Setting of GPT_OPS Start Operation

No.	Step Name	Description
1	Set Operation Mode of GPT320	Set the PWM output operation mode of GPT320. See section 21.3.3. PWM Output Operating Mode
2	Start Count Operation of GTP320	Start count operation of GPT320, and outputs PWM waveform.
3	Set GPT_OPS Input Condition	• When the soft setting is selected, set the soft setting values in the UF, VF, WF bits. • When the external input is selected, if necessary, use noise filtering. Select the sampling clock for external input into NFCS[1:0] bits, and set NFEN bit to 1.
4	Select GPT_OPS Input phase and Alignment	Select input phase by FB bit. Select alignment of input phase by ALIGN bit.
5	Set GPT_OPS Output phase Condition	Set the rotation direction by RV bit. Select whether chopping is performed or not by P, N bits. Select the output polarity by INV bit.
6	Set GPT_OPS Output Disable Condition	Select the error group by GRP[1:0] bits Set ON/OFF of the group output disable function by GODF bit.
7	Set GPT_OPS Operation	Set EN bit to 1 to output the 6-phase output to drive the brushless DC motor.

21.3.14 Inter Channel Logical Operation Function

The logical operation function between compare match outputs can be performed.

Figure 21.137 shows the block diagram of inter channel logical operation.

To prevent hazard to the GPT output, the signal after logical operation is latched with GTCLK. After latching, the output disable control is performed.

When the logical operation function which causes the delay of 1 GTCLK is selected, the output enable signal is also delayed with 1 GTCLK and input to the output disable control.

When the same signal (C = A or D = B) to operate logical function AND, OR, EXOR and NOR is selected, C or D is treated as 1. In the case of GTIOCnA pin output, when A of same channel is selected for C, the result of AND is A, the result of OR is 1, the result of EXOR is NOT A, and the result of NOR is 0.

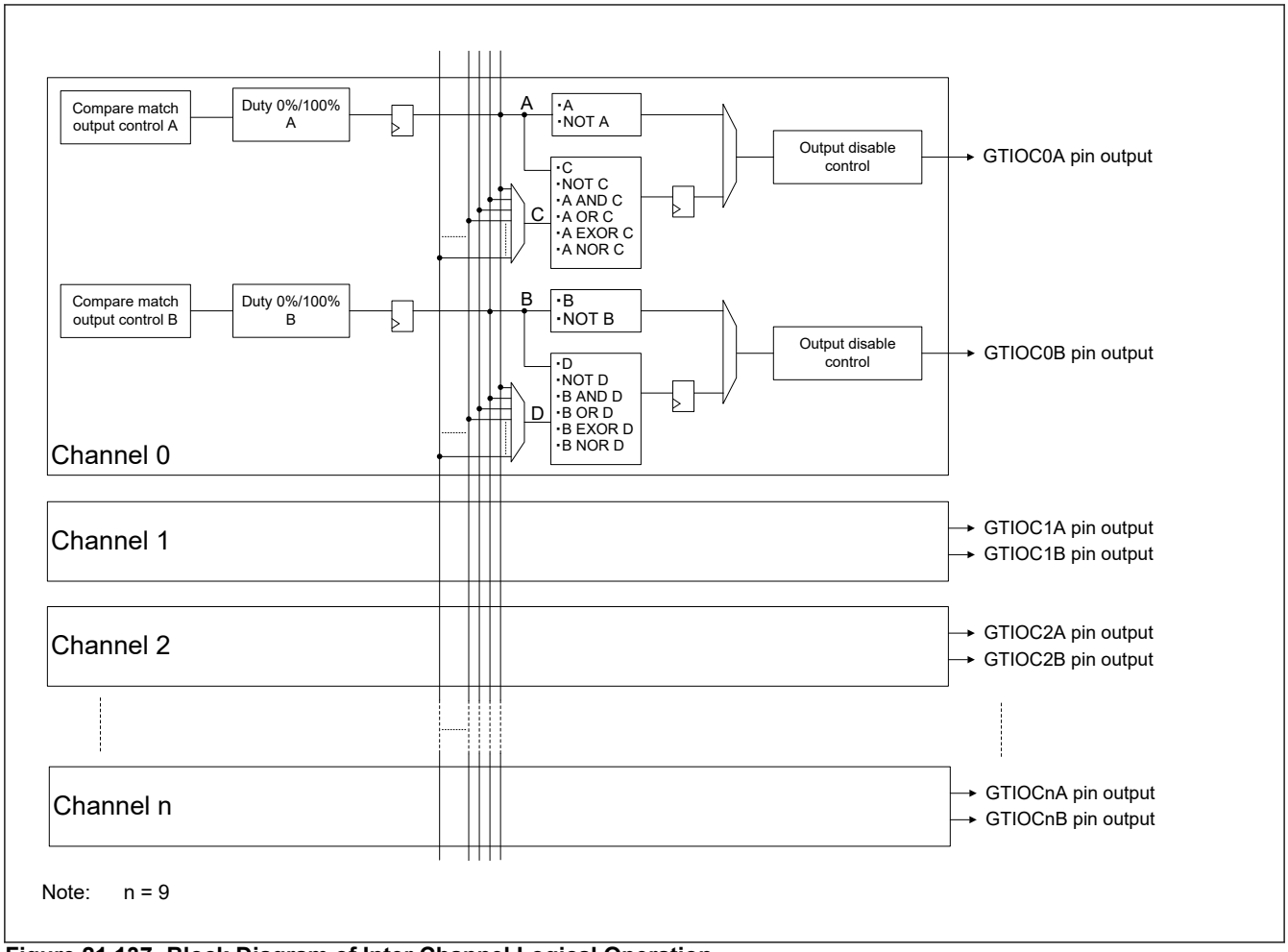


Figure 21.137 Block Diagram of Inter Channel Logical Operation

Figure 21.138 shows an example of inter channel logical operation.

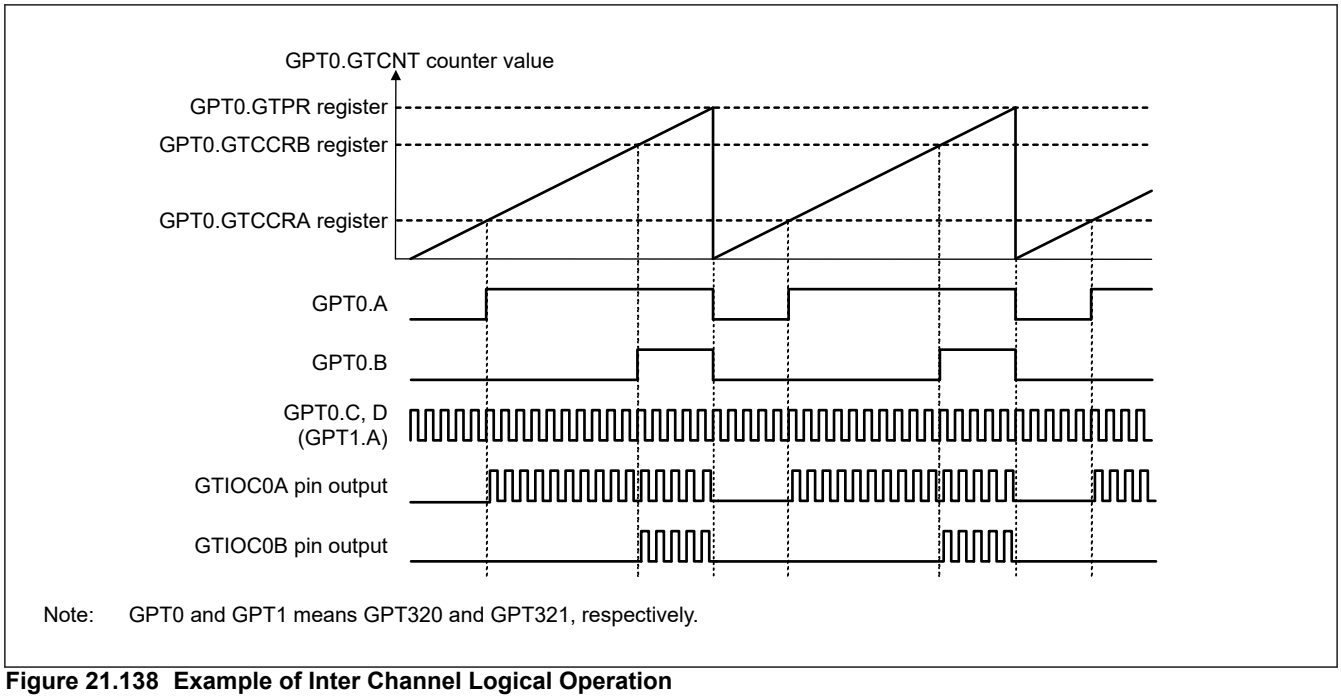


Figure 21.138 Example of Inter Channel Logical Operation

21.4 Interrupt Sources

21.4.1 Interrupt Sources

The GPT provides the following interrupt sources:

- GTCCR input capture/compare match
- GTADTR compare match
- GTCNT counter overflow (GTPR compare match)/underflow.
- period count function finish

Each interrupt source has its own status flag. When an interrupt source signal is generated, the associated status flag in GTST is set to 1, and an interrupt request is generated. The associated status flag in GTST can be cleared by writing 0. If flag set and flag clear occur at the same time, flag clear takes priority over flag set. These flags are automatically updated by the internal state. The Interrupt Controller Unit can change the relative channel priorities. However, the priority within a channel is fixed. For details, see section 12, Interrupt Controller Unit (ICU).

Table 21.68 lists the GPT interrupt sources.

Table 21.68 Interrupt sources

Channel	Name	Interrupt source	Interrupt flag	DTC activation
n = 0 to 9	GPTn_CCMPA	GPT32n.GTCCRA input capture/compare match	GTST[0] (TCFA)	Possible
	GPTn_CCMPB	GPT32n.GTCCRB input capture/compare match	GTST[1] (TCFB)	Possible
	GPTn_CMPC	GPT32n.GTCCRC compare match	GTST[2] (TCFC)	Possible
	GPTn_CMPD	GPT32n.GTCCRD compare match	GTST[3] (TCFD)	Possible
	GPTn_CMPE	GPT32n.GTCCRE compare match	GTST[4] (TCFE)	Possible
	GPTn_CMPF	GPT32n.GTCCRF compare match	GTST[5] (TCFF)	Possible
	GPTn_OVF	GPT32n.GTCNT overflow (GPT32n.GTPR compare match)	GTST[6] (TCFPO)	Possible
	GPTn_UDF	GPT32n.GTCNT underflow	GTST[7] (TCFPU)	Possible
	GPTn_ADTRGA	GPT32n.GTADTRA compare match	GTST[17:16] (ADTRADF, ADTRAUF)	Possible
	GPTn_ADTRGB	GPT32n.GTADTRB compare match	GTST[19:18] (ADTRBDF, ADTRBUF)	Possible
	GPTn_PC	Period count function finish (n = 0 to 3)	GTST[31] (PCF)	Possible

(1) GPTn_CCMPA interrupt (n = 0 to 9)

An interrupt request is generated under the following conditions:

- When the GTCCRA register functions as a compare match register, the GTCNT counter value (in complementary PWM mode, the GTCNT counter value of master channel) matches with the GTCCRA register
- When the GTCCRA register functions as an input capture register, the input-capture signal causes transfer of the GTCNT counter value to the GTCCRA register. In complementary PWM mode, GTCCRA register does not function as an input capture register.

(2) GPTn_CCMPB interrupt (n = 0 to 9)

An interrupt request is generated under the following conditions:

- When the GTCCRB register functions as a compare match register, the GTCNT counter value matches with the GTCCRB register. In complementary PWM mode, GTCCRB register does not function as a compare match register.
- When the GTCCRB register functions as an input capture register, the input-capture signal causes transfer of the GTCNT counter value to the GTCCRB register. In complementary PWM mode, GTCCRB register does not function as an input capture register.

(3) GPTn_CMPC interrupt (n = 0 to 9)

An interrupt request is generated under the following condition:

- When the GTCCRC register functions as a compare match register, the GTCNT counter value (in complementary PWM mode, GTCNT counter value of master channel) matches with the GTCCRC register.

A compare match is not performed and therefore, an interrupt is not requested in the following conditions:

- GTCR.MD[3:0] = 0001b (saw-wave one-shot pulse mode)
- GTCR.MD[3:0] = 0110b (triangle-wave PWM mode 3)
- GTBER.CCRA[1:0] = 01b, 10b, 11b (buffer operation with the GTCCRC register).

(4) GPTn_CMPD interrupt (n = 0 to 9)

An interrupt request is generated under the following condition:

- When the GTCCRD register functions as a compare match register, the GTCNT counter value (in complementary PWM mode, the GTCNT counter value of master channel) matches with the GTCCRD register.

A compare match is not performed and therefore, an interrupt is not requested in the following conditions:

- GTCR.MD[3:0] = 0001b (saw-wave one-shot pulse mode)
- GTCR.MD[3:0] = 0110b (triangle-wave PWM mode 3)
- GTBER.CCRA[1:0] = 10b, 11b (buffer operation with the GTCCRD register).

(5) GPTn_CMPE interrupt (n = 0 to 9)

An interrupt request is generated under the following condition:

- When the GTCCRE register functions as a compare match register, the GTCNT counter value (in complementary PWM mode, the GTCNT counter value of master channel) matches with the GTCCRE register.

A compare match is not performed and therefore, an interrupt is not requested in the following conditions:

- GTCR.MD[3:0] = 0001b (saw-wave one-shot pulse mode)
- GTCR.MD[3:0] = 0110b (triangle-wave PWM mode 3)
- GTBER.CCRB[1:0] = 01b, 10b, 11b (buffer operation with the GTCCRE register).

(6) GPTn_CMPF interrupt (n = 0 to 9)

An interrupt request is generated under the following condition:

- When the GTCCRF register functions as a compare match register, the GTCNT counter value (in complementary PWM mode, the GTCNT counter value of master channel) matches with the GTCCRF register.

A compare match is not performed and therefore, an interrupt is not requested in the following conditions:

- GTCR.MD[3:0] = 0001b (saw-wave one-shot pulse mode)
- GTCR.MD[3:0] = 0110b (triangle-wave PWM mode 3)
- GTBER.CCRB[1:0] = 10b, 11b (buffer operation with the GTCCRF register).

(7) GPTn_OVF interrupt (n = 0 to 9)

An interrupt request is generated in the following conditions:

- In saw-wave PWM mode 1 and saw-wave one-shot pulse mode, interrupt requests are enabled at overflows (when the GTCNT counter value changes from GTPR to 0 during up-counting)
- In saw-wave PWM mode 2, interrupt requests are enabled at overflows (GTCNT counter value changes from GTCCRm (m = A to F) register value selected with GTCSR.CSCMSC[2:0] bits to 0) or the time when GTCNT counter value matches GTPR register value.
- In triangle-wave mode, interrupt requests are enabled at crests (the GTCNT changes from GTPR to GTPR-1)

- In complementary PWM mode, interrupt requests are enabled at crests (GTCNT counter value of master channel changes from GTPR register value to GTPR register value minus 1).
- In counting by hardware sources (include external pulse width measuring function), an overflow (GTCNT changes from GTPR to 0 in up count) has occurred.

(8) GPTn_UDF interrupt (n = 0 to 9)

An interrupt request is generated in the following conditions.

- In saw-wave PWM mode 1 and saw-wave one-shot pulse mode, interrupt requests are enabled at underflows (when the GTCNT counter value changes from 0 to GTPR during down-counting)
- In triangle-wave mode, interrupt requests are enabled at troughs (the GTCNT changes from 0 to 1)
- In complementary PWM mode, interrupt requests are enabled at troughs (GTCNT counter value of master channel changes from 0 to 1).
- In counting by hardware sources (include external pulse width measuring function), underflow (GTCNT changes from 0 to GTPR in down count) has occurred.

About Interrupt signals and interrupt status flags, see section 21.2.16. GTST : General PWM Timer Status Register.

(9) GPTn_ADTRGA interrupt (n = 0 to 9)

When the GTCNT counter value matches with GTADTRA, an interrupt request is generated under the following condition.

- In Up-counting, the interrupt enable bit (ADTRAUEN) in GTINTAD is 1.
- In Down-counting, the interrupt enable bit (ADTRADEN) in GTINTAD is 1.
  In event count operation performing, this interrupt request isn't generated.

(10) GPTn_ADTRGB interrupt (n = 0 to 9)

When the GTCNT counter value matches with GTADTRB, an interrupt is generated under the following condition.

- In Up-counting, the interrupt enable bit (ADTRBUEN) in GTINTAD is 1.
- In Down-counting, the interrupt enable bit (ADTRBDEN) in GTINTAD is 1.
  In event count operation performing, this interrupt request isn't generated.

(11) GPTn_PC Interrupt (n = 0 to 3)

When the GTPC.PCEN bit is 1 and the GTPC.PCNT counter is 1, an interrupt request is generated at the end of cycle.

21.4.2 DMAC and DTC Activation

The DMAC and DTC can be activated by the interrupt in each channel. For details, see section 12, Interrupt Controller Unit (ICU), section 15, DMA Controller (DMAC), and section 16, Data Transfer Controller (DTC).

21.4.3 Interrupt and A/D Conversion Start Request Skipping Function

21.4.3.1 Interrupt Skipping Function by GTITC Register

By setting the GTITC register, the GTCNT counter overflow (GTPR register compare match) interrupt (GPTn_OVF) and underflow interrupt (GPTn_UDF) can be skipped. Other interrupts and A/D conversion start request signals can be skipped in coordination with the GPTn_OVF/GPTn_UDF skipping function. When the interrupt is skipped, the updating of relevant status flag is also skipped. The interrupt skipping is continued even if the status flag is set to 1.

The interrupt skipping function is related only to the GTITC register setting, and is not related to setting of the GTINTAD register interrupt enable bit. The interrupt skipping is continued even if the interrupt is disabled with GTINTAD register setting.

When both troughs and crests are counted and skipped in triangle-wave mode, if the number of times of skipping is odd, GPTn_OVF/GPTn_UDF interrupt requests cannot be generated at troughs only or at crests only depending on the skipping counter start timing. Therefore, in order to count both troughs and crests and generate the GPTn_OVF/GPTn_UDF interrupts at troughs only or crests only in triangle-wave mode, the number of times of skipping should be even.

Similarly, in saw-wave mode, when both overflows and underflows are counted and skipped with the count direction changed, GPTn_OVF/GPTn_UDF interrupt requests are sometimes not generated at overflows only or at underflows only. Therefore, in order to count both overflows and underflows with the count direction changed and generate the GPTn_OVF/GPTn_UDF interrupts at overflows only or underflows only in saw-wave mode, the skipping state should be carefully checked before using.

When changing the skipping count, be sure to release the skipping count setting (GTITC.IVTC[1:0] bits = 00b).

Figure 21.139 to Figure 21.144 show examples of skipping function operation.

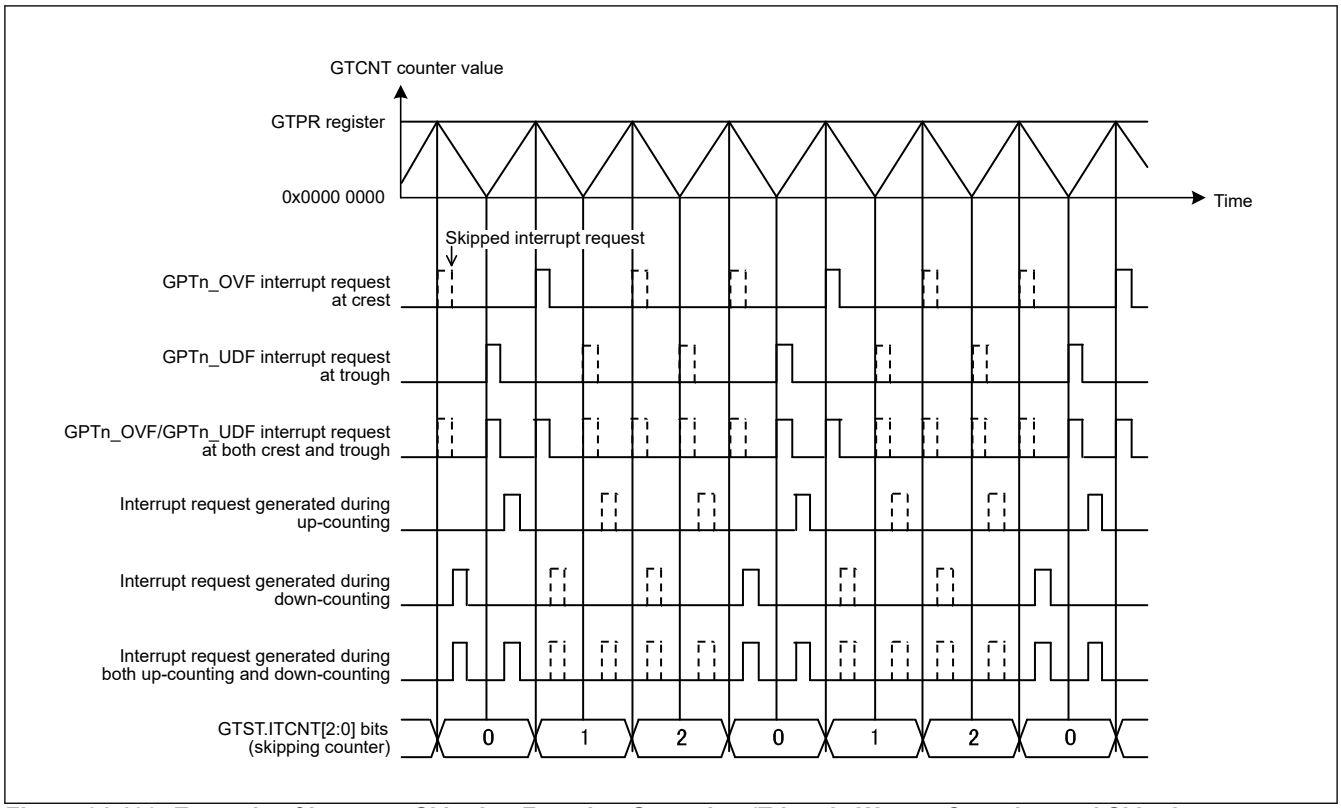


Figure 21.139 Example of Interrupt Skipping Function Operation (Triangle Waves, Counting and Skipping Crests, Skipping Count: 2)

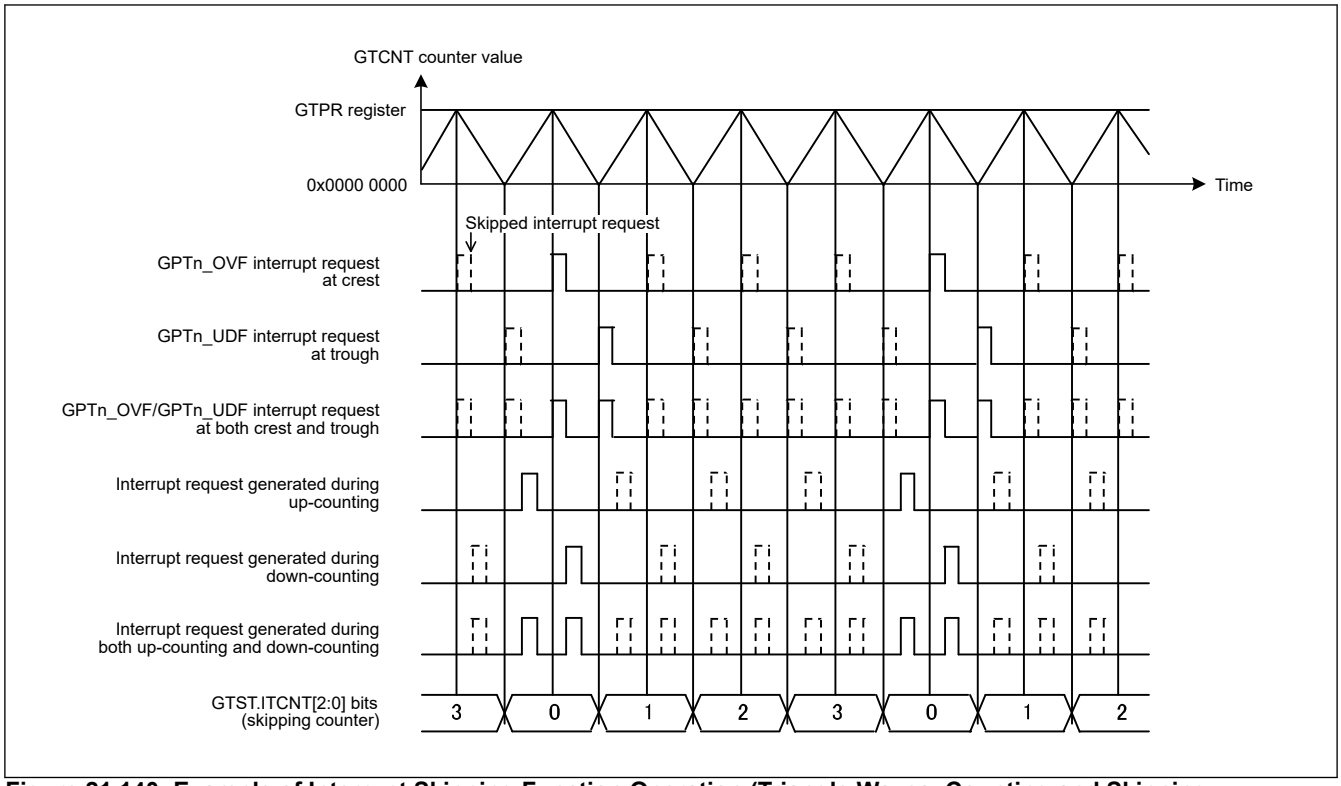


Figure 21.140 Example of Interrupt Skipping Function Operation (Triangle Waves, Counting and Skipping Troughs, Skipping Count: 3)

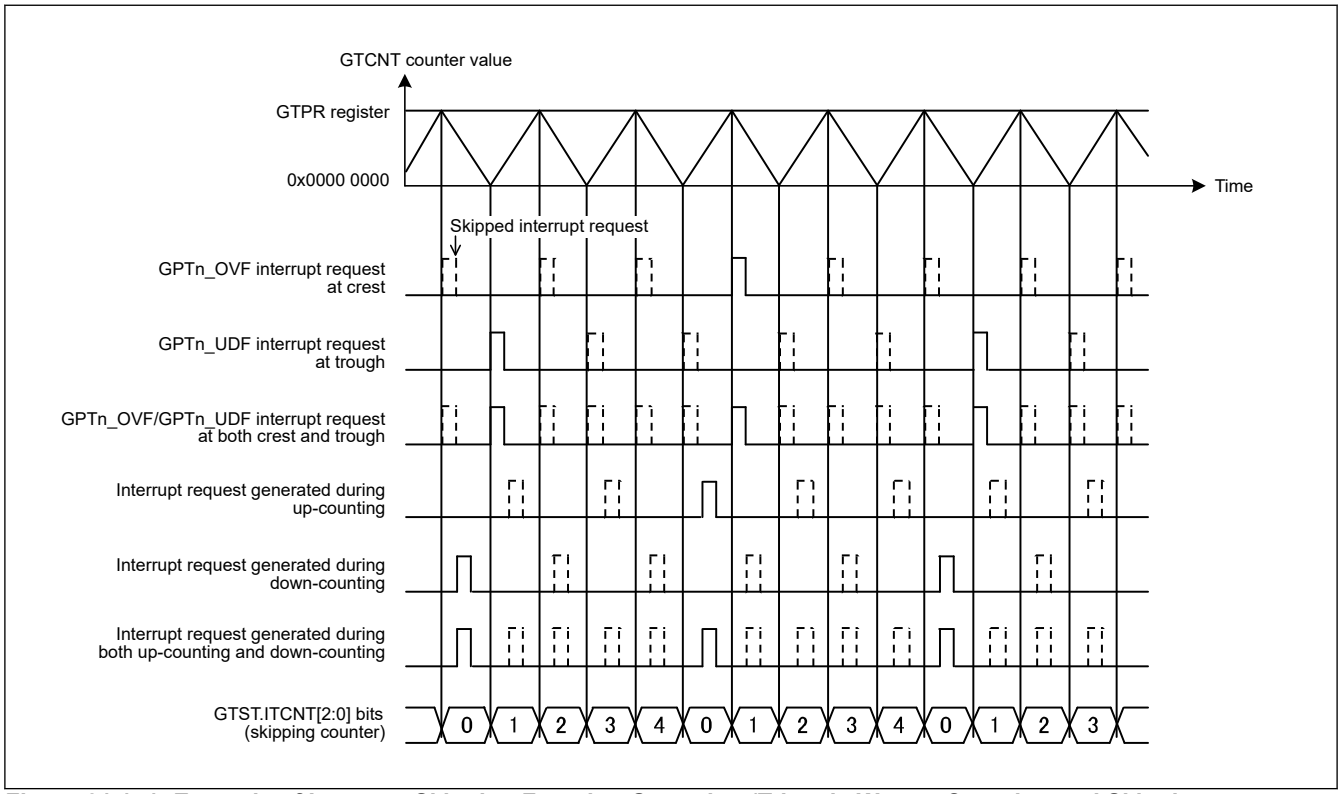


Figure 21.141 Example of Interrupt Skipping Function Operation (Triangle Waves, Counting and Skipping Both Troughs and Crests, Skipping Count: 4)

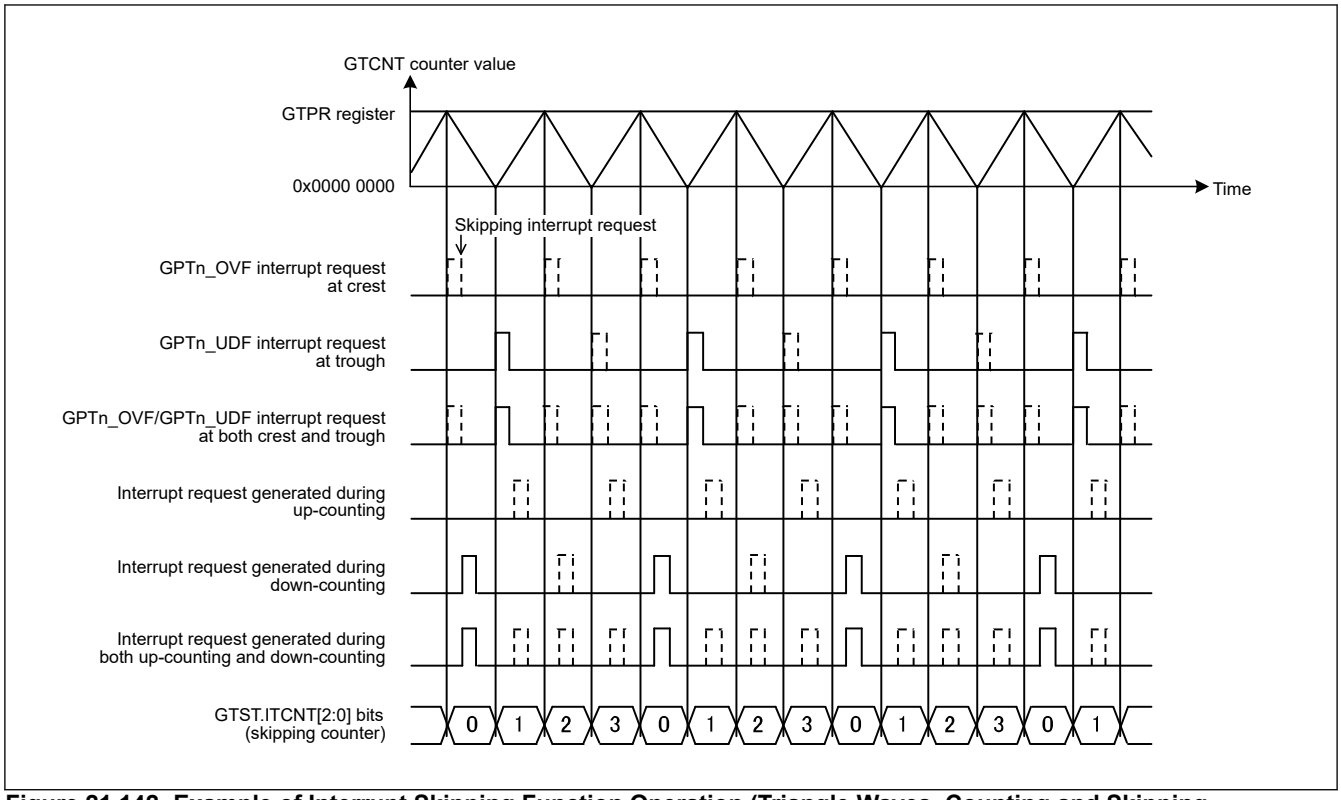


Figure 21.142 Example of Interrupt Skipping Function Operation (Triangle Waves, Counting and Skipping Both Troughs and Crests, Skipping Count: 3, Skipping Started at Up-Counting)

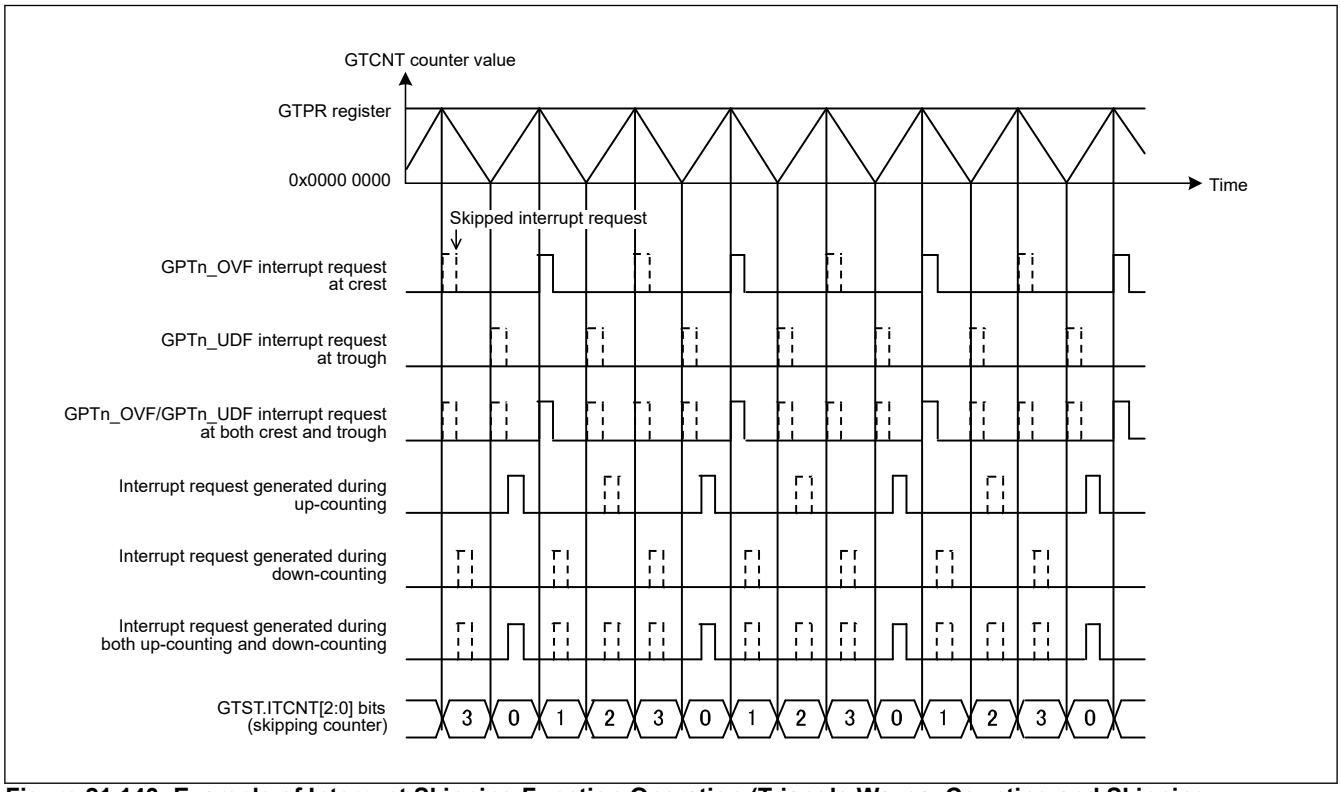


Figure 21.143 Example of Interrupt Skipping Function Operation (Triangle Waves, Counting and Skipping Both Troughs and Crests, Skipping Count: 3, Skipping Started at Down-Counting)

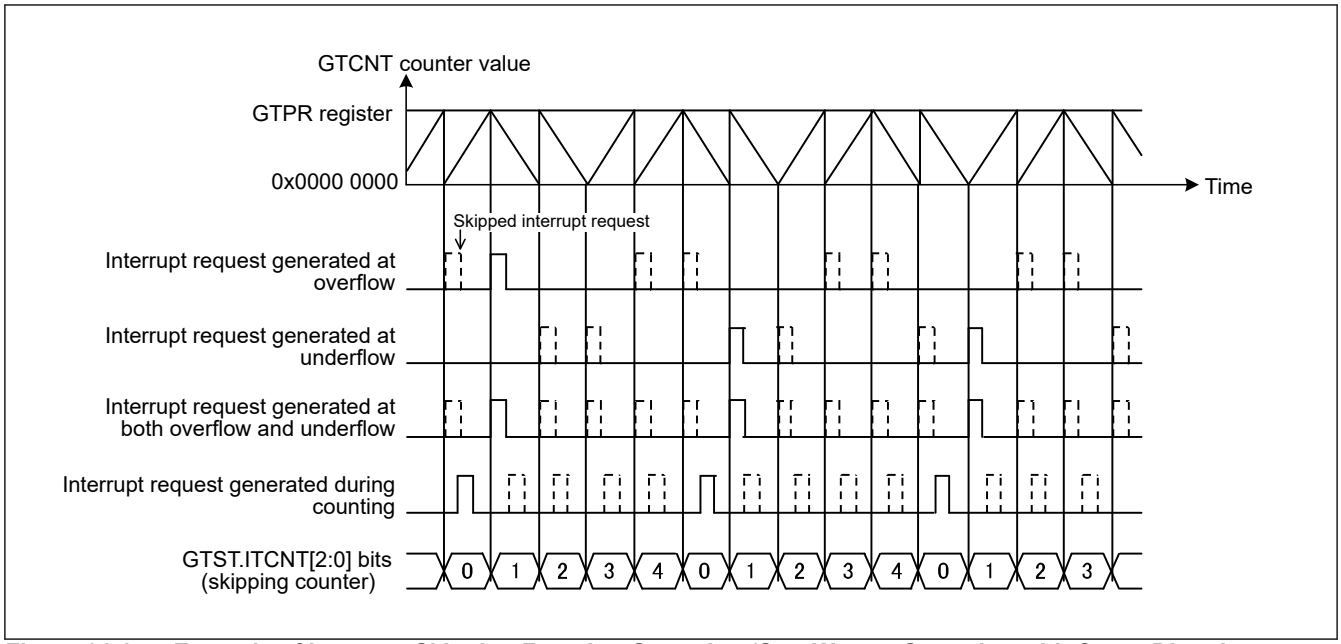


Figure 21.144 Example of Interrupt Skipping Function Operation (Saw Waves, Operation with Count Direction Changed, Counting and Skipping Both Overflows and Underflows, Skipping Count: 4)

21.4.3.2 Extended Interrupt Skipping Function

Overflow/underflow interrupt, compare match/input capture interrupt, A/D conversion start request, and buffer transfer can be skipped by counting the GTCNT counter overflow or underflow based on settings of the GTEITC, GTEITLI1, GTEITLI2 and GTEITLB registers.

Overflow/underflow interrupt, compare match/input capture interrupt, A/D conversion start request, and buffer transfer can individually be set for skipping and the skipping period using the GTEITLI1, GTEITLI2, and GTEITLB registers.

The skipping period is set, related to the operation of the two independent extended interrupt skipping counters (EITCNT1[3:0] and EITCNT2[3:0] bits in the GTEITC registers), as a period that either of the counter value for such skipping counters is other than 0 or other than the skipping count. The skipping period can also be set as a period that the both skipping counters are set as other than 0 or other than the skipping count for the counter value.

Figure 21.145 shows the counter operation for interrupt skipping by the GTITC register and extended interrupt skipping.

The counter operation for extended interrupt skipping is set by the GTEITC register.

The EITCNT1[3:0] bits are set for 0 as initial value and then they count the count source (in Figure 21.145, a crest is selected) selected by the extended interrupt counter 1 count source select bit (GTEITC.EIVTC1[1:0] bits), and repeat the count operation which returns to 0 when the skipping count (in Figure 21.145, count is 2) set by the extended interrupt skipping 1 skipping count setting bit (EIVTT1[3:0] bits) is achieved.

The EITCNT2[3:0] bits can be set for the initial value and they count the count source (in Figure 21.145, a trough is selected) selected by the extended interrupt counter 2 count source select bit (EIVTC2[1:0] bits), and repeat the count operation which return to 0 when the skipping count (in Figure 21.145 count is 2) set by the EIVTT2[3:0] bits is achieved. The initial value is set when the GTEITC register is written by the access of upper 16 bits or 32 bits while the setting for the extended interrupt skipping counter 2 is not counted (EIVTC2[1:0] bits are 00b) and when the write value to the EIVTC2[1:0] bits is other than 00b. When the initial value is set, the write value to the extended interrupt skipping counter 2 initial value bits (EITCNT2IV[3:0] bits) are set as the initial value for the EITCNT2[3:0] bits.

The extended interrupt skipping counter starts up-counting at the first count clock after the setting is modified from not counting to counting.

The interrupt skipping count counter (GTST.ITCNT[2:0] bits) for the interrupt skipping function set by the GTITC register is reset to 000b when the GTCNT counter operation is stopped: however, the EITCNT1[3:0] and EITCNT2[3:0] bits for the extended interrupt skipping function retain the value even after the GTCNT counter operation is stopped, and the counting can be resumed from the value before the counter is stopped. When values for the EITCNT1[3:0] and EITCNT2[3:0] bits are to be reset (0000b), set the EIVTC1[1:0] and EIVTC2[1:0] bits to the setting for not counting (not skipping) (00b).

When the skipping count is to be changed, change the count after stopping the skipping counter operation (set either of the EIVTC1[1:0] bit or the EIVTC2[1:0] bit to 00b).

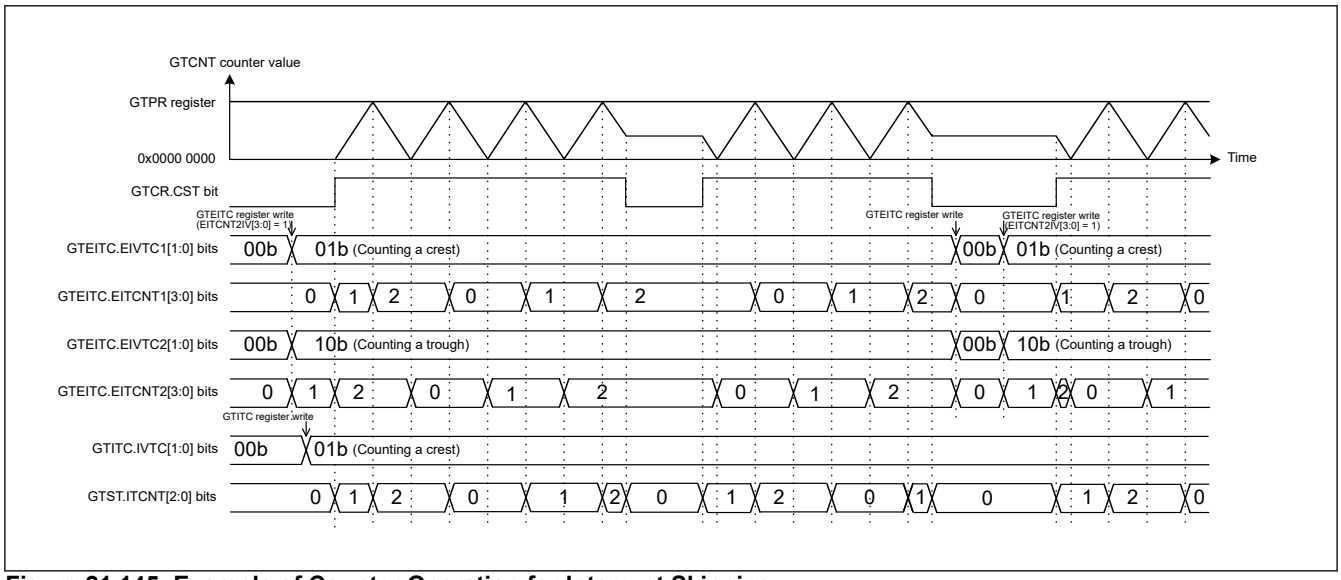


Figure 21.145 Example of Counter Operation for Interrupt Skipping

Interrupt skipping by the GTEITL1 register and A/D conversion start request skipping by the GTEITLI2 register can be performed simultaneously with skipping by the GTITC register or GTADCMSC register. A skipping period in this case is represented by OR-ed skipping periods of respective registers.

Figure 21.146 shows corresponding interrupt skipping operations by different registers are performed simultaneously.

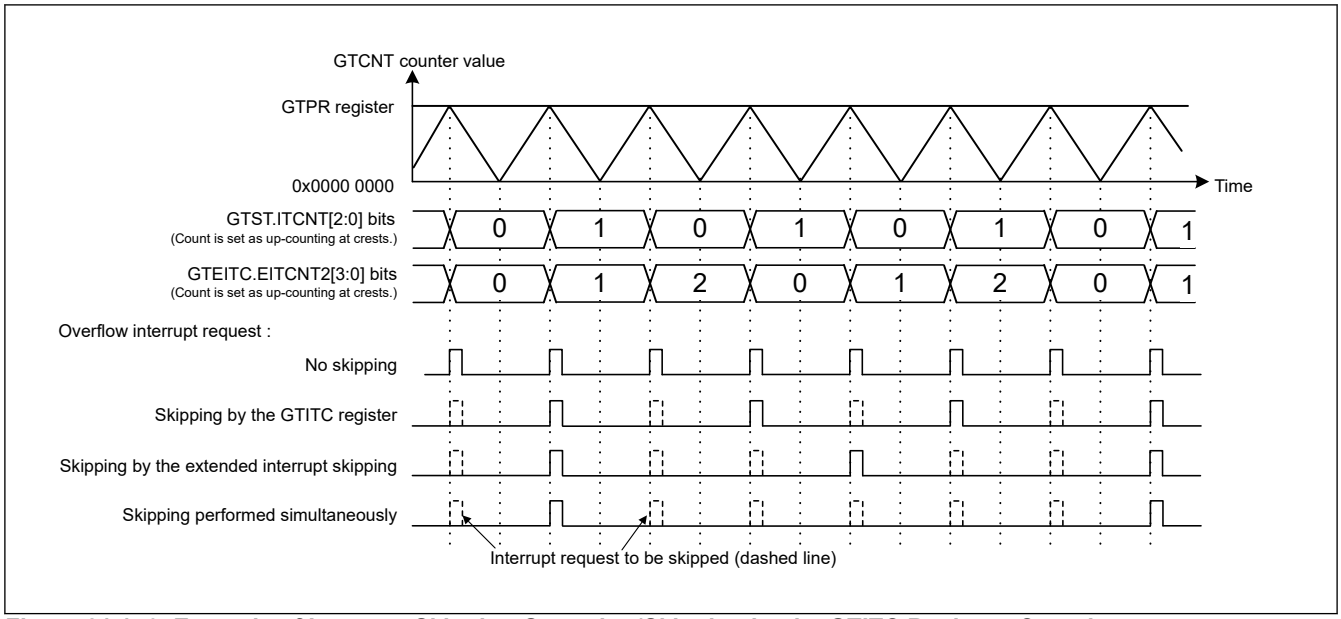


Figure 21.146 Example of Interrupt Skipping Operation(Skipping by the GTITC Register: Counting crests, Extended Interrupt Skipping: EIVTC1[1:0] bits = 00b, EIVTC2[1:0] bits = 01b, EITLV[2:0] bits = 010b)

When the extended interrupt skipping selected by GTEITLI1 is performed, the relevant status flag is also skipped. The skipping function is continued even if the status flag is set to 1.

When A/D conversion start request skipping selected by GTEITLI2 is performed, the relevant status flag is also skipped. The skipping function is continued even if the status flag is set to 1.

Skipping of updating of status flag and ELC event source outputs corresponding to A/D conversion start request for which skipping can be set up in the GTEITLI2 register is only based on the settings of the GTITC register and the extended interrupt skipping register, and has no connection with the setting of A/D conversion start request enable bit in the

GTINTAD register. The A/D conversion start request enable bit is used only for A/D conversion start request output(ELC event source output) after skipping.

A buffer transfer skipping by the GTEITLB register is performed in all of buffer operation which is enabled in the GTBER and GTDTCR registers and GTBER2 register, or all buffer operations performed by saw-wave one-shot pulse mode or triangle-wave PWM mode 3 or complementary PWM mode(excludes buffer transfer from GTCCRC, GTCCRE to GTCCRA).

An interrupt skipping and a buffer transfer skipping are operated individually. An interrupt output without performing a buffer transfer and performing a buffer transfer without an interrupt output can also be operated.

Figure 21.147 to Figure 21.154 show examples of extended skipping function operation.

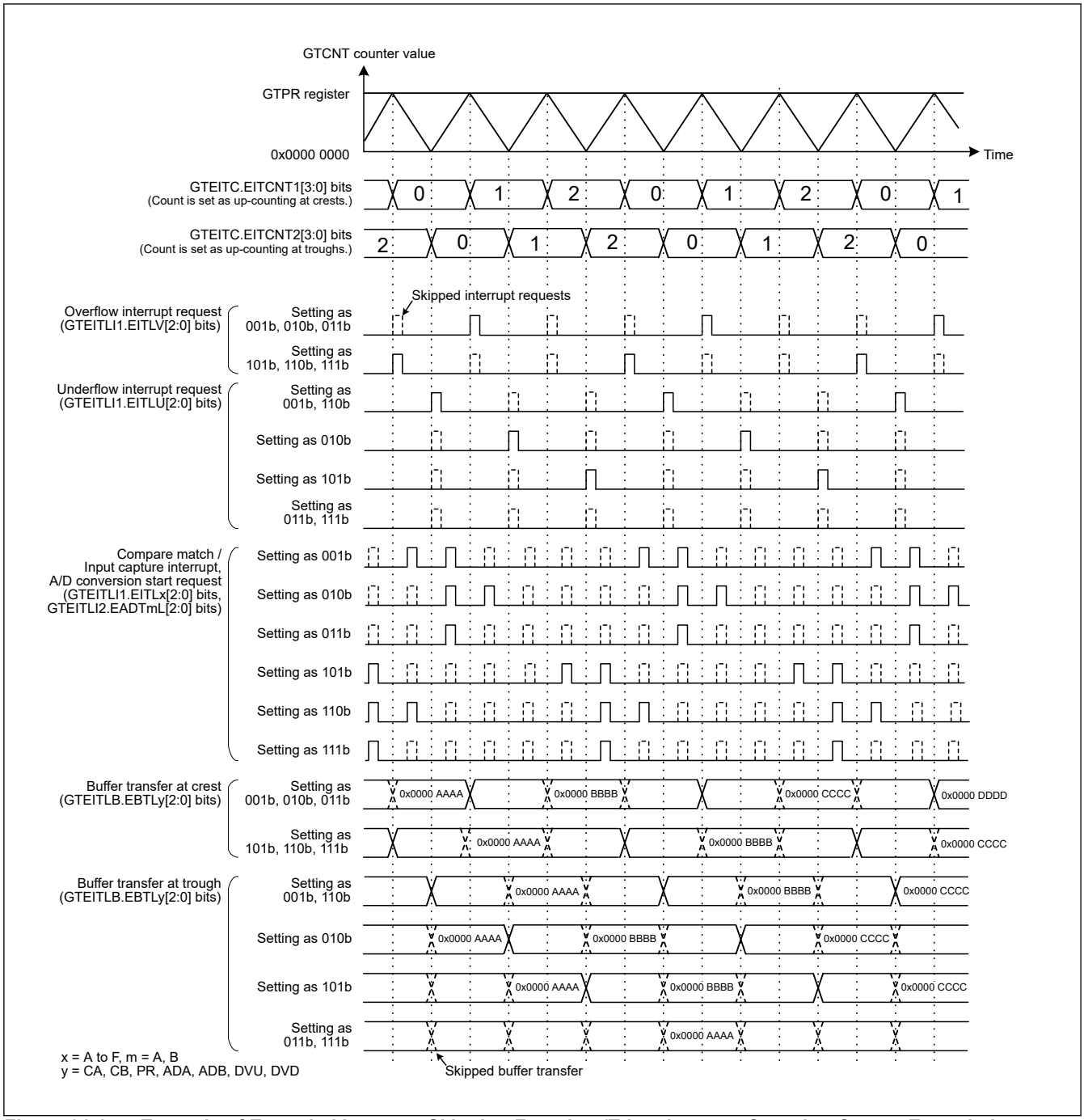


Figure 21.147 Example of Extended Interrupt Skipping Function (Triangle wave, Counting Crests, Extended Interrupt Skipping 1 Skipping Count: 2, Counting Troughs, Extended Interrupt Skipping 2 Skipping Count: 2, Extended Interrupt Skipping Counter 2 Initial Value: 0)

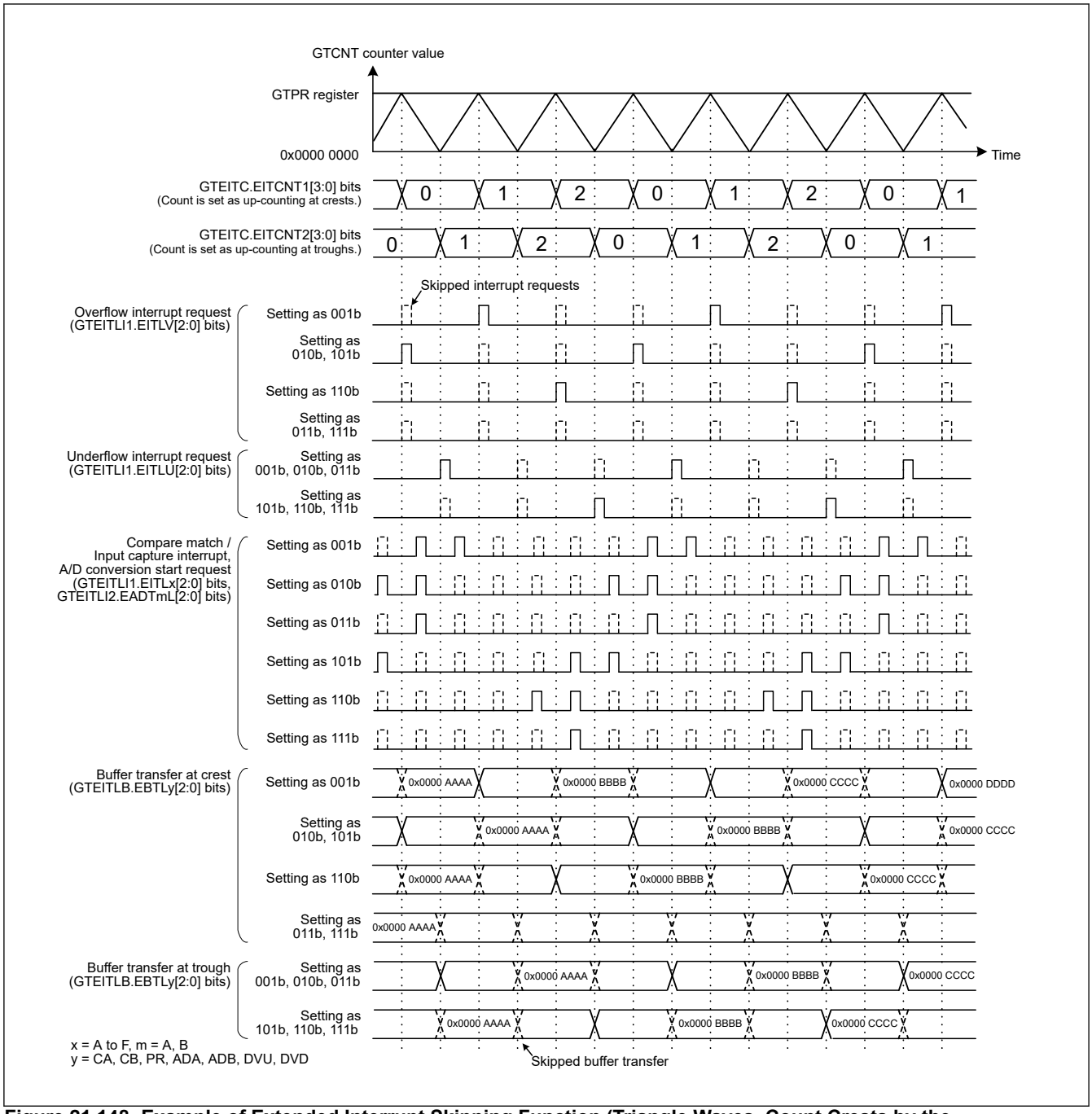


Figure 21.148 Example of Extended Interrupt Skipping Function (Triangle Waves, Count Crests by the Extended Interrupt Skipping 1 with Skipping Count: 2, Count Troughs by the Extended Interrupt Skipping 2 with Skipping Count: 2, Extended Interrupt Skipping Counter 2 Initial Value: 1)

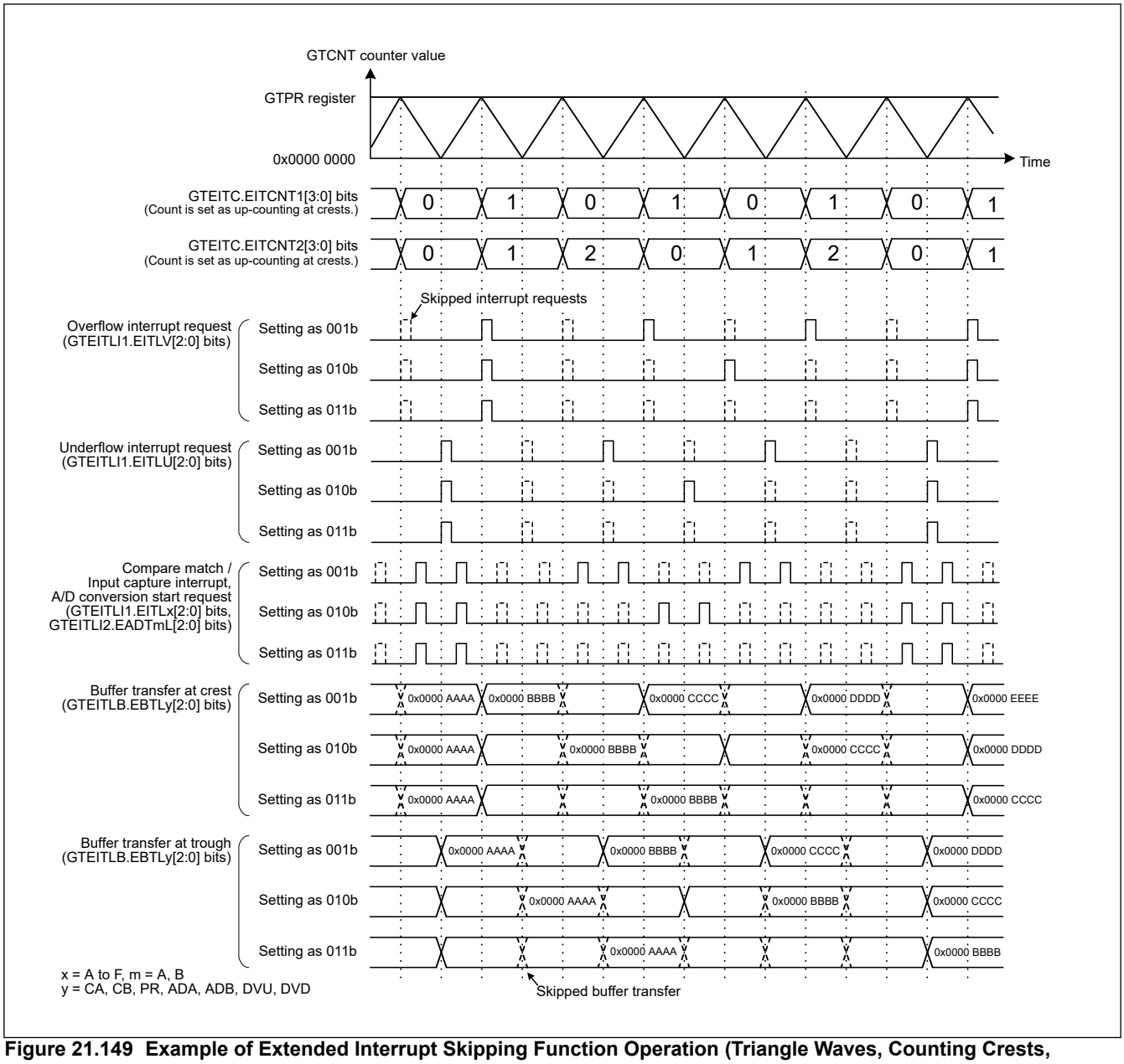


Figure 21.149 Example of Extended Interrupt Skipping Function Operation (Triangle Waves, Counting Crests, Extended Interrupt Skipping 1 Skipping Count: 1, Counting Crests, Extended Interrupt Skipping 2 Skipping Count: 2, Extended Interrupt Skipping Counter 2 Initial Value: 0, Skipped in the Period other than GTEITC.EITCNTk bits (k = 1, 2) as 0)

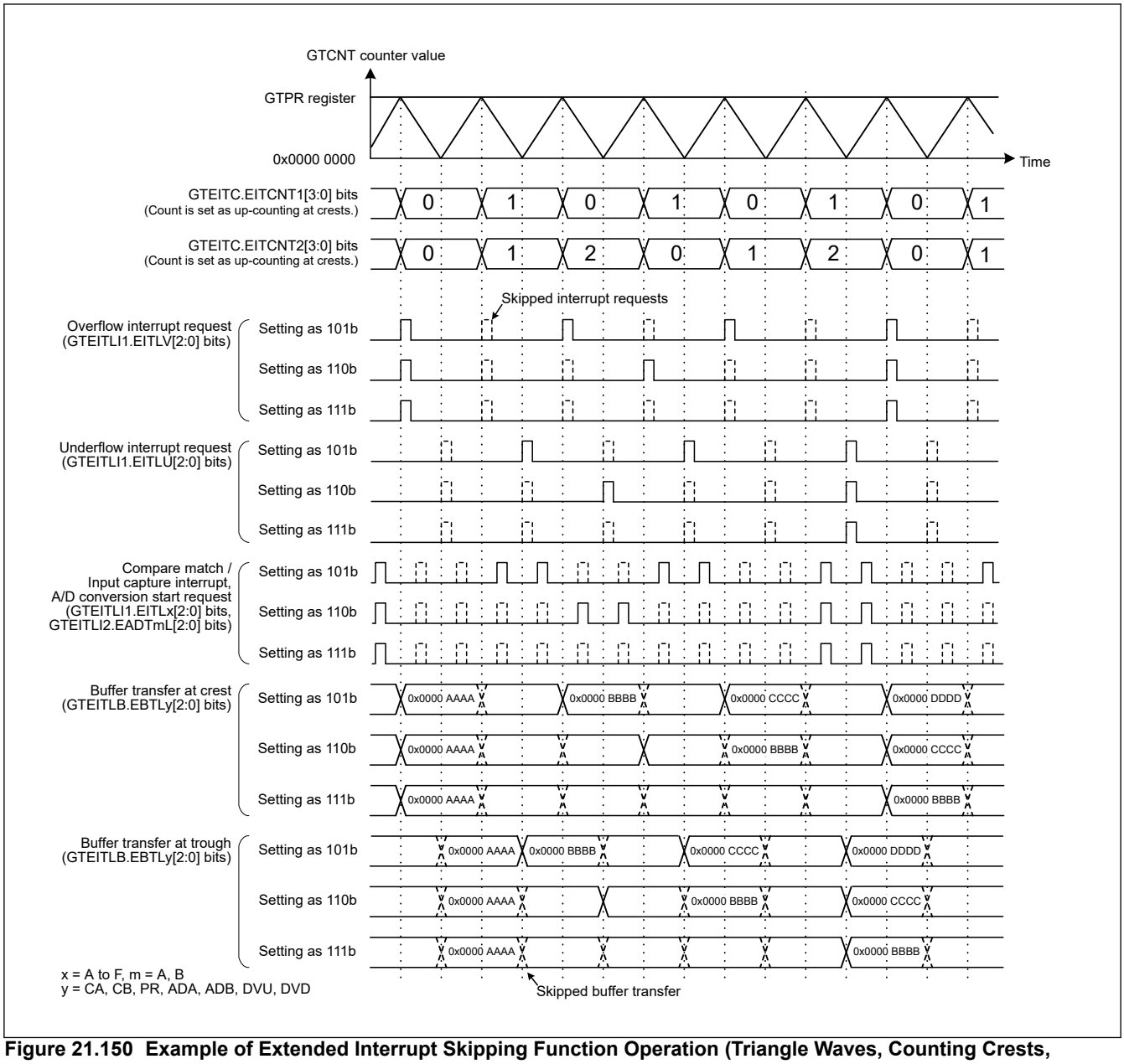


Figure 21.150 Example of Extended Interrupt Skipping Function Operation (Triangle Waves, Counting Crests, Extended Interrupt Skipping 1 Skipping Count: 1, Counting Crests, Extended Interrupt Skipping 2 Skipping Count: 2, Extended Interrupt Skipping Counter 2 Initial Value: 0, Skipped in the Period other than GTEITC.EITCNTk bits (k = 1, 2) as GTEITC.EIVTTk bits.)

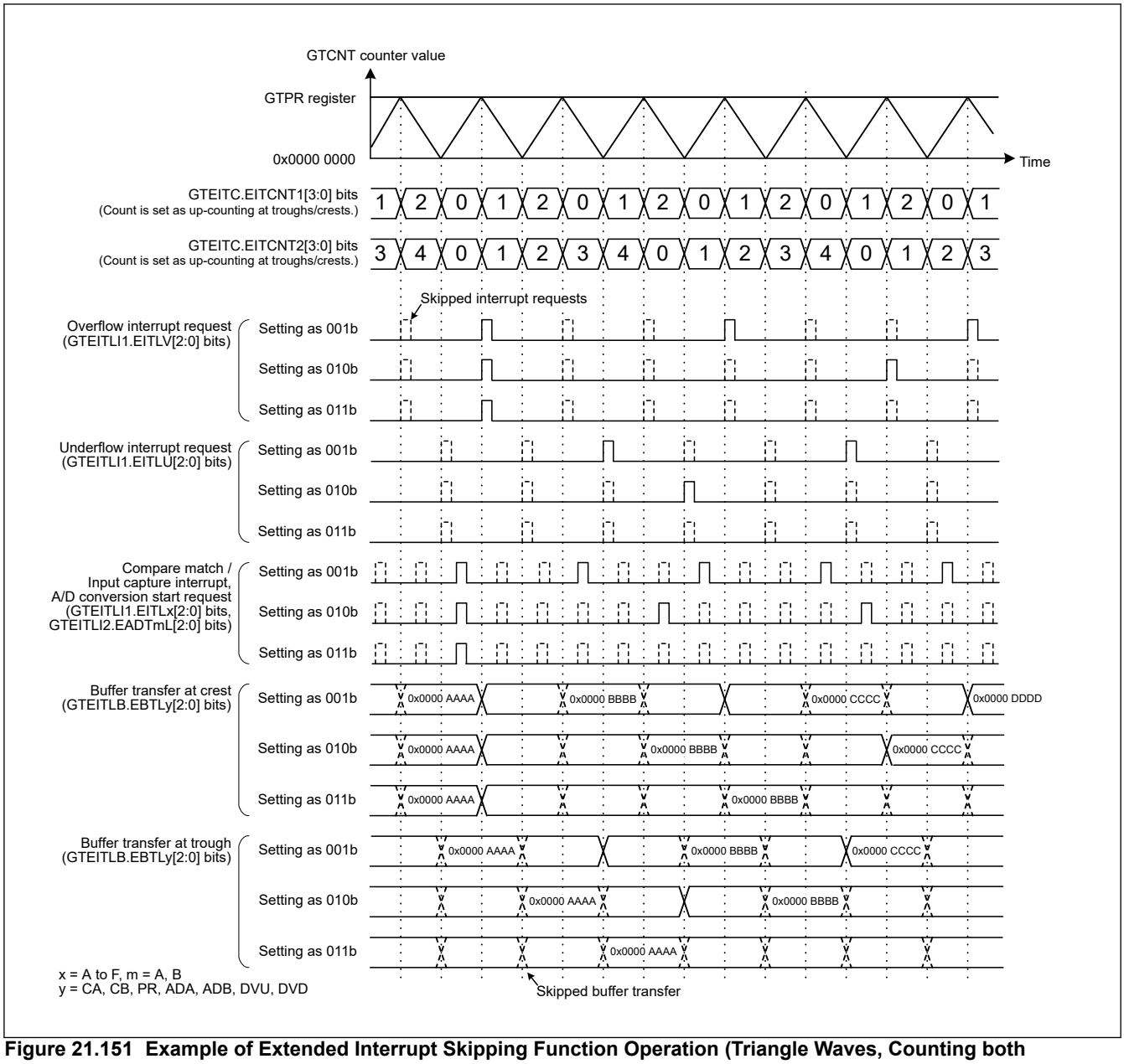


Figure 21.151 Example of Extended Interrupt Skipping Function Operation (Triangle Waves, Counting both Crests and Troughs, Extended Interrupt Skipping 1 Skipping Count: 2, Counting both Crests and Troughs, Extended Interrupt Skipping 2 Skipping Count: 4, Extended Interrupt Skipping Counter 2 Initial Value: 0, Skipped in the Period other than GTEITC.EITCNTk bits (k = 1, 2) as 0)

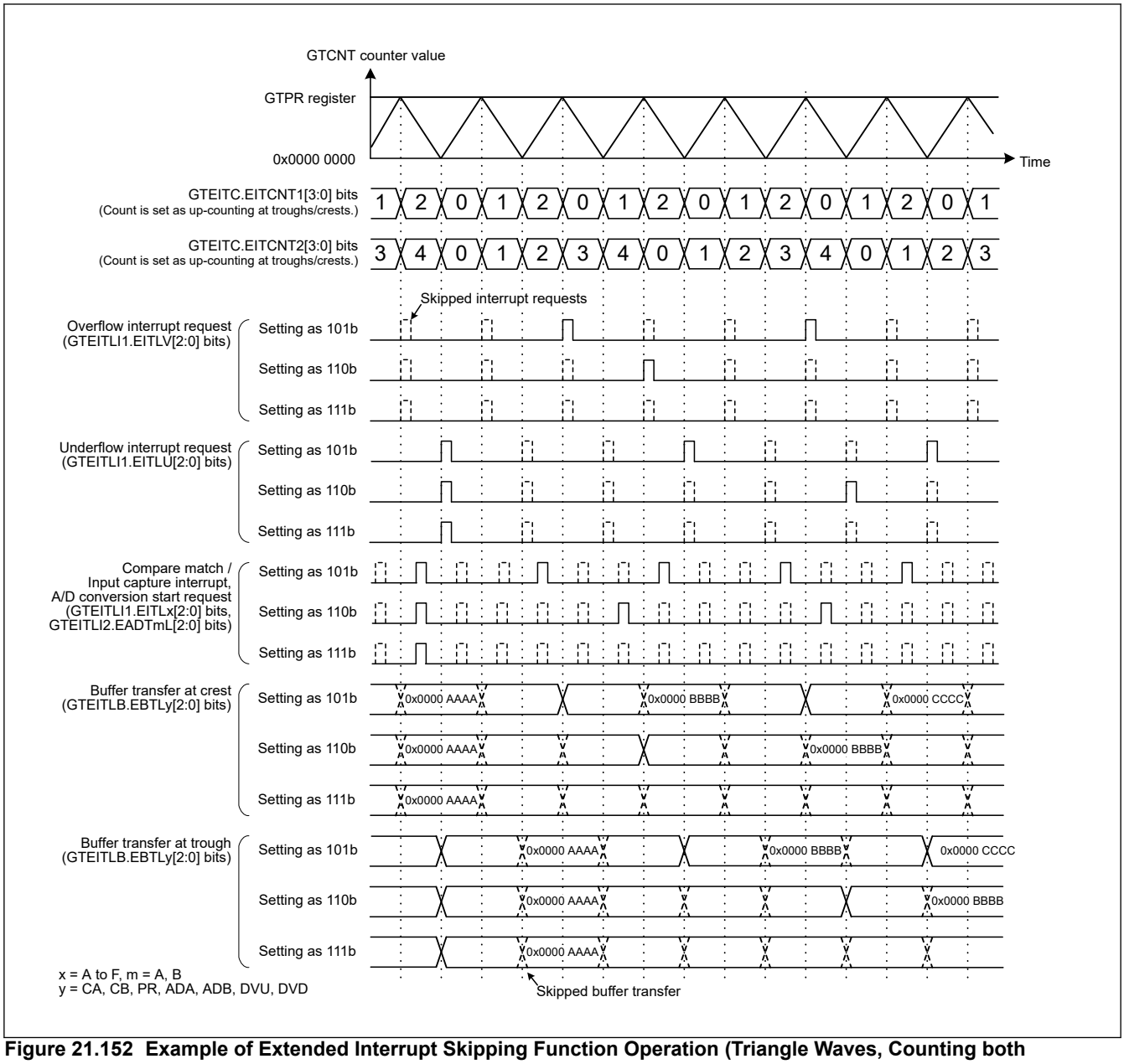


Figure 21.152 Example of Extended Interrupt Skipping Function Operation (Triangle Waves, Counting both Crests and Troughs, Extended Interrupt Skipping 1 Skipping Count: 2, Counting both Crests and Troughs, Extended Interrupt Skipping 2 Skipping Count: 4, Extended Interrupt Skipping Counter 2 Initial Value: 0, Skipped in the Period other than GTEITC.EITCNTk bits (k = 1, 2) as GTEITC.EIVTTk bit.)

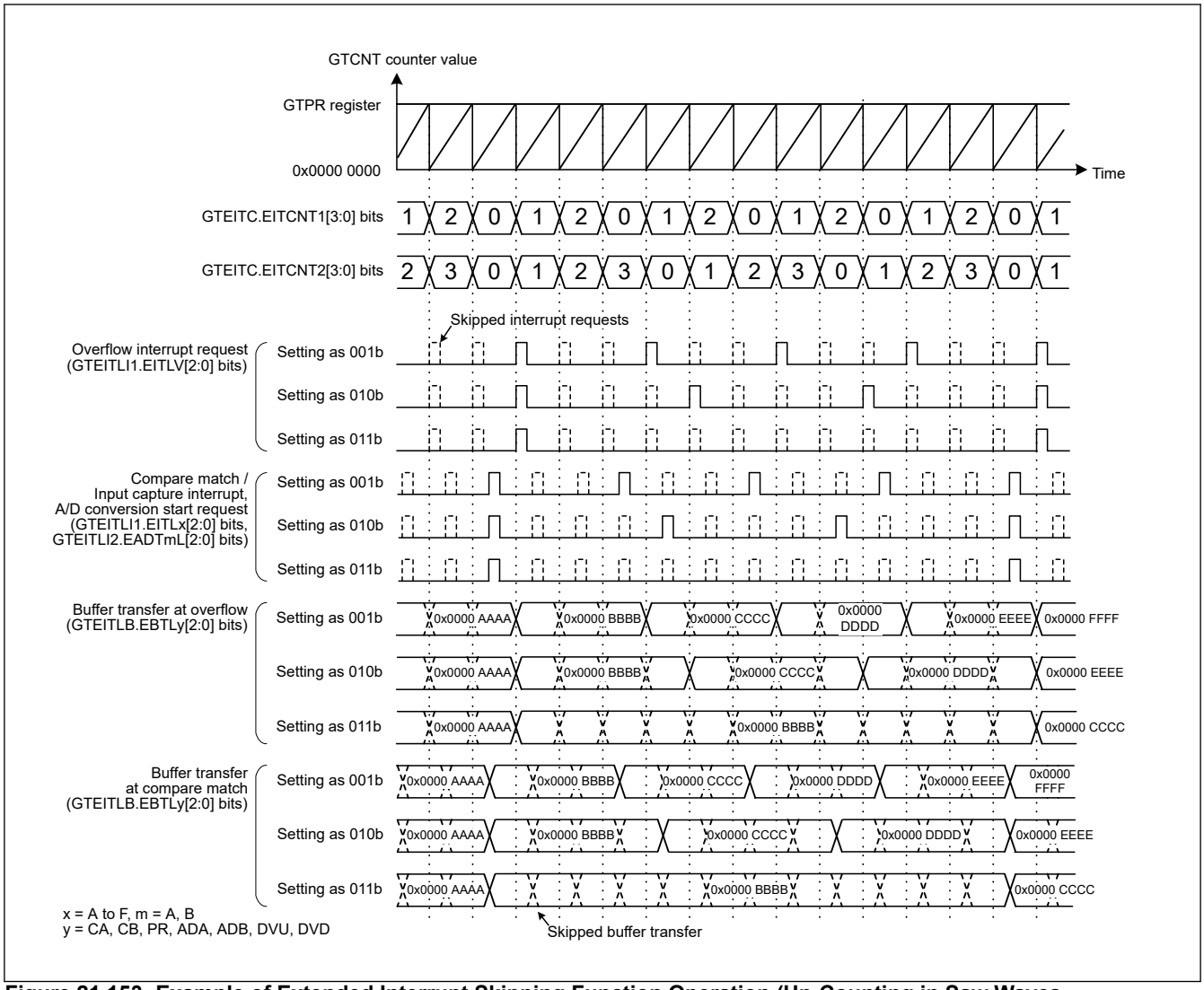


Figure 21.153 Example of Extended Interrupt Skipping Function Operation (Up-Counting in Saw Waves, Counting Overflow, Extended Interrupt Skipping 1 Skipping Count: 2, Counting Overflow, Extended Interrupt Skipping 2 Skipping Count: 3, Extended Interrupt Skipping Counter 2 Initial Value: 0, Skipped in the Period other than GTEITC.EITCNTk bits (k = 1, 2) as 0)

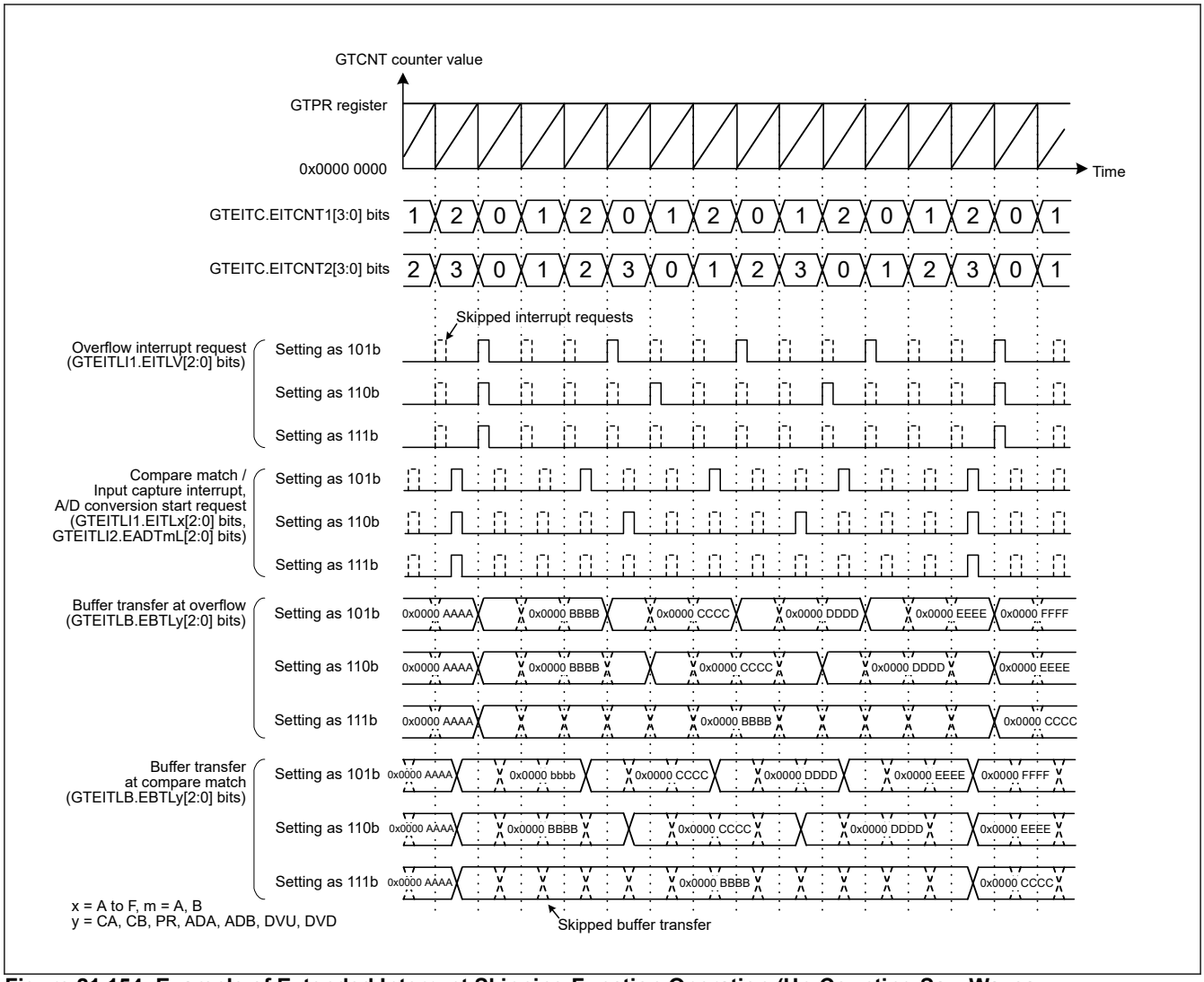


Figure 21.154 Example of Extended Interrupt Skipping Function Operation (Up-Counting Saw Waves, Counting Overflow, Extended Interrupt Skipping 1 Skipping Count: 2, Counting Overflow, Extended Interrupt Skipping 2 Skipping Count: 3, Extended Interrupt Skipping Counter 2 Initial Value: 0, Skipped in the Period other than GTEITC.EITCNTk bits (k = 1, 2) as GTEITC.EIVTTk bits.)

Figure 21.155 shows an example of the extended interrupt skipping operation on the input capture. When the setting is for the input capture operation setting (GTCR.ICDS bit = 0) at count stop of the GTCNT counter, the interrupt by the input capture and the extended skipping on buffer transfer is enabled even at count stop of the GTCNT counter.

When the input capture is generated at count stop of the GTCNT counter by setting the ICDS bit as 0, an interrupt and a buffer transfer are skipped if the skipping counter value is the same with the skipping period set in a corresponding interrupt skipping function select bit.

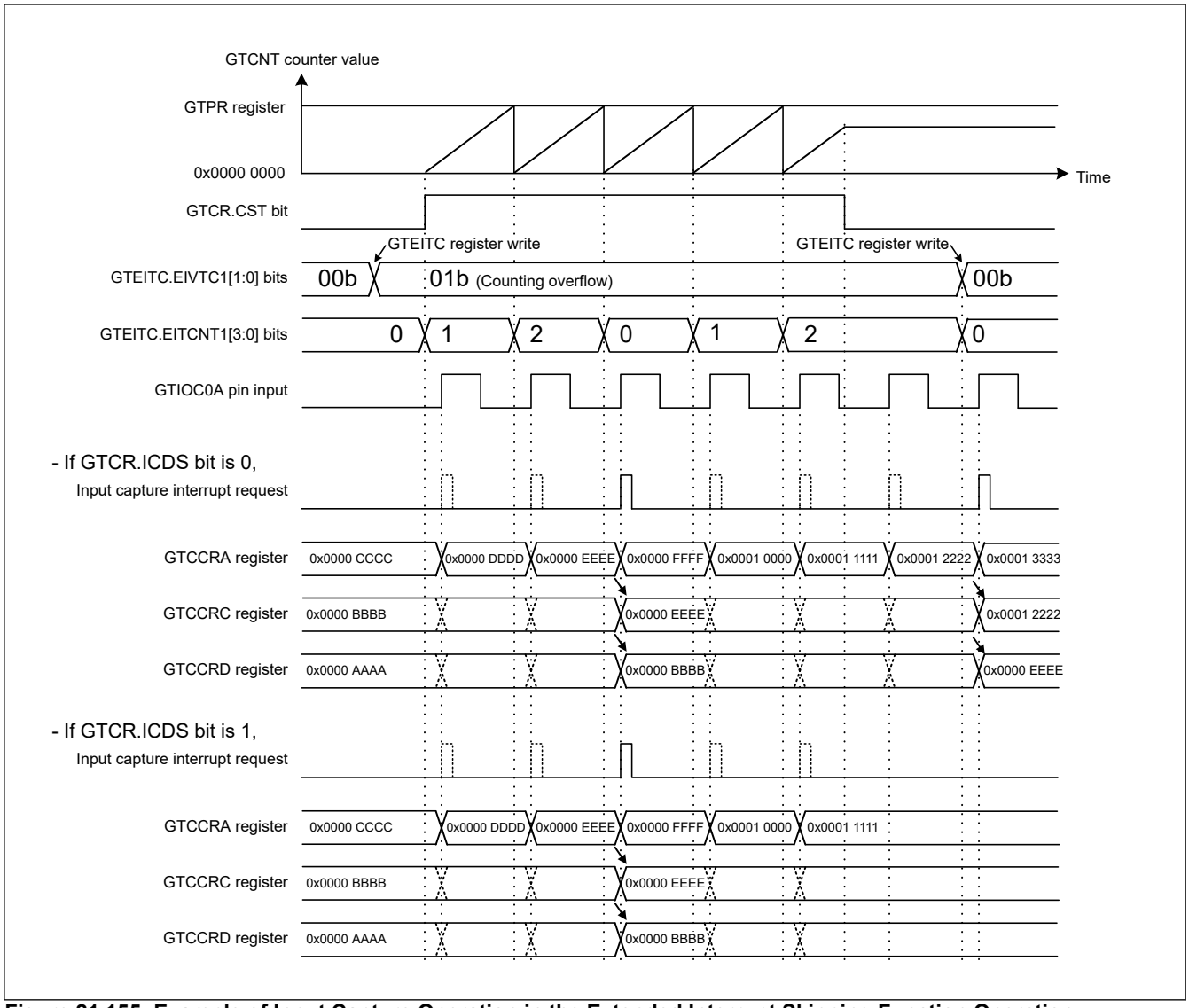


Figure 21.155 Example of Input Capture Operation in the Extended Interrupt Skipping Function Operation (Up-Counting Saw Waves, Counting Overflow, Extended Interrupt Skipping 1 Skipping Count: 2, Skipped in the Period other than the EITCNT1 as 0, Input Capture at the Input Rising)

Table 21.69 shows an example of setting the extended interrupt skipping.

The extended interrupt skipping counter 2 initial value is set by the written value to the ETICNT2IV[3:0] bits which is applied to change the setting from not counting the extended interrupt skipping counter 2 count source (GTEITC.EIVTC2[1:0] bits = 00b) to counting (EIVTC2[1:0] bits = other than 00b). Writing to the extended interrupt skipping counter 2 initial value bit (ETICNT2IV[3:0] bits) is performed only when the setting of the above mentioned extended interrupt skipping counter 2 initial value is written.

Table 21.69 Example for Setting the Extended Interrupt Skipping (1 of 2)

No.	Step Name	Description
1	Set GTCNT counter operation, Set buffer operation, Set compare match value	See section 21.3.2. Buffer Operation, section 21.3.3. PWM Output Operating Mode, and so on.
2	Set the extended interrupt skipping function	Select the skipping counter used for skipping and the skipping period with the skipping function select bit for the interrupt, for the A/D conversion start request, and for a buffer transfer, all which are to be skipped, in the GTEITLI1, GTEITLI2, and GTEITLB registers respectively.

Table 21.69 Example for Setting the Extended Interrupt Skipping (2 of 2)

No.	Step Name	Description
3	Set the extended skipping counter	With the GTEITC register, set a count source of the skipping counter used for skipping, skipping count, and the skipping counter 2 initial value as the following order. Set EIVTCk[1:0] bits (k = 1,2) as value other than 00b, and set EIVTTk[3:0] bits as value other than 0000b. When the skipping counter 2 is used, change EIVTC2[1:0] bits from 00b to value other than 00b as well as set EITCNT2IV[3:0] bits to the skipping counter 2 initial value.
4	Start count operation	Set GTCR.CST to 1 to start count operation.
5	Set buffer value for each cycle	See section 21.3.2. Buffer Operation, section 21.3.3. PWM Output Operating Mode, and so on.

21.4.3.3 A/D conversion Start Request Compare Match Skipping Function

The A/D conversion start request and GTADTR register buffer transfer can be skipped by counting the compare matches of GTADTRA register and GTADTRB register based on settings of the GTADCMSC register and GTADCMSS register.

The skipping period is set, related to the operation of the two independent A/D conversion start request compare match skipping counters (ADCMSCNT1[3:0] and ADCMSCNT2[3:0] bits in the GTADCMSC register), as a period that either of the counter value for such skipping counters is other than 0 or other than the skipping count. The skipping period can also be set as a period that the both skipping counters are set as other than 0 or other than the skipping count for the counter value.

Figure 21.156 shows the counter operation for A/D conversion start request compare match skipping.

The counter operation for A/D conversion start request compare match skipping is set by the GTADCMSC register.

The ADCMSCNT1[3:0] bits count the count source (in Figure 21.156, a crest is selected) selected by the A/D conversion start request compare match skipping counter 1 count source select bit (GTADCMSC.ADCMSC1[1:0] bits), and repeat the count operation which returns to 0 when the skipping count (in Figure 21.156, count is 2) set by the A/D conversion start request compare match skipping 1 skipping count setting bit (ADCMST1[3:0] bits) is achieved.

The ADCMSCNT2[3:0] bits count the count source (in Figure 21.156, a trough is selected) selected by the A/D conversion start request compare match skipping counter 2 count source select bit (GTADCMSC.ADCMSC2[1:0] bits), and repeat the count operation which returns to 0 when the skipping count (in Figure 21.156, count is 2) set by the A/D conversion start request compare match skipping 2 skipping count setting bit (ADCMST2[3:0] bits) is achieved.

The ADCMSCNTk[3:0] bits (k = 1, 2) can be set for the initial value. The initial value is set when the GTADCMSC register is written by the access of 16 bits or 32 bits while the setting for the A/D conversion start request compare match skipping counter k is not counted (ADCMSCk [1:0] bits are 00b) and the write value to the ADCMSCk[1:0] bits is other than 00b. When the initial value is set, the write value to the A/D conversion start request compare match skipping counter k initial value bits (ADCMSCNTkIV[3:0] bits) are set as the initial value for the ADCMSCNT2[3:0] bits.

The A/D conversion start request compare match skipping counter starts up-counting at the first count clock after the setting is modified from not counting to counting.

The ADCMSCNT1[3:0] and ADCMSCNT2[3:0] bits for the A/D conversion start request compare match skipping function retain the value even after the GTCNT counter operation is stopped, and the counting can be resumed from the value before the counter is stopped. When values for the ADCMSCNT1[3:0] and ADCMSCNT2[3:0] bits are to be reset (0000b), set the ADCMSC1[1:0] and ADCMSC2[1:0] bits to the setting for not counting (not skipping) (00b).

When the skipping count is to be changed, change the count after stopping the skipping counter operation (set either of the ADCMSC1[1:0] bit or the ADCMSC 2[1:0] bit to 00b).

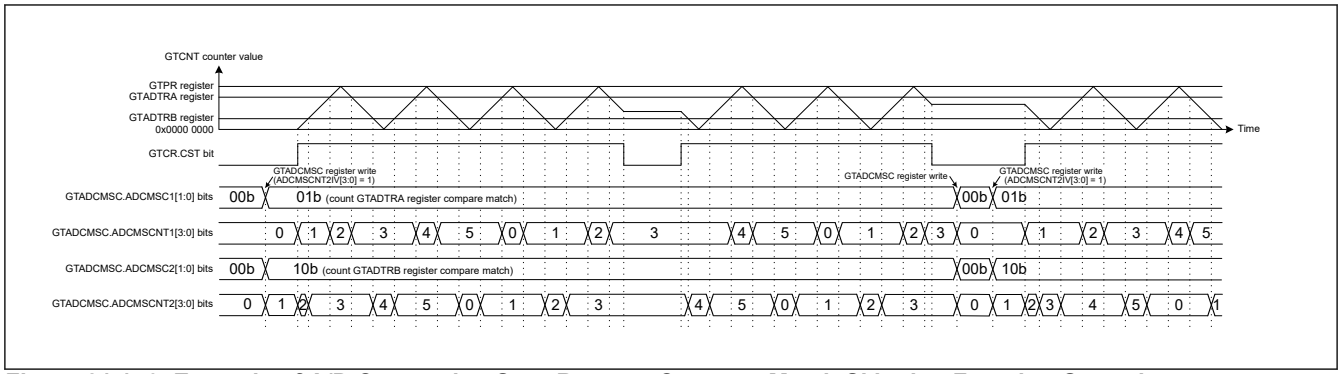


Figure 21.156 Example of A/D Conversion Start Request Compare Match Skipping Function Operation

A/D conversion start request skipping by the ADCMSC register can be performed simultaneously with skipping by the GTITC register or GTEITC register. A skipping period in this case is represented by OR-ed skipping periods of respective registers.

Figure 21.157 shows corresponding interrupt skipping operations and A/D conversion start request skipping operation are performed simultaneously.

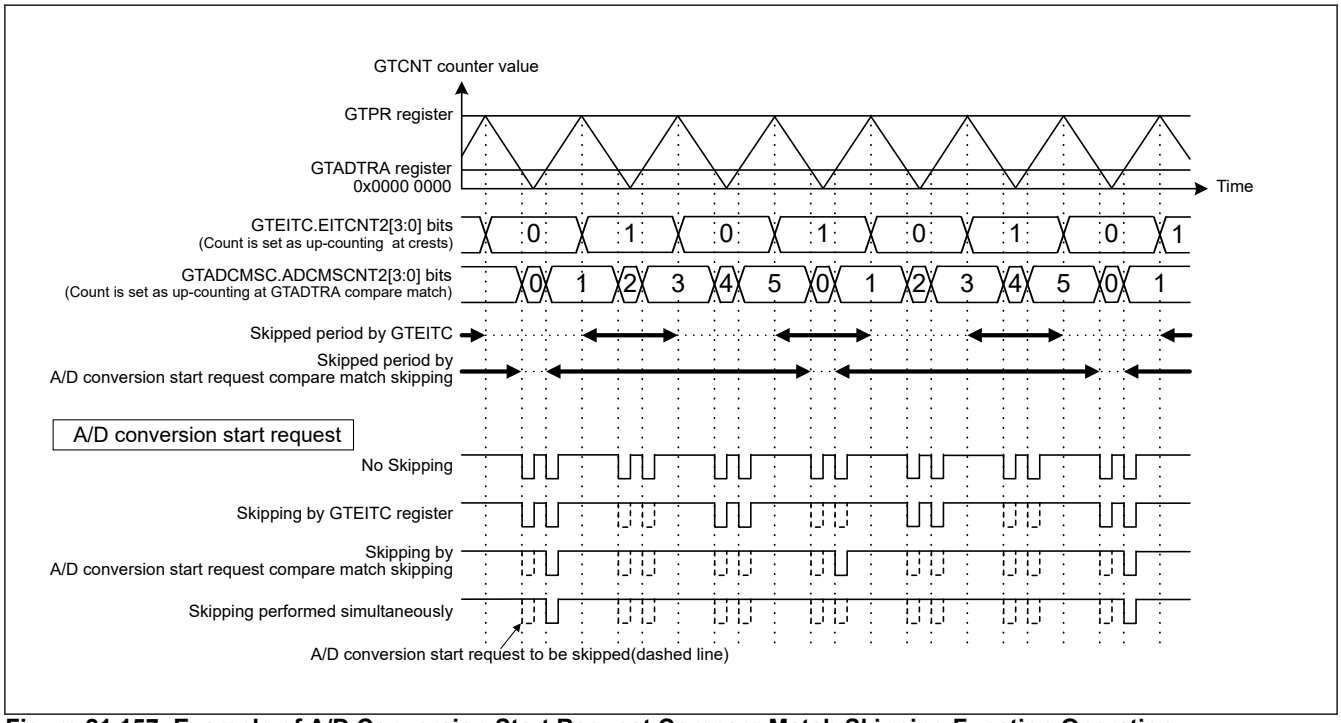


Figure 21.157 Example of A/D Conversion Start Request Compare Match Skipping Function Operation (Extended Interrupt Skipping: EIVTC2[1:0] = 01b, EADTAL[2:0] = 010b, A/D Conversion Start Request Compare Match Skipping: ADCMSC2[1:0] = 01b, ADCMSAL[2:0] = 010b)

When A/D conversion start request skipping which can be set by the GTADCMSS register is performed, updating of status flag and the ELC event output depend on the A/D conversion start request enable bit in the GTINTAD register. All the operations by A/D conversion start request which are set as disable by the GTINTAD register is not performed.

A buffer transfer skipping by the GTADCMSS register is performed in all of buffer operation which is enabled in the GTBER and GTBER2 register, or all buffer operations performed by saw-wave one-shot pulse mode or triangle-wave PWM mode 3 or complementary PWM mode(excludes buffer transfer from GTCCRC, GTCCRE to GTCCRA).

An A/D conversion start request skipping and a buffer transfer skipping are operated individually.

Figure 21.158, Figure 21.159 show examples of extended skipping function operation.

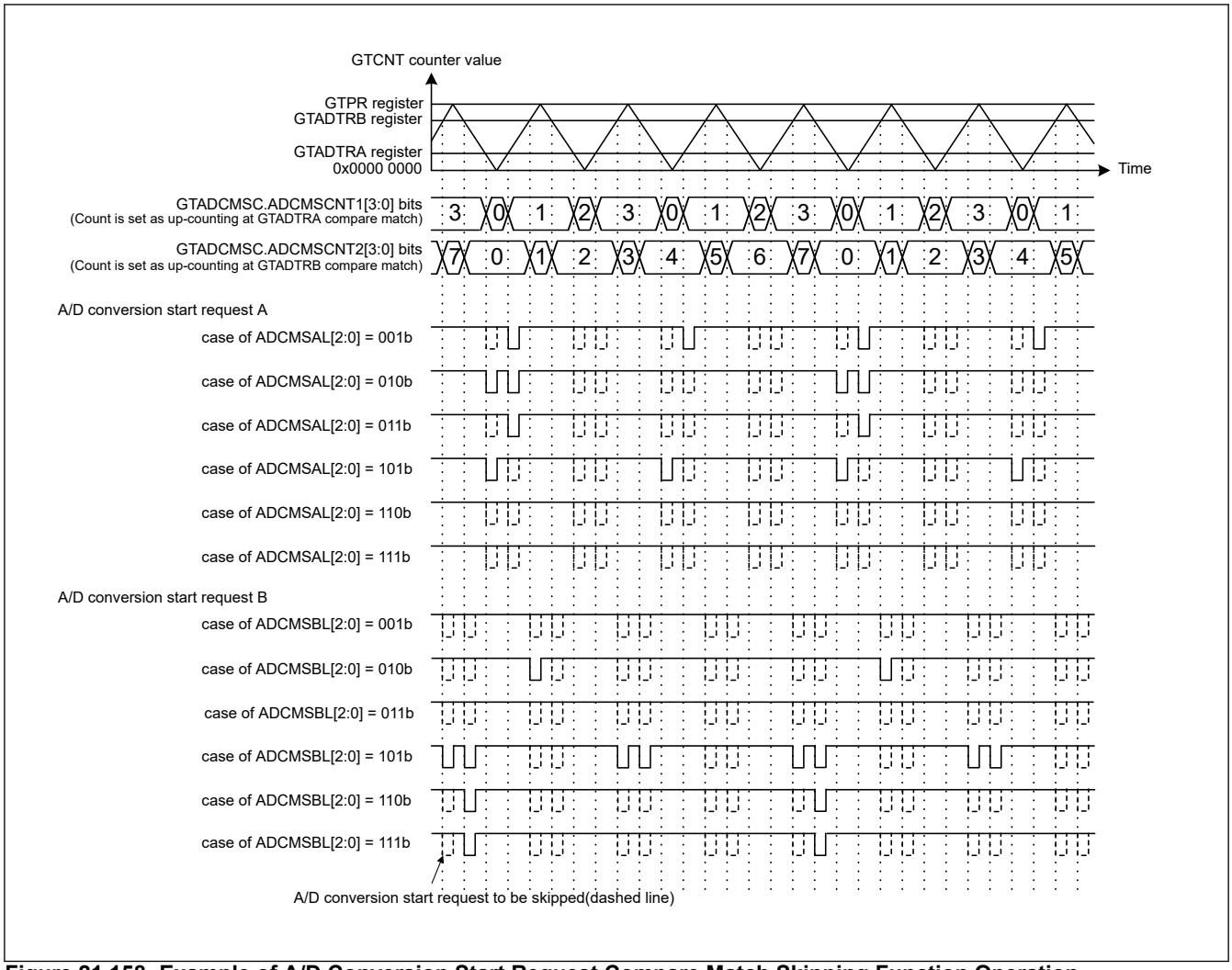


Figure 21.158 Example of A/D Conversion Start Request Compare Match Skipping Function Operation (Triangle wave, A/D Conversion Start Request Compare Match Skipping 1 Skipping Count: 3 Counting GTADTRA compare match, A/D Conversion Start Request Compare Match Skipping 2 Skipping Count: 7 Counting GTADTRB compare match)

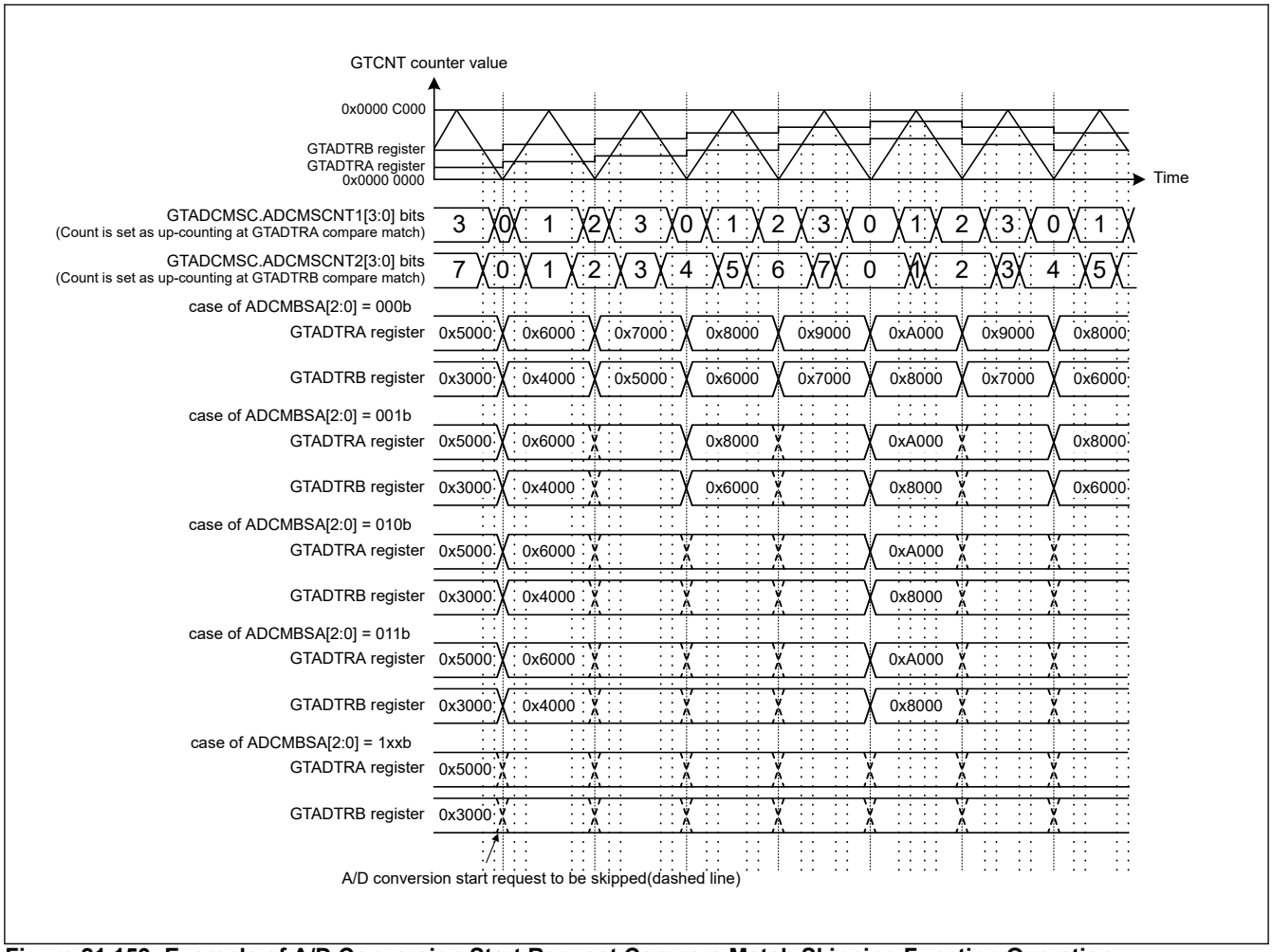


Figure 21.159 Example of A/D Conversion Start Request Compare Match Skipping Function Operation (Triangle wave, A/D Conversion Start Request Compare Match Skipping 1 Skipping Count: 3 Counting GTADTRA compare match, A/D Conversion Start Request Compare Match Skipping 2 Skipping Count: 7 Counting GTADTRB compare match, Buffer transfer of GTADTRA and GTADTRB at trough)

Table 21.70 shows an example of setting the A/D conversion start request compare match skipping.

The A/D conversion start request compare match skipping counter 2 initial value is set by the written value to the ADCMSCNT2IV[3:0] bits which is applied to change the setting from not counting the extended interrupt skipping counter 2 count source (GTADCMSC.ADCMSC2[1:0] bits = 00b) to counting (ADCMSC2[1:0] bits = other than 00b). Writing to the A/D conversion start request compare match skipping counter 2 initial value bit (ADCMSCNT2IV[3:0] bits) is performed only when the setting of the above mentioned A/D conversion start request compare match skipping counter 2 initial value is written.

Table 21.70 Example for Setting the A/D Conversion Start Request Compare Match Skipping (1 of 2)

No.	Step Name	Description
1	Set GTCNT counter operation, Set buffer operation, Set compare match value	See section 21.3.2. Buffer Operation, section 21.3.3. PWM Output Operating Mode, section 21.5. A/D Conversion Start Request, and so on.
2	Set the A/D conversion start request compare match skipping function	Select the skipping counter and the skipping period with the skipping function select bit for the A/D conversion start request or the bit for the buffer transfer to be skipped in the GTADCMSS register.

Table 21.70 Example for Setting the A/D Conversion Start Request Compare Match Skipping (2 of 2)

No.	Step Name	Description
3	Set the A/D conversion start request compare match skipping counter k (k = 1, 2)	With the GTADCMSC register, set a count source of the skipping counter, skipping count, and the skipping counter k initial value as the following order. Set the GTADCMSC.ADCMSCk[1:0] bits as value other than 00b, and set the ADCMSTk[3:0] bits as value other than 0000b. Change the ADCMSCk[1:0] bits from 00b to value other than 00b as well as set the ADCMSCNTkIV[3:0] bits to the skipping counter k initial value.
4	Start count operation	Set GTCR.CST to 1 to start count operation.
5	Set buffer value for each cycle	See section 21.3.2. Buffer Operation, section 21.3.3. PWM Output Operating Mode, section 21.5. A/D Conversion Start Request, and so on.

21.5 A/D Conversion Start Request

The A/D conversion start request can be issued at a compare match between the GTCNT counter and the GTADTRA or GTADTRB register. Up-counting only, down-counting only, or both up-counting and down-counting can be specified by setting the GTINTAD register.

In complementary PWM mode, the A/D conversion start request can be issued at a compare match with the GTCNT counter of master channel.

During event count operation, the A/D conversion start request cannot be generated.

The A/D conversion start request is output as event signals to ELC.

The GTADTRA and GTADTRB registers each has two buffer registers. Buffer operation with the GTADTRA register used together with the GTADTBRA and GTADTDBRA registers, and buffer operation with the GTADTRB register used together with the GTADTBRB and GTADTDBRB registers can be performed.

The timing of the generation of requests to start A/D conversion can be monitored by an external pin. When the A/D conversion start request signal to be monitored is selected in the GTADSMR.ADSMSk bit (k = 0, 1) and when the output is enabled in the ADSMENk bit, a signal is output synchronized with a cycle frame of the timer used to generate the A/D conversion start request signal, of which the output is driven high at the generation of the A/D conversion start request signal by the GTADSMk pin, or at the end of the cycle of which the output is driven low. When a signal to request the start of A/D conversion is generated at the end of the cycle, the generation of this signal has priority in terms of monitoring output and the output remains at the high level till the end of the next cycle. The registers (GTADTRA and GTADTRB) that are sources of generating the A/D conversion start request signals and their counting directions can be checked by the A/D conversion start request flags (ADTRAUF, ADTRADF, ADTRBUF, and ADTRBDF) in the GTST register. When the output of the same A/D conversion start request signal monitoring output is enabled for multiple channels, ORed signals will be output from the GPT32.

Figure 21.160 shows an example of A/D conversion start request operation, Table 21.71 shows example for setting A/D conversion start request operation.

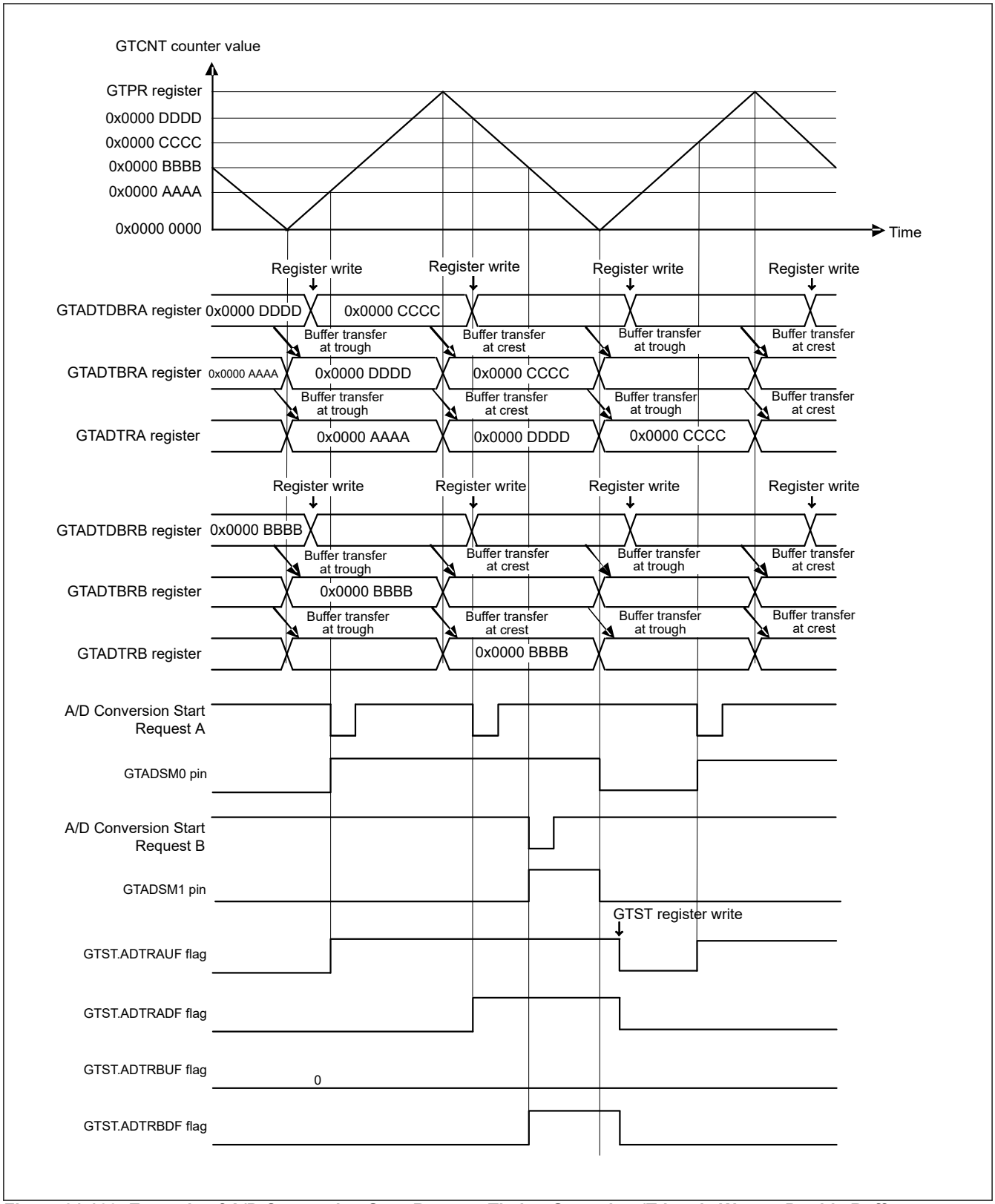


Figure 21.160 Example of A/D Conversion Start Request Timing Operation (Triangle Waves, Double Buffer Operation, Buffer Transfer at Both Troughs and Crests, A/D Conversion Start Request by GTADTRA Register at Both Up-Counting and Down-Counting, A/D Conversion Start Request by GTADTRB Register at Down-Counting, Monitoring of the GTADTRA Register Up-Counting by the GTADSM0 Pin, Monitoring of the GTADTRB Register Down-Counting by the GTADSM1 Pin)

Table 21.71 Example for Setting A/D Conversion Start Request Timing Operation

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits or GTCR.MD[3:0] bits. In Figure 21.160, 100b, 101b, or 110b (0100b, 0101b, or 0110b) (triangle-wave PWM mode) is set.
2	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
3	Set cycle	Set the cycle in the GTPR register.
4	Set initial value for counter	Set the initial value in the GTCNT counter.
5	Set buffer operation	Set buffer operation with the ADTTA[1:0], ADTTB[1:0], ADTDA, and ADTDB bits in the GTBER register. In Figure 21.160, ADTTA[1:0] = 11b, ADTTB[1:0] = 11b, ADTDA = 1, and ADTDB = 1.
6	Set compare match value	Set the A/D conversion start request point in the GTADTRA and GTADTRB registers.
7	Set buffer value	For buffer operation, set the A/D conversion start request point in one cycle after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or half cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTADTBRA and GTADTBRB registers. For double buffer operation, also set the A/D conversion start request point in two cycles after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or one cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTADTDBRA and GTADTDBRB registers.
8	Set A/D conversion start request for monitoring	Select the A/D conversion start request signal to be monitored with ADSMS0[1:0] and ADSMS1[1:0] bits in GTADSMR from GTADSM0 and GTADSM1 pins and enable output of the A/D conversion start request signal being monitored to ADSMEN0 and ADSMEN1 bits in GTADSMR. In Figure 21.160, ADSMS0[1:0] = 00b, ADSMS1[1:0] = 11b, ADSMEN0 = 1, and ADSMEN1 = 1.
9	Enable A/D conversion start request	Set to enable A/D conversion start request with the ADTRAUEN, ADTRADEN, ADTRBUEN, and ADTRBDEN bits in the GTINTAD register. In Figure 21.160, ADTRAUEN = 1, ADTRADEN = 1, ADTRBUEN = 0, and ADTRBDEN = 1.
10	Start count operation	Set the GTCR.CST bit to 1 to start count operation.
11	Set buffer value for each cycle	For buffer operation, set the A/D conversion start request point in one cycle after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or half cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTADTBRA and GTADTBRB registers. For double buffer operation, also set the A/D conversion start request point in two cycles after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or one cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTADTDBRA and GTADTDBRB registers.

Figure 21.161 shows an example for A/D conversion start request timing operation.

This shows an example of the output of A/D conversion start request A by the ELC as start source 0 (ELC_AD00) for the A/D converter. The A/D conversion start request A signal is output by the ELC in response to a match in comparison with the GTADTRA register.

The same timing applies when the A/D conversion start request A signal from GPT is directly input to ADC without going through ELC.

If GPT is operating with PCLKD and ADC is operating with PCLKA, A/D conversion start request A is passed to ELC on the next rising edge of PCLKA.

If GPT is operating with GPTCLK and ADC is also operating with GPTCLK, A/D conversion start request A is passed to ELC without delay.

For all other clock combinations, A/D conversion start request A is synchronized and passed to ELC.

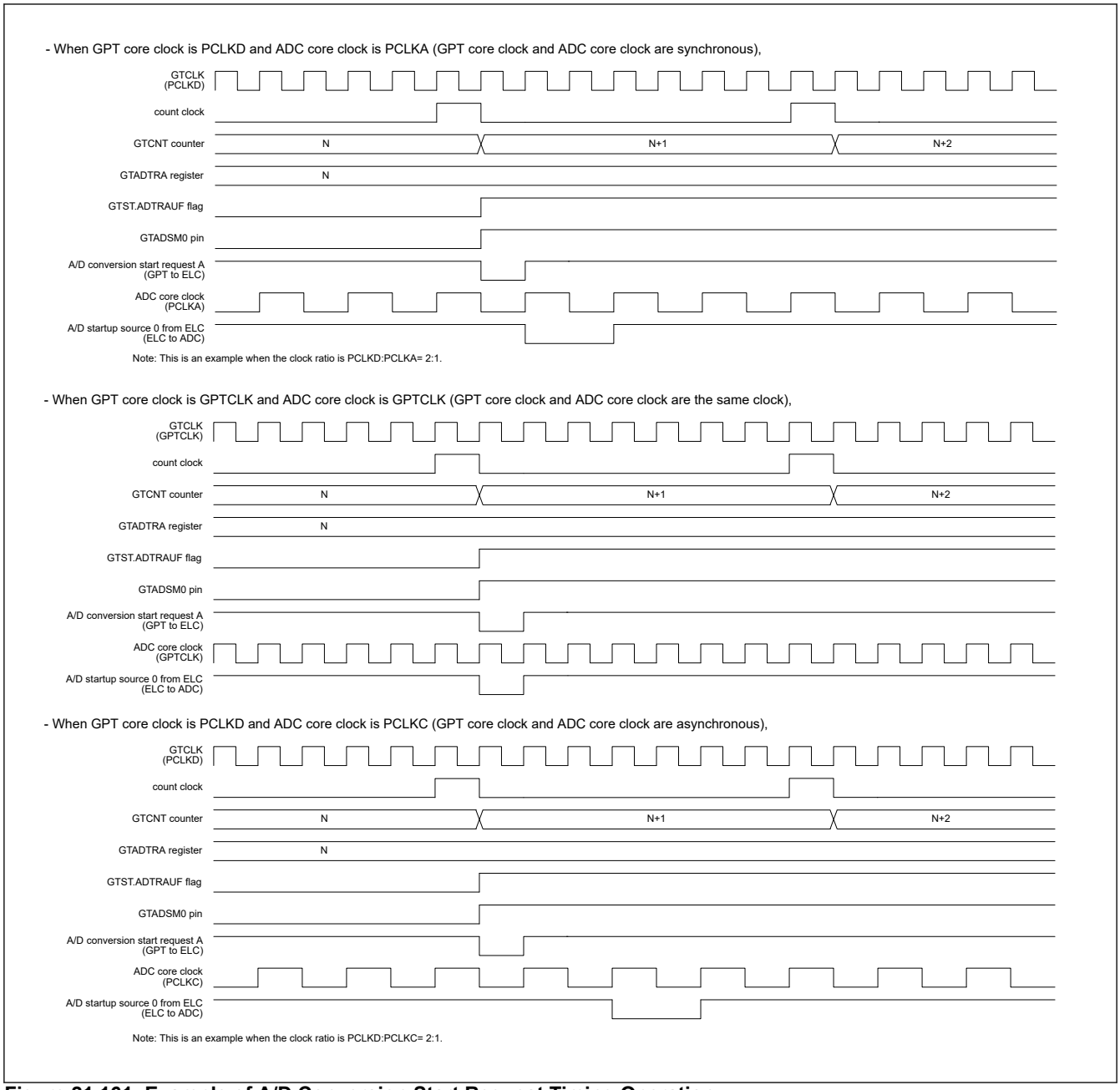


Figure 21.161 Example of A/D Conversion Start Request Timing Operation

For the restriction of A/D Conversion Start Request, see section 17, Event Link Controller (ELC), and section 21.10.6. Interval of interrupt request.

21.6 Operations Linked by ELC

21.6.1 Event Signal Output to ELC

The GPT can perform operation linked with another module set in advance when its interrupt request signal is used as an event signal by the Event Link Controller (ELC).

The A/D conversion start request during up-counting/down-counting can be enabled/disabled individually with the A/D conversion start request enable bit to output events output to ELC.

The GPT has the following ELC event signals:

- Generation of compare match and input capture A interrupt (GPTn_CCMPA)
- Generation of compare match and input capture B interrupt (GPTn_CCMPB)

- Generation of compare match C interrupt (GPTn_CMPC)
- Generation of compare match D interrupt (GPTn_CMPD)
- Generation of compare match E interrupt (GPTn_CMPE)
- Generation of compare match F interrupt (GPTn_CMPF)
- Generation of overflow interrupt (GPTn_OVF)
- Generation of underflow interrupt (GPTn_UDF)
- Generation of A/D conversion start request A (GPTn_ADTRGA)
- Generation of A/D conversion start request B (GPTn_ADTRGB)
- Finish of period count function (GPTm_PC)

Note: n = 0 to 9
m = 0 to 3

21.6.2 Event Signal Inputs from ELC

The GPT can perform the following operations in response to a maximum of 8 events from the ELC:

- Start counting, stop counting, clear counting
- Up-counting, down-counting
- Input capture.

See section 17, Event Link Controller (ELC) for the connection between the ELC and the event signal input.

21.7 Noise Filter Function

Each pin for use in input capture and Hall sensor input to the GPT is equipped with a noise filter. The noise filter samples input signals at the sampling clock and removes the pulses whose length is less than 3 sampling cycles.

The noise filter functionality includes enabling and disabling the noise filter for each pin and setting of the sampling clock for each channel.

Figure 21.162 shows the timing of noise filtering.

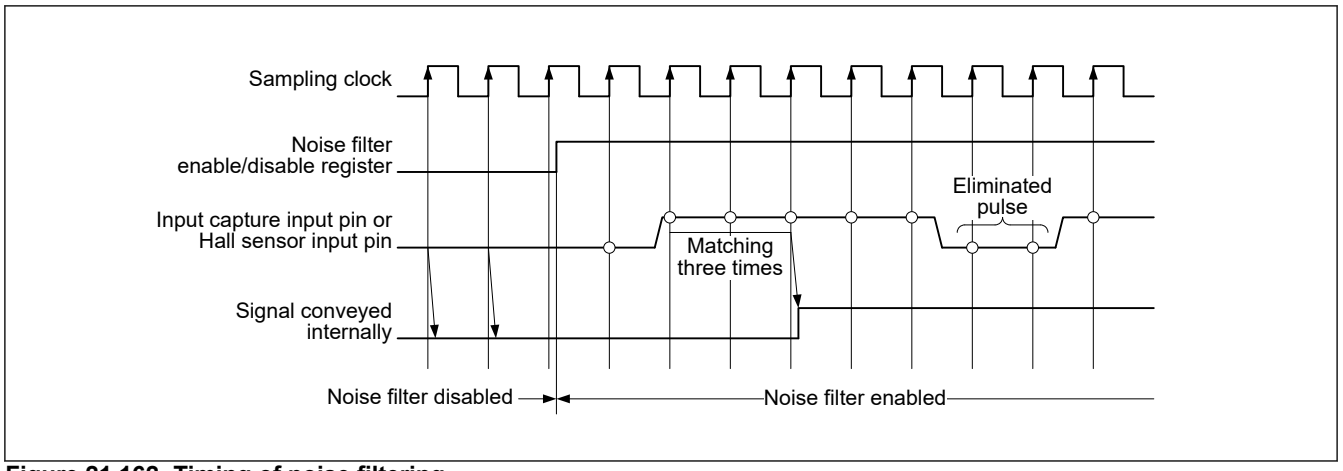


Figure 21.162 Timing of noise filtering

If noise filtering is enabled, the input capture operation or hall sensor input operation is performed on the edges of the noise filtered signal after a delay of (sampling interval × 2 + GTCLK) at the shortest. This is due to the noise filtering for the input capture input or hall sensor input.

21.8 Protection Function

21.8.1 Write-Protection for Registers

To prevent registers from being accidentally modified, registers can be write-protected in channel units by setting GTWP.WP. Write-protection can be set for the following registers:

GTSSR, GTPSR, GTCSR, GTUPSR, GTDNSR, GTICASR, GTICBSR, GTCR, GTUDDTYC, GTIOR, GTINTAD, GTST, GTBER, GTITC, GTCNT, GTCCRA, GTCCRB, GTCCRC, GTCCRD, GTCCRE, GTCCRF, GTPR, GTPBR, GTPDBR, GTADTRA, GTADTBRA, GTADTDBRA, GTADTRB, GTADTBRB, GTADTDBRB, GTDTCR, GTDVU, GTDVD, GTDBU, GTDBD, GTSOS, GTSOTR, GTADSMR, GTEITC, GTEITLI1, GTEITLI2, GTEITLB, GTICLF, GTPC, GTADCMSC, GTADCMSS, GTBER2, GTOLBR, GTICCR.

Every bit in registers GTSTR, GTSTP and GTCLR which can update the corresponding registers in other channels and can be updated by any of the corresponding registers in other channels conversely, can be protected by setting the GTWP.STRWP, STPWP, and CLRWP bits, respectively, per channel.

Likewise, writing to the GTSECSR and GTSECR registers, which can control all channels by writing to the GTSECSR and GTSECR registers of a given channel, can by enabled or disabled by the setting of the GTWP.CMNWP bit.

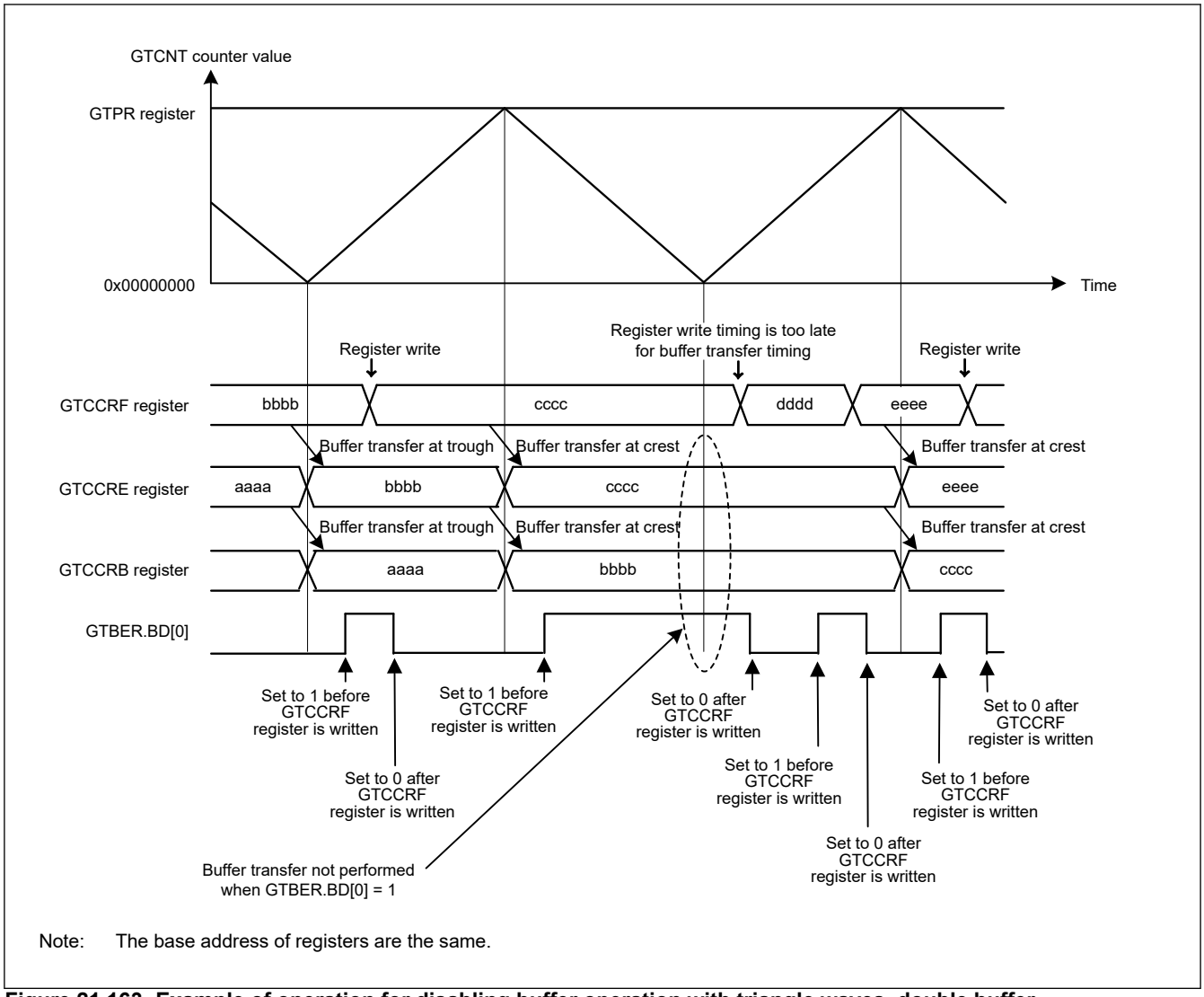
Protection using the GTWP register is only for write operations by the CPU. This protection does not cover updates to registers that occur in association with CPU writes.

21.8.2 Disabling of Buffer Operation

If the timing of the buffer register write is delayed relative to the timing for the buffer transfer, buffer operation can be suspended with the GTBER.BD[3], BD[2], BD[1] and BD[0] bits settings. Specifically, buffer transfer can be temporarily disabled even though a buffer transfer condition is generated during buffer register write, by setting the BD[3], BD[2], BD[1] and BD[0] bits to 1 (buffer operation disabled) before buffer register write, and setting the bits to 0 (buffer operation enabled) after completion of writing to all the buffer registers.

The BD[3], BD[2], BD[1] and BD[0] bits can be set on channel basis by writing directly to the GTBER register or it can be set to 0 simultaneously by setting the GTSECR register for multiple channels which were set by the GTSECSR register.

Figure 21.163 shows an example of operation for disabling buffer operation by writing to the GTBER register.



Note: The base address of registers are the same.

Figure 21.163 Example of operation for disabling buffer operation with triangle waves, double buffer operation, and buffer transfer at both troughs and crests

21.8.2.1 Simultaneous Control of Buffer Operations of Multiple Channels

The GTBER.BD bit can be set by writing directly to the GTBER register per channel or by making settings in the GTSECR register for multiple channels that have already set in the GTSECSR register.

Follow the procedure below to simultaneously set the GTBER.BD bits of multiple channels.

1. Select the channels for simultaneously setting by the GTSECSR register
   Set the GTSECSR register so that the values at the bit positions for the corresponding channels for simultaneously setting of the GTBER.BD bits become 1. All GTSECSR registers can be updated by writing to the GTSECSR register of any channel.
2. Simultaneously set the GTBER.BD bits by updating the GTSECR register
   In the GTSECR register, set the operation of the GTBER.BD bits (enabling or disabling of buffer operation) which are to be simultaneously set. Writing to a GTSECR register from any channel updates the GTBER.BD bits in all channels corresponding to the bits set as 1 in the GTSECSR register, in accordance with the value of the GTSECR register.

Figure 21.164 and Figure 21.165 show examples of simultaneously controlling the enabling or disabling of buffer operation for multiple channels.

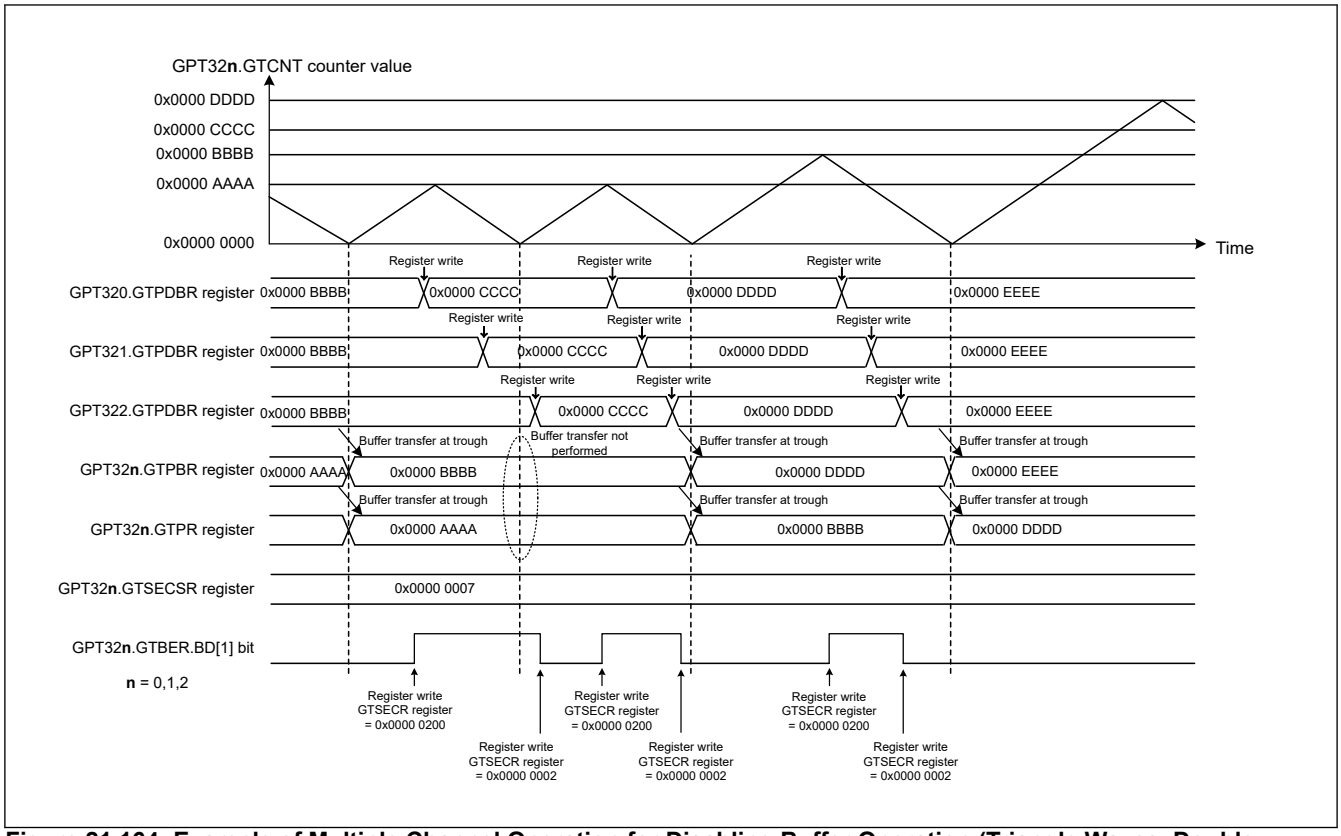


Figure 21.164 Example of Multiple Channel Operation for Disabling Buffer Operation (Triangle Waves, Double Buffer Operation)

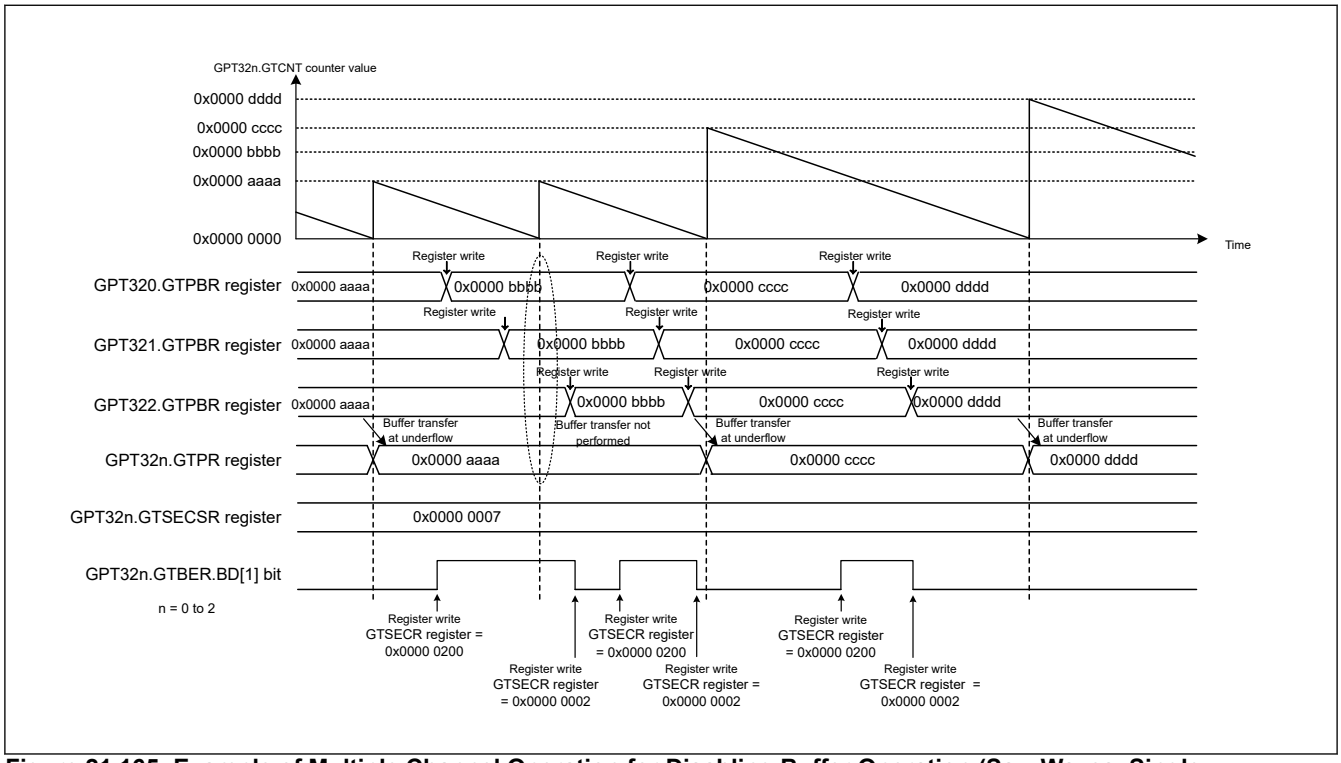


Figure 21.165 Example of Multiple Channel Operation for Disabling Buffer Operation (Saw Waves, Single Buffer Operation)

21.8.2.2 Repeated Double-Buffered Operation When Disabling GTCCR Buffer Transfer

When a GTBER.DBRTECm (m = A, B) bit is set to 1 in saw-wave one-shot pulse mode or triangle-wave PWM mode 3, transfer from the intermediate buffer to the GTCCRm (m = A, B) register is repeated on a cyclic basis even while buffer transfer is disabled by the setting of the GTBER.BD[0] bit or by the buffer transfer extended skipping function.

(1) In saw-wave one-shot pulse mode

In saw-wave one-shot pulse mode, the compare match value for the first half of a waveform is stored in temporary register x (x = C, E) as the intermediate buffer for the GTCCRx (x = C, E) register and the compare match value for the second half of a waveform is stored in temporary register m (m = A, B) as the intermediate buffer for the GTCCRy (y = D, F) register, respectively, for compare match values during repeated buffer operation, and the given values are alternately transferred to the GTCCRm (m = A, B) registers.

Table 21.72 lists the types of buffer transfer of the GTCCR register during counting in saw-wave one-shot pulse mode.

While counting is stopped, the setting of the value in the temporary register is transferred through forcible buffer transfer. In forcible buffer transfer, the values of the GTCCRy (y = D, F) registers are transferred to temporary registers m (m = A, B), and the values of the GTCCRx (x = C, E) are transferred to temporary registers x (x = C, E) when the setting of the corresponding GTBER.DBRTECm (m = A, B) bit is 1, respectively.

When the setting of the GTBER.DBRTECm (m = A, B) bit is 1, values written by the CPU to the GTCCRm (m = A, B) registers are reflected as the values of temporary registers x (x = C, E).

Table 21.72 GTCCR Buffer Transfer Operation in Saw-Wave One Shot Pulse Mode during GTCNT Counting

GTBER.DBRTECm	Buffer Transfer	Timing of transfer				
		GTCCRx ↓ GTCCRm	GTCCRx ↓ Temporary register x	Temporary register x ↓ GTCCRm	GTCCRy ↓ Temporary register m	Temporary register m ↓ GTCCRm
0	Transfer enabled period	Overflow or Underflow	No transfer	No transfer	Overflow or Underflow	GTCCRm compare match
	Transfer disabled period	No transfer	No transfer	No transfer	No transfer	No transfer
1	Transfer enabled period	Overflow or Underflow	Overflow or Underflow	No transfer	Overflow or Underflow	GTCCRm compare match
	Transfer disabled period	No transfer	No transfer	Overflow or Underflow	No transfer	GTCCRm compare match

Note: m = A, B
x = C, E
y = D, F

Figure 21.166 shows the operation of generating transfer-disabled period by extended buffer transfer skipping as an example of repeated double buffer operations when the GTCCR buffer transfer disabled in saw-wave one-shot pulse mode.

Figure 21.167 shows the operation of generating transfer-disabled period by updating the GTBER.BD[0] bit as an example of repeated double-buffer operations when the GTCCR buffer transfer is disabled in saw-wave one-shot pulse mode.

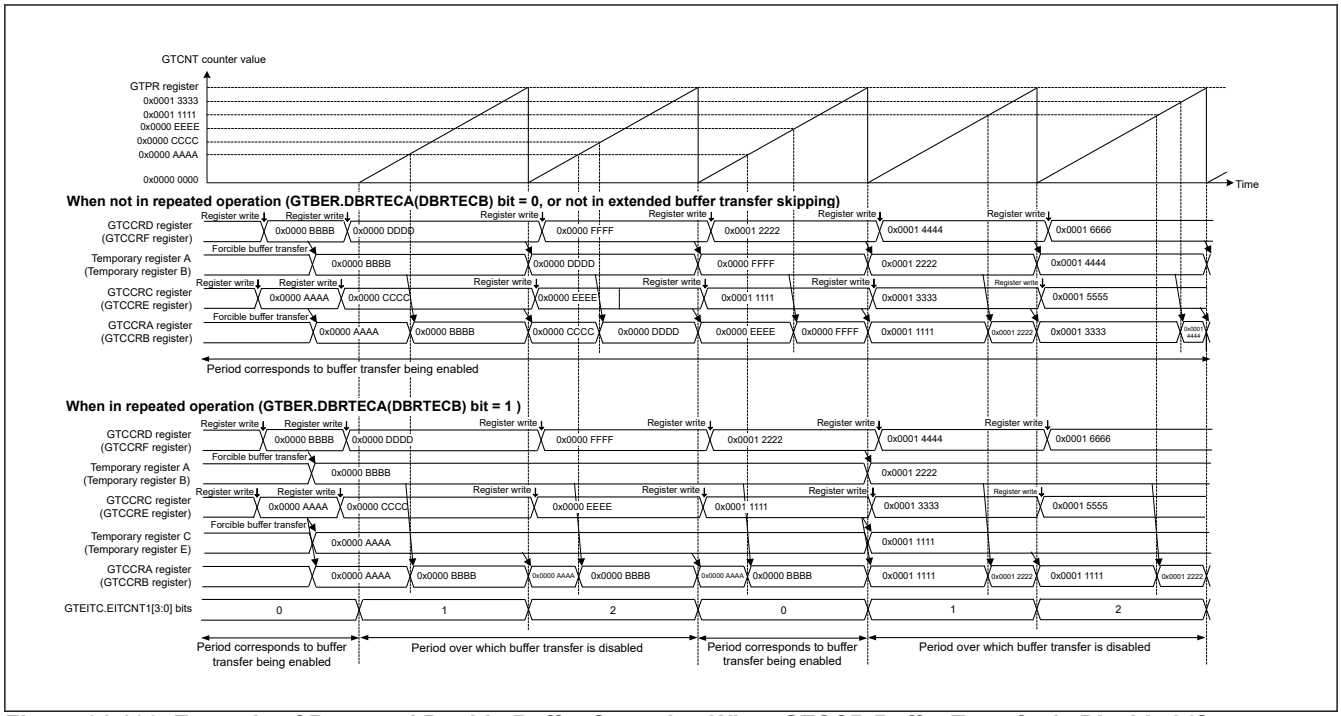


Figure 21.166 Example of Repeated Double-Buffer Operation When GTCCR Buffer Transfer is Disabled (Saw-Wave One-Shot Pulse Mode, Using Extended Buffer Transfer Skipping, GTBER.BD[0] is Constantly 0)

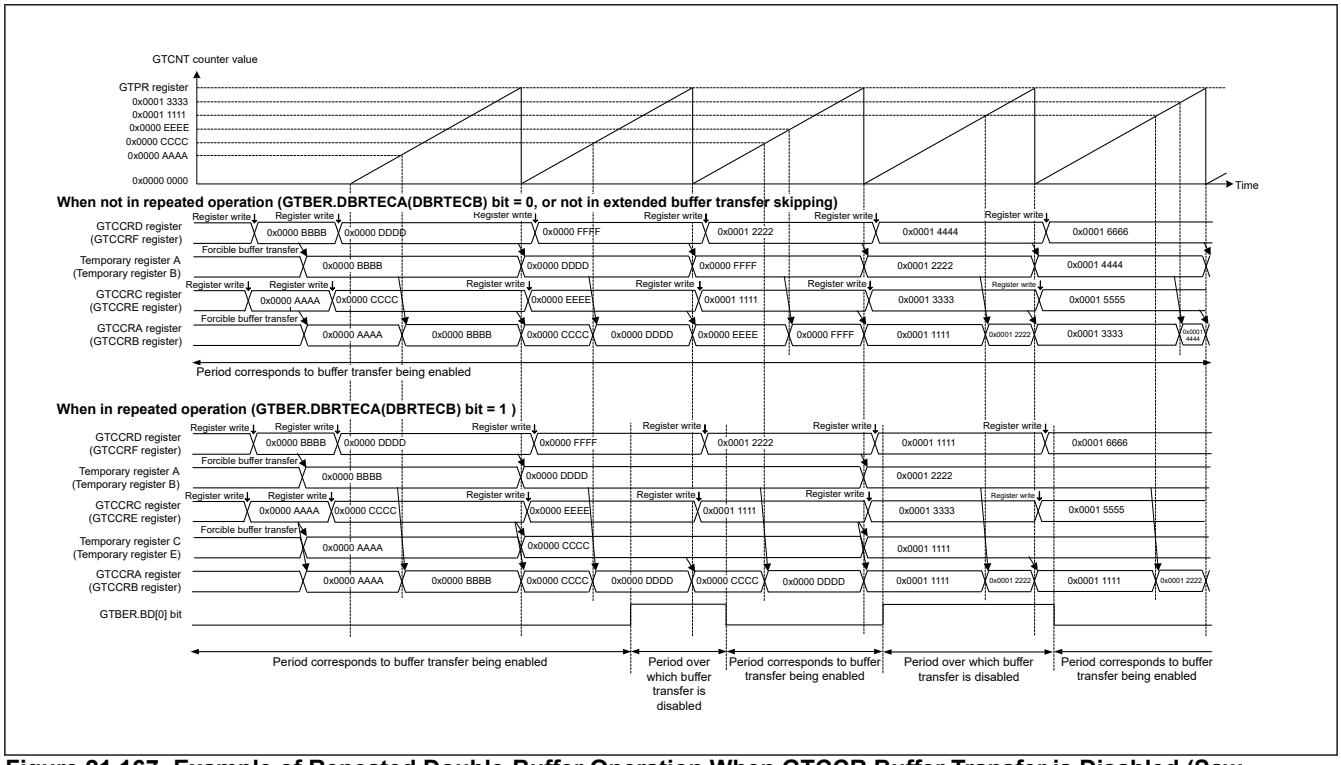


Figure 21.167 Example of Repeated Double-Buffer Operation When GTCCR Buffer Transfer is Disabled (Saw-Wave One-Shot Pulse Mode, Updating the GTBER.BD[0] Bit)

(2) In triangle-wave PWM mode 3

In triangle-wave PWM mode 3, the compare match value for the first half of a waveform is stored in the temporary register x (x = C, E) as the intermediate buffer for the GTCCRx (x = C, E) register and the compare match value for the second half of a waveform is stored in temporary register m (m = A, B) as the intermediate buffer for the GTCCRy (y = D, F) register,

respectively, for compare match values during repeated buffer operation, and the given values are alternately transferred to the GTCCRm (m = A, B) register.

Table 21.73 lists the types of buffer transfer of the GTCCR register during counting operation in triangle-wave PWM mode 3.

While counting is stopped, the setting of the value in the temporary register is transferred through forcible buffer transfer. In forcible buffer transfer, the values of the GTCCRy (y = D, F) registers are transferred to temporary registers m (m = A, B), and the values of the GTCCRx (x = C, E) are transferred to temporary registers x (x = C, E), when the setting of the corresponding GTBER.DBRTECm (m = A, B) bit is 1, respectively.

When the setting of the GTBER.DBRTECm (m = A, B) bit is set to 1, values written by the CPU to the GTCCRm (m = A, B) registers are reflected as the values of temporary registers x (x = C, E).

Table 21.73 GTCCR Buffer Transfer Operation in Triangle-Wave PWM Mode 3 during GTCNT Counting

GTBER.DBRTECm	Buffer Transfer	Timing of transfer				
		GTCCRx ↓ GTCCRm	GTCCRx ↓ Temporary register x	Temporary register x ↓ GTCCRm	GTCCRy ↓ Temporary register m	Temporary register m ↓ GTCCRm
0	Transfer enabled period	Trough	No transfer	No transfer	Trough	Crest
	Transfer disabled period	No transfer	No transfer	No transfer	No transfer	No transfer
1	Transfer enabled period	Trough	Trough	No transfer	Trough	Crest
	Transfer disabled period	No transfer	No transfer	Trough	No transfer	Crest

Note: m = A, B
x = C, E
y = D, F

Figure 21.168 shows the operation of generating transfer-disabled period by extended buffer transfer skipping as an example of repeated double buffer operations when the GTCCR buffer transfer disabled in triangle-wave PWM mode 3.

Figure 21.169 shows the operation of generating transfer-disabled period by updating the GTBER.BD[0] bit as an example of repeated double-buffer operations when the GTCCR buffer transfer is disabled in triangle-wave PWM mode 3.

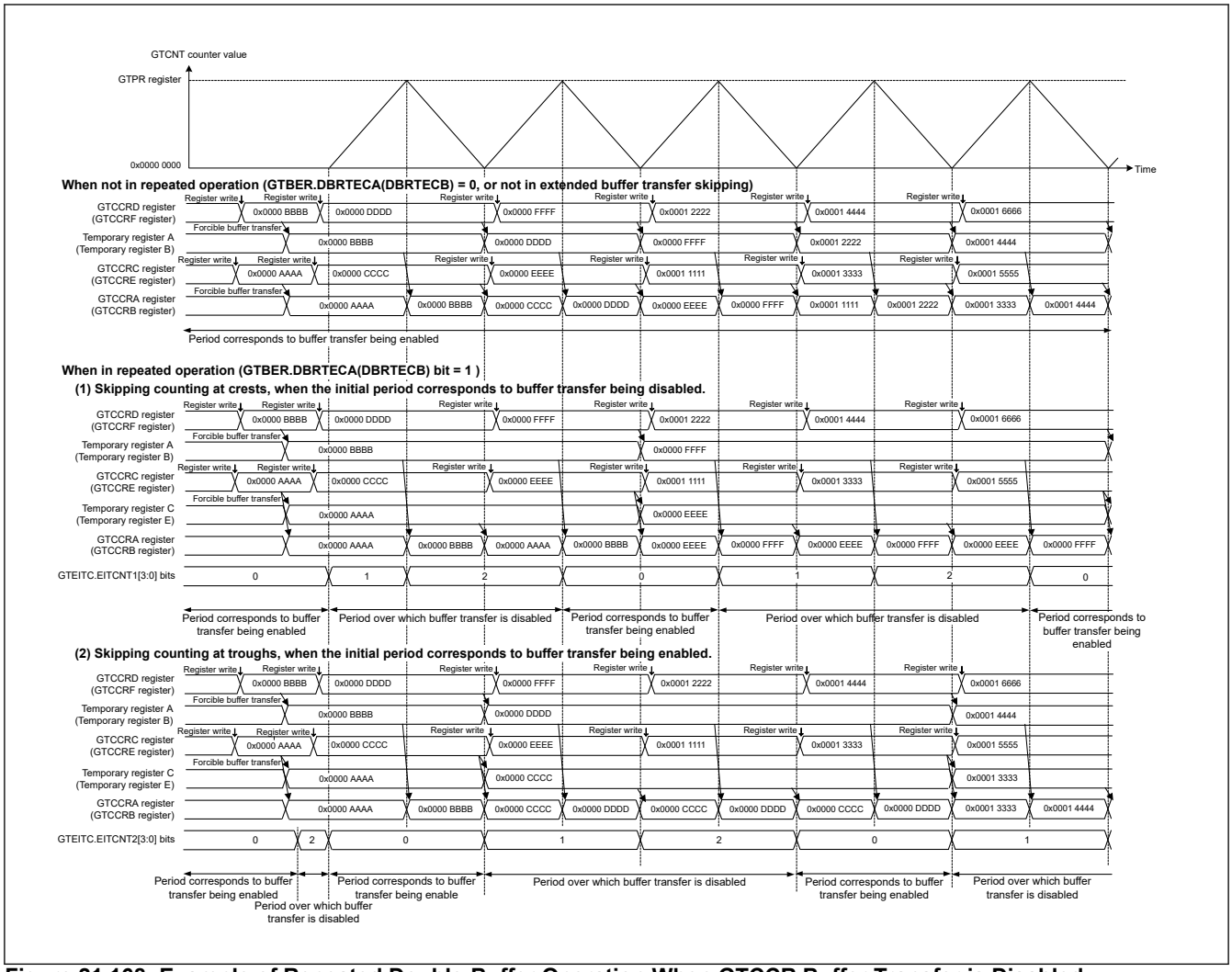


Figure 21.168 Example of Repeated Double-Buffer Operation When GTCCR Buffer Transfer is Disabled (Triangle-Wave PWM Mode 3, Using Extended Buffer Transfer Skipping, GTBER.BD[0] is Constantly 0)

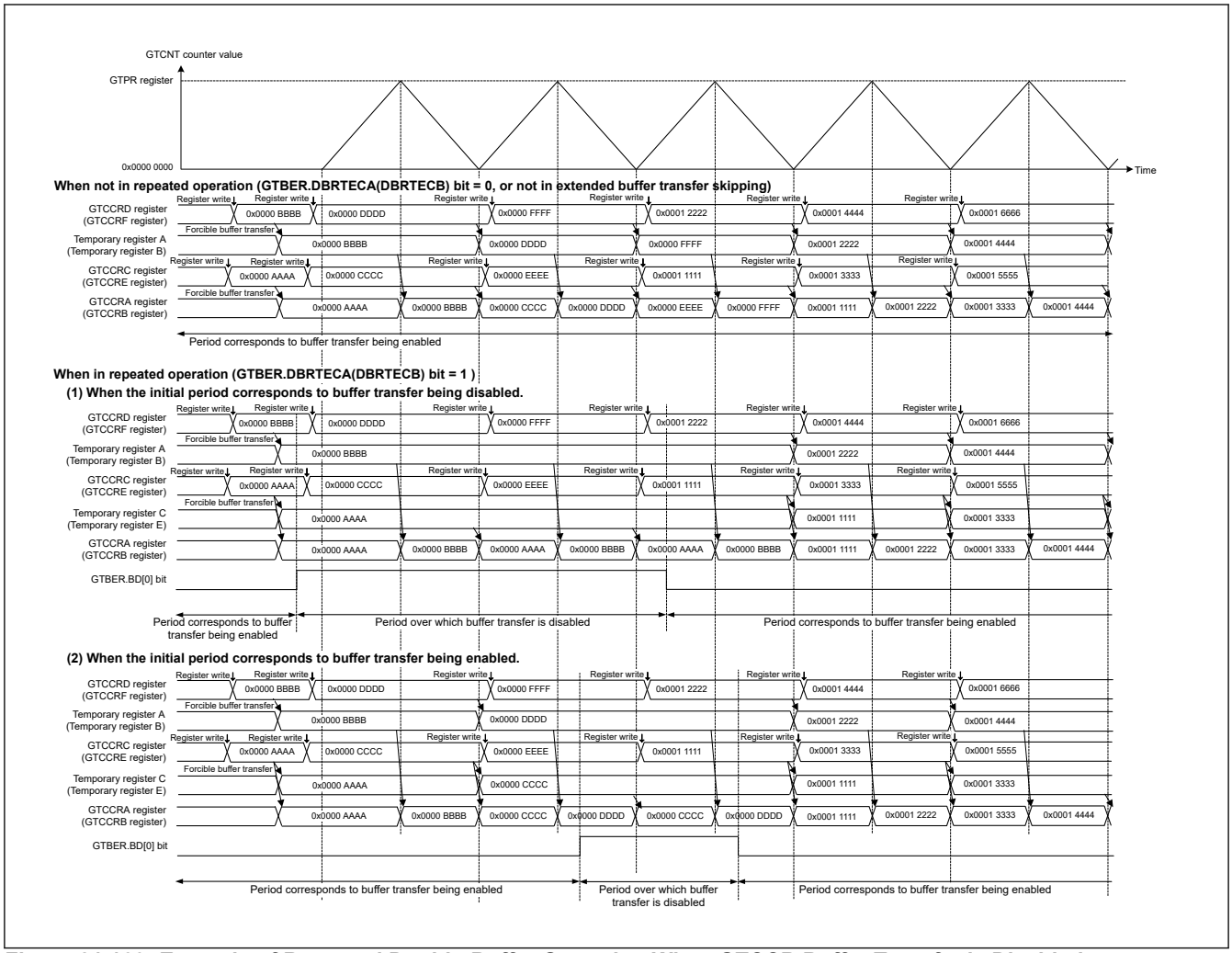


Figure 21.169 Example of Repeated Double-Buffer Operation When GTCCR Buffer Transfer is Disabled (Triangle-Wave PWM Mode 3, Updating the GTBER.BD[0] Bit)

21.8.3 GTIOCnm Pin Output Negate Control (n = 0 to 9, m = A, B)

For protection from system failure, the output disable control that changes the GTIOCnm pin output value forcibly is provided for GTIOCnm pin output by the request of output disable from POEG. Output protection is required when a dead time error or the same output level being on the GTIOCnA and GTIOCnB pins is detected. GPT detects this condition and generates output disable requests to POEG according to the setting of the output disable request permission bits, such as GTINTAD.GRPDTE, GTINTAD.GRPABH, GTINTAD.GRPABL. After the POEG performs the logical OR of the output disable request from each channel and the output disable request from the external input, the POEG generates output disable requests to GPT.

One output disable signal (representing the shared output disable request signal of the GTIOCnA pin and the GTIOCnB pin) out of 4 output disable requests generated by the POEG is selected by setting GTINTAD.GRP[1:0]. The status of the selected disable output request is monitored by reading the GTST.ODF bit. The output level during output disable is set based on the GTIOR.OADF[1:0] bits for the GTIOCnA pin and the GTIOR.OBDF[1:0] setting for the GTIOCnB pin.

The change to the output disable state is performed asynchronously by generating the output disable request from the POEG. The release of the output disable state is performed at end of cycle by terminating the output disable request. It is after 3 GTCLK at shortest when the output disable condition is released after the output disable request becomes no longer satisfied. To reliably control output disabling, clear the flag of POEG for which the condition for the request to disable the output is no longer satisfied after 4 cycles of GTCLK.

When event count is performed or when the operating mode is saw-wave PWM mode 2 or when the output disable state should be released immediately without waiting for end of cycle, GTIOR.OADF[1:0] should be set to 00b (for GTIOCnA pin) or GTIOR.OBDF[1:0] should be set to 00b (for the GTIOCnB pin).

Figure 21.170 shows an example of the GTIOCnm pin output disable control operation. (n = 0 to 9, m = A, B)

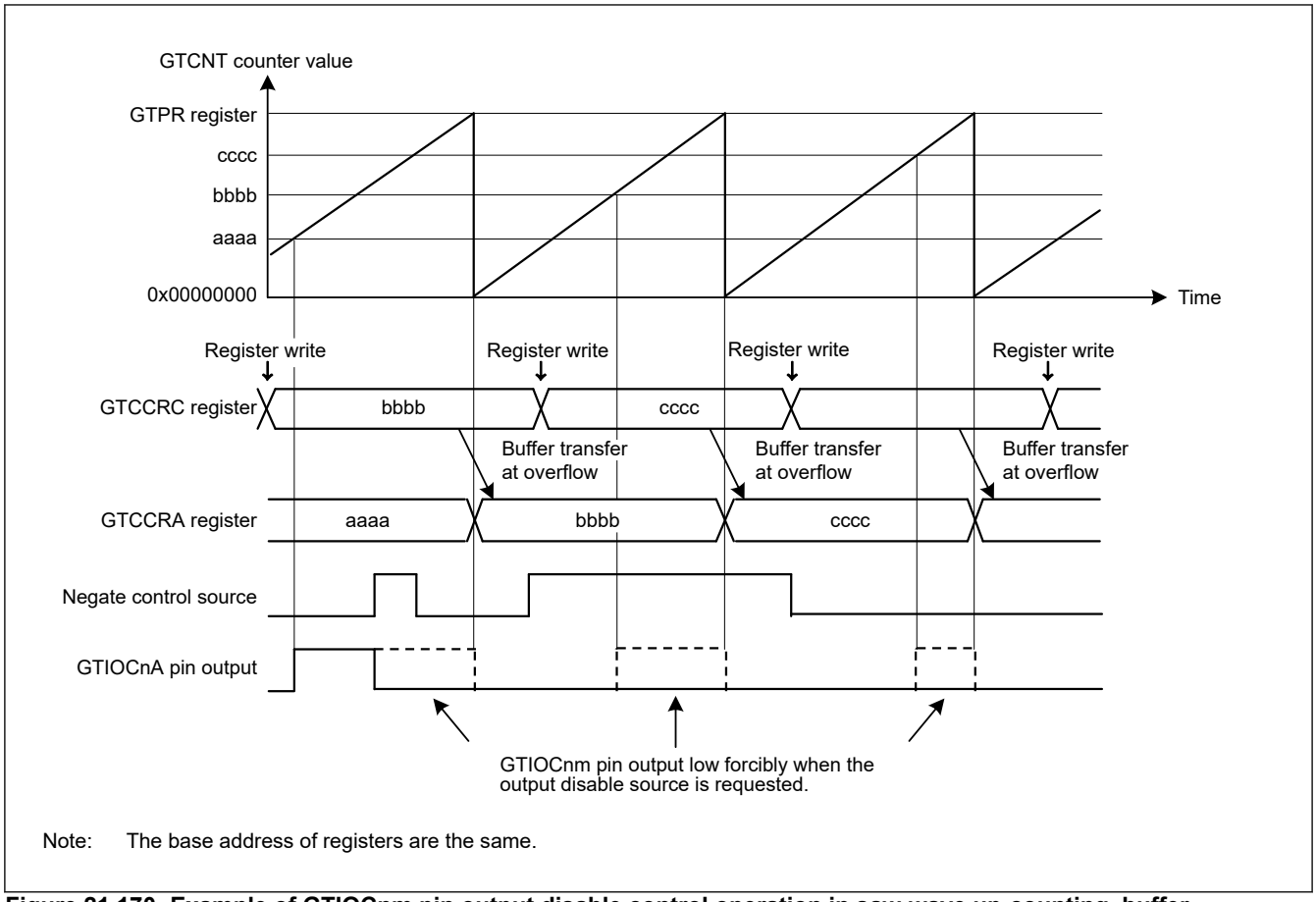


Figure 21.170 Example of GTIOCnm pin output disable control operation in saw-wave up-counting, buffer operation, active level 1, high output at GTCCRA compare match, low output at cycle end, and low output at output disable (n = 0 to 9, m = A, B)

21.8.4 Output Protection Function for GTIOCnm Pin Output (n = 0 to 9; m = A, B)

To prepare for a case when an incorrect value (0x0000 0000 or a value greater than or equal to the GTPR register value) is set in the GTCCRA register, the output protection function for the GTIOCnm pin output (disabling function) is activated when the automatic dead time is set (GTDTCR.TDE bit = 1) in triangle-wave PWM mode.

The status of the output protection function can be read from the GTSOS.SOS[1:0] bits.

Figure 21.171 shows the output protection function state transition.

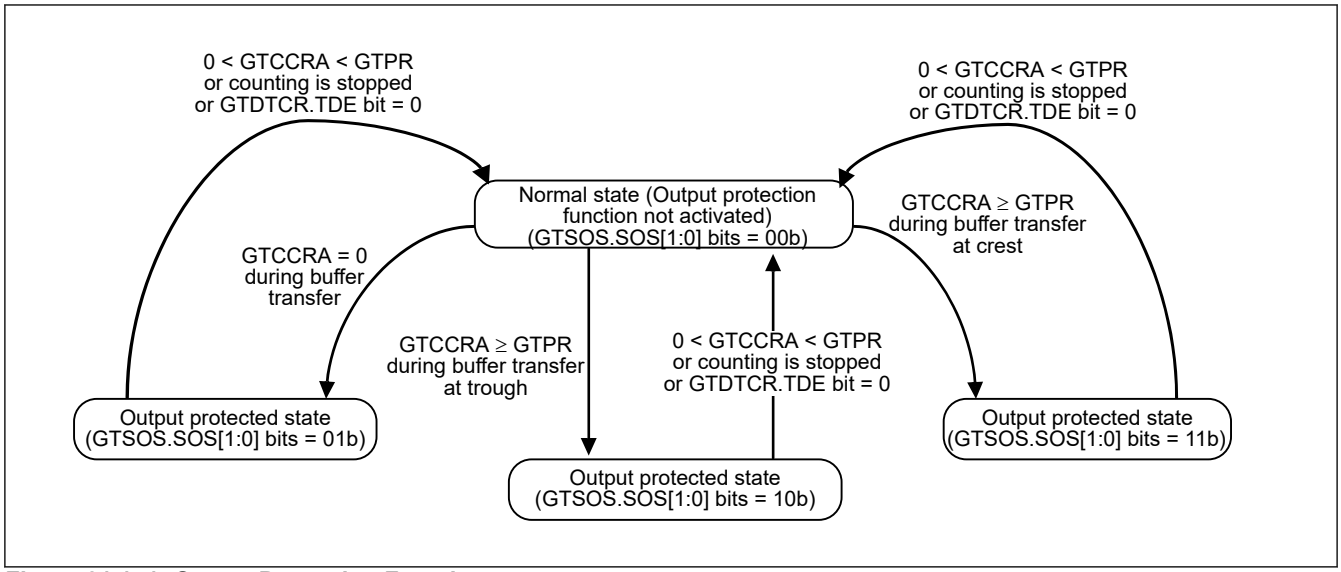
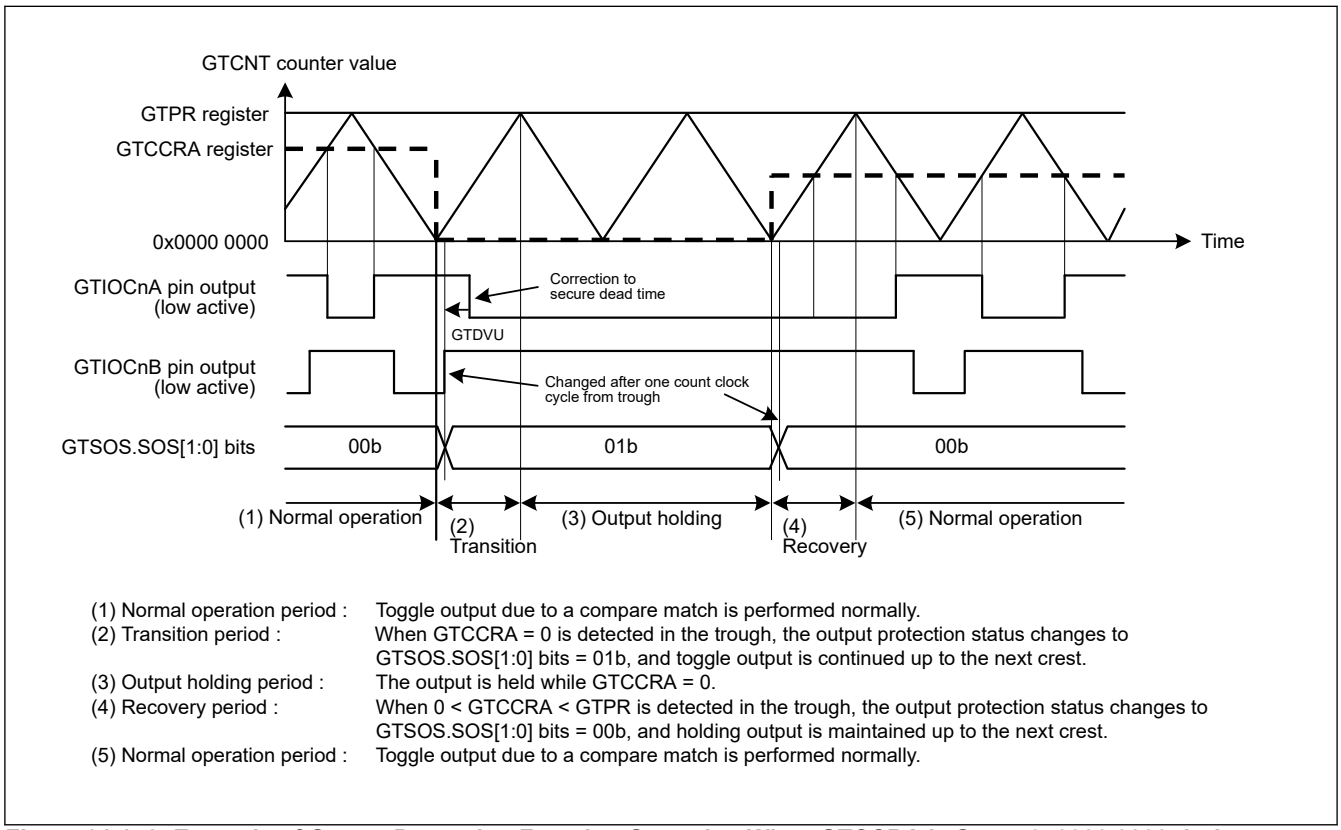


Figure 21.171 Output Protection Function

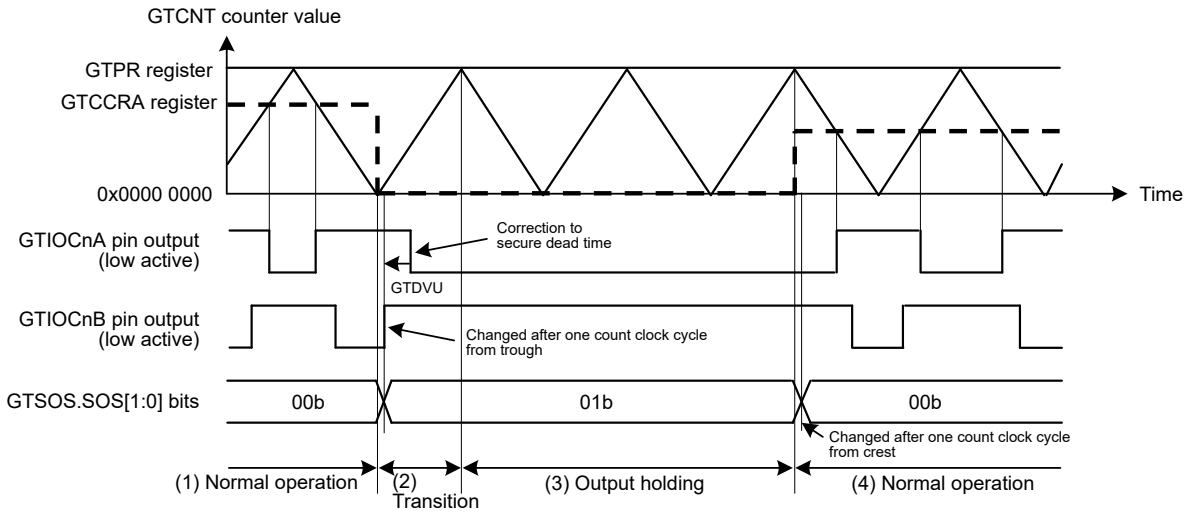
(1) Output Protection Function When the GTCCRA Register is Set to 0x0000 0000 during Buffer Transfer

Figure 21.172 and Figure 21.173 show examples of output protection function operation when the GTCCRA register is set to 0x0000 0000 during buffer transfer at troughs, and Figure 21.174 and Figure 21.175 show examples when the GTCCRA register is set to 0x0000 0000 during buffer transfer at crests.



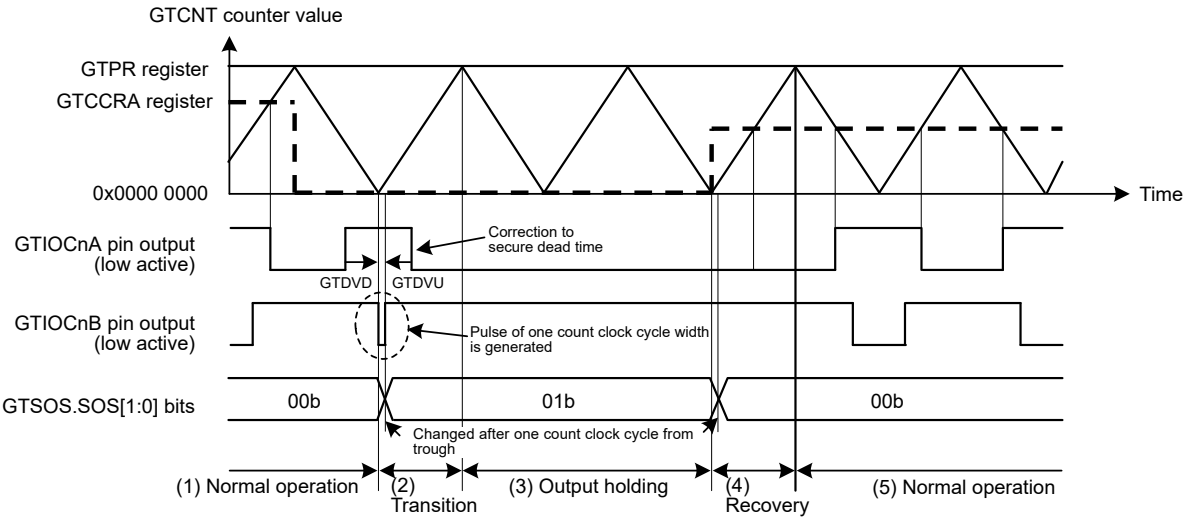
(1) Normal operation period : Toggle output due to a compare match is performed normally.
(2) Transition period : When GTCCRA = 0 is detected in the trough, the output protection status changes to GTSOS.SOS[1:0] bits = 01b, and toggle output is continued up to the next crest.
(3) Output holding period : The output is held while GTCCRA = 0.
(4) Recovery period : When 0 < GTCCRA < GTPR is detected in the trough, the output protection status changes to GTSOS.SOS[1:0] bits = 00b, and holding output is maintained up to the next crest.
(5) Normal operation period : Toggle output due to a compare match is performed normally.

Figure 21.172 Example of Output Protection Function Operation When GTCCRA is Set to 0x0000 0000 during Buffer Transfer at Troughs (Restored to 0 < GTCCRA < GTPR during Buffer Transfer at Troughs, Active Level: Low) (n = 0 to 9)



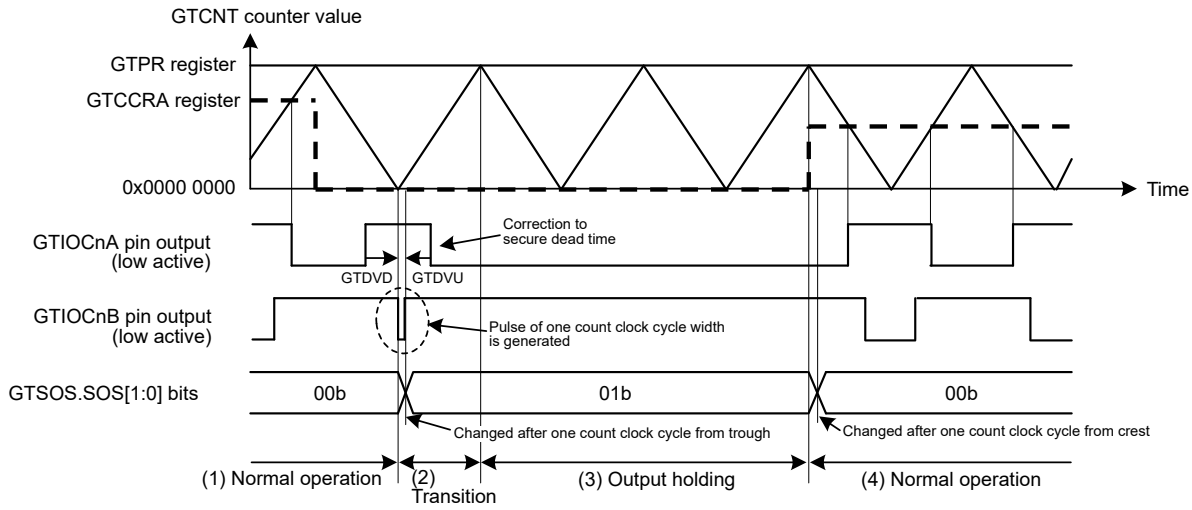
(1) Normal operation period : Toggle output due to a compare match is performed normally.
(2) Transition period : When GTCCRA = 0 is detected in the trough, the output protection status changes to GTSOS.SOS[1:0] bits = 01b, and toggle output is continued up to the next crest.
(3) Output holding period : The output is held while GTCCRA = 0.
(4) Normal operation period : When 0 < GTCCRA < GTPR is detected in the crest, the output protection status changes to GTSOS.SOS[1:0] bits = 00b, and toggle output due to a compare match is performed normally.

Figure 21.173 Example of Output Protection Function Operation When GTCCRA is Set to 0x0000 0000 during Buffer Transfer at Troughs (Restored to 0 < GTCCRA < GTPR during Buffer Transfer at Crests, Active Level: Low) (n = 0 to 9)



(1) Normal operation period : Toggle output due to a compare match is performed normally.
(2) Transition period : When GTCCRA = 0 is detected in the trough, the output protection status changes to GTSOS.SOS[1:0] bits = 01b, and toggle output is continued up to the next crest.
(3) Output holding : The output is held while GTCCRA = 0.
(4) Recovery period : When 0 < GTCCRA < GTPR is detected in the trough, the output protection status changes to GTSOS.SOS[1:0] bits = 00b, and holding output is maintained up to the next crest.
(5) Normal operation period : Toggle output due to a compare match is performed normally.

Figure 21.174 Example of Output Protection Function Operation When GTCCRA is Set to 0x0000 0000 during Buffer Transfer at Crests (Restored to 0 < GTCCRA < GTPR during Buffer Transfer at Troughs, Active Level: Low) (n = 0 to 9)

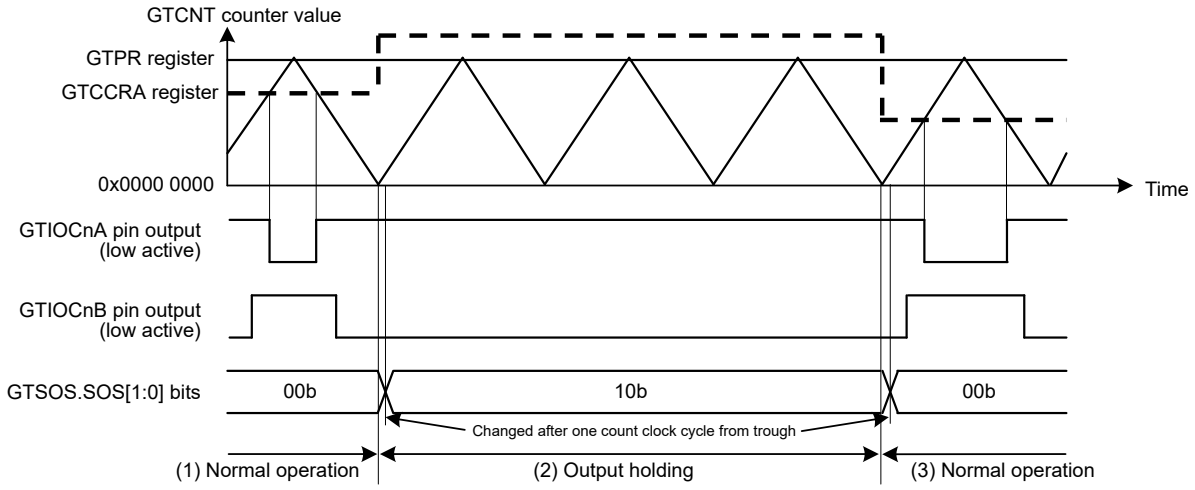


(1) Normal operation period : Toggle output due to a compare match is performed normally.
(2) Transition period : When GTCCRA = 0 is detected in the trough, the output protection status changes to GTSOS.SOS[1:0] bits = 01b, and toggle output is continued up to the next crest.
(3) Output holding period : The output is held while GTCCRA = 0.
(4) Normal operation period : When 0 < GTCCRA < GTPR is detected in the crest, the output protection status changes to GTSOS.SOS[1:0] bits = 00b, and toggle output due to a compare match is performed normally.

Figure 21.175 Example of Output Protection Function Operation When the GTCCRA is Set to 0x0000 0000 during Buffer Transfer at Crests (Restored to 0 < GTCCRA < GTPR during Buffer Transfer at Crests, Active Level: Low) (n = 0 to 9)

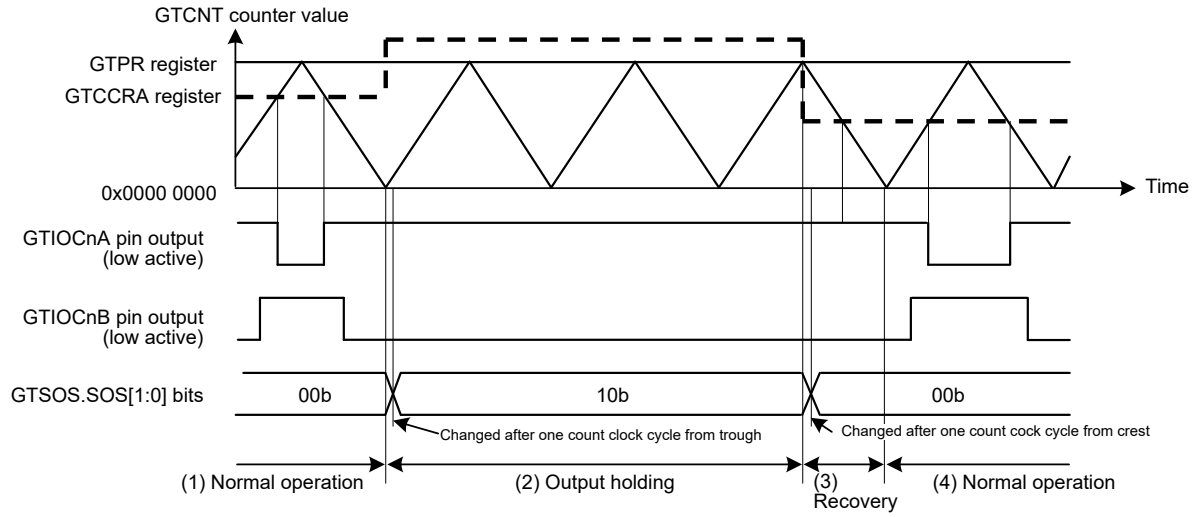
(2) Output Protection Function When GTCCRA Register ≥ GTPR Register is Set during Buffer Transfer at Troughs

Figure 21.176 and Figure 21.177 show examples of output protection function operation when GTCCRA register ≥ GTPR register is set during buffer transfer at troughs.



(1) Normal operation period : Toggle output due to a compare match is performed normally.
(2) Output holding period : When GTCCRA ≥ GTPR is detected in the trough, the output protection status changes to GTSOS.SOS[1:0] bits = 10b, and the output is held while GTCCRA ≥ GTPR.
(3) Normal operation period : When 0 < GTCCRA < GTPR is detected in the trough, the output protection status changes to GTSOS.SOS[1:0] bits = 00b, and toggle output due to a compare match is performed normally.

Figure 21.176 Example of Output Protection Function Operation When GTCCRA ≥ GTPR is set during Buffer Transfer at Troughs (Restored to 0 < GTCCRA < GTPR during Buffer Transfer at Troughs, Active Level: Low) (n = 0 to 9)

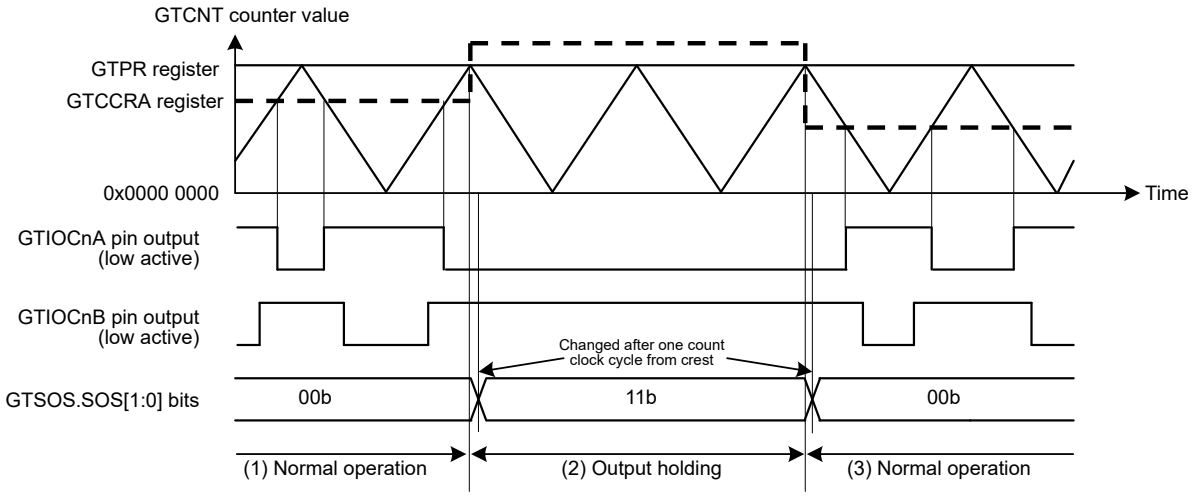


(1) Normal operation period : Toggle output due to a compare match is performed normally.
(2) Output holding period : When GTCCRA ≥ GTPR is detected in the trough, the output protection status changes to GTSOS.SOS[1:0] bits = 10b, and the output is held while GTCCRA ≥ GTPR.
(3) Recovery period : When 0 < GTCCRA < GTPR is detected in the crest, the output protection status changes to GTSOS.SOS[1:0] bits = 00b, and holding output is maintained up to the next trough.
(4) Normal operation period : Toggle output due to a compare match is performed normally.

Figure 21.177 Example of Output Protection Function Operation When GTCCRA ≥ GTPR is Set during Buffer Transfer at Troughs (Restored to 0 < GTCCRA < GTPR during Buffer Transfer at Crests, Active Level: Low) (n = 0 to 9)

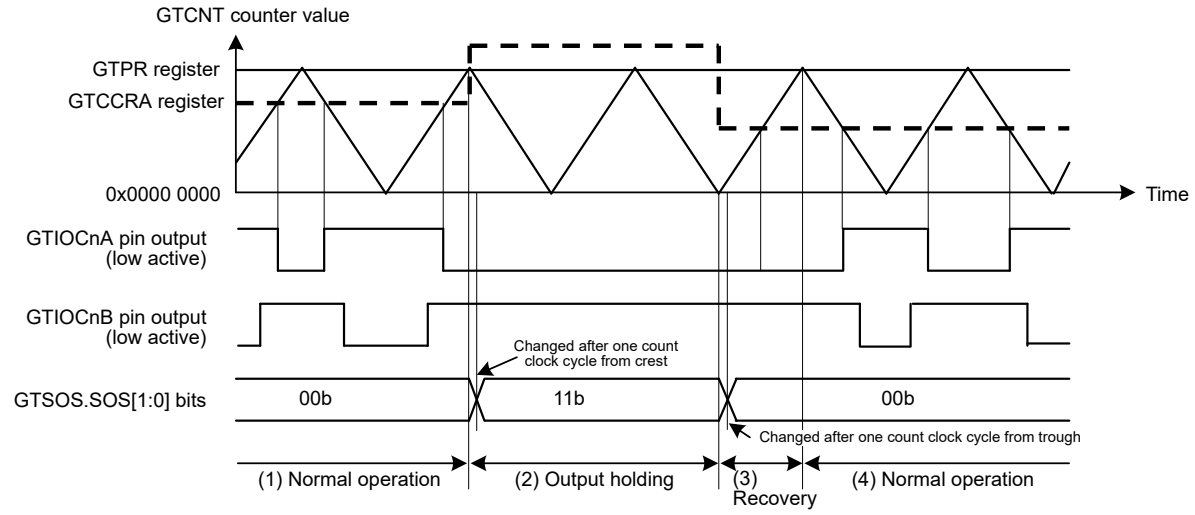
(3) Output Protection Function When GTCCRA ≥ GTPR is Set during Buffer Transfer at Crests

Figure 21.178 and Figure 21.179 show examples of output protection function operation when GTCCRA ≥ GTPR is set during buffer transfer at crests.



(1) Normal operation period : Toggle output due to a compare match is performed normally.
(2) Output holding period : When GTCCRA ≥ GTPR is detected in the crest, the output protection status changes to GTSOS.SOS[1:0] bits = 11b, and the output is held while GTCCRA ≥ GTPR.
(3) Normal operation period : When 0 < GTCCRA < GTPR is detected in the crest, the output protection status changes to GTSOS.SOS[1:0] bits = 00b, and toggle output due to a compare match is performed normally.

Figure 21.178 Example of Output Protection Function Operation When GTCCRA ≥ GTPR is Set during Buffer Transfer at Crests (Restored to 0 < GTCCRA < GTPR during Buffer Transfer at Crests, Active Level: Low) (n = 0 to 9)



(1) Normal operation period : Toggle output due to a compare match is performed normally.
(2) Output holding period : When GTCCRA ≥ GTPR is detected in the crest, the output protection status changes to GTSOS.SOS[1:0] = 11b, and the output is held while GTCCRA ≥ GTPR.
(3) Recovery period : When 0 < GTCCRA < GTPR is detected in the trough, the output protection status changes to GTSOS.SOS[1:0] = 00b, and holding output is maintained up to the next crest.
(4) Normal operation period : Toggle output due to a compare match is performed normally.

Figure 21.179 Example of Output Protection Function Operation When GTCCRA ≥ GTPR is Set during Buffer Transfer at Crests (Restored to 0 < GTCCRA < GTPR during Buffer Transfer at Troughs, Active Level: Low) (n = 0 to 9)

(4) Restricted Specification of Output Protection Function

Even if an incorrect value (0x0000 0000 or a value greater than or equal to the GTPR register value) is set in the GTCCRA register during count operation, the output protection functions in a specific way such that one of the positive- and negative-phase outputs becomes non-active. However, if the following condition is not satisfied, the output protection does not operate normally.

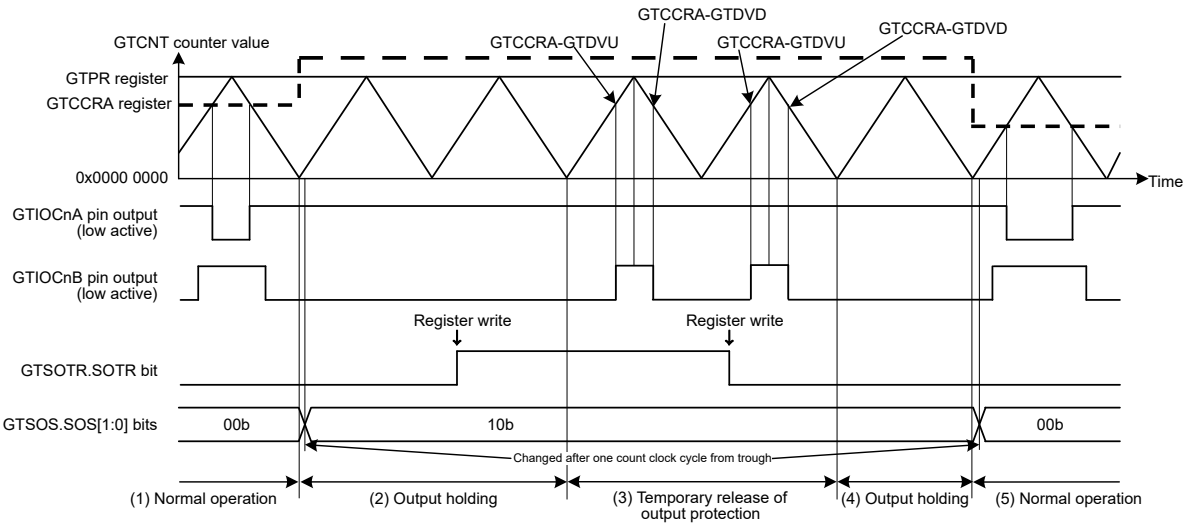
- When the GTCCRA register value at the start of count operation is greater than 0x0000 0000, and less than the setting value of the GTPR register

(5) Temporary Release of Output Protection Function

When the GTSOS.SOS[1:0] bits = 10b (protected state in which GTCCRA register ≥ GTPR register has occurred during transfer at trough), the protected state of the GTIOCnB pin output can be temporarily released by setting the GTSOTR.SOTR bit to 1. The SOS[1:0] bits retain 10b even if the output protection function is released.

When the SOTR bit is set to 0, the GTIOCnB pin output protection can be restarted.

Figure 21.180 shows examples of the operation of temporary release of output protection when the setting of the GTCCRA register ≥ GTPR register during buffer transfer at troughs.



(1) Normal operation period:	Toggle output due to a compare match is performed normally.
(2) Output holding period:	When GTCCRA ≥ GTPR is detected in the trough, the output protection status changes to GTSOS.SOS[1:0] bits = 10b, and the output is held while GTCCRA ≥ GTPR. Even when the GTSOTR.SOTR bit is set to 1 during the output holding period, the GTIOCnB pin output is held until the first trough after the setting.
(3) Output protection temporary release period:	After GTSOTR.SOTR bit is set to 1, the GTIOCnB pin output is performed normally from the first trough after the setting. Even when the GTSOTR.SOTR bit is set to 0 during the output protection releasing period, the GTIOCnB pin output is held until the first trough after the setting.
(4) Output holding period:	The output is held until 0 < GTCCRA < GTPR is detected in the trough.
(5) Normal operation period:	When 0 < GTCCRA < GTPR is detected in the trough, the output protection status changes to GTSOS.SOS[1:0] bits = 00b, and toggle output due to a compare match is performed normally.

Figure 21.180 Example of Temporary Release of Output Protection When the Setting of the GTCCRA ≥ GTPR during Buffer Transfer at Trough s (Restored to 0 < GTCCRA < GTPR during Buffer Transfer at Troughs, Active Level: Low) (n = 0 to 9)

21.9 Initialization Method of Output Pins

21.9.1 Pin Settings after Reset

The GPT registers are initialized at a reset. Start counting after selecting the port pin function with the PmnPFS register, setting GTIOR.OAE and GTIOR.OBE bits, and outputting the GPT function to external pins.

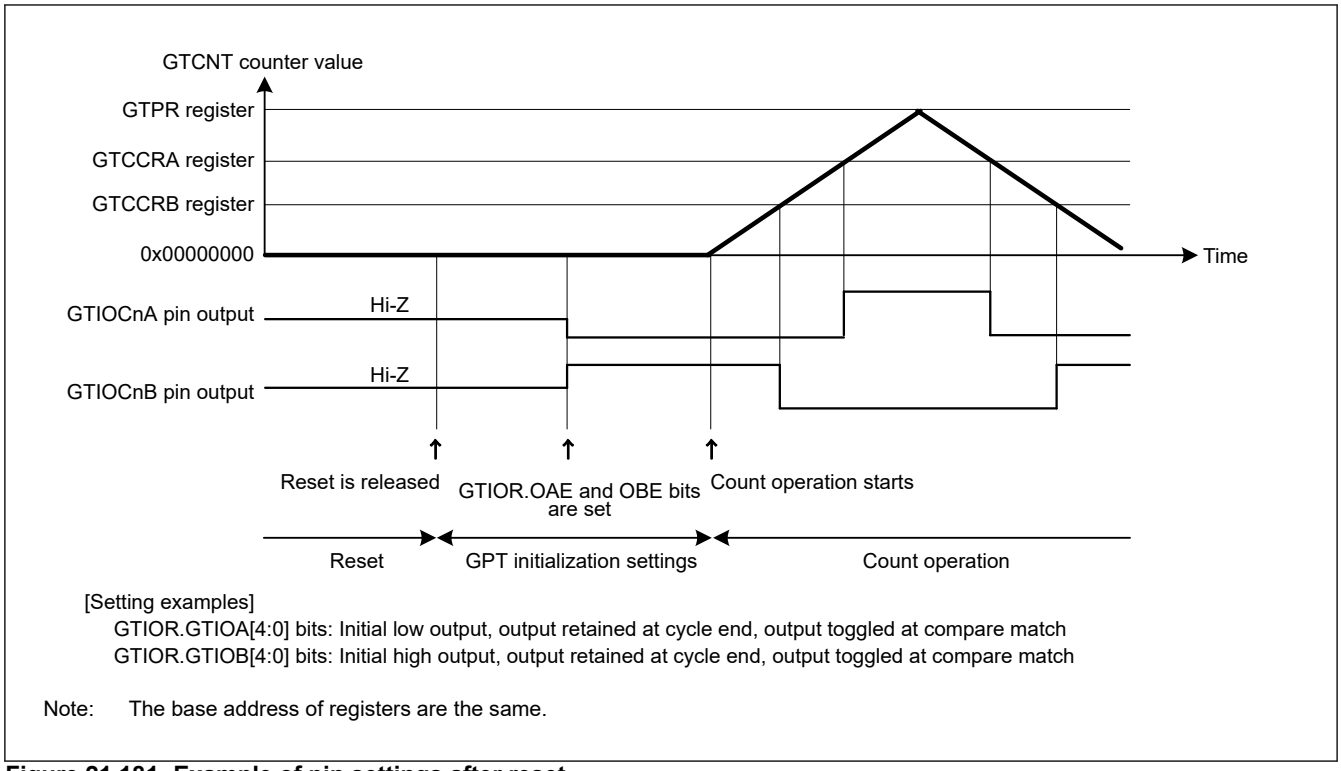


Figure 21.181 Example of pin settings after reset

21.9.2 Pin Initialization Due to Error during Operation

If an error occurs during GPT operation, the following four types of pin control can be performed before pin initialization:

- Set the OAHLD and OBHLD bits in GTIOR to 1 and retain the outputs at count stop
- Set the OAHLD and OBHLD bits in GTIOR to 0, specify arbitrary output values at OADFLT and OBDFLT in GTIOR, and output the arbitrary values at count stop
- Set the pin to output an arbitrary value as a general output port by setting the PDR, PODR registers and PmnPFS.PMR bit of the I/O port in advance. Set the OAE and OBE bits in GTIOR to 0, and the control bit associated with the pin in the PMR to 0 to allow arbitrary values to be output from the pin set as a general output port when an error occurs.
- Drive the output to a high impedance state using the POEG function.

If the automatic dead time setting is made, clear the GTDTCR.TDE bit to 0 after counting stops. When counting stops, only the values of registers that are changed by a GPT external source change. If counting is resumed, operation continues from where it stopped. If counting is stopped, the registers must be initialized before counting starts.

21.10 Usage Notes

21.10.1 Module-Stop Function Setting

The Module Stop Control Register can enable or disable GPT operation. The GPT is initially stopped after a reset. Releasing the module-stop state enables access to the registers. For details, see section 10, Low Power Modes.

Set GTCLKCR register before releasing the module-stop state.

21.10.2 GTCCRn Settings during Compare Match Operation (n = A to F)

(1) When automatic dead time setting is made in triangle-wave PWM mode

The GTCCRA register must satisfy all of the following conditions:

- GTDVU < GTCCRA
- GTCCRA > GTDVD
- 0 < GTCCRA < GTPR

When the setting of GTCCRA = 0 or GTCCRA ≥ GTPR is made for the GTCCRA register during count operation, the output protection function is activated. However, if the following condition is not satisfied, the output protection function does not work normally:

- The value of the GTCCRA register at the start of counting is larger than 0 and less than GTPR.

For details, see section 21.8.4. Output Protection Function for GTIOCnm Pin Output (n = 0 to 9; m = A, B)

(2) When automatic dead time setting is not made in triangle-wave PWM mode

The GTCCRA register must be set within the range of 0 < GTCCRA < GTPR. If GTCCRA = 0 or GTCCRA = GTPR is set, a compare match occurs within the cycle only when GTCCRA = 0 or GTCCRA = GTPR is satisfied. When GTCCRA > GTPR, no compare match occurs.

Similarly, GTCCRB must be set within the range of 0 < GTCCRB < GTPR. If GTCCRB = 0 or GTCCRB = GTPR is set, a compare match occurs within the cycle only when GTCCRB is 0 or GTCCRB = GTPR is satisfied. When GTCCRB > GTPR, no compare match occurs.

(3) When automatic dead time setting is made in saw-wave one-shot pulse mode

The GTCCRC and GTCCRD registers must be set to satisfy the following restrictions. If the restrictions are not satisfied, the correct output waveforms with secured dead time may not be obtained.

- In up-counting: GTCCRC < GTCCRD, GTCCRC > GTDVU, GTCCRD < GTPR - GTDVD
- In down-counting: GTCCRC > GTCCRD, GTCCRC < GTPR - GTDVU, GTCCRD > GTDVD

(4) When automatic dead time setting is not made in saw-wave one-shot pulse mode

The GTCCRC and GTCCRD registers must be set to satisfy the following restrictions. If the restrictions are not satisfied, two compare matches do not occur and pulse output cannot be performed.

- In up-counting: 0 < GTCCRC < GTCCRD < GTPR
- In down-counting: GTPR > GTCCRC > GTCCRD > 0

Similarly, GTCCRE and GTCCRF must be set to satisfy the following restrictions. If the restrictions are not satisfied, two compare matches do not occur and pulse output cannot be performed.

- In up-counting: 0 < GTCCRE < GTCCRF < GTPR
- In down-counting: GTPR > GTCCRE > GTCCRF > 0.

(5) In saw-wave PWM mode

The GTCCRA register must be set with the range of 0 < GTCCRA < GTPR. If GTCCRA = 0 or GTCCRA = GTPR is set, a compare match occurs within the cycle only when GTCCRA = 0 or GTCCRA = GTPR is satisfied. If GTCCRA > GTPR is set, no compare match occurs.

Similarly, GTCCRB must be set with the range of 0 < GTCCRB < GTPR. If GTCCRB = 0 or GTCCRB = GTPR is set, a compare match occurs within the cycle only when GTCCRB = 0 or GTCCRB = GTPR is satisfied. If GTCCRB > GTPR is set, no compare match occurs.

(6) In Complementary PWM mode 1, 2, 3

The GTCCRn register must be set with the range of 0 ≤ GTCCRn ≤ GTPR + GTDVU.

(7) In Complementary PWM mode 4

In single buffer operation, the GTCCRn register must be set with the range of 0 ≤ GTCCRn ≤ GTPR + GTDVU.

In double buffer operation, the GTCCRn register must be set with the range of GTDVU < GTCCRn < GTPR.

21.10.3 Setting Range for GTCNT Counter

Other than the Saw-wave PWM mode 2 and Complementary PWM mode, the GTCNT counter register must be set with the range of 0 ≤ GTCNT ≤ GTPR.

21.10.4 Starting and Stopping the GTCNT Counter

The control timing of starting and stopping the GTCNT counter by the GTCR.CST bit synchronizes the count clock that is selected in GTCR.TPCS[3:0]. When GTCR.CST is updated, the GTCNT counter starts/stops after a count clock that is selected in GTCR.TPCS[3:0]. Therefore, an event generated before the GTCNT counter actually starts is ignored, resulting in situations in which an event is accepted or an interrupt occurs after GTCR.CST is set to 0.

21.10.5 Priority Order of Each Event

(1) GTCNT register

Table 21.74 shows a priority order of events updating the GTCNT register.

Table 21.74 Priority order of sources updating GTCNT

Source updating GTCNT	Priority order
Writing by CPU (writing to GTCNT/GTCLR)	High
Clear by hardware sources set in GTCSR	↑
Count up or down by hardware sources set in GTUPSR/GTDNSR	↑
Count operation	Low

If up-counting and down-counting by hardware sources occur at the same time, the GTCNT counter value does not change. When there is a conflict between updating the GTCNT register and reading by the CPU, pre-update data is read.

(2) GTCR.CST bit

When there is a conflict between starting/stopping by hardware sources set in the GTSSR/GTPSR registers and writing by the CPU (writing to GTCR/GTSTR/GTSTP registers), the writing by CPU has priority over the starting/stopping by hardware sources.

In case that stop by the period count function conflicts with start by the CPU writing (GTCR register writing/GTSTR register writing), the period count function is finished with setting the GTST.PCF flag. The CST bit is not changed and the GTCNT continues to count.

When there is a conflict between starting by hardware sources set in the GTSSR register and stopping by hardware sources set in GTPSR register, the GTCR.CST bit value does not change. When there is a conflict between updating the GTCR.CST bit and reading by the CPU (reading from GTCR/GTSTR/GTSTP registers), pre-update data is read.

(3) GTCCRm registers (m = A to F)

When there is a conflict between input capture/buffer transfer operation and writing to the GTCCRm registers, the writing to GTCCRm registers has priority over input capture/buffer transfer operation. When there is a conflict between input capture and writing to the counter register by the CPU or updating the counter register by hardware sources, the pre-update counter value is captured. When there is a conflict between updating the GTCCRm registers and reading by the CPU, pre-update data is read.

(4) GTPR register

When there is a conflict between buffer transfer operation and writing to the GTPR register, writing to GTPR register has priority over buffer transfer operation. When there is a conflict between updating GTPR register and reading by the CPU, pre-update data is read.

(5) GTADTRm registers (m = A, B)

When there is a conflict between buffer transfer operation and writing to GTADTRm register, writing to GTADTRm register has priority over buffer transfer operation.

When there is a conflict between updating the GTADTRm register and reading by the CPU, pre-update data is read.

(6) GTDVm registers (m = U, D)

When there is a conflict between buffer transfer operation and writing to GTDVm register, writing to GTDVm register has priority over buffer transfer operation.

When there is a conflict between updating the GTDVm register and reading by the CPU, pre-update data is read.

(7) GTIOR.GTIOm registers (m = A, B)

When there is a conflict between buffer transfer operation and writing to GTIOR.GTIOm register, writing to GTIOR.GTIOm register has priority over buffer transfer operation.

When there is a conflict between updating the GTIOR.GTIOm and reading by the CPU, pre-update data is read.

21.10.6 Interval of interrupt request

When the core clock of GPT is GPTCLK, interrupt may be lost if the interval between the same interrupt signal is shorter than the following value. However, this restriction does not apply to different interrupt signals.

Interrupt_Interval [ns] = Period_of_GPTCLK [ns] * 6 + Period_of_PCLKA [ns] * 4

For the restriction of event signal, see section 17, Event Link Controller (ELC).

Also, ADC can receive the A/D conversion start request from GPT without going through ELC by setting ADTRGGPTx register (x = 0 to 8).

Again, when the clocks of GPT and ADC are combined as shown in Table 21.75, the A/D conversion start request may be lost if the interval between one A/D conversion start request and the next is less than the following value for the same A/D conversion start request.

However, this restriction does not apply to different A/D conversion start request.

Event_Interval [ns] = Period_of_GPT_core_clock [ns] * 6 + Period_of_ADC_core_clock [ns] * 4

Table 21.75 Combination of clocks with the restricted Event Interval

GPT core clock	ADC core clock
GPTCLK	PCLKA or PCLKC
PCLKD	PCLKC or GPTCLK

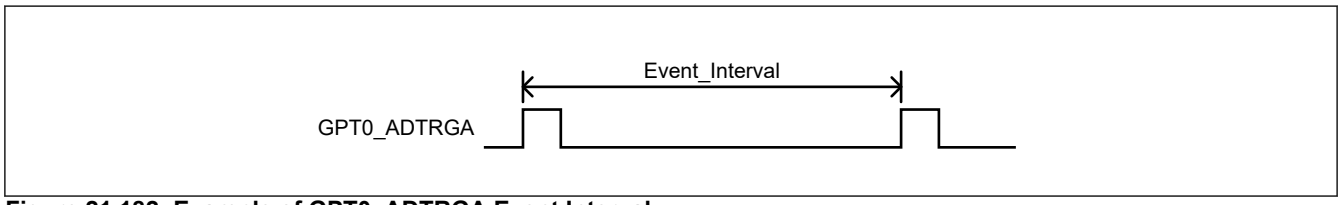


Figure 21.182 Example of GPT0_ADTRGA Event Interval

21.10.7 Notes on the GTIOCnm signal input to PWM Delay Generation Circuit (n = 0 to 3, m = A, B)

When controlling the delay of PWM waveform in PWM Delay Generation Circuit, the following limitations exist.

- In saw-wave mode, It is prohibited to change the GTIOCnm signal during the three clock cycles immediately before overflow or underflow.
- In saw-wave mode, it is prohibited to clear the GTCNT register by GTCSR during counting operation.
- In triangle-wave mode, It is prohibited to change the GTIOCnm signal during the three clock cycles immediately before trough.

If the above limitations are not followed, the edge of signal waveform output from PWM Delay Generation Circuit may disappear.

Figure 21.183 shows an example of the change timing of an unacceptable GTIOCnm signal.

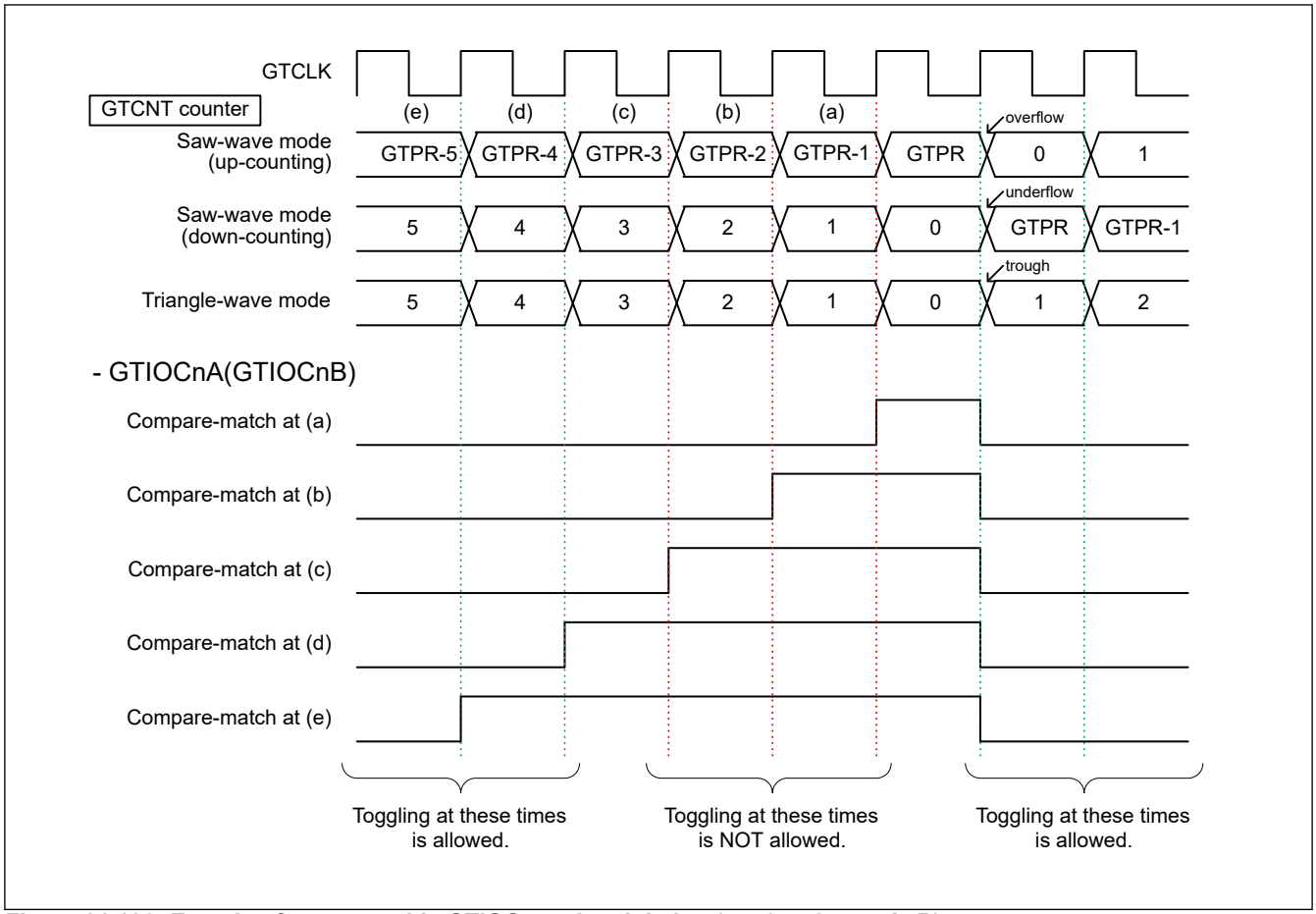


Figure 21.183 Eample of unacceptable GTIOCnm signal timing (n = 0 to 3, m = A, B)

22. PWM Delay Generation Circuit (PDG)

22.1 Overview

The PWM Delay Generation circuit (PDG) has 4 channels delay circuits that can connect to the GPT. The PDG can control the rise and fall edge timing with which the PWM output for the GPT320 through the GPT323.

Table 22.1 lists the specifications for the PWM Delay Generation Circuit, Figure 22.1 shows a block diagram, and Table 22.2 lists the I/O pins.

Table 22.1 Specifications of the PWM Delay Generation Circuit

Parameter	Specifications
Function	The circuit can control the timing with which signals on the two PWM output pins for channel 0/1/2/3 rise and fall to an accuracy of up to 1/32 times the period of the GPT core clock (GTCLK). The GPT core clock (GTCLK) can be selected from PCLKD or GPTCLK.

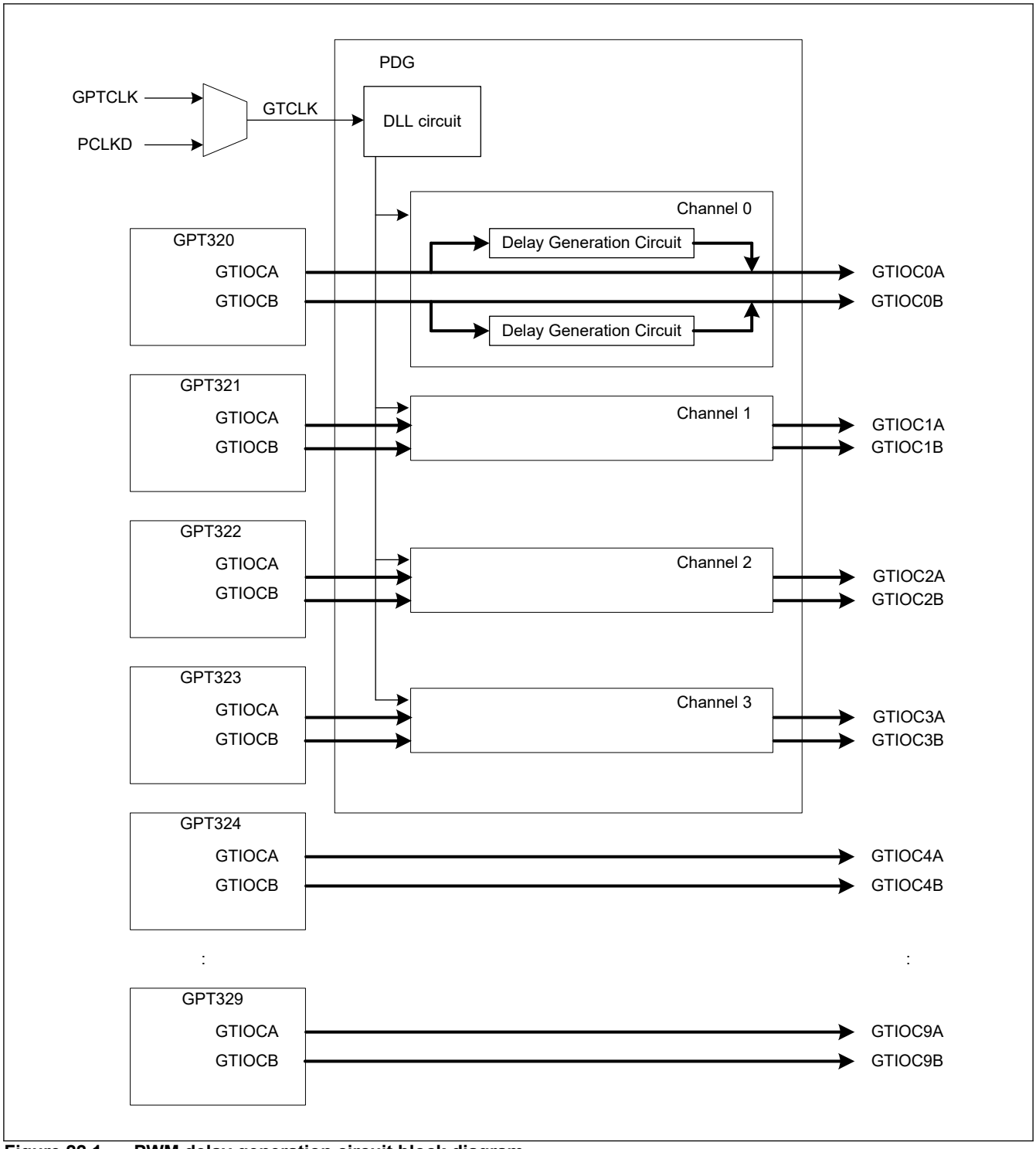


Figure 22.1 PWM delay generation circuit block diagram

Table 22.2 PWM delay generation circuit I/O pins (1 of 2)

I/O pin	I/O	Function
GTIOC0A	Output	Delayed output of GTIOCA pin of GPT channel 0
GTIOC0B	Output	Delayed output of GTIOCB pin of GPT channel 0
GTIOC1A	Output	Delayed output of GTIOCA pin of GPT channel 1
GTIOC1B	Output	Delayed output of GTIOCB pin of GPT channel 1
GTIOC2A	Output	Delayed output of GTIOCA pin of GPT channel 2

Table 22.2 PWM delay generation circuit I/O pins (2 of 2)

I/O pin	I/O	Function
GTIOC2B	Output	Delayed output of GTIOCB pin of GPT channel 2
GTIOC3A	Output	Delayed output of GTIOCA pin of GPT channel 3
GTIOC3B	Output	Delayed output of GTIOCB pin of GPT channel 3

22.2 Register Descriptions

22.2.1 GTDLYCR : PWM Output Delay Control Register

Base address: PDG = 0x4016_A000

Offset address: 0x0000

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

Bit	Symbol	Function	R/W
0	DLLEN	DLL Operation Enable 0: DLL operation disabled 1: DLL operation enabled	R/W
1	DLYRST	PWM Delay Generation Circuit Reset 0: Normal operation 1: Reset	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
8	FRANGE	GPT core clock Frequency Range 0: GPT core clock frequency is 115 MHz to 200 MHz 1: GPT core clock frequency is 80 MHz to 120 MHz	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W

The GTDLYCR register controls the PWM delay generation circuit, which applies delays to the PWM outputs. GTDLYCR register can be written when register write protection is disabled (GPT320.GTWP.WP = 0).

When changing GTDLYCR after changing the value of GPT320.GTWP.WP bit, be sure to read back the value of GTWP register before changing the value of GTDLYCR.

DLLEN bit (DLL Operation Enable)

The DLLEN bit selects whether the on-chip DLL in the PWM delay generation circuit is activated or not.

DLYRST bit (PWM Delay Generation Circuit Reset)

The DLYRST bit resets the internal state of the PWM delay generation circuit.

FRANGE bit (GPT core clock Frequency Range)

The FRANGE bit sets the frequency range of the GPT core clock.

Set the FRANGE bit only when the DLLEN bit is 0.

22.2.2 GTDLYCR2 : PWM Output Delay Control Register 2

Base address: PDG = 0x4016_A000

Offset address: 0x0002

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	DLYEN3	DLYEN2	DLYEN1	DLYEN0	—	—	—	—	DLYBS3	DLYBS2	DLYBS1	DLYBS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DLYBS0	PWM Delay Generation Circuit bypass for channel 0 0: Delay generation circuit of channel 0 bypassed 1: Delay generation circuit of channel 0 not bypassed	R/W
1	DLYBS1	PWM Delay Generation Circuit bypass for channel 1 0: Delay generation circuit of channel 1 bypassed 1: Delay generation circuit of channel 1 not bypassed	R/W
2	DLYBS2	PWM Delay Generation Circuit bypass for channel 2 0: Delay generation circuit of channel 2 bypassed 1: Delay generation circuit of channel 2 not bypassed	R/W
3	DLYBS3	PWM Delay Generation Circuit bypass for channel 3 0: Delay generation circuit of channel 3 bypassed 1: Delay generation circuit of channel 3 not bypassed	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
8	DLYEN0	PWM Delay Generation Circuit enable for channel 0 0: Delay generation circuit of channel 0 enabled 1: Delay generation circuit of channel 0 disabled	R/W
9	DLYEN1	PWM Delay Generation Circuit enable for channel 1 0: Delay generation circuit of channel 1 enabled 1: Delay generation circuit of channel 1 disabled	R/W
10	DLYEN2	PWM Delay Generation Circuit enable for channel 2 0: Delay generation circuit of channel 2 enabled 1: Delay generation circuit of channel 2 disabled	R/W
11	DLYEN3	PWM Delay Generation Circuit enable for channel 3 0: Delay generation circuit of channel 3 enabled 1: Delay generation circuit of channel 3 disabled	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W

The GTDLYCR2 register controls each channel of PWM delay generation circuit. GTDLYCR2 can be written when register write protection is disabled (GPT320.GTWP.WP = 0).

When changing GTDLYCR2 after changing the value of GPT320.GTWP.WP bit, be sure to read back the value of GTWP register before changing the value of GTDLYCR2.

DLYBSn (n = 0 to 3) bit (PWM Delay Generation Circuit Bypass for channel n)

The DLYBSn bit selects whether delays are applied to PWM output signals from the GTIOCnA and GTIOCnB pins (n = 0 to 3) by the PWM delay generation circuit or whether the circuit is bypassed.

A signal delayed in the PWM delay generation circuit is output 3 cycles of GPT core clock (GTCLK) later than if it bypasses the PWM delay generation circuit.

DLYENn (n = 0 to 3) bit (PWM Delay Generation Circuit Enable for channel n)

The DLYENn bit selects whether channel n (n = 0 to 3) of PWM delay generation circuit is power on or off. If channel n of the PWM delay generation circuit is not used, set to 1.

22.2.3 GTDLYRnA : GTIOCnA Rising Output Delay Register (n = 0 to 3)

Base address: PDG = 0x4016_A000

Offset address: 0x018 + 0x4 × n

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

Bit	Symbol	Function	R/W
4:0	DLY[4:0]	GTIOCnA Output Rising Edge Delay Setting 0x00: Delay on rising edges is not applied 0x01: Delay of 1/32 times GTCLK period applied 0x02: Delay of 2/32 times GTCLK period applied 0x03: Delay of 3/32 times GTCLK period applied 0x04: Delay of 4/32 times GTCLK period applied 0x05: Delay of 5/32 times GTCLK period applied 0x06: Delay of 6/32 times GTCLK period applied 0x07: Delay of 7/32 times GTCLK period applied 0x08: Delay of 8/32 times GTCLK period applied 0x09: Delay of 9/32 times GTCLK period applied 0x0A: Delay of 10/32 times GTCLK period applied 0x0B: Delay of 11/32 times GTCLK period applied 0x0C: Delay of 12/32 times GTCLK period applied 0x0D: Delay of 13/32 times GTCLK period applied 0x0E: Delay of 14/32 times GTCLK period applied 0x0F: Delay of 15/32 times GTCLK period applied 0x10: Delay of 16/32 times GTCLK period applied 0x11: Delay of 17/32 times GTCLK period applied 0x12: Delay of 18/32 times GTCLK period applied 0x13: Delay of 19/32 times GTCLK period applied 0x14: Delay of 20/32 times GTCLK period applied 0x15: Delay of 21/32 times GTCLK period applied 0x16: Delay of 22/32 times GTCLK period applied 0x17: Delay of 23/32 times GTCLK period applied 0x18: Delay of 24/32 times GTCLK period applied 0x19: Delay of 25/32 times GTCLK period applied 0x1A: Delay of 26/32 times GTCLK period applied 0x1B: Delay of 27/32 times GTCLK period applied 0x1C: Delay of 28/32 times GTCLK period applied 0x1D: Delay of 29/32 times GTCLK period applied 0x1E: Delay of 30/32 times GTCLK period applied 0x1F: Delay of 31/32 times GTCLK period applied	R/W
15:5	—	These bits are read as 0. The write value should be 0.	R/W

The GTDLYRnA register sets a delay to be applied to rising edges of output signals on the GTIOCnA pin. On the timing for the transfer of settings, see section 22.3.2. Timing for Transfer of GTDLYRnA, GTDLYRnB, GTDLYFnA, and GTDLYFnB Register Settings.

GTDLYRnA can be written when register write protection is disabled (GPT32n.GTWP.WP = 0).

When changing GTDLYRnA after changing the value of GPT32n.GTWP.WP bit, be sure to read back the value of GTWP register before changing the value of GTDLYRnA.

22.2.4 GTDLYFnA : GTIOCnA Falling Output Delay Register (n = 0 to 3)

Base address: PDG = 0x4016_A000

Offset address: 0x028 + 0x4 × n

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

Bit	Symbol	Function	R/W
4:0	DLY[4:0]	GTIOCnA Output Falling Edge Delay Setting 0x00: Delay on falling edges is not applied 0x01: Delay of 1/32 times GTCLK period applied 0x02: Delay of 2/32 times GTCLK period applied 0x03: Delay of 3/32 times GTCLK period applied 0x04: Delay of 4/32 times GTCLK period applied 0x05: Delay of 5/32 times GTCLK period applied 0x06: Delay of 6/32 times GTCLK period applied 0x07: Delay of 7/32 times GTCLK period applied 0x08: Delay of 8/32 times GTCLK period applied 0x09: Delay of 9/32 times GTCLK period applied 0x0A: Delay of 10/32 times GTCLK period applied 0x0B: Delay of 11/32 times GTCLK period applied 0x0C: Delay of 12/32 times GTCLK period applied 0x0D: Delay of 13/32 times GTCLK period applied 0x0E: Delay of 14/32 times GTCLK period applied 0x0F: Delay of 15/32 times GTCLK period applied 0x10: Delay of 16/32 times GTCLK period applied 0x11: Delay of 17/32 times GTCLK period applied 0x12: Delay of 18/32 times GTCLK period applied 0x13: Delay of 19/32 times GTCLK period applied 0x14: Delay of 20/32 times GTCLK period applied 0x15: Delay of 21/32 times GTCLK period applied 0x16: Delay of 22/32 times GTCLK period applied 0x17: Delay of 23/32 times GTCLK period applied 0x18: Delay of 24/32 times GTCLK period applied 0x19: Delay of 25/32 times GTCLK period applied 0x1A: Delay of 26/32 times GTCLK period applied 0x1B: Delay of 27/32 times GTCLK period applied 0x1C: Delay of 28/32 times GTCLK period applied 0x1D: Delay of 29/32 times GTCLK period applied 0x1E: Delay of 30/32 times GTCLK period applied 0x1F: Delay of 31/32 times GTCLK period applied	R/W
15:5	—	These bits are read as 0. The write value should be 0.	R/W

The GTDLYFnA register sets a delay to be applied to falling edges of output signals on the GTIOCnA pin. On the timing for the transfer of settings, see section 22.3.2. Timing for Transfer of GTDLYRnA, GTDLYRnB, GTDLYFnA, and GTDLYFnB Register Settings.

GTDLYFnA can be written when register write protection is disabled (GPT32n.GTWP.WP = 0).

When changing GTDLYFnA after changing the value of GPT32n.GTWP.WP bit, be sure to read back the value of GTWP register before changing the value of GTDLYFnA.

22.2.5 GTDLYRnB : GTIOCnB Rising Output Delay Register (n = 0 to 3)

Base address: PDG = 0x4016_A000

Offset address: 0x01A + 0x4 × n

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

Bit	Symbol	Function	R/W
4:0	DLY[4:0]	GTIOCnB Output Rising Edge Delay Setting 0x00: Delay on rising edges is not applied 0x01: Delay of 1/32 times GTCLK period applied 0x02: Delay of 2/32 times GTCLK period applied 0x03: Delay of 3/32 times GTCLK period applied 0x04: Delay of 4/32 times GTCLK period applied 0x05: Delay of 5/32 times GTCLK period applied 0x06: Delay of 6/32 times GTCLK period applied 0x07: Delay of 7/32 times GTCLK period applied 0x08: Delay of 8/32 times GTCLK period applied 0x09: Delay of 9/32 times GTCLK period applied 0x0A: Delay of 10/32 times GTCLK period applied 0x0B: Delay of 11/32 times GTCLK period applied 0x0C: Delay of 12/32 times GTCLK period applied 0x0D: Delay of 13/32 times GTCLK period applied 0x0E: Delay of 14/32 times GTCLK period applied 0x0F: Delay of 15/32 times GTCLK period applied 0x10: Delay of 16/32 times GTCLK period applied 0x11: Delay of 17/32 times GTCLK period applied 0x12: Delay of 18/32 times GTCLK period applied 0x13: Delay of 19/32 times GTCLK period applied 0x14: Delay of 20/32 times GTCLK period applied 0x15: Delay of 21/32 times GTCLK period applied 0x16: Delay of 22/32 times GTCLK period applied 0x17: Delay of 23/32 times GTCLK period applied 0x18: Delay of 24/32 times GTCLK period applied 0x19: Delay of 25/32 times GTCLK period applied 0x1A: Delay of 26/32 times GTCLK period applied 0x1B: Delay of 27/32 times GTCLK period applied 0x1C: Delay of 28/32 times GTCLK period applied 0x1D: Delay of 29/32 times GTCLK period applied 0x1E: Delay of 30/32 times GTCLK period applied 0x1F: Delay of 31/32 times GTCLK period applied	R/W
15:5	—	These bits are read as 0. The write value should be 0.	R/W

The GTDLYRnB register sets a delay to be applied to rising edges of output signals on the GTIOCnB pin. On the timing for the transfer of settings, see section 22.3.2. Timing for Transfer of GTDLYRnA, GTDLYRnB, GTDLYFnA, and GTDLYFnB Register Settings.

GTDLYRnB can be written when register write protection is disabled (GPT32n.GTWP.WP = 0).

When changing GTDLYRnB after changing the value of GPT32n.GTWP.WP bit, be sure to read back the value of GTWP register before changing the value of GTDLYRnB.

22.2.6 GTDLYFnB : GTIOCnB Falling Output Delay Register (n = 0 to 3)

Base address: PDG = 0x4016_A000

Offset address: 0x02A + 0x4 × n

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

Bit	Symbol	Function	R/W
4:0	DLY[4:0]	GTIOCnB Output Falling Edge Delay Setting 0x00: Delay on falling edges is not applied 0x01: Delay of 1/32 times GTCLK period applied 0x02: Delay of 2/32 times GTCLK period applied 0x03: Delay of 3/32 times GTCLK period applied 0x04: Delay of 4/32 times GTCLK period applied 0x05: Delay of 5/32 times GTCLK period applied 0x06: Delay of 6/32 times GTCLK period applied 0x07: Delay of 7/32 times GTCLK period applied 0x08: Delay of 8/32 times GTCLK period applied 0x09: Delay of 9/32 times GTCLK period applied 0x0A: Delay of 10/32 times GTCLK period applied 0x0B: Delay of 11/32 times GTCLK period applied 0x0C: Delay of 12/32 times GTCLK period applied 0x0D: Delay of 13/32 times GTCLK period applied 0x0E: Delay of 14/32 times GTCLK period applied 0x0F: Delay of 15/32 times GTCLK period applied 0x10: Delay of 16/32 times GTCLK period applied 0x11: Delay of 17/32 times GTCLK period applied 0x12: Delay of 18/32 times GTCLK period applied 0x13: Delay of 19/32 times GTCLK period applied 0x14: Delay of 20/32 times GTCLK period applied 0x15: Delay of 21/32 times GTCLK period applied 0x16: Delay of 22/32 times GTCLK period applied 0x17: Delay of 23/32 times GTCLK period applied 0x18: Delay of 24/32 times GTCLK period applied 0x19: Delay of 25/32 times GTCLK period applied 0x1A: Delay of 26/32 times GTCLK period applied 0x1B: Delay of 27/32 times GTCLK period applied 0x1C: Delay of 28/32 times GTCLK period applied 0x1D: Delay of 29/32 times GTCLK period applied 0x1E: Delay of 30/32 times GTCLK period applied 0x1F: Delay of 31/32 times GTCLK period applied	R/W
15:5	—	These bits are read as 0. The write value should be 0.	R/W

The GTDLYFnB register sets a delay to be applied to falling edges of output signals on the GTIOCnB pin. On the timing for the transfer of settings, see section 22.3.2. Timing for Transfer of GTDLYRnA, GTDLYRnB, GTDLYFnA, and GTDLYFnB Register Settings.

GTDLYFnB can be written when register write protection is disabled (GPT32n.GTWP.WP = 0).

When changing GTDLYFnB after changing the value of GPT32n.GTWP.WP bit, be sure to read back the value of GTWP register before changing the value of GTDLYFnB.

22.3 Operation

22.3.1 Adjustments to the Timing of Rising and Falling Edges in PWM Waveforms

The timing of rising and falling edges in PWM waveforms which are output from the GTIOCnA and GTIOCnB pins, where n = channel number, can be delayed to an accuracy of 1/32 of the GPT core clock (GTCLK) period.

If the timing of rising or falling edges in PWM waveforms output from the GTIOCnA and GTIOCnB pins must be adjusted, initialize the PWM generation circuit as shown in the procedure in Figure 22.2.

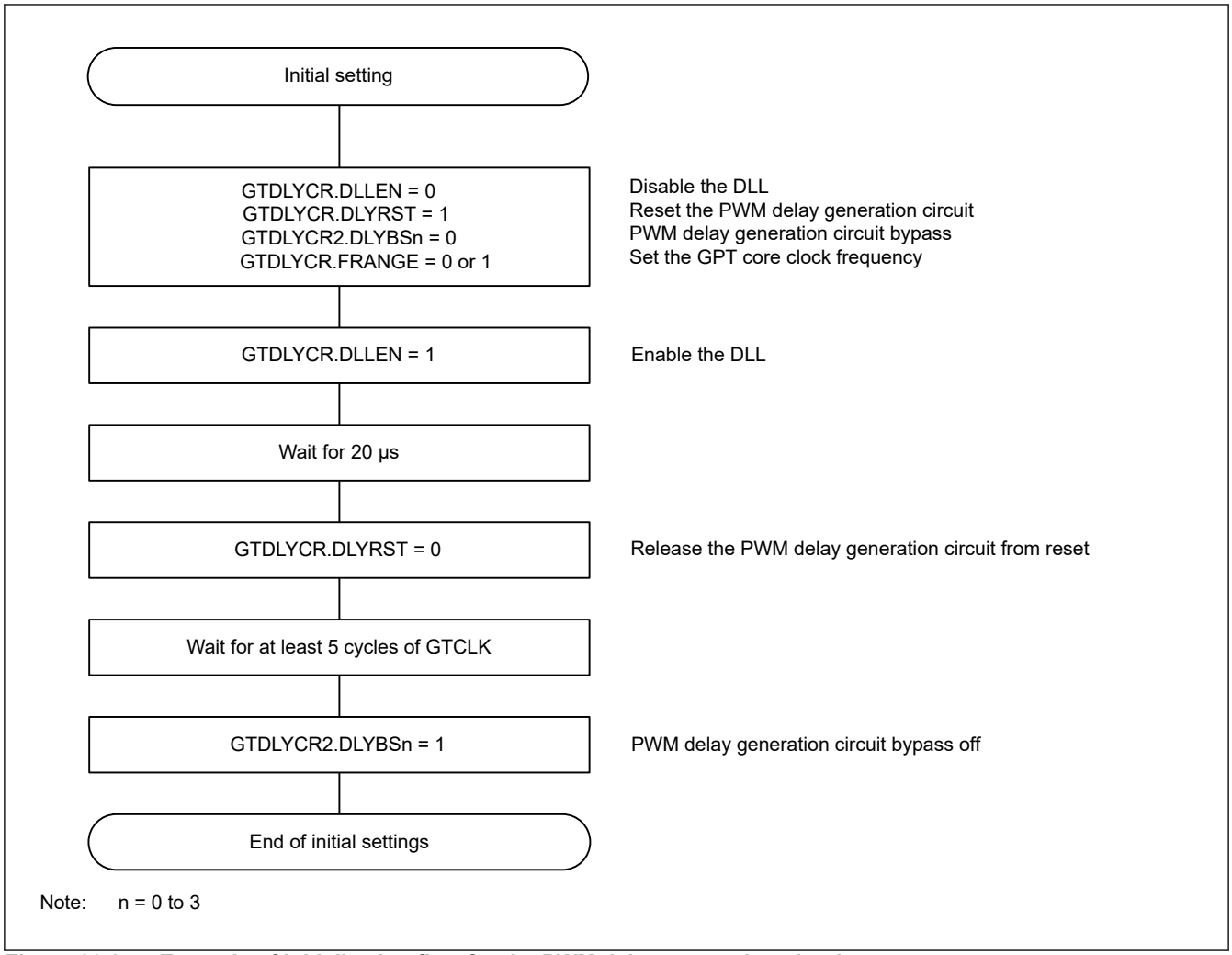


Figure 22.2 Example of initialization flow for the PWM delay generation circuit

In the PWM delay generation circuit, delay can be applied to rising and falling edges of the PWM output to an accuracy of 1/32 of the period of the GPT core clock (GTCLK). This is described in section 21.3.3. PWM Output Operating Mode. Delays associated with the settings are reflected in the PWM output with the timing described in section 22.3.2. Timing for Transfer of GTDLYRnA, GTDLYRnB, GTDLYFnA, and GTDLYFnB Register Settings. Table 22.3 shows the association between the GTDLYRnA, GTDLYRnB, GTDLYFnA, and GTDLYFnB registers and the PWM outputs.

Table 22.3 Association between PWM output pins and delay setting registers

PWM output pin	Rising-edge delay setting register	Falling-edge delay setting register
GTIOC0A	GTDLYR0A	GTDLYF0A
GTIOC0B	GTDLYR0B	GTDLYF0B
GTIOC1A	GTDLYR1A	GTDLYF1A
GTIOC1B	GTDLYR1B	GTDLYF1B
GTIOC2A	GTDLYR2A	GTDLYF2A
GTIOC2B	GTDLYR2B	GTDLYF2B
GTIOC3A	GTDLYR3A	GTDLYF3A
GTIOC3B	GTDLYR3B	GTDLYF3B

When the PWM delay generation circuit is in use, the timing with which a PWM output signal rises and falls can be controlled to an accuracy of 1/32 of the period of the GPT core clock (GTCLK). When this option is not in use, the period of the PWM output waveform is controlled to an accuracy of one period of the input clock for the timer counter, which is GTCLK. With the PWM delay generation circuit, the output can be controlled to an accuracy 32 times better. Additionally,

the delay settings also control the periods at high and low level for the PWM waveform to the given accuracy. PWM delay generation circuit channels can be individually enabled or disabled.

22.3.2 Timing for Transfer of GTDLYRnA, GTDLYRnB, GTDLYFnA, and GTDLYFnB Register Settings

Settings for the GTDLYRnA, GTDLYRnB, GTDLYFnA, and GTDLYFnB registers are initially transferred to temporary registers, and then reflected in the delay on the GTIOCnA and GTIOCnB (n = 0 to 3) outputs. Transfer of the settings takes place on overflows (in up-counting) or underflows (in down-counting) for saw waves, and in the troughs of triangle waves.

Figure 22.3 and Figure 22.4 show examples of the operation of the GTDLYR0A and GTDLYF0A registers.

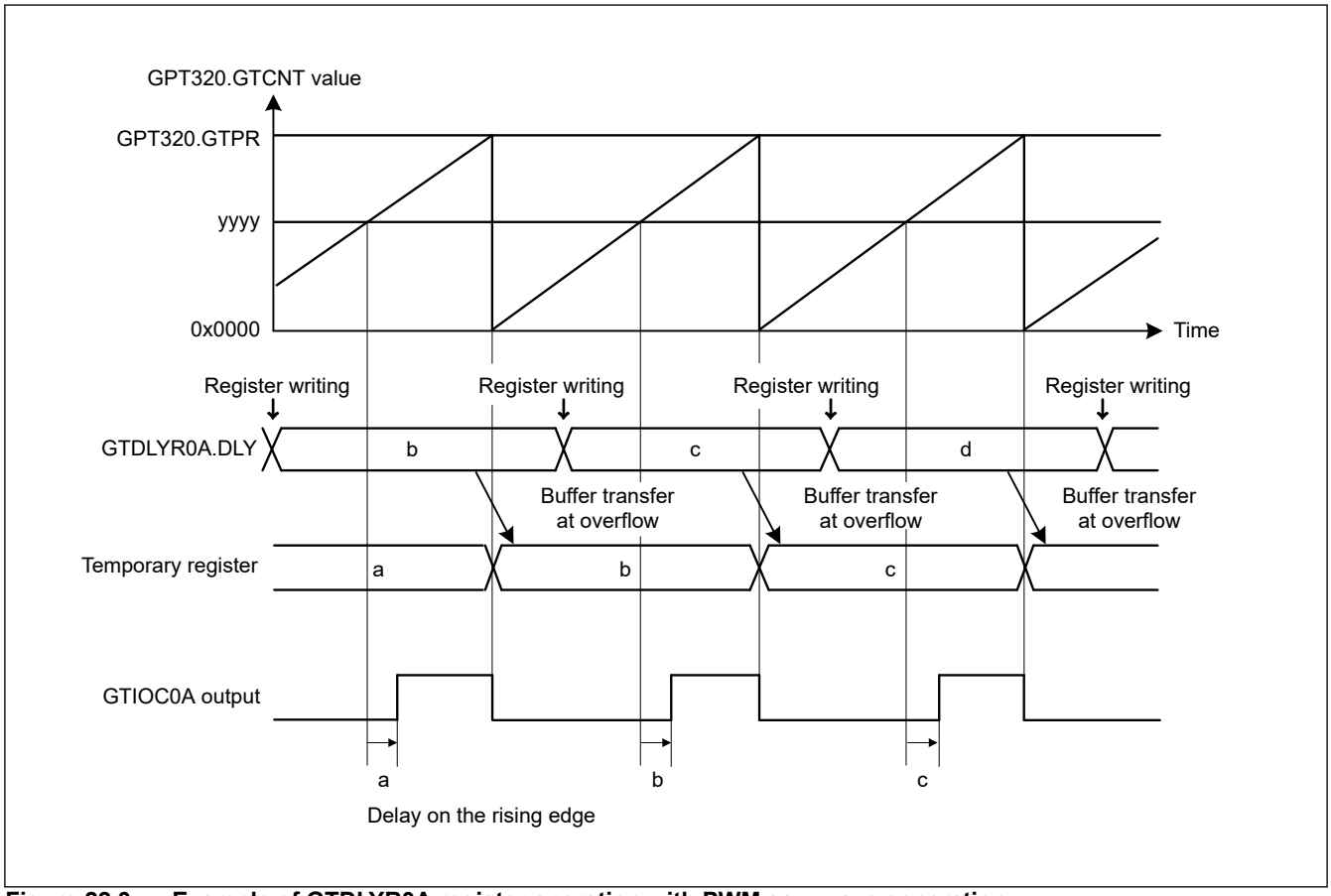


Figure 22.3 Example of GTDLYR0A register operation with PWM saw-wave generation

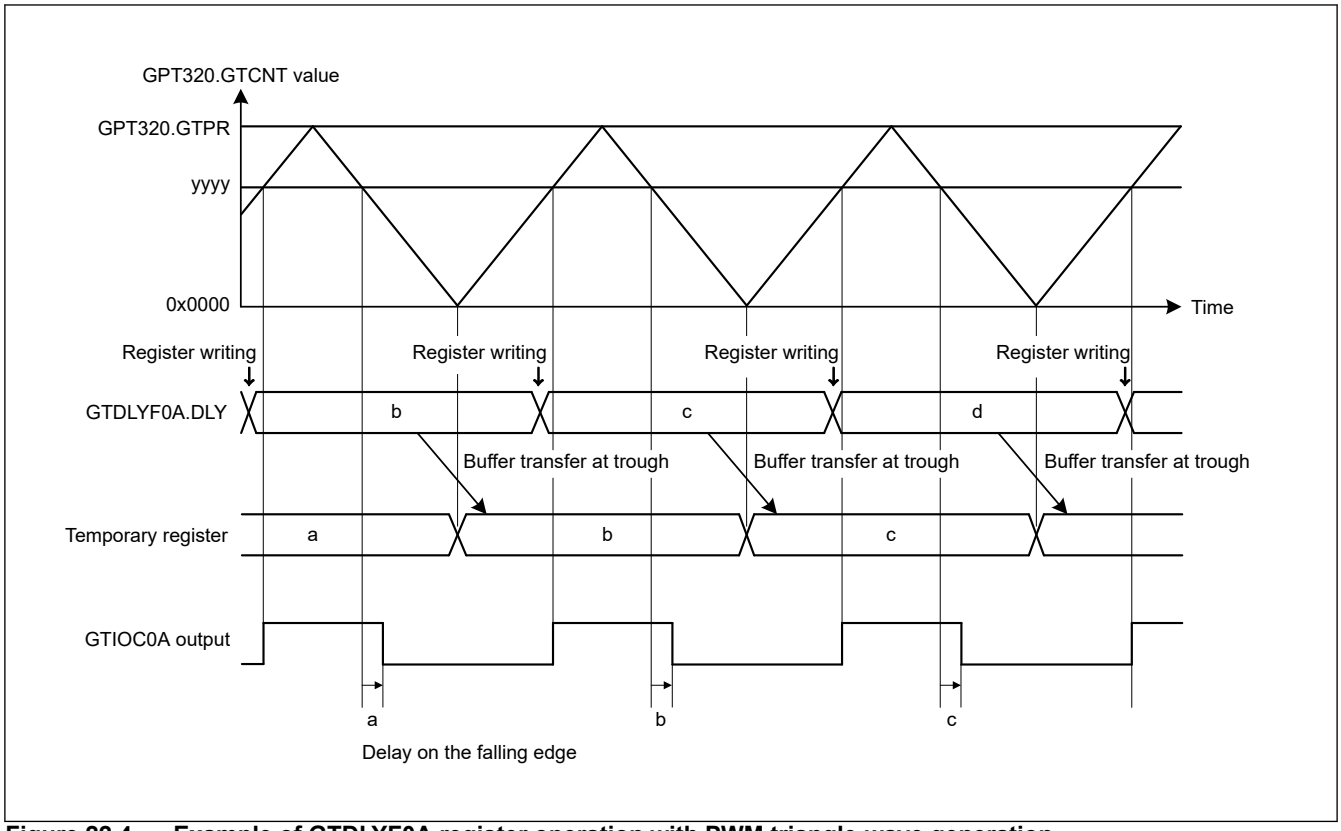


Figure 22.4 Example of GTDLYF0A register operation with PWM triangle-wave generation

22.4 Usage Notes

22.4.1 Settings for the Module-Stop Function

The Module Stop Control Register D (MSTPCRD) can enable or disable operation of the PWM delay generation circuit. The PWM delay generation circuit is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 10, Low Power Modes.

22.4.2 Notes on Delay Settings for PWM Delay Generation Circuit

When the PWM delay generation circuit generates delays for a PWM output waveform and the waveform is toggled in response to compare-matches, do not change the settings for delay while the compare-match value is within the ranges listed in Table 22.4. This constraint applies to the GTDLYFnA, GTDLYRnA, GTDLYFnB, and GTDLYRnB registers.

Table 22.4 Constraints on delay settings

Mode	Direction of counting	Compare-match value
Saw-wave mode	Up	GTPR - 2 or above
	Down	2 or below
Triangle-wave mode	Down	2 or below

Figure 22.5 shows an example of how the constraints apply to the timing of setting GTDLYFnA in saw-wave waveform one-shot pulse mode (counting up). Do not change the value set in GTDLYFnA while GTCCRD ≥ GTPR - 2.

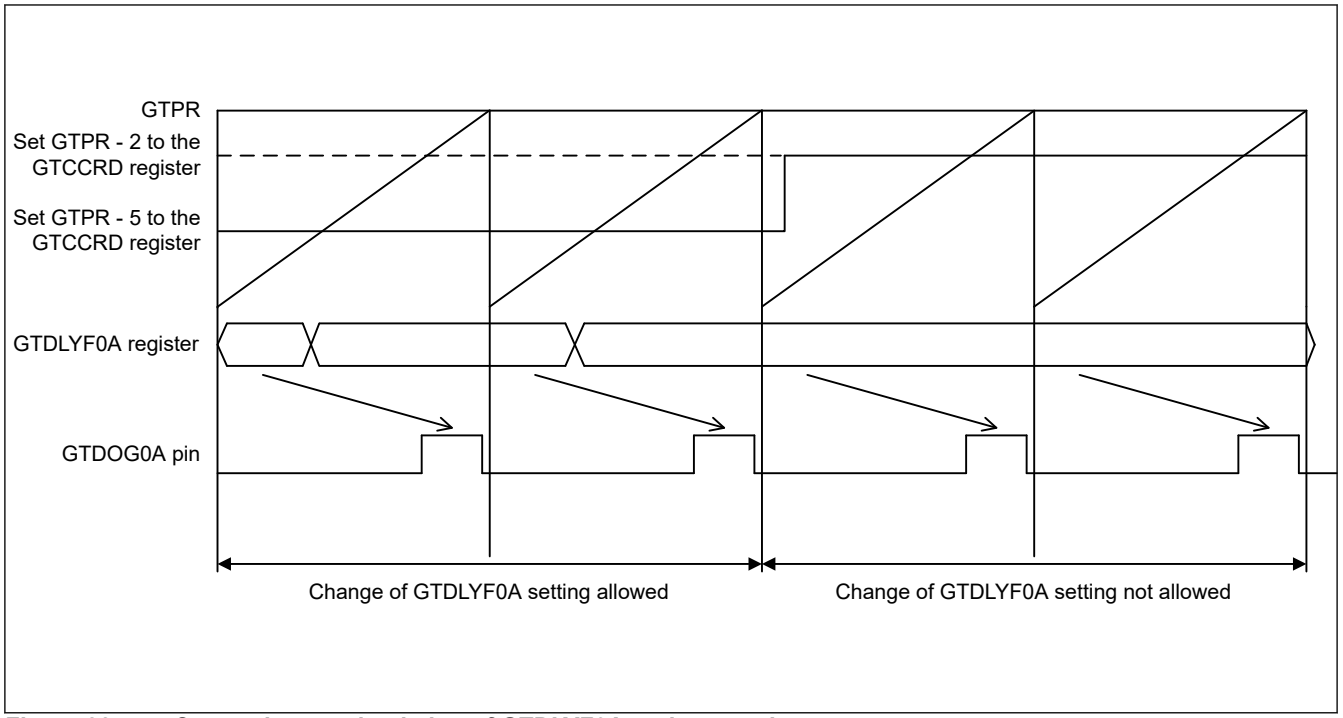


Figure 22.5 Constraints on the timing of GTDLYF0A register settings

Changing the values in the GTDLYFnA, GTDLYRnA, GTDLYFnB, and GTDLYRnB registers during periods where changes to settings are not allowed, might lead to faulty output waveforms such as shifts in the timing of output waveform transitions from the expected values.

22.4.3 Register Write Interval

When GPT core clock is GPTCLK, the writing value may not be reflected if the interval between writing to the GTDLYRnA/GTDLYFnA/GTDLYRnB/GTDLYFnA register is shorter than the following interval time. This restriction only applies to successive writes to the same register.

Write_Interval [ns] = Period_of_PCLKA [ns] × 6 + Period_of_GPTCLK [ns] × 4

23. Low Power Asynchronous General Purpose Timer (AGTW)

This is the AGTW_B version of the AGTW peripheral module.

AGTW_B is referred to as AGTW in this chapter.

23.1 Overview

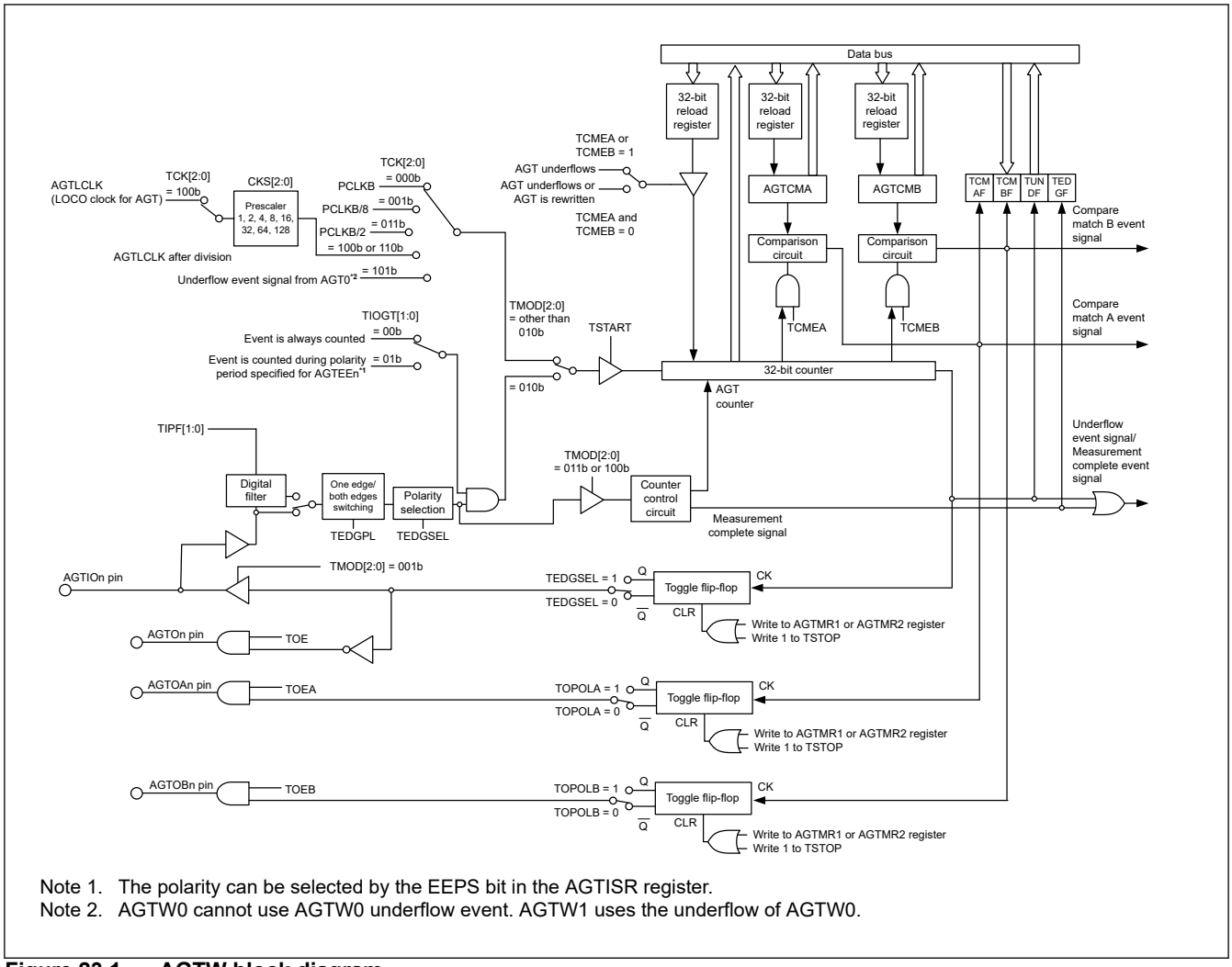
The low power Asynchronous General Purpose Timer (AGT) is a 32-bit timer that can be used for pulse output, external pulse width or period measurement, and counting external events. This timer consists of a reload register and a down counter. The reload register and the down counter are allocated to the same address, and can be accessed with the AGT register.

Table 23.1 lists the AGTW specifications, Figure 23.1 shows a block diagram, and Table 23.2 lists the I/O pins.

Table 23.1 AGTW specifications

Parameter		Description
Operating modes	Timer mode	The count source is counted
	Pulse output mode	The count source is counted and the output is inverted at each timer underflow
	Event counter mode	An external event is counted
	Pulse width measurement mode	An external pulse width is measured
	Pulse period measurement mode	An external pulse period is measured
Number of Channels		32 bits × 2 channels (AGTWn (n = 0, 1))
Count source (operating clock)*2	Timer mode	PCLKB, PCLKB/2, PCLKB/8, AGTLCLK/d (d = 1, 2, 4, 8, 16, 32, 64, or 128), or underflow signal of AGTW0 selectable.*1
	Pulse output mode	
	Pulse width measurement mode	
	Pulse period measurement mode	
	Event counting mode	External event input
Interrupt and Event Link function		• Underflow event signal or measurement complete event signal – When the counter underflows – When the measurement of the active width of the external input pin (AGTWIOn) completes in pulse width measurement mode – When the set edge of the external input pin (AGTWIOn) is input in pulse period measurement mode. • Compare match A event signal – When the values of AGT register and AGTCMA register matched (compare match A function enabled). • Compare match B event signal – When the values of AGT and AGTCMB registers matched (compare match B function enabled). • Return from Snooze mode or Software Standby mode can be performed with AGT1_AGTI, AGT1_AGTCMAI, or AGT1_AGTCMBI*3
Selectable functions		• Compare match function One or two of the AGT Compare Match A register and AGT Compare Match B register is selectable.
TrustZone Filter		Security attribution can be set for each channels

Note 1. AGTW0 cannot use underflow signal. AGTW1 connects directly with the underflow event signal from the AGTW0 timer.
Note 2. Satisfy the frequency of the peripheral module clock (PCLKB) ≥ the frequency of the count source clock.
Note 3. For details, see section 10, Low Power Modes.



Note 1. The polarity can be selected by the EEPS bit in the AGTISR register.
Note 2. AGTW0 cannot use AGTW0 underflow event. AGTW1 uses the underflow of AGTW0.

Figure 23.1 AGTW block diagram

Table 23.2 AGTW I/O pins

Pin name	I/O	Function
AGTEEn	Input	External event input enable for AGTW
AGTIOn	Input/output	External event input and pulse output for AGTW
AGTOn	Output	Pulse output for AGTW
AGTOAn	Output	Compare match A output for AGTW
AGTOBn	Output	Compare match B output for AGTW

Note: Channel number: n = 0, 1

23.2 Register Descriptions

23.2.1 AGT : AGT Counter Register

Base address: AGTW_Bn = 0x400E_8000 + 0x0100 × n (n = 0, 1)

Offset address: 0x00

Bit position:	31																												0
Bit field:																													
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
31:0	n/a	32-bit counter and reload register Setting range : 0x00000000 to 0xFFFFFFFF	R/W

AGTWn.AGT is a 32-bit register. The write value is written to the reload register and the read value is read from the counter.

The states of the reload register and the counter change according to the TSTART bit in the AGTCR register and TCMEA/TCMEB bit in the AGTCMSR register. For details, see section 23.3.1. Reload Register and Counter Rewrite Operation.

When 1 is written to the TSTOP bit in the AGTCR register, AGT counter is forcibly stopped and set to 0xFFFFFFFF.

When the TCK[2:0] bits setting in the AGTMR1 register are a value other than 001b (PCLKB/8) or 011b (PCLKB/2), if the AGT register is set to 0x00000000, a request signal to the ICU, the DTC, the DMAC, and the ELC is generated once immediately after the count starts. The and AGTWOn, AGTWIOn pin output are toggled.

When the AGT register is set to 0x00000000 in event counter mode, regardless of the value of TCK[2:0] bits, a request signal to the ICU, the DTC, the DMAC, and the ELC is generated once immediately after the count starts.

In addition, the AGTWOn pin output is toggled even during a period other than the specified count period. When the AGT register is set to 0x00000001 or more, a request signal is generated each time AGT underflows.

23.2.2 AGTCMA : AGT Compare Match A Register

Base address: AGTW_Bn = 0x400E_8000 + 0x0100 × n (n = 0, 1)

Offset address: 0x04

Bit position:	31																												0
Bit field:																													
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
31:0	n/a	32-bit compare match A data is stored.*1 Setting range : 0x00000000 to 0xFFFFFFFF	R/W

Note 1. Set the AGTCMA register to 0xFFFFFFFF when compare match A is not used.

The AGTCMA register is a read/write register to set a value for compare match with the AGT counter. The states of the reload register and compare register A change according to the TSTART bit in the AGTCR register. For details, see section 23.3.2. Reload Register and AGT Compare Match A/B Register Rewrite Operation.

23.2.3 AGTCMB : AGT Compare Match B Register

Base address: AGTW_Bn = 0x400E_8000 + 0x0100 × n (n = 0, 1)

Offset address: 0x08

Bit position:	31																												0
Bit field:																													
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
31:0	n/a	32-bit compare match B data is stored.*1 Setting range : 0x00000000 to 0xFFFFFFFF	R/W

Note 1. Set the AGTCMB register to 0xFFFFFFFF when compare match B is not used.

The AGTCMB register is a read/write register to set a value for compare match with the AGT counter. The states of the reload register and compare register B change according to the TSTART bit in the AGTCR register. For details, see section 23.3.2. Reload Register and AGT Compare Match A/B Register Rewrite Operation.

23.2.4 AGTCR : AGT Control Register

Base address: AGTW_Bn = 0x400E_8000 + 0x0100 × n (n = 0, 1)

Offset address: 0x0C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TCMBF	TCMAF	TUNDF	TEDGF	—	TSTOP	TCSTF	TSTART
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TSTART	AGT Count Start*2 0: Count stops 1: Count starts	R/W
1	TCSTF	AGT Count Status Flag*2 0: Count stopped 1: Count in progress	R
2	TSTOP	AGT Count Forced Stop*1 0: Writing is invalid 1: The count is forcibly stopped	W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	TEDGF	Active Edge Judgment Flag 0: No active edge received 1: Active edge received	R/(W)*3
5	TUNDF	Underflow Flag 0: No underflow 1: Underflow	R/(W)*3
6	TCMAF	Compare Match A Flag 0: No match 1: Match	R/(W)*3
7	TCMBF	Compare Match B Flag 0: No match 1: Match	R/(W)*3

Note 1. When 1 (count is forcibly stopped) is written to the TSTOP bit, the TSTART bit and TCSTF flag are initialized at the same time. The pulse output level is also initialized. The read value is 0.
Note 2. For information on using the TSTART bit and TCSTF flag, see section 23.4.1. Count Operation Start and Stop Control.
Note 3. Only 0 can be written to clear the flag.

TSTART bit (AGT Count Start)

The count operation is started by writing 1 to the TSTART bit and stopped by writing 0. When the TSTART bit is set to 1 (count starts), the TCSTF flag is set to 1 (count in progress) in synchronization with the count source. Also, after 0 is written to the TSTART bit, the TCSTF flag is set to 0 (count stops) in synchronization with the count source. For details, see section 23.4.1. Count Operation Start and Stop Control.

TCSTF flag (AGT Count Status Flag)

The TCSTF flag indicates the AGT count status.

[Setting condition]

- When 1 is written to the TSTART bit (the TCSTF flag is set to 1 in synchronization with the count source).

[Clearing conditions]

- When 0 is written to the TSTART bit (the TCSTF flag is set to 0 in synchronization with the count source)
- When 1 is written to the TSTOP bit.

TSTOP bit (AGT Count Forced Stop)

When 1 is written to the TSTOP bit, the count is forcibly stopped. The read value is 0.

TEDGF flag (Active Edge Judgment Flag)

The TEDGF flag indicates that an active edge was detected.

[Setting condition]

- When the measurement of the active width of the external input pin (AGTWIOn) is complete in pulse width measurement mode
- When the set edge of the external input pin (AGTWIOn) is input in pulse period measurement mode.

[Clearing condition]

- When 0 is written to this flag by software.

TUNDF flag (Underflow Flag)

The TUNDF flag indicates that the counter underflowed.

[Setting condition]

- When the counter underflows.

[Clearing condition]

- When 0 is written to this flag by software.

TCMAF flag (Compare Match A Flag)

The TCMAF flag indicates that compare match A was detected.

[Setting condition]

- When the value in the AGT register matches the value in the AGTCMA register.

[Clearing condition]

- When 0 is written to this flag by software.

TCMBF flag (Compare Match B Flag)

The TCMBF flag indicates that compare match B was detected.

[Setting condition]

- When the value in the AGT register matches the value in the AGTCMB register.

[Clearing condition]

- When 0 is written to this flag by software.

23.2.5 AGTMR1 : AGT Mode Register 1

Base address: AGTW_Bn = 0x400E_8000 + 0x0100 × n (n = 0, 1)

Offset address: 0x0D

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	TCK[2:0]			TEDGPL	TMOD[2:0]		
Value after reset:	0	1	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	TMOD[2:0]	Operating Mode*3 0 0 0: Timer mode 0 0 1: Pulse output mode 0 1 0: Event counter mode 0 1 1: Pulse width measurement mode 1 0 0: Pulse period measurement mode Others: Setting prohibited	R/W

Bit	Symbol	Function	R/W
3	TEDGPL	Edge Polarity*4 0: Single-edge 1: Both-edge	R/W
6:4	TCK[2:0]	Count Source*1 *2 *5 *7 0 0 0: PCLKB 0 0 1: PCLKB/8 0 1 1: PCLKB/2 1 0 0: Divided clock AGTLCLK specified by CKS[2:0] bits in the AGTMR2 register 1 0 1: Underflow event signal from AGTW0*6 1 1 0: Setting prohibited Others: Setting prohibited	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

Note: Write access to the AGTMR1 register initializes the output from the AGTWOn, AGTWIOn, AGTWOAn, and AGTWOBn pins. For details on the output level at initialization, see section 23.2.7. AGTIOC : AGT I/O Control Register.
Note 1. When event counter mode is selected, the external input pin (AGTWIOn) is selected as the count source regardless of the setting of TCK[2:0] bits.
Note 2. Do not switch count sources during count operation. Only switch count sources when both the TSTART bit and TCSTF flag in the AGTCR register are set to 0 (count stops).
Note 3. The operating mode can only be changed when the count is stopped while both the TSTART bit and TCSTF flag in the AGTCR register are set to 0 (count is stopped). Do not change the operating mode during count operation.
Note 4. The TEDGPL bit is enabled only in event counter mode.
Note 5. To run AGT in Software Standby mode, Snooze mode, select AGTLCLK (TCK[2:0] = 100b).
Note 6. AGTW0 cannot use AGTW0 underflow (setting prohibited). AGTW1 uses the AGTW0 underflow.
Note 7. Do not change the TCK[2:0] bits when the CKS[2:0] bits in the AGTMR2 register is not 000b. First, change the CKS[2:0] bits in the AGTMR2 register to 000b. Then change the TCK[2:0] bits and wait for one cycle of the count source.

23.2.6 AGTMR2 : AGT Mode Register 2

Base address: AGTW_Bn = 0x400E_8000 + 0x0100 × n (n = 0, 1)

Offset address: 0x0E

Bit position:	7	6	5	4	3	2	1	0
Bit field:	LPM	—	—	—	—	CKS[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	CKS[2:0]	AGTLCLK Count Source Clock Frequency Division Ratio*1 *2 *3 0 0 0: 1/1 0 0 1: 1/2 0 1 0: 1/4 0 1 1: 1/8 1 0 0: 1/16 1 0 1: 1/32 1 1 0: 1/64 1 1 1: 1/128	R/W
6:3	—	These bits are read as 0. The write value should be 0.	R/W
7	LPM	Low Power Mode 0: Normal mode 1: Low power mode	R/W

Note 1. Do not rewrite the CKS[2:0] bits during count operation. Only rewrite the CKS[2:0] bits when both the TSTART bit and TCSTF flag in the AGTCR register are set to 0 (count stops).
Note 2. When count source is AGTLCLK, the switch of CKS[2:0] bits is valid.
Note 3. Do not switch the TCK[2:0] bits in the AGTMR1 register when CKS[2:0] bits are not 000b. Switch the TCK[2:0] bits in the AGTMR1 register after CKS[2:0] bits are set to 000b, and wait for 1 cycle of the count source.

CKS[2:0] bit (AGTLCLK Count Source Clock Frequency Division Ratio)

CKS[2:0] bits select the Count Source Clock Frequency Division Ratio for AGTLCLK.

LPM bit (Low Power Mode)

The LPM bit sets the low power operation, which impacts access to certain AGT registers. Set this bit to 1 to operate in low power.

When this bit is 1, access to the following registers is prohibited:

- AGT/AGTCMA/AGTCMB/AGTCR.

After this bit is switched from 1 to 0, the first access to the register is constrained as follows:

- When reading from the AGT register, read AGT register twice. Only the second reading of data is valid.
- When writing to the AGT, AGTCMA, AGTCMB, and AGTCR register, allow at least 2 cycles of the count source clock when writing to the register.
- When confirm the value written to the AGT, AGTCMA, AGTCMB, and AGTCR registers.
  - When the count operation is stopped; after writing data, it can be read in the next cycle.
  - When the count operation is operating; after writing data, it can be read 4 cycles after the count source clock.

Figure 23.2 shows the flow of how to write LPM bit

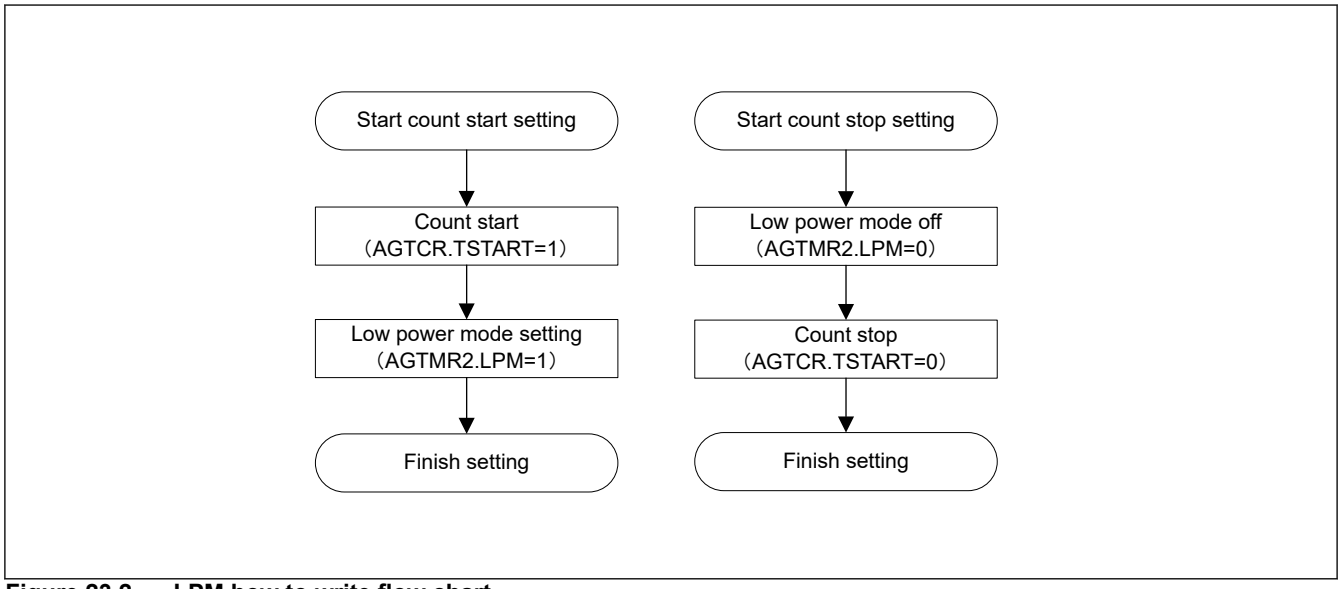


Figure 23.2 LPM how to write flow chart

23.2.7 AGTIOC : AGT I/O Control Register

Base address: AGTW_Bn = 0x400E_8000 + 0x0100 × n (n = 0, 1)

Offset address: 0x10

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TIOGT[1:0]		TIPF[1:0]		—	TOE	—	TEDG SEL
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TEDGSEL	I/O Polarity Switch Function varies depending on the operating mode (see Table 23.3 and Table 23.4).	R/W
1	—	This bit is read as 0. The write value should be 0.	R/W
2	TOE	AGTWOn pin Output Enable 0: AGTWOn pin output disabled 1: AGTWOn pin output enabled	R/W

Bit	Symbol	Function	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
5:4	TIPF[1:0]	Input Filter*3 These bits specifies the sampling frequency of the filter for the AGTWIOn input. If the input to the AGTWIOn pin is sampled and the value matches three successive times, that value is taken as the input value. 0 0: No filter 0 1: Filter sampled at PCLKB 1 0: Filter sampled at PCLKB/8 1 1: Filter sampled at PCLKB/32	R/W
7:6	TIOGT[1:0]	Count Control*1 *2 0 0: Event is always counted 0 1: Event is counted during polarity period specified for AGTWEEn pin Others: Setting prohibited	R/W

Note 1. When AGTWEEn pin is used, the polarity to count an event can be selected with the EEPS bit in the AGTISR register.
Note 2. TIOGT[1:0] bits are enabled only in event counter mode.
Note 3. When event counter mode operation is performed during Software Standby mode, the digital filter function cannot be used.

TEDGSEL bit (I/O Polarity Switch)

The TEDGSEL bit switches the AGTWOn pin output polarity and the AGTWIOn pin input/output edge and polarity.

In pulse output mode, it only controls polarity of the AGTWOn pin output and AGTWIOn pin output. AGTWOn pin output and AGTWIOn pin output are initialized when the AGTMR1 register is written or the TSTOP bit in the AGTCR register is written with 1.

TOE bit (AGTWOn pin Output Enable)

The TOE bit selects whether the AGTOn pin output is disabled or enabled.

TIPF[1:0] bits (Input Filter)

The TIPF[1:0] bits specify the sampling frequency of the AGTIOn pin input filter. When the input to the AGTIOn pin is sampled and the values match three times in succession, the value is regarded as the input value.

TIOGT[1:0] bits (Count Control)

The TIOGT[1:0] bits control the event count.

Table 23.3 AGTWIOn pin I/O edge and polarity switching

Operating mode	Function
Timer mode	Not used
Pulse output mode	0: Output is started at high (initialization level: high) i.e. inverted output 1: Output is started at low (initialization level: low). i.e. normal output
Event counter mode	0: Count on rising edge 1: Count on falling edge.
Pulse width measurement mode	0: Low-level width is measured 1: High-level width is measured.
Pulse period measurement mode	0: Measure from one rising edge to the next rising edge 1: Measure from one falling edge to the next falling edge.

Table 23.4 AGTWOn pin output polarity switching

Operating mode	Function
All modes	0: Output is started at low (initial level: low): Normal output 1: Output is started at high (initial level: high): Inverted output

23.2.8 AGTISR : AGT Event Pin Select Register

Base address: AGTW_Bn = 0x400E_8000 + 0x0100 × n (n = 0, 1)

Offset address: 0x11

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

Bit	Symbol	Function	R/W
1:0	—	These bits are read as 0. The write value should be 0.	R/W
2	EEPS	AGTWEEn Polarity Selection 0: An event is counted during the low-level period 1: An event is counted during the high-level period	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

EEPS bit (AGTWEEn Polarity Selection)

The EEPS bit selects the polarity of events to be counted.

23.2.9 AGTCMSR : AGT Compare Match Function Select Register

Base address: AGTW_Bn = 0x400E_8000 + 0x0100 × n (n = 0, 1)

Offset address: 0x12

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	TOPOLB	TOEB	TCMEB	—	TOPOLA	TOEA	TCMEA
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TCMEA	AGT Compare Match A Register Enable*1 *2 *3 0: AGT Compare match A register disabled 1: AGT Compare match A register enabled	R/W
1	TOEA	AGTWOAn Pin Output Enable*1 *2 0: AGTWOAn pin output disabled 1: AGTWOAn pin output enabled	R/W
2	TOPOLA	AGTWOAn Pin Polarity Select*1 *2 0: AGTWOAn pin output is started on low. i.e. normal output 1: AGTWOAn pin output is started on high. i.e. inverted output	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	TCMEB	AGT Compare Match B Register Enable*1 *2 *3 0: Compare match B register disabled 1: Compare match B register enabled	R/W
5	TOEB	AGTWOBn Pin Output Enable*1 *2 0: AGTWOBn pin output disabled 1: AGTWOBn pin output enabled	R/W
6	TOPOLB	AGTWOBn Pin Polarity Select*1 *2 0: AGTWOBn pin output is started on low. i.e. normal output 1: AGTWOBn pin output is started on high. i.e. inverted output	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

Note 1. Do not rewrite the AGTCMSR register during a count operation. Only rewrite the AGTCMSR register when both the TSTART bit and TCSTF flag in the AGTCR register are set to 0 (count stops).

Note 2. Do not set 1 when in pulse width measurement mode or pulse period measurement mode.

Note 3. When 1 is written to the TSTOP bit in the AGTCR register, TCMEA and TCMEB is forcibly stopped and set to 0.

23.2.10 AGTIOSEL : AGT Pin Select Register

Base address: AGTW_Bn = 0x400E_8000 + 0x0100 × n (n = 0, 1)

Offset address: 0x13

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

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
4	TIES	AGTWIOn Pin Input Enable 0: External event input is disabled during Software Standby mode 1: External event input is enabled during Software Standby mode	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

The AGTIOSEL register sets the AGTWIOn pin when using the AGTWIOn pin in Software Standby mode.

TIES bit (AGTWIOn Pin Input Enable)

The TIES bit enables or disables an external event input.

23.3 Operation

23.3.1 Reload Register and Counter Rewrite Operation

Regardless of the operating mode, the timing of the rewrite operation to the reload register and the counter differs depending on the value of the TSTART bit in the AGTCR register and of the TCMEA or TCMEB bit in the AGTCMSR register. When the TSTART bit is 0 (count stops), the count value is directly written to the reload register and the counter. When the TSTART bit is 1 (count starts) and the TCMEA bit and TCMEB bit are 0 (AGT compare match A/B register are invalid), the value is written to the reload register in synchronization with the count source, and then to the counter in synchronization with the next count source. When the TSTART bit is 1 (count starts) and the TCMEA bit or the TCMEB bit is 1 (AGT compare match A register or compare match B register is valid), the value is written to the reload register in synchronization with the count source, and then to the counter in synchronization with the underflow of the counter.

Figure 23.3 and Figure 23.4 show the timing of rewrite operation with TSTART bit value and TCMEA/TCMEB bit value.

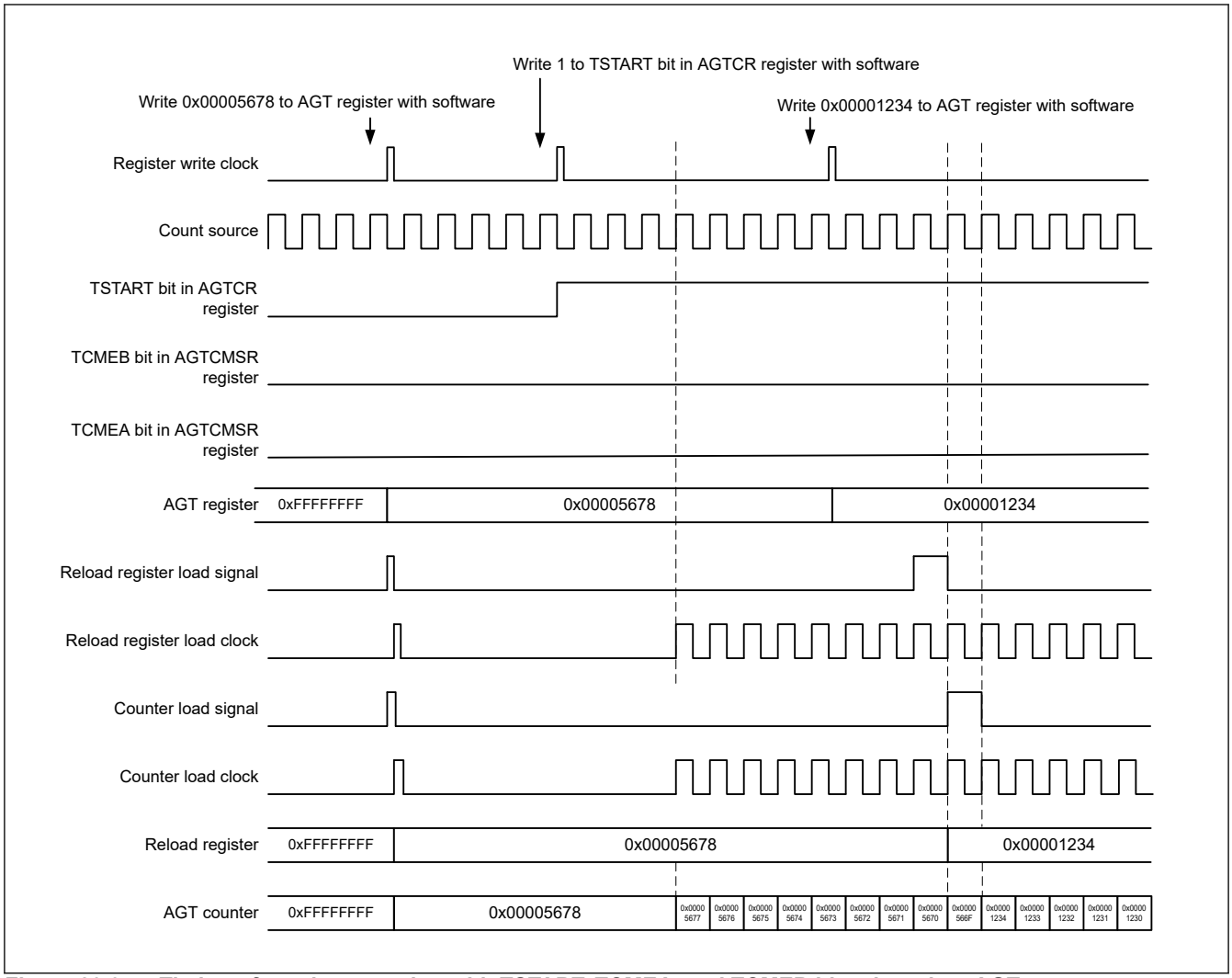


Figure 23.3 Timing of rewrite operation with TSTART, TCMEA, and TCMEB bit value when AGT compare match A register and AGT compare match B register is invalid

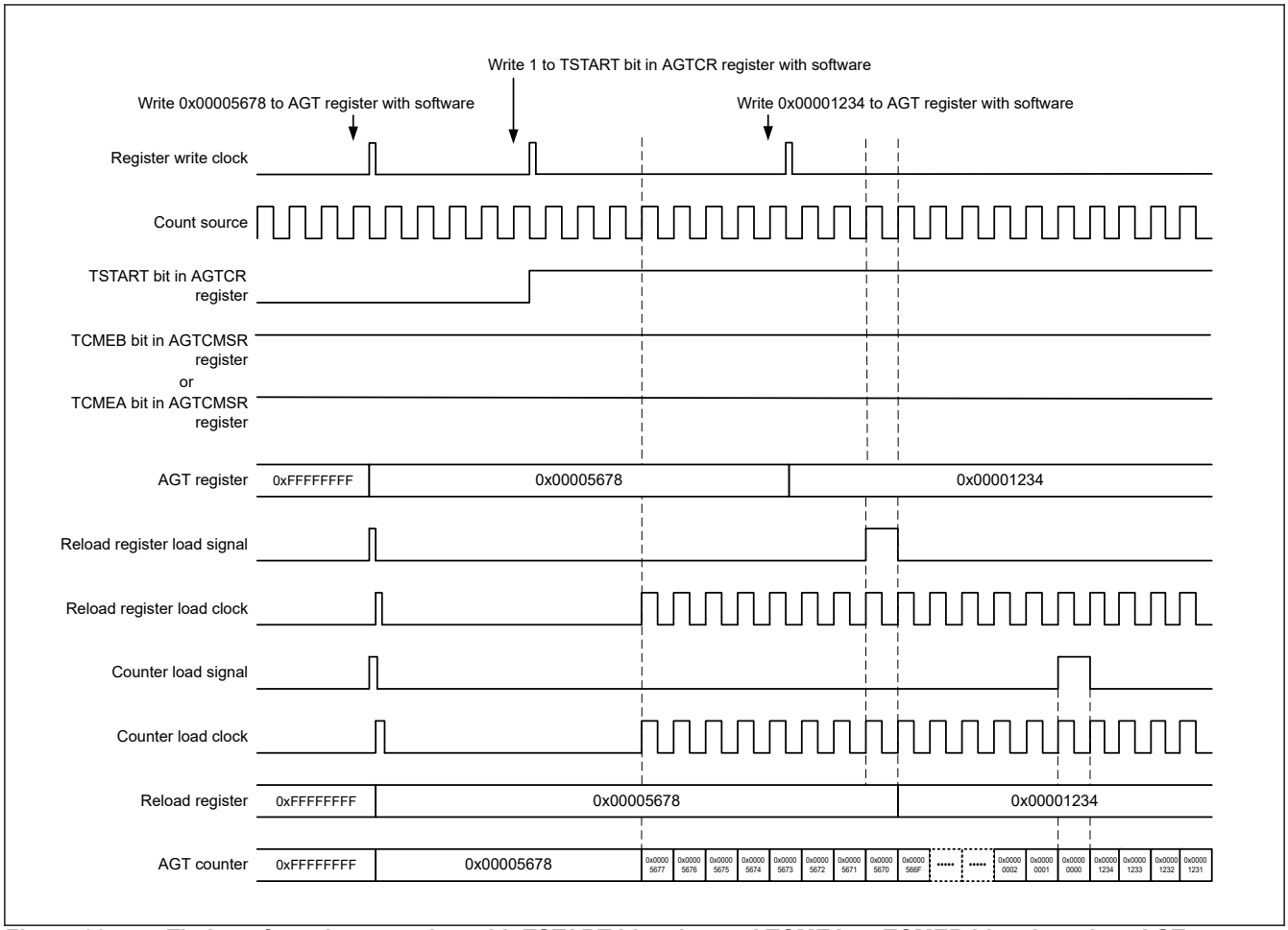


Figure 23.4 Timing of rewrite operation with TSTART bit value and TCMEA or TCMEB bit value when AGT compare match A register or AGT compare match B register is valid

23.3.2 Reload Register and AGT Compare Match A/B Register Rewrite Operation

Regardless of the operating mode, the timing of the rewrite operation to the reload register and AGT compare register A/B depends on the value of the TSTART bit in the AGTCR register. When the TSTART bit is 0 (count stops), the count value is directly written to the reload register and AGT compare register A/B. When the TSTART bit is 1 (count starts), the value is written to the reload register in synchronization with the count source, and then to the compare register in synchronization with the underflow of the counter.

Figure 23.5 shows the timing of rewrite operation with TSTART bit value for compare register A. AGT Compare register B is of the same timing as AGT compare register A.

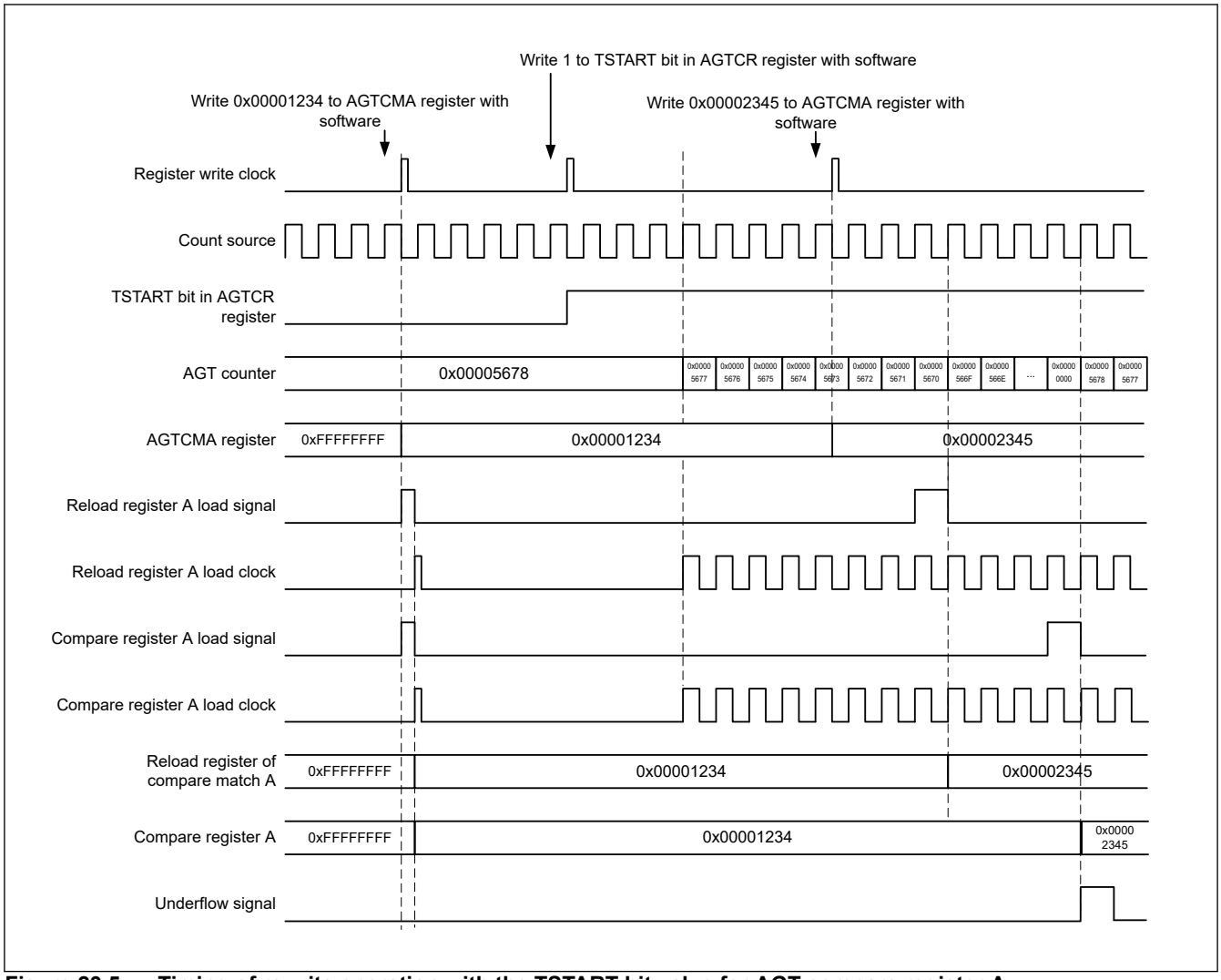


Figure 23.5 Timing of rewrite operation with the TSTART bit value for AGT compare register A

23.3.3 Timer Mode

In this mode, the AGT counter is decremented by the count source selected with the TCK[2:0] bits in the AGTMR1 register. In timer mode, the count value is decremented by 1 on each rising edge of the count source. When the count value reaches 0x00000000 and the next count source is input, an underflow occurs and an interrupt request is generated.

Figure 23.6 shows the operation example in timer mode.

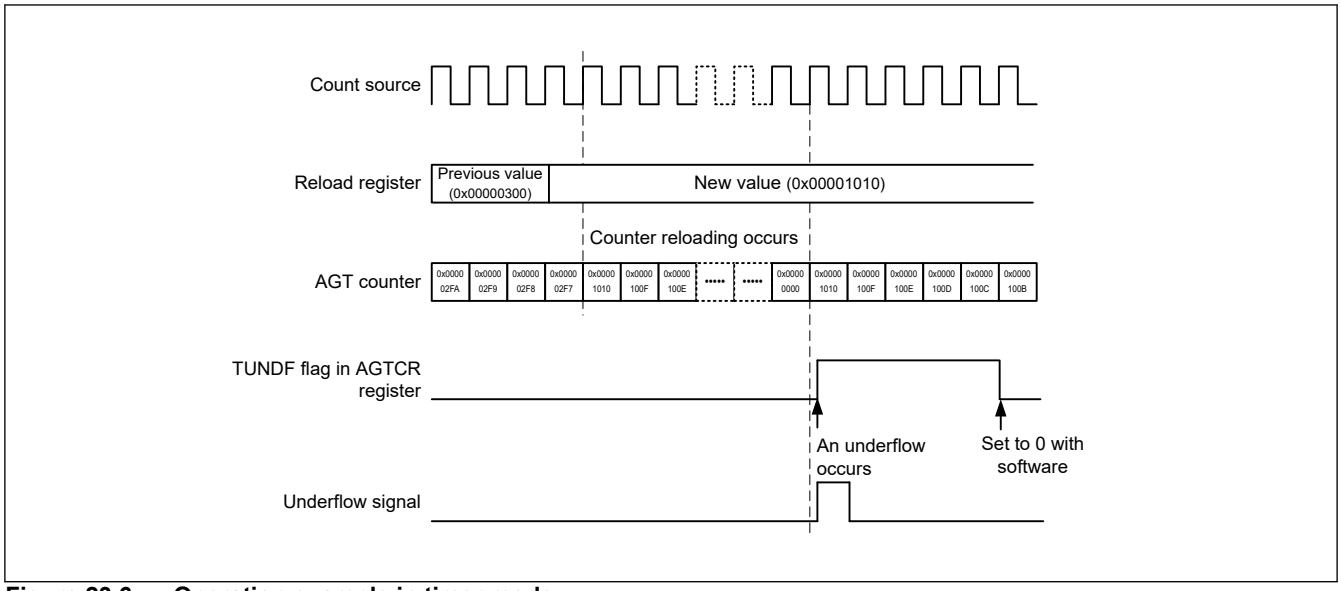


Figure 23.6 Operation example in timer mode

23.3.4 Pulse Output Mode

In pulse output mode, the counter is decremented by the count source selected with the TCK[2:0] bits in the AGTMR1 register, and the output level of the AGTWIOn and AGTOn pins inverted each time an underflow occurs.

In pulse output mode, the count value is decremented by 1 on each rising edge of the count source. When the count value reaches 0x00000000 and the next count source is input, an underflow occurs and an interrupt request is generated. In addition, a pulse can be output from the AGTWIOn and AGTOn pins. The output level is inverted each time an underflow occurs. The pulse output from the AGTWOn pin can be stopped with the TOE bit in the AGTIOC register. The output level can be selected with the TEDGSEL bit in the AGTIOC register.

Figure 23.7 shows the operation example in pulse output mode.

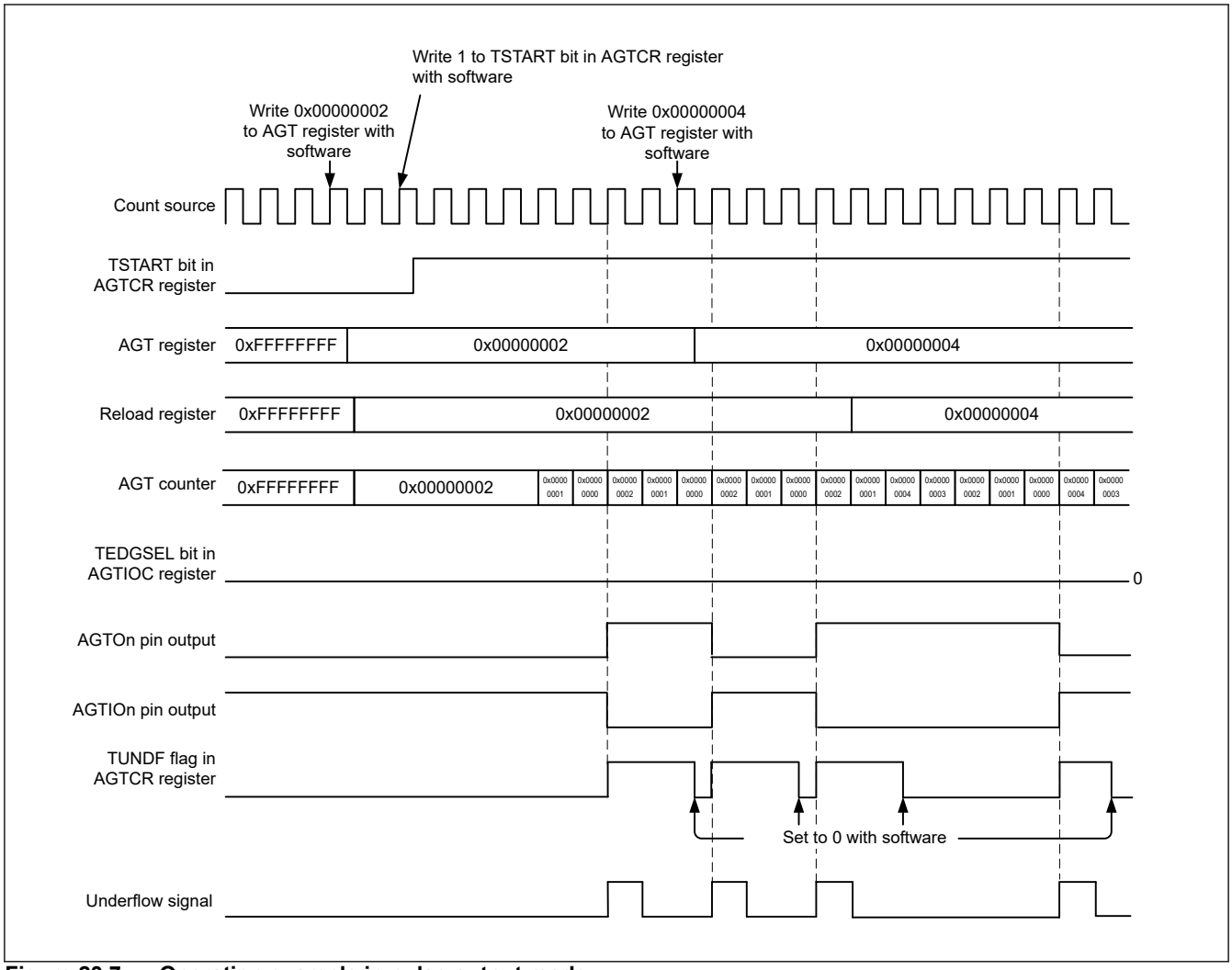


Figure 23.7 Operation example in pulse output mode

23.3.5 Event Counter Mode

In event counter mode, the counter is decremented by an external event signal (count source) input to the AGTWIOn pin. Various periods for counting events can be set with the TIOGT[1:0] bits in the AGTIOC register and AGTISR registers. In addition, the filter function for the AGTWIOn pin input can be specified with bits TIPF[1:0] in the AGTIOC register. The output from the AGTWOn pin can be toggled even in event counter mode.

Figure 23.8 shows the operation example in event counter mode.

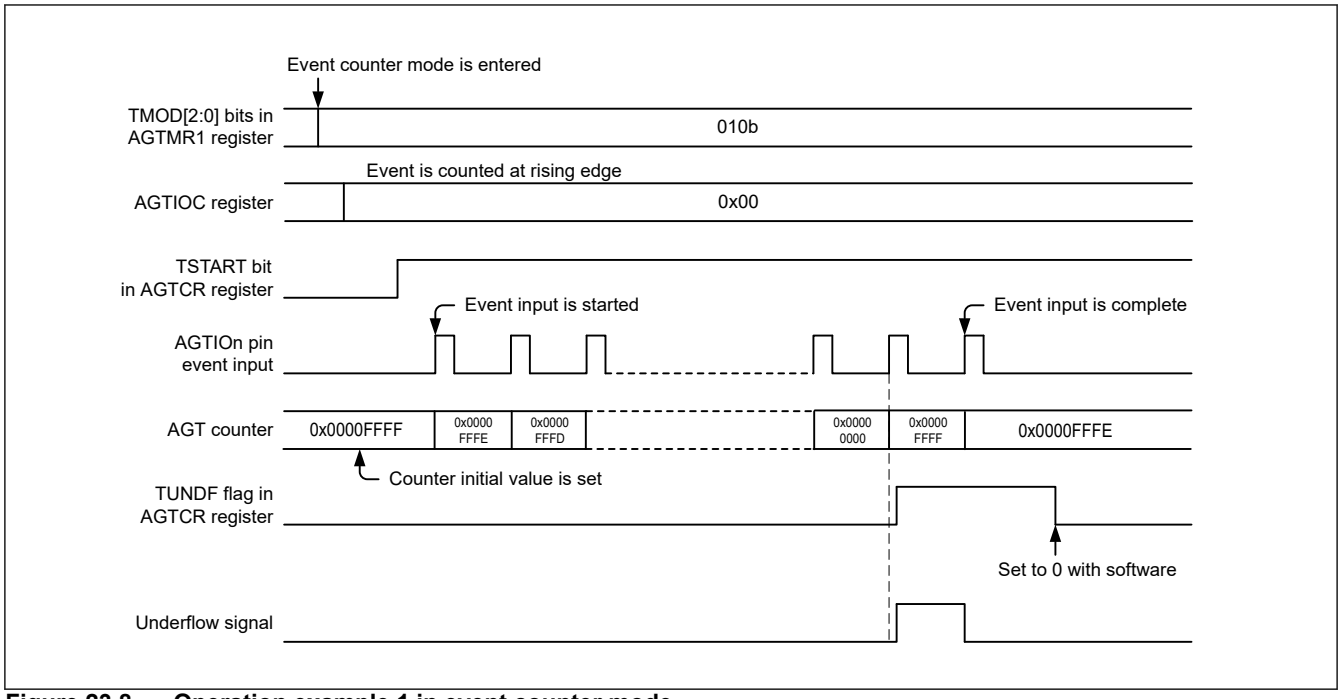
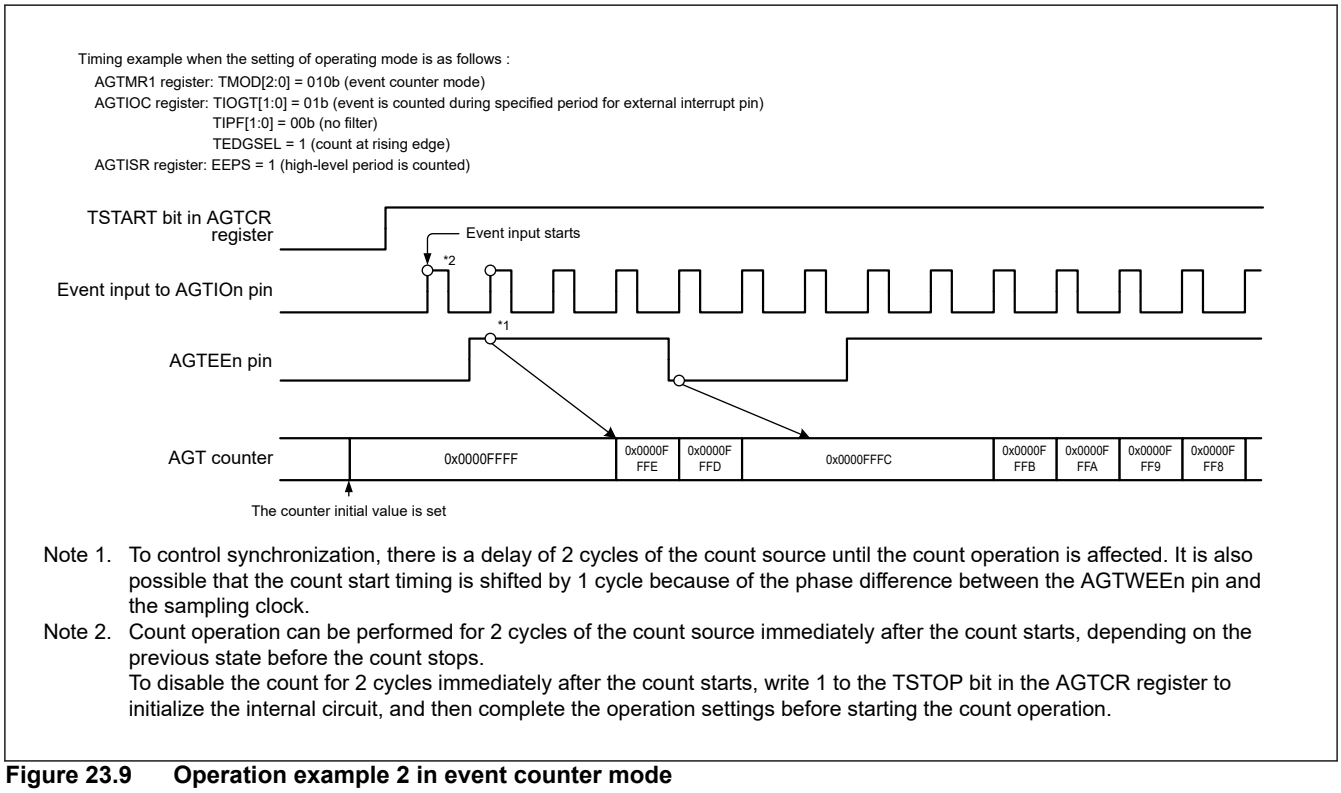


Figure 23.8 Operation example 1 in event counter mode

Figure 23.9 shows an operation example for counting during the specified period in event counter mode (TIOGT[1:0] bits in the AGTIOC register are set to 01b).



Note 1. To control synchronization, there is a delay of 2 cycles of the count source until the count operation is affected. It is also possible that the count start timing is shifted by 1 cycle because of the phase difference between the AGTWEEn pin and the sampling clock.

Note 2. Count operation can be performed for 2 cycles of the count source immediately after the count starts, depending on the previous state before the count stops.
To disable the count for 2 cycles immediately after the count starts, write 1 to the TSTOP bit in the AGTCR register to initialize the internal circuit, and then complete the operation settings before starting the count operation.

Figure 23.9 Operation example 2 in event counter mode

23.3.6 Pulse Width Measurement Mode

In pulse width measurement mode, the pulse width of an external signal input to the AGTWIOn pin is measured. When the level specified by the TEDGSEL bit in the AGTIOC register is input to the AGTWIOn pin, the counter is decremented by the count source selected with the TCK[2:0] bits in the AGTMR1 register. When the specified level on the AGTWIOn pin ends, the counter is stopped, the TEDGF flag in the AGTCR register is set to 1 (active edge received), and an interrupt request is generated. The measurement of pulse width data is performed by reading the count value while the counter is

stopped. Also, when the counter underflows during measurement, the TUNDF flag in the AGTCR register is set to 1 and an interrupt request is generated.

Figure 23.10 shows the operation example in pulse width measurement mode.

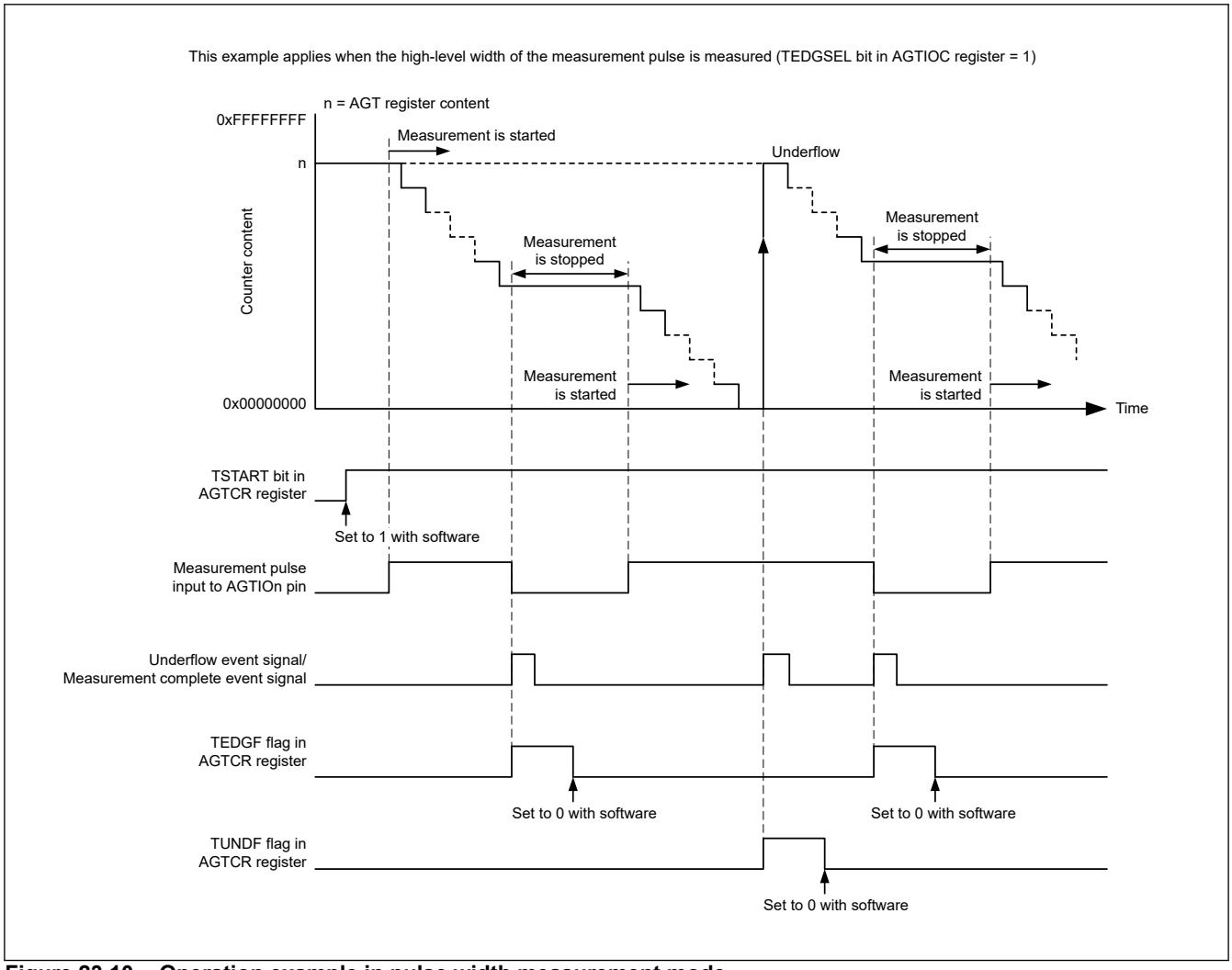


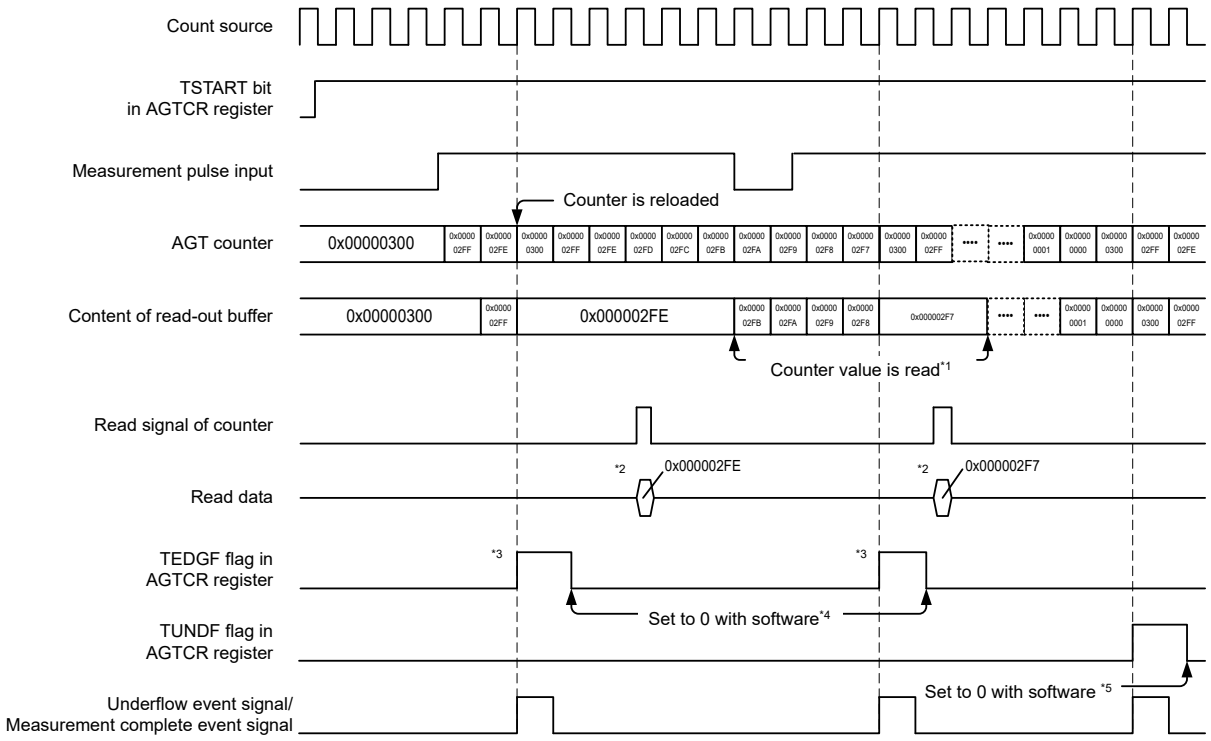
Figure 23.10 Operation example in pulse width measurement mode

23.3.7 Pulse Period Measurement Mode

In pulse period measurement mode, the pulse period of an external signal input to the AGTWIOn pin is measured. The counter is decremented by the count source selected with TCK[2:0] bits in the AGTMR1 register. When a pulse with the period specified by the TEDGSEL bit in the AGTIOC register is input to the AGTWIOn pin, the count value is transferred to the read-out buffer on the rising edge of the count source. The value in the reload register is loaded to the counter at the next rising edge. Simultaneously, the TEDGF flag in the AGTCR register is set to 1 (active edge received) and an interrupt request is generated. The read-out buffer (AGT register) is read at this time and the difference from the reload value (see section 23.4.6. How to Calculate Event Number, Pulse Width, and Pulse Period) is the period data of the input pulse. The period data is retained until the read-out buffer is read. When the counter underflows, the TUNDF flag in the AGTCR register is set to 1 (underflow) and an interrupt request is generated.

Figure 23.11 shows the operation example in pulse period measurement mode.

Only input pulses with a period longer than twice the period of the count source are measured. Also, the low-level and high-level widths must both be longer than the period of the count source. If a pulse period shorter than these conditions is input, the input might be ignored.



This example applies when the initial value of the AGT register is set to A, the TEDGSEL bit in the AGTIOC register is set to 0, and the period from one rising edge to the next edge of the measurement pulse is measured.

Note 1. Reading from the AGT register must be performed during the period from when the TEDGF flag is set to 1 (active edge received) until the next active edge is input. The content of the read-out buffer is retained until the AGT register is read. If it is not read before the active edge is input, the measurement result of the previous period is retained.
Note 2. When the AGT register is read in pulse period measurement mode, the content of the read-out buffer is read.
Note 3. When the active edge of the measurement pulse is input and then the set edge of an external pulse is input, the TEDGF flag in the AGTCR register is set to 1 (active edge received).
Note 4. To set to 0 with software, write 0 to the TEDGF flag in the AGTCR register with an 8-bit memory manipulation instruction.
Note 5. To set to 0 with software, write 0 to the TUNDF flag in the AGTCR register with an 8-bit memory manipulation instruction.

Figure 23.11 Operation example in pulse period measurement mode

23.3.8 Compare Match function

The compare match function detects matches (compare match) between the content of the AGTCMA or AGTCMB register and the content of the AGT register. This function is enabled when the TCMEA or TCMEB bit in the AGTCMSR register is 1 (compare match A register or compare match B register is valid). The counter is decremented by the count source selected with the TCK[2:0] bits in the AGTMR1 register, and when the values of AGT and AGTCMA or AGTCMB match, the TCMAF/TCMBF flag in the AGTCR register is set to 1 (match), and an interrupt request is generated.

When the compare match function is enabled, the timing of the rewrite operation to the reload register and the counter differs. See section 23.3.1. Reload Register and Counter Rewrite Operation for details. In addition, the output level of the AGTWOAn, AGTWOBn pins is inverted by the match and by the underflow. The output level can be selected with the TOPOLA or TOPOLB bit in the AGTCMSR register.

Figure 23.12 shows the operation example in compare match function.

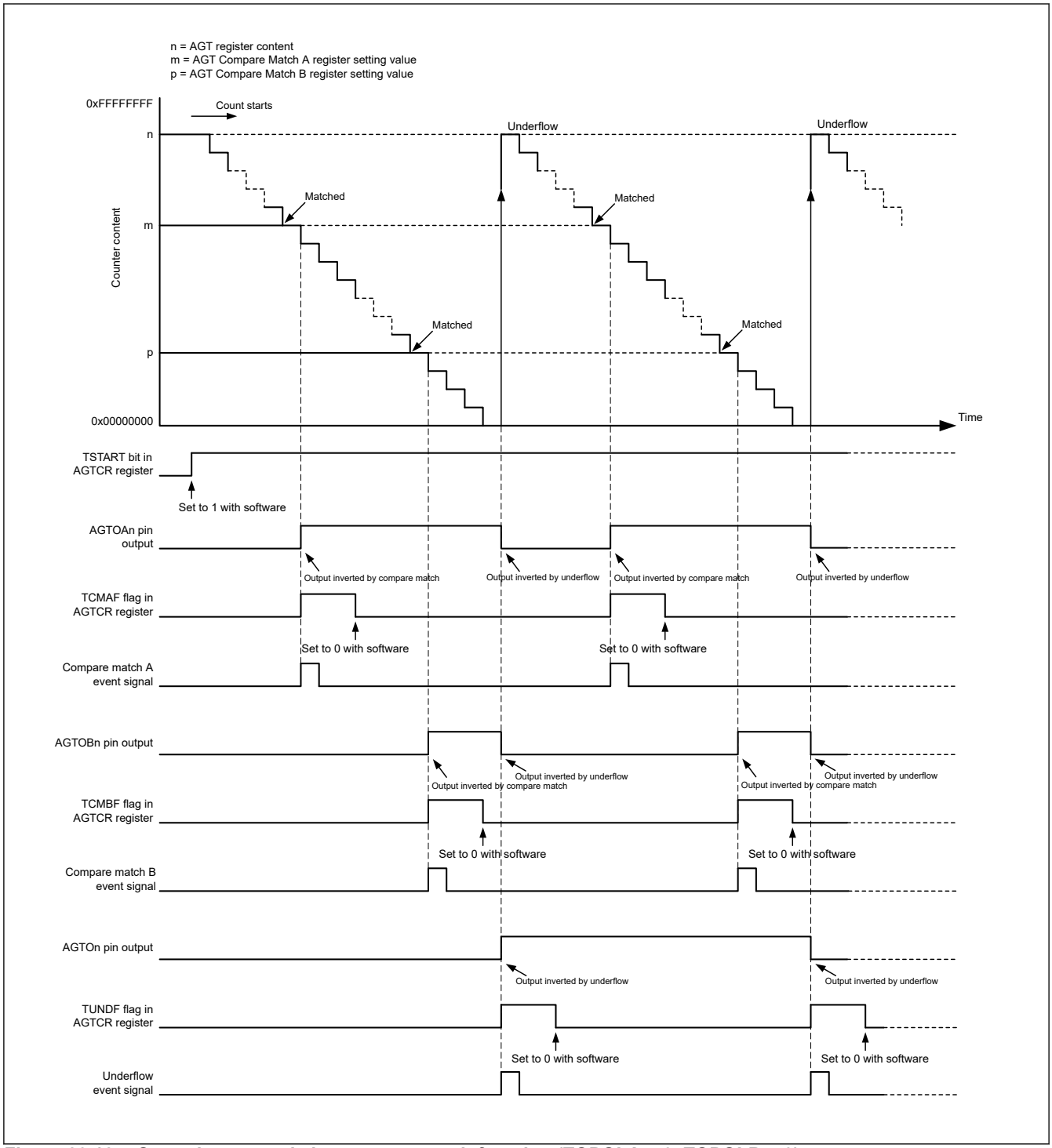


Figure 23.12 Operation example in compare match function (TOPOLA = 0, TOPOLB = 0)

23.3.9 Output Settings for Each Mode

Table 23.5 to Table 23.8 list the states of pins AGTWOn, AGTWIOn, AGTWOAn, and AGTWOBn pins in each mode.

Table 23.5 AGTWOn pin setting

Operating mode	AGTIOC register		AGTWOn pin output
	TOE bit	TEDGSEL bit	
All modes	1	1	Inverted output
		0	Normal output
	0	0 or 1	Output disabled

Table 23.6 AGTWIOn pin setting

Operating mode	AGTIOC register	AGTWIOn pin I/O
	TEDGSEL bit	
Timer mode	0 or 1	Input (not used)
Pulse output mode	1	Normal output
	0	Inverted output
Event counter mode	0 or 1	Input
Pulse width measurement mode		
Pulse period measurement mode		

Table 23.7 AGTWOAn pin setting

Operating mode	AGTCMSR register		AGTWOAn pin output
	TOEA bit	TOPOLA bit	
Timer mode	1	1	Inverted output
		0	Normal output
	0	0 or 1	Output disabled (not used)
Pulse output mode	1	1	Inverted output
		0	Normal output
	0	0 or 1	Output disabled (not used)
Event counter mode	1	1	Inverted output
		0	Normal output
	0	0 or 1	Output disabled (not used)
Pulse width measurement mode	0	0	Prohibited
Pulse period measurement mode			

Table 23.8 AGTWOBn pin setting (1 of 2)

Operating mode	AGTCMSR register		AGTWOBn pin output
	TOEB bit	TOPOLB bit	
Timer mode	1	1	Inverted output
		0	Normal output
	0	0 or 1	Output disabled (not used)
Pulse output mode	1	1	Inverted output
		0	Normal output
	0	0 or 1	Output disabled (not used)
Event counter mode	1	1	Inverted output
		0	Normal output
	0	0 or 1	Output disabled (not used)

Table 23.8 AGTWOBn pin setting (2 of 2)

Operating mode	AGTCMSR register		AGTWOBn pin output
	TOEB bit	TOPOLB bit	
Pulse width measurement mode	0	0	Prohibited
Pulse period measurement mode			

23.3.10 Standby Mode

The AGT can operate in Software Standby mode. Set it to Software Standby mode with count operation start (TSTART = 1, TCSTF = 1).

Table 23.9 and Table 23.10 show the setting that can be used in Software Standby mode.

Table 23.9 Usable settings in Software Standby mode (AGTW0)

Operating mode	AGTMR1.TCK[2:0]	Operating clock	Resurgence factor of CPU
Timer mode	100b	AGTLCLK	—
Pulse output mode	100b	AGTLCLK	—
Event counter mode	—	AGTWIOn (n = 0)*1	—
Pulse width measurement mode	100b	AGTLCLK	—
Pulse period measurement mode	100b	AGTLCLK	—

Note: —: invalid
Note 1. When using the AGTWIOn pin for external event input in Software Standby mode, set AGTIOSEL.TIES = 1.

Table 23.10 Usable settings in Software Standby mode (AGTW1)

Operating mode	AGTMR1.TCK[2:0]	Operating clock	Resurgence factor of CPU
Timer mode	100b or 101b*1	AGTLCLK or AGTW0 underflow	• Underflow • Compare match A/B
Pulse output mode	100b or 101b*1	AGTLCLK or AGTW0 underflow	• Underflow • Compare match A/B
Event counter mode	—	AGTWIOn (n = 1)*2	• Underflow • Compare match A/B
Pulse width measurement mode	100b or 101b*1	AGTLCLK or AGTW0 underflow	• Underflow • Active edge
Pulse period measurement mode	100b or 101b*1	AGTLCLK or AGTW0 underflow	• Underflow • Active edge

Note: —: invalid
Note: Release of Software Standby mode is only AGT1.
Note: Compare match A/B is resurgence factor of CPU from Software Standby mode.
Note 1. Only when AGTW0 operates in Table 23.9
Note 2. When using the AGTWIOn pin for external event input in Software Standby mode, set AGTIOSEL.TIES = 1.

23.3.11 Interrupt Sources

The AGTWn has three interrupt sources as listed in Table 23.11.

Table 23.11 AGTW interrupt sources

Name	Interrupt source	DMAC/DTC activation
AGTn_AGTI	• When the counter underflows • When measurement of the active width of the external input pin (AGTWIOn) is complete in pulse width measurement mode • When the set edge of the external input pin (AGTWIOn) is input in pulse period measurement mode.	Possible
AGTn_AGTCMAI	• When the values of AGT register and AGTCMA register match	Possible
AGTn_AGTCMBI	• When the values of AGT register and AGTCMB register match	Possible

Note: Channel number (n = 0, 1)

23.3.12 Event Signal Output to ELC

The AGTWn (n = 0, 1) uses the Event Link Controller (ELC) to perform a link operation to a specified module using the interrupt request signal as the event signal. The AGTWn (n = 0, 1) outputs compare match A, compare match B, and underflow/measurement complete signals as event signals. For details, see section 17, Event Link Controller (ELC).

23.4 Usage Notes

23.4.1 Count Operation Start and Stop Control

- When the operating mode (see Table 23.1) is set to other than the event counter mode, or the count source is set to other than AGTWn underflow event signal (TCK[2:0] = 101b):
  - After 1 (count starts) is written to the TSTART bit in the AGTCR register while the count is stopped, the TCSTF flag in the AGTCR register remains 0 (count stops) for 3 cycles of the count source. Do not access the registers associated with AGTW other than the TCSTF flag until this flag is set to 1 (count in progress).
  - After 0 (count stops) is written to the TSTART bit during a count operation, the TCSTF flag remains 1 for 3 cycles of the count source. When the TCSTF flag is set to 0, the count is stopped. Do not access the registers associated with AGTW other than the TCSTF flag until this flag is set to 0.
- When the operating mode (see Table 23.1) is set to event counter mode, or the count source is set to AGTW1 underflow event signal (TCK[2:0] = 101b):
  - After 1 (count starts) is written to the TSTART bit in the AGTCR register while the count is stopped, the TCSTF flag in the AGTCR register remains 0 (count stops) for 2 PCLKB cycles. Do not access the registers associated with AGTW other than the TCSTF flag until this flag is set to 1 (count in progress).
  - After 0 (count stops) is written to the TSTART bit during a count operation, the TCSTF flag remains 1 for 2 PCLKB cycles. When the TCSTF flag is set to 0, the count is stopped. Do not access the registers associated with AGTW other than the TCSTF flag until this flag is set to 0.

23.4.2 Access to Counter Register

When the TSTART bit and TCSTF flag in the AGTCR register are both 1 (count starts), allow at least 3 cycles of the count source clock between writes when writing to the AGT register successively.

23.4.3 When Changing Mode

The registers associated with AGT operating mode (AGTMR1, AGTMR2, AGTIOC, AGTISR, and AGTCMSR) can be changed only when the count is stopped with both the TSTART bit and TCSTF flag set to 0 (count stops). Do not change these registers during count operation.

When the registers associated with AGT operating mode are changed, the values of TEDGF, TUNDF, TCMAF, and TCMBF flags are undefined. Before starting the count, write 0 to the following flags:

- TEDGF (no active edge received)
- TUNDF (no underflow)
- TCMAF (no match)
- TCMBF (no match).

23.4.4 Output pin setting

When using the AGTWOn, AGTWIOn, AGTWOAn, or AGTWOBn as an output pin, set up the Operation and determine the initial output values. Then set an output mode in the port register.

When using the AGTWIOn as an input pin in pulse width measurement mode or pulse period measurement mode, set up the Operation and start count operation. Then start to enter external events from the AGTWIOn pin. Invalidate the first measurement and validate the second and later completed measurements.

23.4.5 Digital Filter

When using the digital filter, do not start the timer operation for 5 cycles of the digital filter clock after setting TIPF[1:0] bits and when the TEDGSEL bit in the AGTIOC register changes.

23.4.6 How to Calculate Event Number, Pulse Width, and Pulse Period

- In event counter mode, event number is expressed mathematically as follows:
  Event number = initial value of counter [AGT register] - counter value of active event end
- In pulse width measurement mode, pulse width is expressed mathematically as follows:
  Pulse width = counter value of stopping measurement - counter value of next stopping measurement
- In pulse period measurement mode, input pulse period is expressed mathematically as follows:
  Period of input pulse = (initial value of counter [AGT register] - reading value of the read-out buffer) + 1.

23.4.7 When Count is Forcibly Stopped by TSTOP Bit

After the counter is forcibly stopped by the TSTOP bit in the AGTCR register, do not access the following I/O registers for 1 cycle of the count source:

- AGT
- AGTCMA
- AGTCMB
- AGTCR
- AGTMR1
- AGTMR2.

23.4.8 When Selecting AGTW0 Underflow as the Count Source

Operate according to the following procedures described in this section when selecting the underflow event signal as the count source.

(1) Procedure for starting operation

1. Set AGTW.
2. Start the count operation of AGTW1.
3. Start the count operation of AGTW0.

(2) Procedure for stopping operation

1. Stop the count operation of AGTW0.
2. Stop the count operation of AGTW1.
3. Stop the count source clock of AGTW1 (write 000b in the AGTMR1.TCK[2:0] bits).

23.4.9 Module-stop function

AGTW operation can be disabled or enabled using Module Stop Control Register D (MSTPCRD). The AGTW module is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 10, Low Power Modes

23.4.10 When Switching Source Clock

When switching a clock source by changing SCKSCR.CKSEL[2:0], the clock output from the selector stops for 4 cycles of the switched clock. Therefore, when using the AGTWIOn, AGTWEEn, or both input as external event input, the clock source should not be switched. If switching the clock source while using the external event input, extend the input pulse width by 4 clock cycles of the switched source clock cycles.

24. Watchdog Timer (WDT)

24.1 Overview

The Watchdog Timer (WDT) is a 14-bit down counter that can be used to reset the MCU when the counter underflows because the system has run out of control and is unable to refresh the WDT. In addition, the WDT can be used to generate a non-maskable interrupt or an underflow interrupt.

Table 24.1 lists the WDT specifications and Figure 24.1 shows a block diagram.

Table 24.1 WDT specifications

Parameter	Specifications
Count source*1	Peripheral clock (PCLKB)
Clock division ratio	Division by 4, 64, 128, 512, 2048, or 8192
Counter operation	Counting down using a 14-bit down-counter
Condition for starting the counter	• Auto start mode: Counting automatically starts after a reset or after an underflow or refresh error occurs • Register start mode: Counting is started with a refresh by writing to the WDTRR register • Only secure developer can select Auto-start mode or Register-start mode
Conditions for stopping the counter	• Reset (the down-counter and other registers return to their initial values) • A counter underflows or a refresh error is generated
Window function	Window start and end positions can be specified (refresh-permitted and refresh-prohibited periods)
Watchdog timer reset sources	• Down-counter underflows • Refreshing outside the refresh-permitted period (refresh error)
Non-maskable interrupt/interrupt sources	• Down-counter underflows • Refreshing outside the refresh-permitted period (refresh error)
Reading of the counter value	The down-counter value can be read by the WDTSR register
Event link function (output)	• Down-counter underflow event output • Refresh error event output
Output signal (internal signal)	• Reset output • Interrupt request output • Sleep-mode count stop control output
TrustZone Filter	Security attribution can be set

Note 1. Satisfy the frequency of the peripheral module clock (PCLKB) ≥ 4 × (the frequency of the count clock source after division).

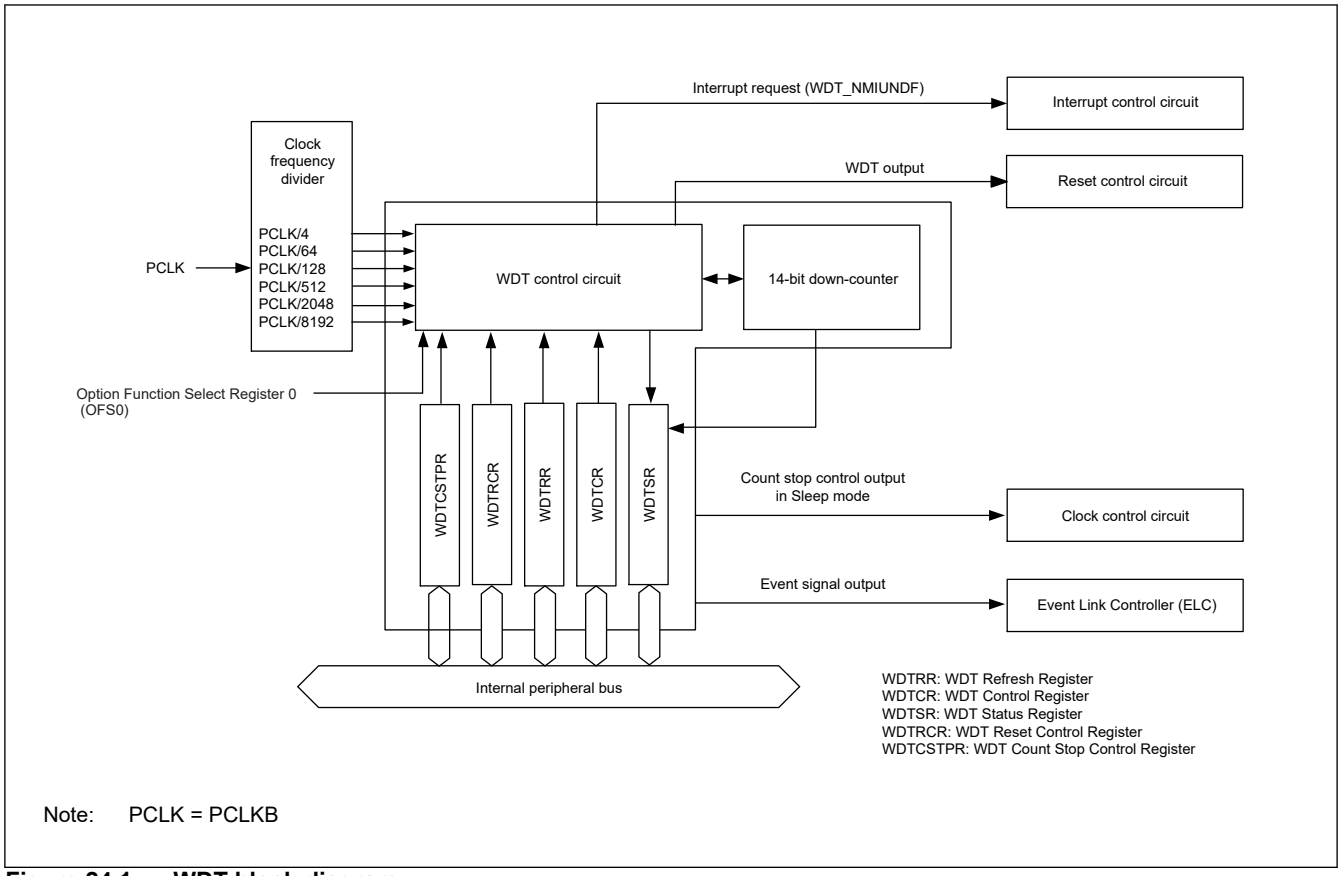


Figure 24.1 WDT block diagram

24.2 Register Descriptions

24.2.1 WDTRR : WDT Refresh Register

Base address: WDT = 0x4008_3400

Offset address: 0x00

Bit position:	7							0
Bit field:								
Value after reset:	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
7:0	n/a	The down-counter is refreshed by writing 0x00 and then writing 0xFF to this register.	R/W

The WDTRR register refreshes the down-counter of the WDT.

The down-counter of the WDT is refreshed by writing 0x00 and then writing 0xFF to WDTRR register (refresh operation) within the refresh-permitted period.

After the down-counter is refreshed, it starts counting down from the value selected by setting the WDT Timeout Period Select bits (OFS0.WDTTOPS[1:0]) in the Option Function Select Register 0 in auto start mode. In register start mode, counting down starts from the value selected by setting the Timeout Period Select bits (WDTCR.TOPS[1:0]) in the WDT Control Register.

When 0x00 is written, the read value is 0x00. When a value other than 0x00 is written, the read value is 0xFF. For details of the refresh operation, see section 24.3.3. Refresh Operation.

24.2.2 WDTCR : WDT Control Register

Base address: WDT = 0x4008_3400

Offset address: 0x02

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

Bit	Symbol	Function	R/W
1:0	TOPS[1:0]	Timeout Period Select 0 0: 1024 cycles (0x03FF) 0 1: 4096 cycles (0x0FFF) 1 0: 8192 cycles (0x1FFF) 1 1: 16384 cycles (0x3FFF)	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
7:4	CKS[3:0]	Clock Division Ratio Select 0x1: PCLKB/4 0x4: PCLKB/64 0xF: PCLKB/128 0x6: PCLKB/512 0x7: PCLKB/2048 0x8: PCLKB/8192 Others: Setting prohibited	R/W
9:8	RPES[1:0]	Window End Position Select 0 0: 75% 0 1: 50% 1 0: 25% 1 1: 0% (do not specify window end position).	R/W
11:10	—	These bits are read as 0. The write value should be 0.	R/W
13:12	RPSS[1:0]	Window Start Position Select 0 0: 25% 0 1: 50% 1 0: 75% 1 1: 100% (do not specify window start position).	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W

The WDTCR register is used to set the clock division ratio, and window start and end positions for refresh, and the timeout period until the down-counter underflows in register start mode.

Some constraints apply to writes to the WDTCR register. For details, see section 24.3.2. Controlling Writes to the WDTCR, WDTRCR, and WDTCSTPR Registers.

In auto start mode, the settings in the WDTCR register are disabled, and the settings in the Option Function Select Register 0 (OFS0) are enabled. The settings for the WDTCR register can also be made in the OFS0 register. For details, see section 24.3.8. Association between Option Function Select Register 0 (OFS0) and WDT Registers.

TOPS[1:0] bits (Timeout Period Select)

The TOPS[1:0] bits select the timeout period, the period until the down-counter underflows, from 1024, 4096, 8192, and 16384 cycles, taking the divided clock specified in the CKS[3:0] bits as 1 cycle. After the down-counter is refreshed, the combination of the CKS[3:0] and TOPS[1:0] bits determines the number of PCLKB cycles until the counter underflows.

Table 24.2 lists the relationship between the CKS[3:0] and TOPS[1:0] bit settings, the timeout period, and the number of PCLKB cycles.

Table 24.2 Timeout period settings

CKS[3:0] bits	TOPS[1:0] bits	Clock division ratio	Timeout period (number of cycles)	PCLKB clock cycles
0x1	00b	PCLKB/4	1024	4096
	01b		4096	16384
	10b		8192	32768
	11b		16384	65536
0x4	00b	PCLKB/64	1024	65536
	01b		4096	262144
	10b		8192	524288
	11b		16384	1048576
0xF	00b	PCLKB/128	1024	131072
	01b		4096	524288
	10b		8192	1048576
	11b		16384	2097152
0x6	00b	PCLKB/512	1024	524288
	01b		4096	2097152
	10b		8192	4194304
	11b		16384	8388608
0x7	00b	PCLKB/2048	1024	2097152
	01b		4096	8388608
	10b		8192	16777216
	11b		16384	33554432
0x8	00b	PCLKB/8192	1024	8388608
	01b		4096	33554432
	10b		8192	67108864
	11b		16384	134217728

CKS[3:0] bits (Clock Division Ratio Select)

The CKS[3:0] bits specify the division ratio of the clock used for the down-counter. The division ratio can be selected from the PCLKB divided by 4, 64, 128, 512, 2048, and 8192. Combined with the TOPS[1:0] bit setting, this allows the WDT to be configured to a count period between 4096 and 134217728 PCLKB clock cycles.

RPES[1:0] bits (Window End Position Select)

The RPES[1:0] bits specify the window end position that indicates the refresh-permitted period. 75%, 50%, 25%, or 0% of the timeout period can be selected for the window end position. Set the window end position to a value less than the value for the window start position (window start position > window end position). If the window start position is set to a value less than or equal to the window end position, the window start position setting is enabled and the window end position is set to 0%.

RPSS[1:0] bits (Window Start Position Select)

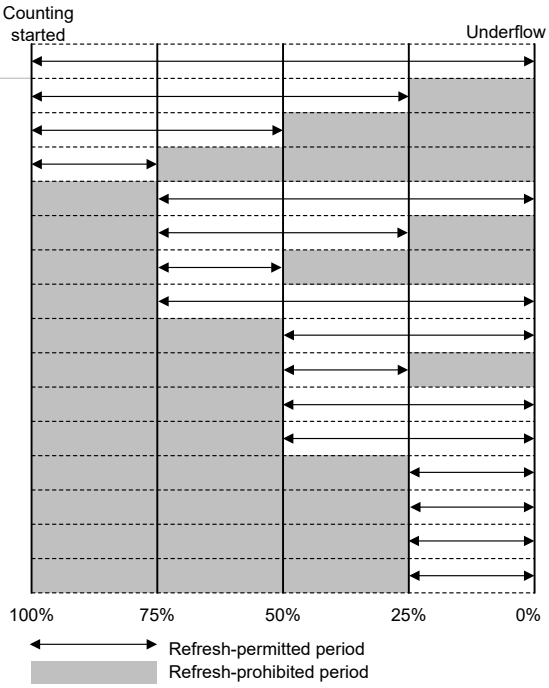
The RPSS[1:0] bits specify the window start position that indicates the refresh-permitted period. 100%, 75%, 50%, or 25% of the timeout period can be selected for the window start position. Set the window start position to a value greater than the value for the window end position. If the window start position is set to a value less than or equal to the window end position, the window start position setting is enabled and the window end position is set to 0%.

Table 24.3 lists the counter values for the window start and end positions, and Figure 24.2 shows the refresh-permitted period set in the RPSS[1:0], RPES[1:0], and TOPS[1:0] bits.

Table 24.3 Relationship between the timeout period and window start and end counter values

TOPS[1:0]	Timeout period		Window start and end counter value			
	Cycles	Counter value	100%	75%	50%	25%
00b	1024	0x03FF	0x03FF	0x02FF	0x01FF	0x00FF
01b	4096	0x0FFF	0x0FFF	0x0BFF	0x07FF	0x03FF
10b	8192	0x1FFF	0x1FFF	0x17FF	0x0FFF	0x07FF
11b	16384	0x3FFF	0x3FFF	0x2FFF	0x1FFF	0x0FFF

RPSS[1:0] bits		RPES[1:0] bits		Window	
b13	b12	b9	b8	Start (%)	End (%)
1	1	1	1	100	0
		1	0		25
		0	1		50
		0	0		75
1	0	1	1	75	0
		1	0		25
		0	1		50
		0	0		75
0	1	1	1	50	0
		1	0		25
		0	1		50
		0	0		75
0	0	1	1	25	0
		1	0		25
		0	1		50
		0	0		75



Note: If window end position setting ≥ window start position setting, the window end position setting is set to 0%.

Figure 24.2 RPSS[1:0] and RPES[1:0] bits setting and refresh-permitted period

24.2.3 WDTSR : WDT Status Register

Base address: WDT = 0x4008_3400

Offset address: 0x04

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

Bit	Symbol	Function	R/W
13:0	CNTVAL[13:0]	Down-Counter Value Value counted by the down-counter	R
14	UNDFF	Underflow Flag 0: No underflow occurred 1: Underflow occurred	R/W*1

Bit	Symbol	Function	R/W
15	REFEF	Refresh Error Flag 0: No refresh error occurred 1: Refresh error occurred	R/W*1

Note 1. Only 0 can be written to clear the flag.

The WDTSR register indicates the counter value of the down-counter and the status of whether an underflow or refresh error occurred in the down-counter.

CNTVAL[13:0] bits (Down-Counter Value)

Read the CNTVAL[13:0] bits to confirm the value of the down-counter. The read value might differ from the actual count by 1.

UNDFF flag (Underflow Flag)

Read the UNDFF flag to confirm whether an underflow occurred in the counter. A value of 1 indicates that the down counter underflowed. Write 0 to the flag to set the value to 0. Writing 1 has no effect.

Clearing of the UNDFF flag takes (N+1) PCLKB cycles. In addition, clearing of the flag is ignored for (N+1) PCLKB cycles after an underflow. N is specified in the WDTCR.CKS[3:0] bits as follows:

- When WDTCR.CKS[3:0] = 0x1, N = 4
- When WDTCR.CKS[3:0] = 0x4, N = 64
- When WDTCR.CKS[3:0] = 0xF, N = 128
- When WDTCR.CKS[3:0] = 0x6, N = 512
- When WDTCR.CKS[3:0] = 0x7, N = 2048
- When WDTCR.CKS[3:0] = 0x8, N = 8192

REFEF flag (Refresh Error Flag)

Read the REFEF flag to confirm whether a refresh error occurred, indicating that a refresh operation was performed during a prohibited period. A value of 1 indicates that a refresh error occurred. Write 0 to the flag to set the value to 0. Writing 1 has no effect.

Clearing of the REFEF flag takes (N+1) PCLKB cycles. In addition, clearing of the flag is ignored for (N+1) PCLKB cycles after a refresh error. N is specified in the WDTCR.CKS[3:0] bits as follows:

- When WDTCR.CKS[3:0] = 0x1, N = 4
- When WDTCR.CKS[3:0] = 0x4, N = 64
- When WDTCR.CKS[3:0] = 0xF, N = 128
- When WDTCR.CKS[3:0] = 0x6, N = 512
- When WDTCR.CKS[3:0] = 0x7, N = 2048
- When WDTCR.CKS[3:0] = 0x8, N = 8192

24.2.4 WDTRCR : WDT Reset Control Register

Base address: WDT = 0x4008_3400

Offset address: 0x06

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

Bit	Symbol	Function	R/W
6:0	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
7	RSTIRQS	WDT Behavior Selection 0: Interrupt 1: Reset	R/W

The WDTRCR register controls reset output by a WDT down-counter underflow or interrupt request output.

Some constraints apply to writes to the WDTRCR register. For details, see section 24.3.2. Controlling Writes to the WDTCR, WDTRCR, and WDTCSTPR Registers.

In auto start mode, the WDTRCR register settings are disabled, and the settings in the Option Function Select register 0 (OFS0) are enabled. The settings for the WDTRCR register can also be made for the OFS0 register. For details, see section 24.3.8. Association between Option Function Select Register 0 (OFS0) and WDT Registers.

24.2.5 WDTCSTPR : WDT Count Stop Control Register

Base address: WDT = 0x4008_3400

Offset address: 0x08

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

Bit	Symbol	Function	R/W
6:0	—	These bits are read as 0. The write value should be 0.	R/W
7	SLCSTP	Sleep-Mode Count Stop Control Register 0: Disable count stop 1: Stop count on transition to Sleep mode	R/W

The WDTCSTPR register controls whether to stop the WDT counter in Sleep mode. Some constraints apply to writes to the WDTCSTPR register. For details, see section 24.3.2. Controlling Writes to the WDTCR, WDTRCR, and WDTCSTPR Registers.

In auto start mode, the WDTCSTPR register settings are disabled, and the settings in the Option Function Select register 0 (OFS0) are enabled. The settings for the WDTCSTPR register can also be made for the OFS0 register. For details, see section 24.3.8. Association between Option Function Select Register 0 (OFS0) and WDT Registers.

SLCSTP bit (Sleep-Mode Count Stop Control Register)

The SLCSTP bit selects whether to stop counting on transition to sleep mode.

24.2.6 Option Function Select Register 0 (OFS0)

For information on the OFS0 register, see section 24.3.8. Association between Option Function Select Register 0 (OFS0) and WDT Registers.

24.3 Operation

24.3.1 Count Operation in each Start Mode

The WDT has two start modes:

- Auto start mode, in which counting automatically starts after a release from the reset state
- Register start mode, in which counting starts with a refresh by writing to the register.

In auto start mode, counting automatically starts after a release from the reset state according to the settings in the Option Function Select register 0 (OFS0) in the flash.

In register start mode, counting starts with a refresh by writing to the WDTRR register after the respective registers are set after a release from the reset state.

Select auto start mode or register start mode by setting the WDT Start Mode Select bit (OFS0.WDTSTRT) in the OFS0 register.

When the auto start mode is selected, the settings in the WDT Control Register (WDTCR), WDT Reset Control Register (WDTRCR), and WDT Count Stop Control Register (WDTCSTPR) are disabled while the settings in the OFS0 register are enabled.

When the register start mode is selected, the setting for the OFS0 register is disabled while the settings for the WDT Control Register (WDTCR), WDT Reset Control Register (WDTRCR), and WDT Count Stop Control Register (WDTCSTPR) are enabled.

24.3.1.1 Register start mode

When the WDT Start Mode Select bit (OFS0.WDTSTRT) is 1, register start mode is selected, the OFS0 register setting is invalid, and the WDT control register (WDTCR), WDT Reset Control Register (WDTRCR), and WDT Count Stop Control Register (WDTCSTPR) are enabled.

After the reset state is released, set the following:

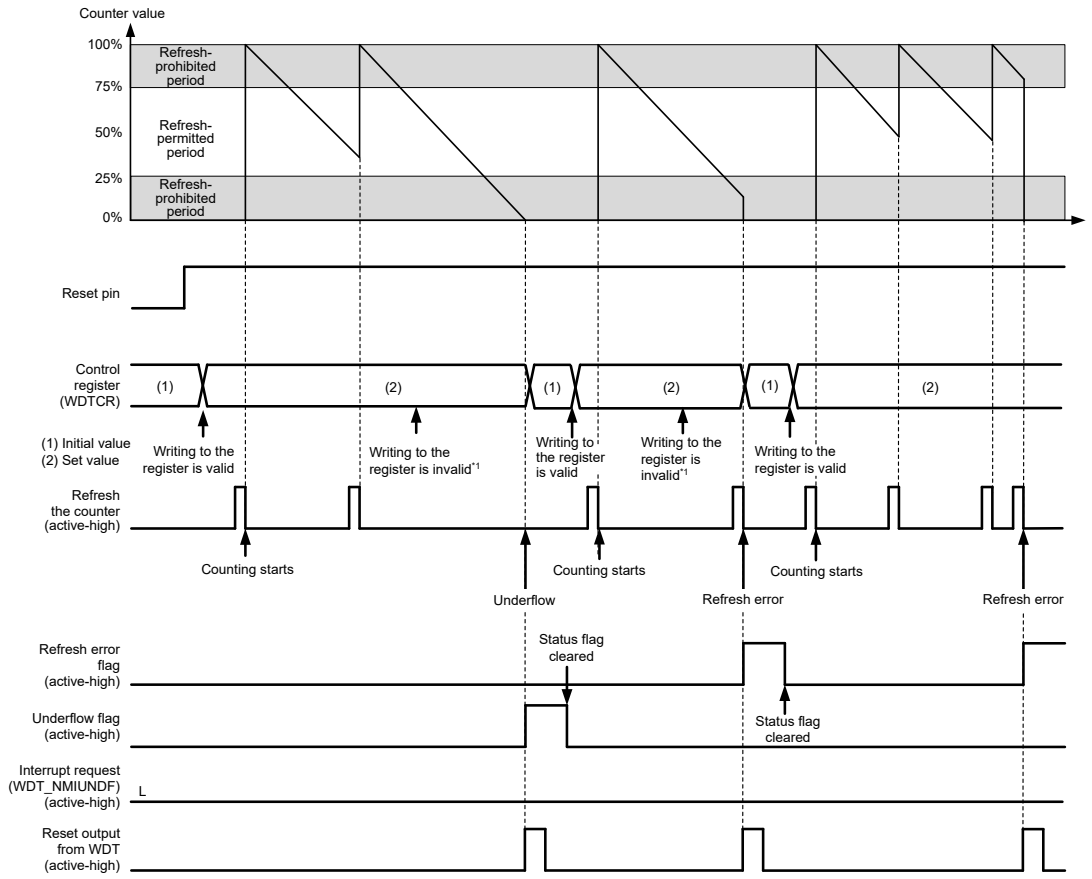
- Clock division ratio in the WDTCR register
- Window start and end positions in the WDTCR register
- Timeout period in the WDTCR register
- Reset output or interrupt request output in the WDTRCR register
- Counter stop control during transitions to Sleep mode in the WDTCSTPR register

The WDT refresh register (WDTRR) refreshes the down counter. As a result, the downcount starts at the value set by the timeout period selection bit (WDTCR.TOPS[1:0]).

Thereafter, as long as the counter is refreshed in the refresh-permitted period, the value in the counter is reset each time the counter is refreshed and counting down continues. The WDT does not output the reset signal or non-maskable interrupt request/interrupt request as long as counting continues. However, if the down-counter underflows because the down-counter cannot be refreshed due to a program runaway, or if a refresh error occurs because the counter was refreshed outside the refresh-permitted period, the WDT outputs the reset signal or a non-maskable interrupt request/interrupt request (WDT_NMIUNDF). Reset output or interrupt request output can be selected in the WDT Reset Interrupt Request Select bit (WDTRCR.RSTIRQS). The interrupt enabled for operating the NMI can be selected in the WDT Underflow/Refresh Error Interrupt Enable bit (NMIER.WDTEN).

Figure 24.3 shows an example of operation under the following conditions:

- Register start mode (OFS0.WDTSTRT = 1)
- WDT reset interrupt request selection (WDTRCR.RSTIRQS = 1)
- The window start position is 75% (WDTCR.RPSS[1:0] = 10b)
- The window end position is 25% (WDTCR.RPES[1:0] = 10b)



Note: Refreshing the counter requires up to 4 count signal cycles from refresh request to execution.
Note 1. See section 24.3.2. Controlling Writes to the WDTCR, WDTRCR, and WDTCSTPR Registers.
Reset pin = RES

Figure 24.3 Operation example in register start mode

24.3.1.2 Auto start mode

When the WDT Start Mode Select bit (OFS0.WDTSTRT) in the Option Function Select Register 0 (OFS0) is 0, auto start mode is selected, the WDT Control Register (WDTCR), WDT Reset Control Register (WDTRCR), and WDT Count Stop Control Register (WDTCSTPR) are disabled, and the settings in the OFS0 register are enabled.

Within the reset state, the setting values for the following in the Option Function Select Register 0 (OFS0) are set in the WDT registers:

- Clock division ratio
- Window start and end positions
- Timeout period
- Reset output or interrupt request
- Counter stop control during transition to Sleep mode

When the reset state is released, the down-counter automatically starts counting down from the value set in the WDT Timeout Period Select bits (OFS0.WDTTOPS[1:0]).

Thereafter, as long as the counter is refreshed in the refresh-permitted period, the value in the counter is reset each time the counter is refreshed and counting down continues. The WDT does not output the reset signal or non-maskable interrupt request/interrupt request (WDT_NMIUNDF) as long as the counting continues. However, if the down-counter underflows

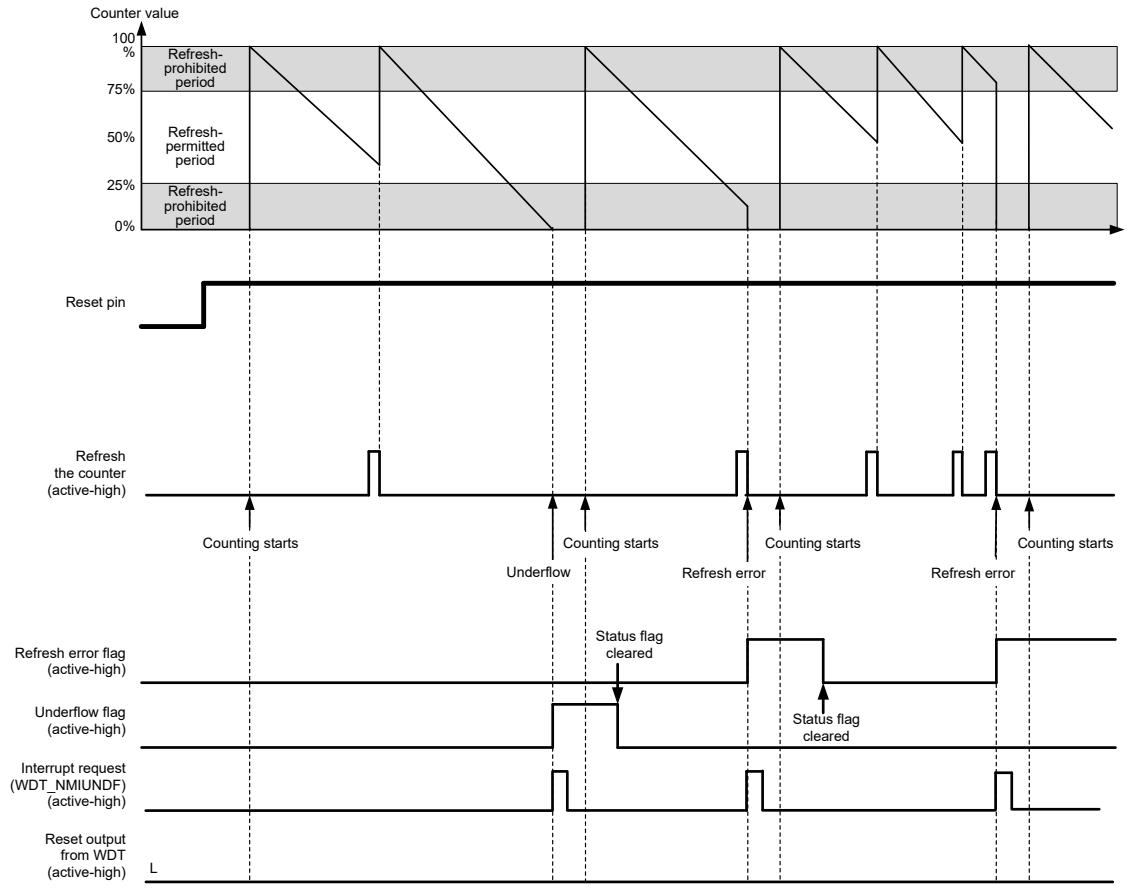
because refreshing of the down-counter is not possible due to a runaway program or if a refresh error occurs due to refreshing outside the refresh-permitted period, the WDT outputs the reset signal or non-maskable interrupt request/interrupt request (WDT_NMIUNDF).

After the reset signal or non-maskable interrupt request/interrupt request is generated, the counter reloads the timeout period after counting for 1 cycle. The value of the timeout period is set in the down-counter and counting restarts.

Reset output or interrupt request output can be selected by setting the WDT Reset Interrupt Request Select bit (OFS0.WDTRSTIRQS). Non-maskable interrupt request or interrupt request can be selected in the WDT Underflow/ Refresh Error Interrupt Enable bit (NMIER.WDTEN).

Figure 24.4 shows an example of operation (non-maskable interrupt) under the following conditions:

- Auto start mode (OFS0.WDTSTRT = 0)
- WDT behavior selection: interrupt (OFS0.WDTRSTIRQS = 0)
- Non-maskable Interrupt: IWDT Underflow/Refresh Error Interrupt Enabled (NMIER.WDTEN = 1)
- The window start position is 75% (OFS0.WDTRPSS[1:0] = 10b)
- The window end position is 25% (OFS0.WDTRPES[1:0] = 10b)



Note: Refreshing the counter requires up to 4 count signal cycles from refresh request to execution.
Note: Reset pin = RES

Figure 24.4 Operation example in auto start mode

24.3.2 Controlling Writes to the WDTCR, WDTRCR, and WDTCSTPR Registers

Writing to the WDT Control Register (WDTCR), WDT Reset Control Register (WDTRCR), or WDT Count Stop Control Register (WDTCSTPR) is possible once each between the release from the reset state and the first refresh operation.

After a refresh (counting starts) or a write to WDTCR, WDTRCR or WDTCSTPR register, the protection signal in the WDT becomes 1 to protect WDTCR, WDTRCR and WDTCSTPR register against subsequent write attempts. This protection is released by the reset source of the WDT. With other reset sources, the protection is not released.

Figure 24.5 shows control waveforms produced in response to writing to the WDTCR.

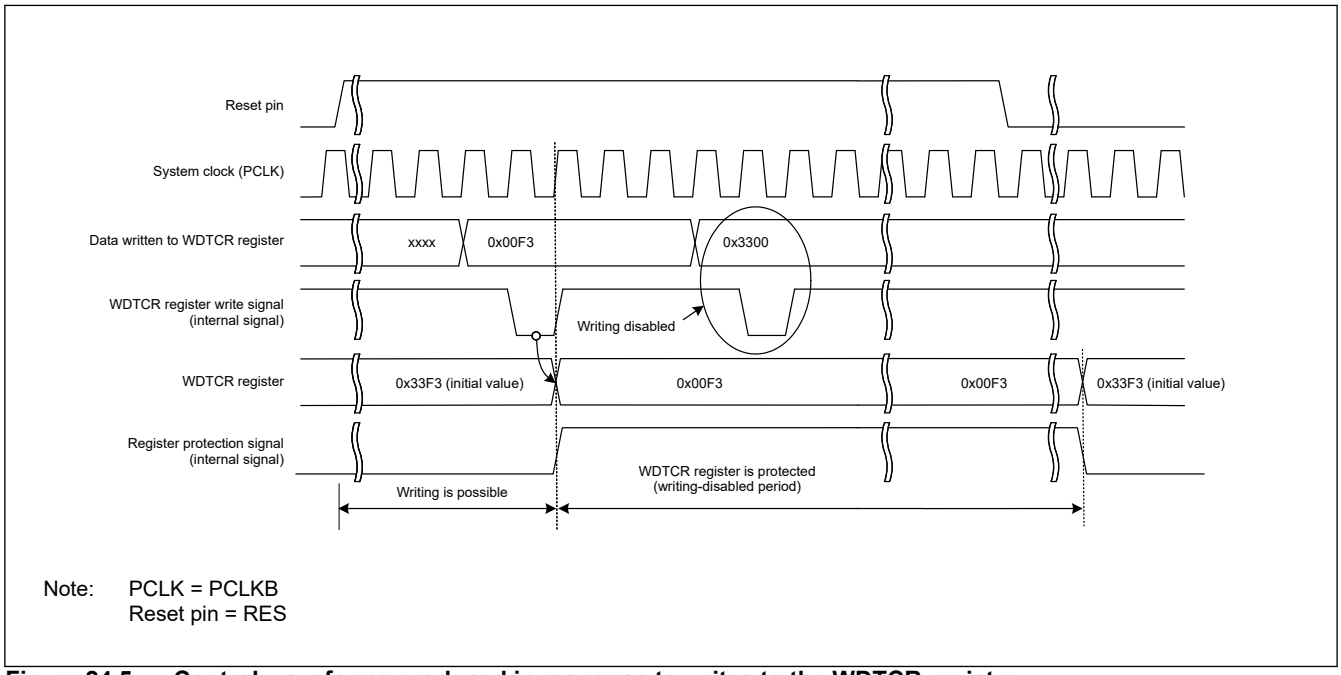


Figure 24.5 Control waveforms produced in response to writes to the WDTCR register

24.3.3 Refresh Operation

To refresh the down counter and start the counting operation, write to the WDT Refresh Register (WDTRR) in the order of values from 0x00 to 0xFF. If a value other than 0xFF is written after 0x00, the down-counter is not refreshed. If an invalid value is written, refreshing is performed normally by writing to the WDTRR register in the order of values from 0x00 to 0xFF.

Correct refreshing is also performed when a register other than WDTRR is accessed or WDTRR is read between writing 0x00 and writing 0xFF to WDTRR. Writes to refresh the counter must be made within the refresh-permitted period, and this is determined by the 0xFF write. For this reason, correct refreshing is performed even when 0x00 is written outside the refresh-permitted period.

[Example write sequences that are valid for refreshing the counter]

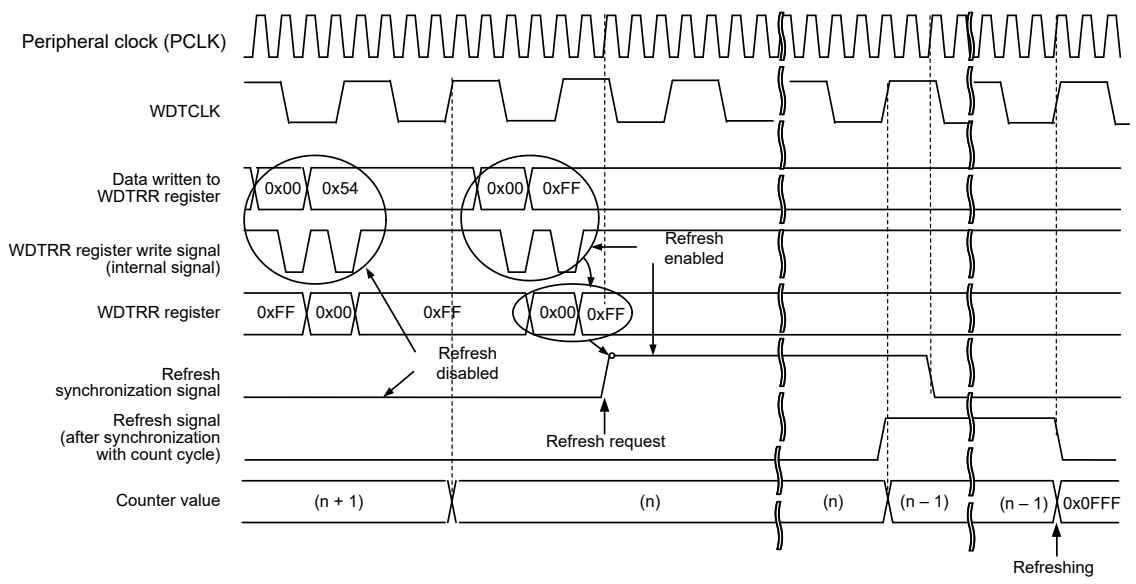
- 0x00 → 0xFF
- 0x00 ((n–1)th time) → 0x00 (nth time) → 0xFF
- 0x00 → access to another register or read from WDTRR → 0xFF

[Example write sequences that are invalid for refreshing the counter]

- 0x23 (a value other than 0x00) → 0xFF
- 0x00 → 0x54 (a value other than 0xFF)
- 0x00 → 0xAA (0x00 and a value other than 0xFF) → 0xFF

After 0xFF is written to the WDT Refresh Register (WDTRR), refreshing the down-counter requires up to 4 cycles of the signal for counting. To meet this requirement, complete writing 0xFF to WDTRR 4 count cycles before the down-counter underflows.

Figure 24.6 shows the WDT refresh-operation waveforms when the clock division ratio is PCLKB/64.



Note: PCLK = PCLKB

Figure 24.6 WDT refresh operation waveforms when WDTCR.CKS[3:0] = 0x4 and WDTCR.TOPS[1:0] = 01b

Note: When setting the refresh time, consider the oscillation accuracy of the clock sources of the PCLKB and WDTCLK. Set values which ensure that refreshing is possible even when the frequency varies in the range of error of the oscillation accuracy.

24.3.4 Status Flags

The refresh error (WDTSR.REFEF) and underflow (WDTSR.UNDFF) flags retain the source of the interrupt request from the WDT. After a release from the interrupt request generation, read the WDTSR.REFEF and WDTSR.UNDFF flags to check for the interrupt source. For each flag, writing 0 clears the bit. Writing 1 has no effect. Leaving the status flags unchanged does not affect operation. If the flags are not cleared at the next interrupt request from the WDT, the earlier interrupt source is cleared and the new interrupt source is written. For the time period between when 0 is written in each flag and when its value is reflected, see section 24.2.3. WDTSR : WDT Status Register.

24.3.5 Reset Output

When the Reset Interrupt Select bit (WDTRCR.RSTIRQS) is set to 1 in register start mode, or when the WDT Reset Interrupt Request Select bit (OFS0.WDTRSTIRQS) in the Option Function Select Register 0 (OFS0) is set to 1 in auto start mode, a reset signal is output for 1 cycle count when an underflow in the down-counter or a refresh error occurs.

In register start mode, the down-counter is initialized (all bits set to 0) and stopped in that state after output of a reset signal. After the reset state is released and the program is restarted, the counter is set up again and counting down starts again with a refresh. In auto start mode, counting down starts automatically after the reset state is released.

24.3.6 Interrupt Sources

When the Reset Interrupt Select bit (WDTRCR.RSTIRQS) is set to 0 in register start mode or when the WDT Reset Interrupt Request Select bit (OFS0.WDTRSTIRQS) in the Option Function Select Register 0 (OFS0) is set to 0 in auto start mode, an interrupt (WDT_NMIUNDF) signal is generated when an underflow in the counter or a refresh error occurs. This interrupt can be used as a non-maskable interrupt or an interrupt. For details, see section 12, Interrupt Controller Unit (ICU).

Table 24.4 WDT interrupt source

Name	Interrupt source	Interrupt to CPU	Start DMAC or DTC
WDT_NMIUNDF	• Down-counter underflow • Refresh error	Possible	Not possible

24.3.7 Reading the Down-Counter Value

The WDT stores the counter value in the down-counter value bits (WDTSR.CNTVAL[13:0]) of the WDT Status Register. Check these bits to obtain the counter value. The read value of the down-counter might differ from the actual count by one.

Figure 24.7 shows the processing for reading the WDT down-counter value when the clock division ratio is PCLKB/64.

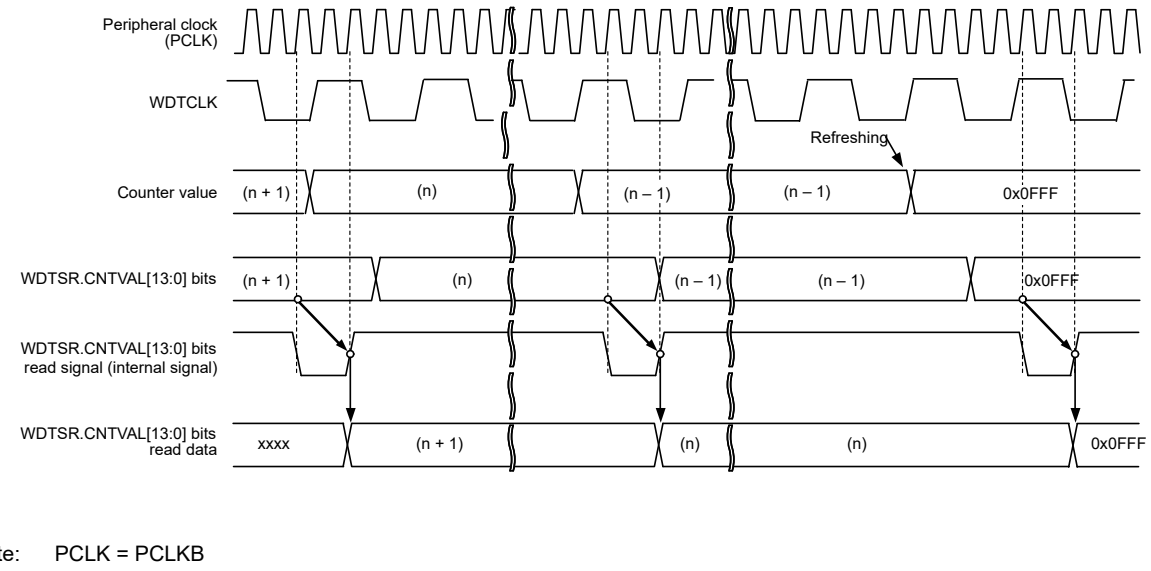


Figure 24.7 Processing for reading WDT down-counter value when WDTCR.CKS[3:0] = 0x4 and WDTCR.TOPS[1:0] = 01b

24.3.8 Association between Option Function Select Register 0 (OFS0) and WDT Registers

Table 24.5 lists the association between the Option Function Select Register 0 (OFS0) used in auto start mode, and the registers used in register start mode. For details on the Option Function Select Register 0 (OFS0), see section 6.2.1. OFS0 : Option Function Select Register 0.

Table 24.5 Association between Option Function Select Register 0 (OFS0) and the WDT registers

Control target	Function	OFS0 register (enabled in auto start mode) OFS0.WDTSTRT = 0	WDT registers (enabled in register start mode) OFS0.WDTSTRT = 1
Down-counter	Timeout period selection	OFS0.WDTTOPS[1:0]	WDTCR.TOPS[1:0]
	Clock division ratio selection	OFS0.WDTCKS[3:0]	WDTCR.CKS[3:0]
	Window start position selection	OFS0.WDTRPSS[1:0]	WDTCR.RPSS[1:0]
	Window end position selection	OFS0.WDTRPES[1:0]	WDTCR.RPES[1:0]
Reset output or interrupt request output	Select a reset interrupt request	OFS0.WDTRSTIRQS	WDTRCR.RSTIRQS
Count stop	Sleep mode count stop control	OFS0.WDTSTPCTL	WDTCSTPR.SLCSTP

24.4 Output to the Event Link Controller (ELC)

The WDT is capable of a link operation for the previously specified module when interrupt request signal is used as an event signal by the ELC. The event signal is output by the counter underflow and refresh error. An event signal is output regardless of the setting of the Reset Interrupt Request Select bit (WDTRCR.RSTIRQS) in register start mode or auto start mode. An event signal can also be output when the next interrupt source is generated while the Refresh Error flag (WDTSR.REFEF) or Underflow flag (WDTSR.UNDFF) is 1. For details, see section 17, Event Link Controller (ELC).

24.5 Usage Notes

24.5.1 ICU Event Link Setting Register n (IELSRn) Setting

Setting 0x53 to ICU Event Link Setting Register n (ICU.IELSRn) is prohibited when WDT reset interrupt request selection resets (OFS0.WDTRSTIRQS = 0 or WDTRCR.RSTIRQS = 0) or when enabling event link operation (ELSRn.ELS[8:0] = 0x53).

25. Independent Watchdog Timer (IWDT)

25.1 Overview

The Independent Watchdog Timer (IWDT) consists of a 14-bit down counter that must be serviced periodically to prevent counter underflow. The IWDT provides functionality to reset the MCU or to generate a non-maskable interrupt or an underflow interrupt. Because the timer operates with an independent, dedicated clock source, it is particularly useful in returning the MCU to a known state as a fail-safe mechanism when the system runs out of control. The IWDT can be triggered automatically by a reset, underflow, refresh error, or a refresh of the count value in the registers.

The IWDT functions differ from those of the WDT in the following respects:

- The divided IWDT-dedicated clock (IWDTCLK) is used as the count source (not affected by PCLKB)
- IWDT does not support register start mode

Table 25.1 lists the IWDT specifications and Figure 25.1 shows a block diagram.

Table 25.1 IWDT specifications

Parameter	Description
Count source*1	IWDT-dedicated clock (IWDTCLK)
Clock division ratio	Division by 1, 16, 32, 64, 128, or 256
Counter operation	Counting down using a 14-bit down-counter
Condition for starting the counter	• Counting automatically starts after a reset • Only secure developer can start the IWDT
Conditions for stopping the counter	• Reset (the down-counter and other registers return to their initial values) • A counter underflows or a refresh error is generated (counting restarts automatically).
Window function	Window start and end positions can be specified (refresh-permitted and refresh-prohibited periods)
Reset output sources	• Down-counter underflows • Refreshing outside the refresh-permitted period (refresh error).
Non-maskable interrupt/interrupt sources	• Down-counter underflows • Refreshing outside the refresh-permitted period (refresh error).
Reading the counter value	The down-counter value can be read by the IWDTSR register
Event link function	• Down-counter underflow event output • Refresh error event output.
Output signal (internal signal)	• Reset output • Interrupt request output • Sleep-mode count stop control output.
Auto start mode	Configurable to the following triggers: • Clock frequency division ratio after a reset (OFS0.IWDTCKS[3:0] bits) • Timeout period of the Independent Watchdog Timer (OFS0.IWDTTOPS[1:0] bits) • Window start position in the Independent Watchdog Timer (OFS0.IWDTRPSS[1:0] bits) • Window end position in the Independent Watchdog Timer (OFS0.IWDTRPES[1:0] bits) • Reset output or interrupt request output (OFS0.IWDTRSTIRQS bit) • Down-count stop function at transition to Sleep, Snooze, or Software Standby mode (OFS0.IWDTSTPCTL bit).
TrustZone Filter	Security attribution can be set

Note 1. Satisfy the frequency of the peripheral module clock (PCLKB) ≥ 4 × (the frequency of the count clock source after division).

The bus interface and registers operate with PCLKB, and the 14-bit counter and control circuits operate with IWDTCLK.

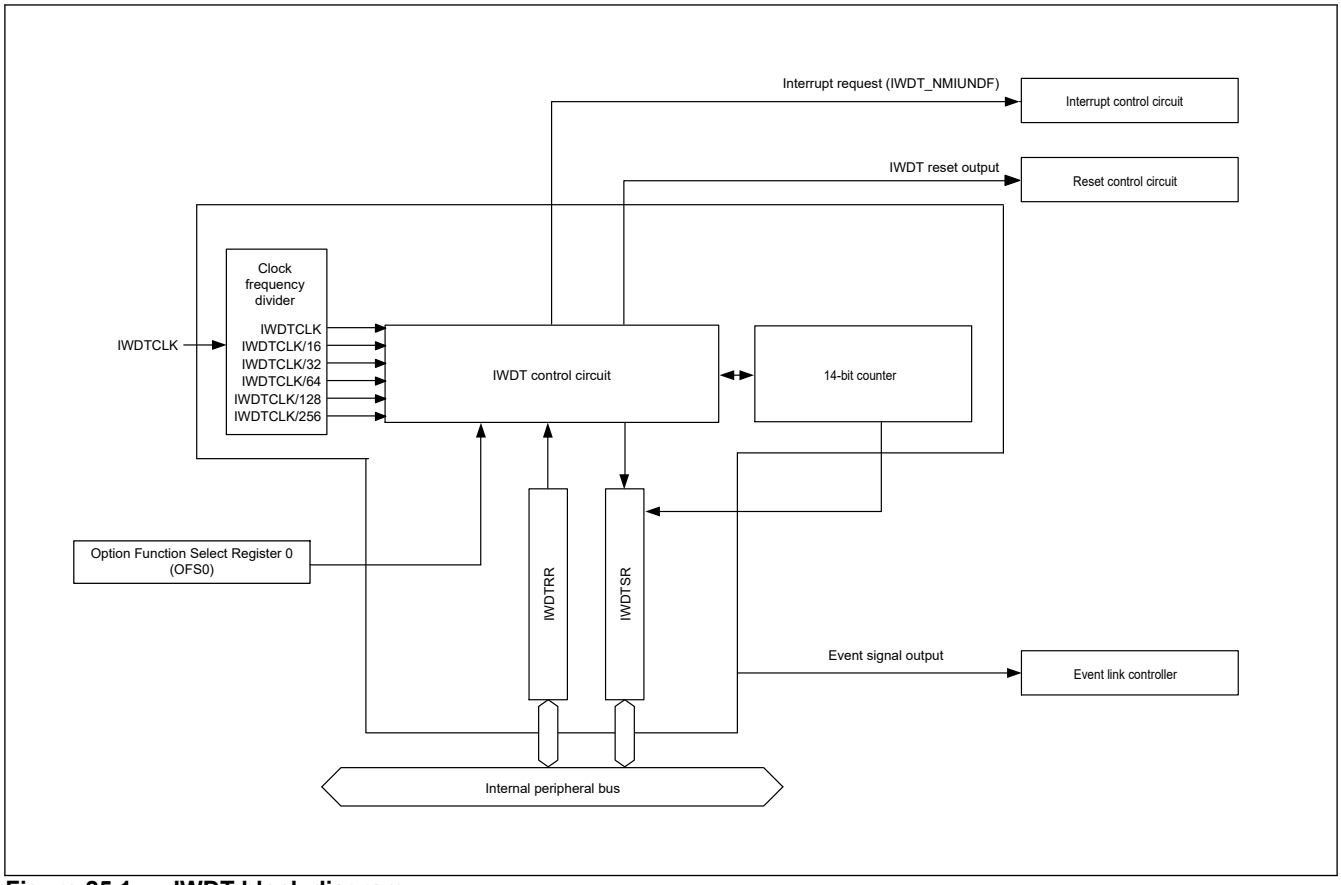


Figure 25.1 IWDT block diagram

25.2 Register Descriptions

25.2.1 IWDTRR : IWDT Refresh Register

Base address: IWDT = 0x4008_3200

Offset address: 0x00

Bit position:	7							0
Bit field:								
Value after reset:	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
7:0	n/a	The down-counter is refreshed by writing 0x00 and then writing 0xFF to this register	R/W

The IWDTRR register refreshes the down-counter of the IWDT. The down-counter of the IWDT is refreshed by writing 0x00 and then writing 0xFF to IWDTRR (refresh operation) within the refresh-permitted period. After the down-counter is refreshed, it starts counting down from the value selected in the IWDT Timeout Period Select bits (OFS0.IWDTTOPS[1:0]) in the Option Function Select Register 0 (OFS0).

When 0x00 is written, the read value is 0x00. When a value other than 0x00 is written, the read value is 0xFF. For details of the refresh operation, see section 25.3.2. Refresh Operation.

25.2.2 IWDTSR : IWDT Status Register

Base address: IWDT = 0x4008_3200

Offset address: 0x04

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

Bit	Symbol	Function	R/W
13:0	CNTVAL[13:0]	Down-counter Value Value counted by the down-counter	R
14	UNDFF	Underflow Flag 0: No underflow occurred 1: Underflow occurred	R/W*1
15	REFEF	Refresh Error Flag 0: No refresh error occurred 1: Refresh error occurred	R/W*1

Note 1. Only 0 can be written to clear the flag.

The IWDTSR register indicates the counter value of the down-counter and whether an underflow or refresh error occurred in the down-counter.

CNTVAL[13:0] bits (Down-counter Value)

Read the CNTVAL[13:0] bits to confirm the value of the down-counter. The read value might differ from the actual count by 1.

UNDFF flag (Underflow Flag)

Read the UNDFF flag to confirm whether an underflow occurred in the down-counter. The value 1 indicates that the down-counter underflowed. Write 0 to the UNDFF flag to set the value to 0. Writing 1 has no effect.

Clearing of the UNDFF flag takes (N + 2) IWDTCLK cycles and 2 PCLKB cycles. In addition, clearing of this flag is ignored for (N + 2) IWDTCLK cycles after an underflow. N is specified in the IWDTCKS[3:0] bits as follows:

- When OFS0.IWDTCKS[3:0] = 0x0, N = 1
- When OFS0.IWDTCKS[3:0] = 0x2, N = 16
- When OFS0.IWDTCKS[3:0] = 0x3, N = 32
- When OFS0.IWDTCKS[3:0] = 0x4, N = 64
- When OFS0.IWDTCKS[3:0] = 0xF, N = 128
- When OFS0.IWDTCKS[3:0] = 0x5, N = 256.

REFEF flag (Refresh Error Flag)

Read the REFEF flag to confirm whether a refresh error occurred. This indicates that a refresh operation was performed during a prohibited period. The value 1 indicates that a refresh error occurred. Write 0 to the REFEF flag to set the value to 0. Writing 1 has no effect.

Clearing of the REFEF flag takes (N + 2) IWDTCLK cycles and 2 PCLKB cycles. In addition, clearing of this flag is ignored for (N + 2) IWDTCLK cycles following a refresh error. N is specified in the IWDTCKS[3:0] bits as follows:

- When OFS0.IWDTCKS[3:0] = 0x0, N = 1
- When OFS0.IWDTCKS[3:0] = 0x2, N = 16
- When OFS0.IWDTCKS[3:0] = 0x3, N = 32
- When OFS0.IWDTCKS[3:0] = 0x4, N = 64
- When OFS0.IWDTCKS[3:0] = 0xF, N = 128
- When OFS0.IWDTCKS[3:0] = 0x5, N = 256.

25.2.3 OFS0 : Option Function Select Register 0

For information on the Option Function Select Register 0 (OFS0), see section 6.2.1. OFS0 : Option Function Select Register 0.

IWDTTOPS[1:0] bits (IWDT Timeout Period Select)

The IWDTTOPS[1:0] bits select the timeout period, that is, the period until the down-counter underflows, from 128, 512, 1024, or 2048 cycles, taking the divided clock specified in the IWDTCKS[3:0] bits as 1 cycle.

After the down-counter is refreshed, the combination of the IWDTCKS[3:0] and IWDTTOPS[1:0] bits determines the number of IWDTCLK cycles until the counter underflows.

Table 25.2 lists the relationship between the IWDTCKS[3:0] and IWDTTOPS[1:0] bit settings, the timeout period, and the number of IWDTCLK cycles.

Table 25.2 Timeout period settings

IWDTCKS[3:0] bits				IWDTTOPS[1:0] bits		Clock division ratio	Timeout period (number of cycles)	IWDTCLK cycles
b7	b6	b5	b4	b1	b0			
0	0	0	0	0	0	IWDTCLK	128	128
				0	1		512	512
				1	0		1024	1024
				1	1		2048	2048
0	0	1	0	0	0	IWDTCLK/16	128	2048
				0	1		512	8192
				1	0		1024	16384
				1	1		2048	32768
0	0	1	1	0	0	IWDTCLK/32	128	4096
				0	1		512	16384
				1	0		1024	32768
				1	1		2048	65536
0	1	0	0	0	0	IWDTCLK/64	128	8192
				0	1		512	32768
				1	0		1024	65536
				1	1		2048	131072
1	1	1	1	0	0	IWDTCLK/128	128	16384
				0	1		512	65536
				1	0		1024	131072
				1	1		2048	262144
0	1	0	1	0	0	IWDTCLK/256	128	32768
				0	1		512	131072
				1	0		1024	262144
				1	1		2048	524288

IWDTCKS[3:0] bits (IWDT-Dedicated Clock Frequency Division Ratio Select)

The IWDTCKS[3:0] bits specify the division ratio of the clock used for the down-counter. The division ratio can be selected from the IWDT-dedicated clock (IWDTCLK) divided by 1, 16, 32, 64, 128, and 256. Combined with the IWDTTOPS[1:0] bit setting, the IWDT can be configured to a count period between 128 and 524,288 IWDTCLK cycles.

IWDTRPES[1:0] bits (IWDT Window End Position Select)

The IWDTRPES[1:0] bits specify the window end position that indicates the refresh-permitted period. 75%, 50%, 25%, or 0% of the timeout period can be selected for the window end position. Set the window end position to a value less than the window start position (window start position > window end position). If the window start position is set to a value less than or equal to the window end position, the window start position setting is enabled and the window end position is set to 0%.

IWDTRPSS[1:0] bits (IWDT Window Start Position Select)

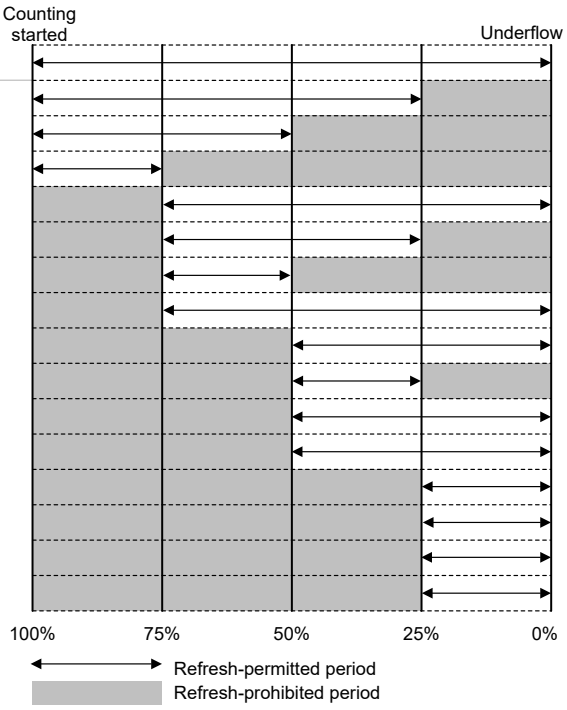
The IWDTRPSS[1:0] bits specify the window start position that indicates the refresh-permitted period. 100%, 75%, 50%, or 25% of the timeout period can be selected for the window start position. Set the window start position to a value greater than the window end position. If the window start position is set to a value less than or equal to the window end position, the window start position setting is enabled and the window end position is set to 0%.

Table 25.3 lists the counter values for the window start and end positions, and Figure 25.2 shows the refresh-permitted period set in the IWDTRPSS[1:0], IWDTRPES[1:0], and IWDTTOPS[1:0] bits.

Table 25.3 Relationship between the timeout period and window start and end counter values

IWDTTOPS[1:0] bits		Timeout period		Window start and end counter value			
b1	b0	Cycles	Counter value	100%	75%	50%	25%
0	0	128	0x007F	0x007F	0x005F	0x003F	0x001F
0	1	512	0x01FF	0x01FF	0x017F	0x00FF	0x007F
1	0	1024	0x03FF	0x03FF	0x02FF	0x01FF	0x00FF
1	1	2048	0x07FF	0x07FF	0x05FF	0x03FF	0x01FF

IWDTRPSS[1:0] bits		IWDTRPES[1:0] bits		Window	
b13	b12	b9	b8	Start (%)	End (%)
1	1	1	1	100	0
		1	0		25
		0	1		50
		0	0		75
1	0	1	1	75	0
		1	0		25
		0	1		50
		0	0		75
0	1	1	1	50	0
		1	0		25
		0	1		50
		0	0		75
0	0	1	1	25	0
		1	0		25
		0	1		50
		0	0		75



Note: If window end position setting ≥ window start position setting, the window end position setting is set to 0%.

Figure 25.2 IWDTRPSS[1:0] and IWDTRPES[1:0] bit settings and refresh-permitted period

IWDTRSTIRQS bit (IWDT Reset Interrupt Request Select)

The IWDTRSTIRQS bit specifies the behavior when an underflow or a refresh error occurs. Setting 1 selects reset output. Setting 0 selects interrupt.

IWDTSTPCTL bit (IWDT Stop Control)

The IWDTSTPCTL bit selects whether to stop counting on transition to Sleep, Snooze, or Software Standby mode.

25.3 Operation

25.3.1 Auto Start Mode

When the IWDT Start Mode Select bit (OFS0.IWDTSTRT) in the Option Function Select Register 0 is 0, auto start mode is selected, otherwise the IWDT is disabled.

Within the reset state, the setting values for the following in the Option Function Select Register 0 (OFS0) are set in the IWDT registers:

- Clock division ratio (OFS0.IWDTCKS[3:0])
- Window start and end positions (OFS0.IWDTRPSS[1:0], OFS0.IWDTRPES[1:0])
- Timeout period (OFS0.IWDTTOPS[1:0])
- Reset output or interrupt request (OFS0.IWDTRSTIRQS)

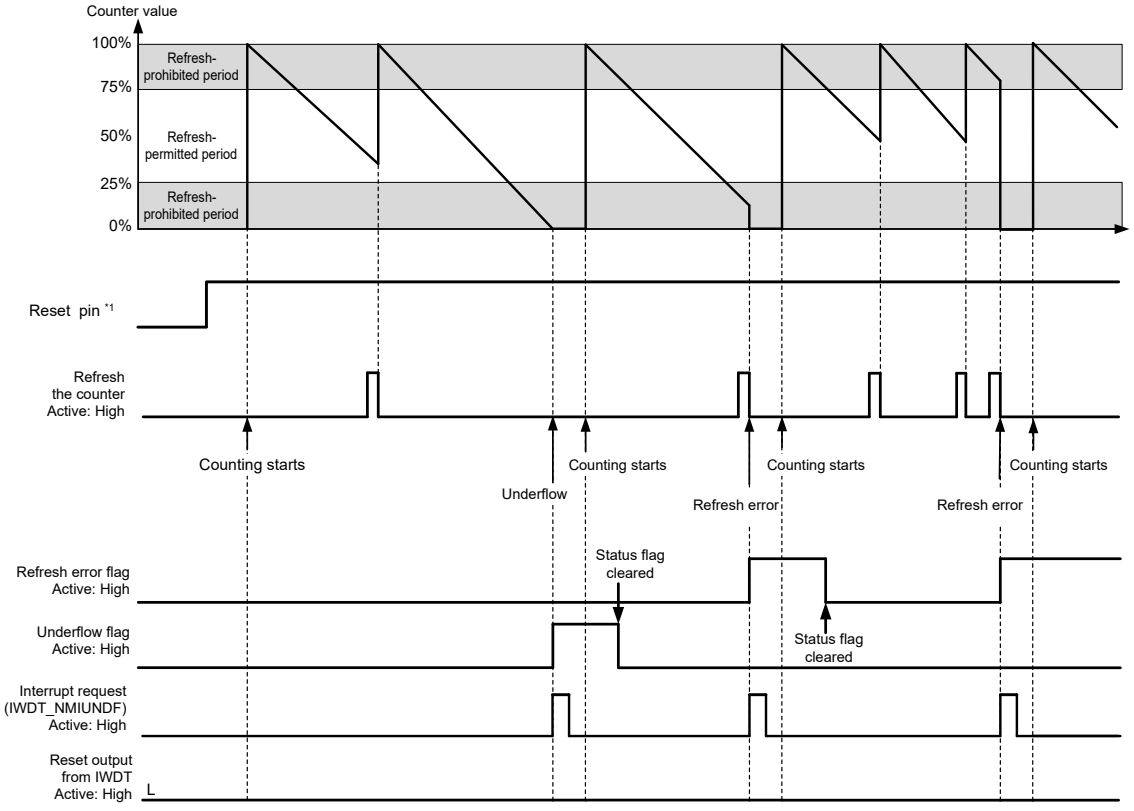
When the reset state is released, the counter automatically starts counting down from the value selected in the IWDT Timeout Period Select bits (OFS0.IWDTTOPS[1:0]).

After that, as long as the program continues normal operation and the counter is refreshed within the refresh-permitted period, the value in the counter is reset each time the counter is refreshed and down-counting continues. The IWDT does not output the reset signal as long as this procedure continues. However, if the counter underflows because the program crashed or because a refresh error occurred when an attempt is made to refresh outside the refresh-permitted period, the IWDT asserts the reset signal or non-maskable interrupt request/interrupt request (IWDT_NMIUNDF).

After the reset signal or non-maskable interrupt request/interrupt request is generated, the counter reloads the timeout period after counting for 1 cycle, the value of the timeout period is set in the down-counter and counting starts. The reset output or interrupt request output can be selected with the IWDT Reset Interrupt Request Select bit (OFS0.IWDTRSTIRQS). The interrupt enabled for operating the NMI can be selected with the IWDT Underflow/Refresh Error Interrupt Enable bit (NMIER.IWDTEN).

Figure 25.3 shows an example of operation under the following conditions:

- Auto start mode (OFS0.IWDTSTRT = 0)
- IWDT behavior selection: interrupt (OFS0.IWDTRSTIRQS = 0)
- Non-maskable Interrupt: IWDT Underflow/Refresh Error Interrupt Enabled (NMIER.IWDTEN = 1)
- The window start position is 75% (OFS0.IWDTRPSS[1:0] = 10b)
- The window end position is 25% (OFS0.IWDTRPES[1:0] = 10b).



Note 1. Reset pin = RES

Figure 25.3 Operation example in auto start mode

25.3.2 Refresh Operation

To refresh the down counter and start the counting operation, write to the IWDT Refresh Register (IWDTRR) in the order of values from 0x00 to 0xFF. If a value other than 0xFF is written after 0x00, the down-counter is not refreshed. If an invalid value is written, refreshing is performed normally by writing to the IWDTRR register in the order of values from 0x00 to 0xFF.

When writes are made in the order of 0x00 (first time) → 0x00 (second time), and if 0xFF is written after that, the writing order 0x00 → 0xFF is satisfied. Writing 0x00 ((n - 1)th time) →0x00 (nth time) → 0xFF is valid, and the refresh is performed correctly. Even when the first value written before 0x00 is not 0x00, correct refreshing is performed as long as the operation contains the write sequence of 0x00 → 0xFF.

Correct refreshing is also performed regardless of whether a register other than IWDTRR is accessed or IWDTRR is read between writing 0x00 and writing 0xFF to IWDTRR. Writes to refresh the counter must be made within the refresh-permitted period. Whether writing is done within the refresh-permitted period is determined when 0xFF is written. For this reason, correct refreshing is performed even when 0x00 is written outside the refresh-permitted period.

[Example write sequences that are valid to refresh the counter]

- 0x00 → 0xFF
- 0x00 ((n - 1)th time) → 0x00 (nth time) → 0xFF
- 0x00 → access to another register or read from IWDTRR → 0xFF.

[Example write sequences that are not valid to refresh the counter]

- 0x23 (a value other than 0x00) → 0xFF

- 0x00 → 0x54 (a value other than 0xFF)
- 0x00 → 0xAA (0x00 and a value other than 0xFF) → 0xFF.

After 0xFF is written to the IWDTRR register, refreshing the counter requires up to 4 cycles of the signal for counting (the IWDT-Dedicated Clock Frequency Division Ratio Select bits (OFS0.IWDTCKS[3:0]) to determine how many cycles of the IWDT-dedicated clock (IWDTCLK) make up 1 cycle for counting. To meet this requirement, writing 0xFF to the IWDTRR must be completed 4 count cycles before the end of the refresh-permitted period or a down-counter underflow. The value of the counter can be checked with the counter bits (IWDTSR.CNTVAL[13:0]).

[Example refreshing timings]

- When the window start position is set to 0x1FFF, even if 0x00 is written to IWDTRR before 0x1FFF is reached at (0x2002, for example), refreshing occurs if 0xFF is written to IWDTRR after the value of the IWDTSR.CNTVAL[13:0] bits reaches 0x1FFF
- When the window end position is set to 0x1FFF, refreshing occurs if 0x2003 (4 count cycles before 0x1FFF) or a greater value is read from the IWDTSR.CNTVAL[13:0] bits immediately after writing 0x00 → 0xFF to IWDTRR
- When the refresh-permitted period continues until count 0x0000, refreshing can be performed immediately before an underflow. In this case, if 0x0003 (4 count cycles before an underflow) or a greater value is read from the IWDTSR.CNTVAL[13:0] bits immediately after writing 0x00 → 0xFF to IWDTRR, no underflow occurs and refreshing is performed.

Figure 25.4 shows the IWDT refresh-operation waveforms when PCLKB > IWDTCLK and the clock division ratio is IWDTCLK.

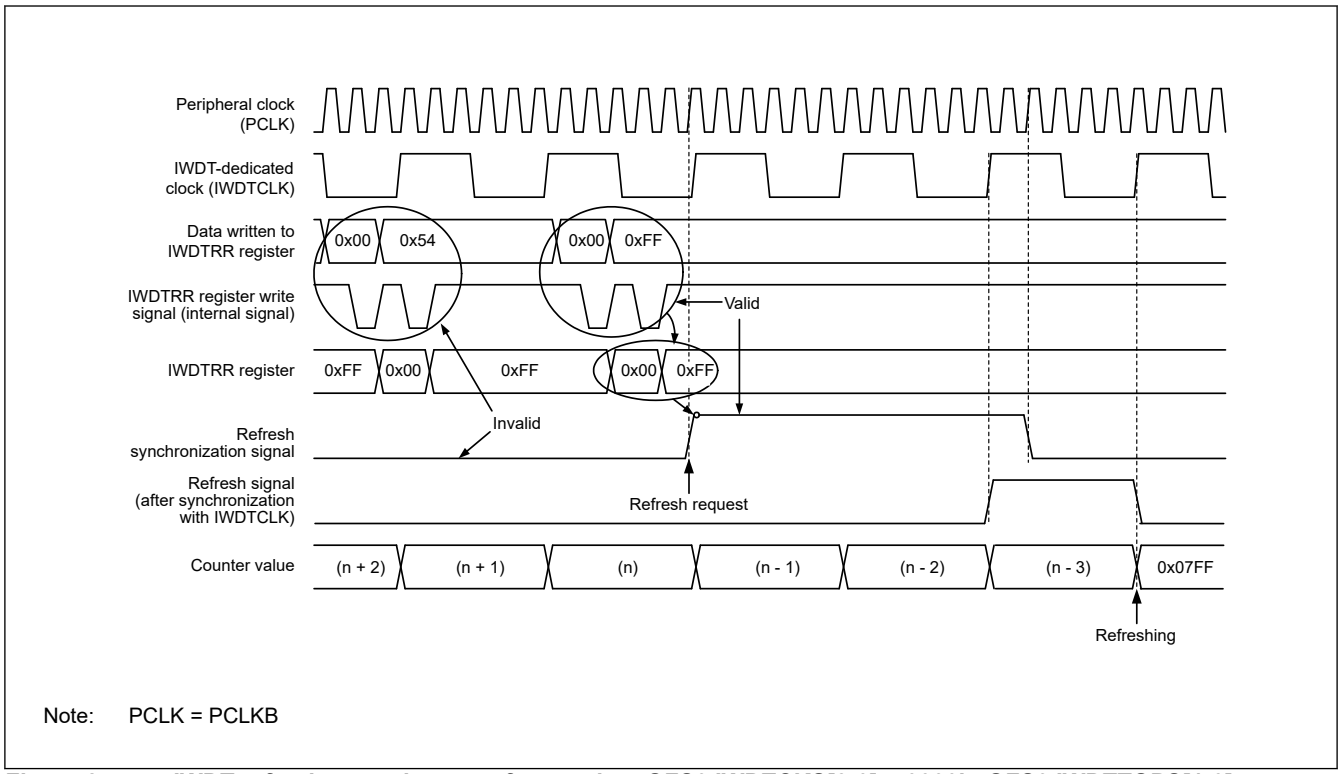


Figure 25.4 IWDT refresh operation waveforms when OFS0.IWDTCKS[3:0] = 0000b, OFS0.IWDTTOPS[1:0] = 11b

25.3.3 Status Flags

The refresh error (IWDTSR.REFEF) and underflow (IWDTSR.UNDFF) flags retain the source of the interrupt request from the IWDT. Therefore, after a release from the interrupt request generation, read the IWDTSR.REFEF and UNDFF flags to check for the interrupt source. For each flag, writing 0 clears the bit and writing 1 has no effect.

Leaving the status flags unchanged does not affect operation. If the flags are not cleared at the time of the next interrupt request from the IWDT, the earlier interrupt source is cleared and the new interrupt source is written. For the time period between when 0 is written in each flag and when its value is reflected, see section 25.2.2. IWDTSR : IWDT Status Register.

25.3.4 Reset Output

When the IWDT Reset Interrupt Request Select bit (OFS0.IWDTRSTIRQS) in the Option Function Select Register 0 (OFS0) is set to 1, a reset signal is output when an underflow in the counter or a refresh error occurs. Counting down automatically starts after the reset output.

25.3.5 Interrupt Sources

When the IWDT Reset Interrupt Request Select bit (OFS0.IWDTRSTIRQS) in the Option Function Select Register 0 (OFS0) is set to 0, an interrupt (IWDT_NMIUNDF) signal occurs when an underflow in the counter or a refresh error occurs. This interrupt can be used as a non-maskable interrupt or an interrupt. For details, see section 12, Interrupt Controller Unit (ICU).

Table 25.4 IWDT interrupt source

Name	Interrupt source	Interrupt to CPU	Start DMAC or DTC
IWDT_NMIUNDF	• Down-counter underflow • Refresh error	Possible	Not possible

25.3.6 Reading the Down-Counter Value

As the counter is a IWDT-dedicated clock (IWDTCLK), the counter value cannot be read directly. The IWDT synchronizes the counter value with the peripheral clock (PCLKB) and stores it in the down-counter value bits (IWDTSR.CNTVAL[13:0]) of the IWDT Status Register. Check these bits to obtain the counter value indirectly.

Reading the counter value requires multiple PCLKB clock cycles (up to 4 clock cycles), and the read counter value might differ from the actual counter value by a value of one count.

Figure 25.5 shows the processing for reading the IWDT counter value when PCLKB > IWDTCLK and the clock division ratio is IWDTCLK.

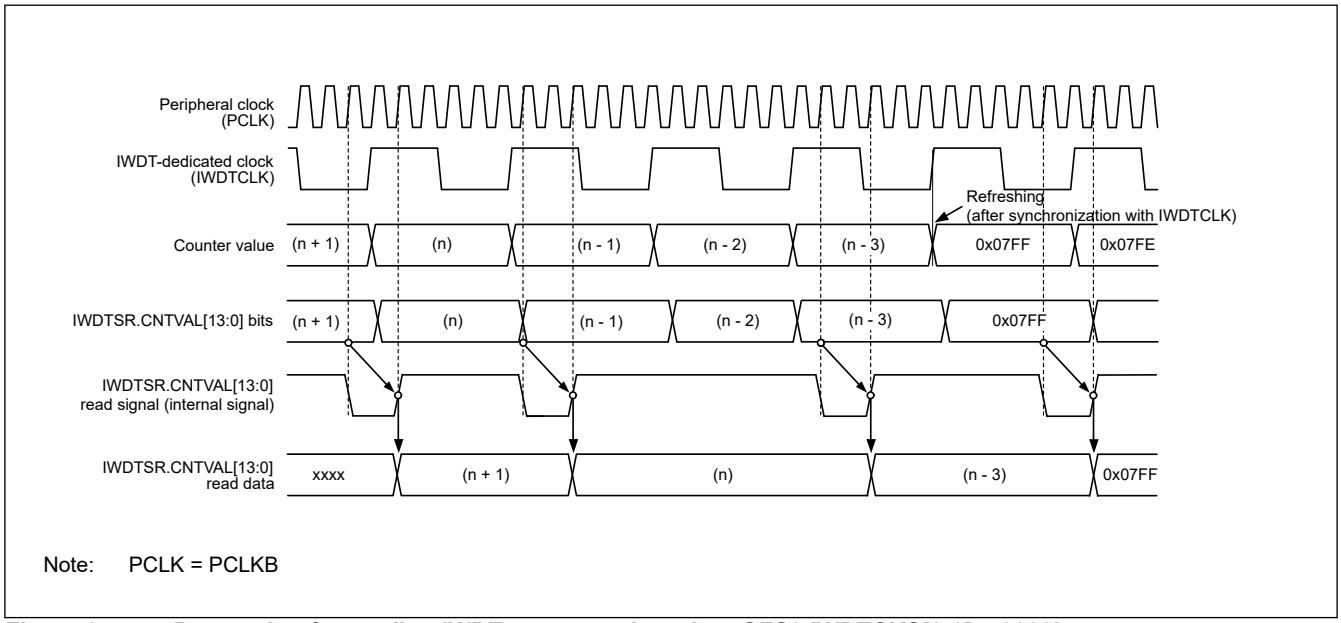


Figure 25.5 Processing for reading IWDT counter value when OFS0.IWDTCKS[3:0] = 0000b, OFS0.IWDTTOPS[1:0] = 11b

25.4 Output to the Event Link Controller (ELC)

The IWDT is capable of link operation for a specified module when the interrupt request signal is used as an event signal by the event link controller (ELC). The event signal is output by the counter underflow or refresh error.

An event signal is output regardless of the setting of the OFS0.IWDTRSTIRQS bit. An event signal can also be output at generation of the next interrupt source while the Refresh Error flag (IWDTSR.REFEF) or Underflow flag (IWDTSR.UNDFF) is 1. For details, see section 17, Event Link Controller (ELC).

25.5 Usage Notes

25.5.1 Refresh Operations

While configuring the refresh time, consider variations in the range of errors given the accuracy of PCLKB and IWDTCLK. Set values that ensure refreshing is possible.

25.5.2 Clock Division Ratio Setting

Satisfy the frequency of the peripheral module clock (PCLKB) ≥ 4 × (the frequency of the count clock source after division).

25.5.3 Constraints on the ICU Event Link Setting Register n (IELSRn) Setting

Setting 0x52 to ICU Event Link Setting Register n (IELSRn.IELS[8:0]) is prohibited when enabling the IWDT reset assertion (OFS0.IWDTRSTIRQS = 0) or when enabling event link operation (ELSRn.ELS[8:0] = 0x52).

26. Serial Communications Interface (SCI)

This is the SCI_B version of the SCI peripheral module.

SCI_B is referred to as SCI in this chapter.

26.1 Overview

The Serial Communications Interface (SCI) × 6 channels have asynchronous and synchronous serial interfaces:

- Asynchronous interfaces (UART and Asynchronous Communications Interface Adapter (ACIA))
- 8-bit clock synchronous interface
- Simple IIC (master-only)
- Simple SPI
- Simple LIN
- Smart card interface
- Manchester interface

The smart card interface complies with the ISO/IEC 7816-3 standard for electronic signals and transmission protocol. SCIn (n = 0 to 4, 9) has FIFO buffers to enable continuous and full-duplex communication, and the data transfer speed can be configured independently using an on-chip baud rate generator.

In this section, PCLK refers to PCLKA, TCLK refers to SCITCLK.

Table 26.1 lists the SCI specifications, Figure 26.1 shows a block diagram of SCI, and Table 26.2 lists the I/O pins.

Table 26.1 SCI specifications (1 of 4)

Parameter		Specifications
Number of modules		6 (SCIn (n = 0 to 4, 9))
Serial communication modes		• Asynchronous • Clock synchronous • Simple IIC • Simple SPI • Simple LIN • Smart card interface • Manchester interface
Transfer speed		Bit rate specifiable with the on-chip baud rate generator
Full-duplex communications		• Transmitter: Continuous transmission possible using double-buffering • Receiver: Continuous reception possible using double-buffering
Half-duplex communications		Half-duplex communication is possible by using only TXDn pins
Data transfer		Selectable as LSB-first or MSB-first transfer
Inverter for communication pins (RXDn, TXDn)		Selectable inverter for each pins (RXDn, TXDn)
Interrupt sources		Transmit end, transmit data empty, receive data full, receive error, receive data ready, address match. Break Field detection/output, Bus collision detection, Active edge detection. Completion of generation of a start condition, restart condition, or stop condition. (for simple IIC mode)
Loop Back function		Self-diagnosis of communication function by IP internal transmission / reception is possible
Synchronizer Bypass function		Ability to bypass synchronization circuit between bus clock and operation clock (TCLK)
Module-stop function		Module-stop state can be set for each channel
Snooze end request		SCI0 address mismatch (SCI0_DCUF)
Clock synchronous mode	Data length	8 bits
	Adjustment of receive sampling timing	Adjustable receive sampling timing after the default timing in master mode only when using internal clock

Table 26.1 SCI specifications (2 of 4)

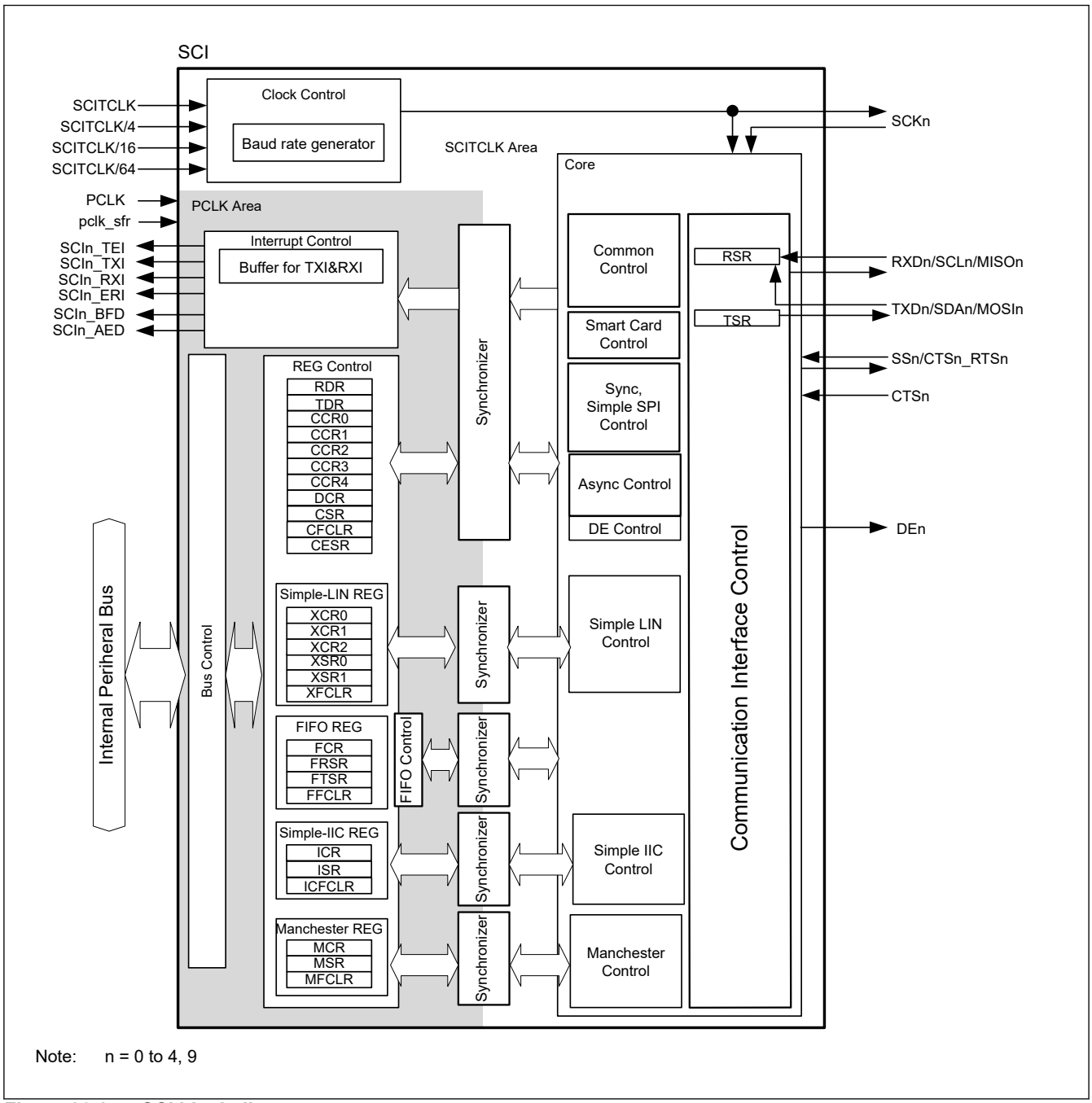
Parameter		Specifications
	Receive error detection	Overrun error
	Clock source	Selectable to internal clock (master mode) or external clock (slave mode)
	Double-speed mode	Baud rate generator double-speed mode is selectable
	Hardware flow control	Transmission and reception controllable with CTSn_RTSn pins
	Transmission and reception	Selectable to 1-stage register or 16-stage FIFO
Asynchronous mode	Data length	7, 8, or 9 bits
	Transmission stop bit	1 or 2 bits
	Adjustment of receive sampling timing	Adjustable receive sampling timing before/after the default timing
	Adjustment of transmit timing	Adjustable edge timing of transmit waveform controlled by the setting value of registers.
	Parity	Even parity, odd parity, or no parity
	Receive error detection	• Parity error • Overrun error • Framing error
	Hardware flow control	Transmission and reception controllable with CTSn_RTSn pin and CTSn pin
	Transmission and reception	Selectable to 1-stage register or 16-stage FIFO
	Address match	Interrupt request/event output can be issued upon detecting a match between received data and the value in the compare match register
	Address mismatch (SCI0 only) receive data	Snooze end request can be issued when detecting a mismatch between the received data and the value in the compare match register
	Start-bit detection	Selectable to low level or falling edge detection
	Break detection	Breaks from framing errors detectable by read from CSR register
	Clock source	Selectable to internal or external clock
	Double-speed mode	Baud rate generator double-speed mode is selectable
	Multi-processor communications function	Serial communication enabled among multiple processors
	RS-485 driver control function	Output DEn signal to enable external transceiver transmit mode
	Noise cancellation	Digital noise filters included on signal paths from the RXDn pin inputs
Smart card interface mode	Error processing	Error signal can be automatically transmitted upon detecting a parity error during reception
		Data can be automatically retransmitted upon receiving an error signal during transmission
	Data type	Both direct and inverse convention supported
Manchester mode	Communication format	Manchester code with the preface and the Start Bit added
	Data length	7,8, or 9 bits
	Transmission stop bit	1 or 2 bits
	Parity function	Even parity, odd parity, or no parity
	Receive error detection	Parity, overrun, framing, Manchester errors
	Hardware flow control	Transmission and reception controllable with CTSn_RTSn pin and CTSn pin
	Clock source	Only internal clock can be used.
	Double-speed mode	Baud rate generator double-speed mode is selectable

Table 26.1 SCI specifications (3 of 4)

Parameter		Specifications
	Multi-processor communication function	Serial communication among multiple processors
	Manchester encoding / decoding function	Function to perform Manchester encoding / decoding of transmission / reception data and communicate using Manchester code
	Noise cancellation	The signal paths from input on the RXDn pins incorporate digital noise filters
	Preface setting / detection function	The function outputs the configured the preface pattern and detects it.
	Start Bit setting / detection function	The function outputs the configured the Start Bit pattern and detects it.
	Reception retiming function	Timing correction is performed for each bit of the received signal
Simple IIC mode	Transfer format	I^2C bus format (MSB-first only)
	Operating mode	Master (single-master operation only)
	Transfer rate	Up to 400 kbps
	Noise cancellation	The signal paths from input on the SCLn and SDAn pins incorporate digital noise filters and provide an adjustable interval for noise cancellation
Simple SPI mode	Data length	8 bits
	Error detection	Overrun error
	Clock source	Selectable to internal clock (master mode) or external clock (slave mode)
	Double-speed mode	Baud rate generator double-speed mode is selectable
	Transmission / Reception	Selectable either 1 stage register or 16-stage FIFO
	Adjustment of receive sampling timing	Adjustable receive sampling timing after the default timing in master mode only when using internal clock
	SSn input pin function	High impedance state can be invoked on the output pins by driving the SSn pin high.
	Clock settings	Configurable among four clock phase and clock polarity settings
Simple LIN	Start Frame Transmission	• Break Field output possible, Break Field output complete interrupt output possible • Bus collision detection possible, bus collision detection interrupt output possible
	Start Frame Reception	• Break Field detectable, Break Field detected interrupt output possible • Control Field 0/1 data comparison function • Control Field 1 can set two types of comparison data of primary and secondary • Priority interrupt bit can be set in Control Field 1 • Handling of Start Frames that do not include a Break Field • Handling of Start Frames that do not include a Control Field 0 • Bit rate measurement function
	Input/Output control function	• Selectable polarity for TXDn and RXDn signals • Selection of a digital filter for the RXDn signal • Half-duplex operation employing RXDn and TXDn signals multiplexed on the same pin • Selectable timing for the sampling of data received through RXDn
Bit rate modulation function		Error reduction through correction of outputs from the on-chip baud rate generator
Event link function		Error event output for receive error or error signal detection (SCIn_ERI) (n = 0 to 4, 9)
		Receive data full event output (SCIn_RXI) (n = 0 to 4, 9)
		Transmit data empty event output (SCIn_TXI) (n = 0 to 4, 9)
		Address match event output (SCIn_AM) (n = 0 to 4, 9)
		Active edge detection event output (SCIn_AED) (n = 0 to 4, 9)

Table 26.1 SCI specifications (4 of 4)

Parameter	Specifications
	Transmit end event output (SCIn_TEI) (n = 0 to 4, 9)
TrustZone Filter	Security attribution can be set for each channels



Note: n = 0 to 4, 9

Figure 26.1 SCI block diagram

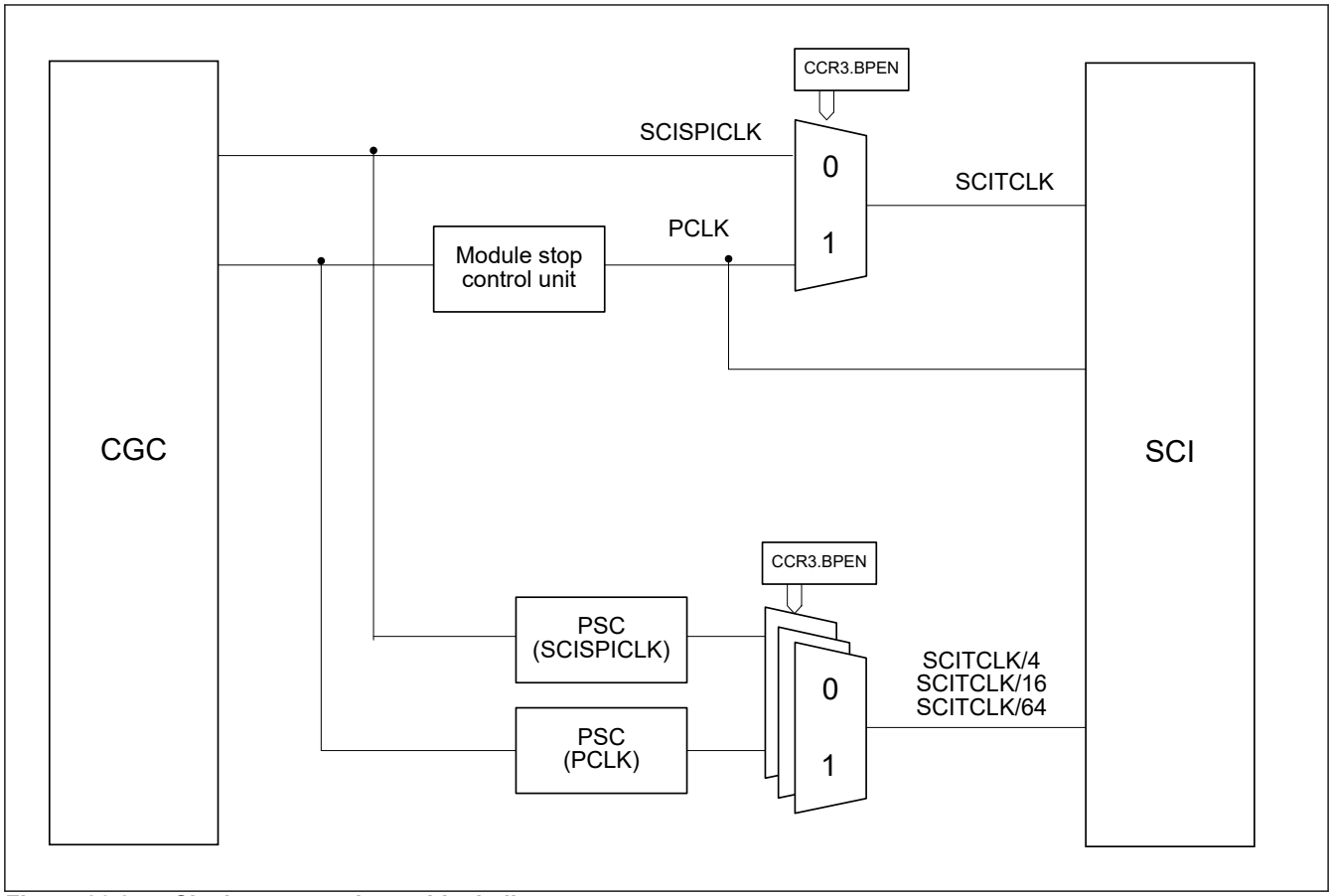


Figure 26.2 Clock source selector block diagram

Table 26.2 SCI I/O pins

Function	Pin name	Input/ Output	Description
SCIn (n = 0 to 4, 9)	RXDn/SCLn/MISOn	Input/ Output	SCIn receive data input SCIn I^2C clock input/output SCIn slave transmit data input/output
	TXDn/SDAn/MOSIn	Input/ Output	SCIn transmit data output SCIn I^2C data input/output SCIn master transmit data input/output
	SSn/CTSn_RTSn	Input/ Output	SCIn chip select input, active-low SCIn transfer start control input/output, active-low
	CTSn	Input	SCIn transfer start control input, active-low
	DEn	Output	Driver Enable signal output
	SCKn	Input/ Output	SCIn clock input/output

26.2 Register Descriptions

26.2.1 RSR : Receive Shift Register

RSR is a shift register that receives serial data input from the RXDn pin and converts it into parallel data. When one frame of data is received, the data is automatically transferred to the RDR register. The RSR register cannot be directly accessed by the CPU.

26.2.2 RDR : Receive Data Register

Base address: SCI_Bn = 0x4011_8000 + 0x0100 × n (n = 0 to 4, 9)

Offset address: 0x00

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

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

Bit	Symbol	Function	R/W
8:0	RDAT[8:0]	Serial receive data RDAT is a 9-bit register for storing received data. Received data is stored in [6:0] when 7-bit data is selected, in [7:0] when 8-bit data is selected, and in [8:0] when 9-bit data is selected. And 0 is stored in the unused bit.	R
9	MPB	Multi-processor flag 0: Data transmission cycles 1: ID transmission cycles	R
10	DR	Receive data ready flag FRSR.DR can be read.	R
11	FPER	FIFO parity error flag Valid only in Asynchronous mode 0: There is no parity error in the data read from the receive-FIFO 1: There is parity error in the data read from the receive-FIFO	R
12	FFER	FIFO framing error flag Valid only in Asynchronous mode 0: There is no framing error in the data read from the receive-FIFO 1: There is framing error in the data read from the receive-FIFO	R
23:13	—	These bits are read as 0.	R
24	ORER	Overrun Error flag CSR.ORER can be read.	R
26:25	—	These bits are read as 0.	R
27	PER	Parity error flag CSR.PER can be read.	R
28	FER	Framing error flag CSR.FER can be read.	R
31:29	—	These bits are read as 0.	R

In FIFO mode (CCR3.FM = 1), this register is 16-stage FIFO buffer configuration.

RDAT[8:0] bit (Serial receive data)

After one frame of data is received, the received data is transferred from the RSR register to this register, thus allowing the RSR register to receive the next data.

The RSR and RDR registers have a double-buffered construction to enable continuous reception.

For Non-FIFO mode, read RDR only once when a receive data full interrupt (SCIn_RXI) request is issued. Without reading received data from RDR, if the next one frame is received, an overrun error occurs.

For FIFO mode, continuous reception is executed until 16 stages are stored. If data is read when there is no received data in the receive-FIFO(RDR), the value is undefined. When the receive-FIFO (RDR) are full of received data, subsequent serial receive data is lost.

The CPU cannot write to RDR.

0 is stored in the bit position which isn't received (RDR.bit8 or RDR.bit7) at the time of 7bit or 8bit communication of Asynchronous and Manchester mode.

MPB bit (Multi-processor flag)

In Asynchronous mode and Manchester mode, during multi-processor communication (CCR3.MP = 1), the value of the multi-processor bit corresponding to the received data (RDAT[8:0]) can be read.

FPER bit (FIFO parity error flag)

Indicates whether the data read from the receive-FIFO has a parity error.

0 is stored for non-FIFO mode.

FFER bit (FIFO framing error flag)

Indicates whether the data read from receive-FIFO has a framing error.

0 is stored for non-FIFO mode.

26.2.3 TDR : Transmit Data Register

Base address: SCI_Bn = 0x4011_8000 + 0x0100 × n (n = 0 to 4, 9)

Offset address: 0x04

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

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

Bit	Symbol	Function	R/W
8:0	TDAT[8:0]	Serial transmit data TDAT is a 9-bit register for setting transmit data. Transmit data is set in [6:0] when 7-bit data is selected, in [7:0] when 8-bit data is selected, and in [8:0] when 9-bit data is selected. When byte access, write TDR [15:8] and then write TDR [7:0].	R/W
9	MPBT	Multi-processor transfer bit flag Value of the multi-processor bit in the transmission frame. This bit is use in Asynchronous and Manchester mode. When writing to this bit when not used, write the initial value. 0: Data transmission cycles 1: ID transmission cycles	R/W
11:10	—	These bits are read as 1. The write value should be 1.	R/W
12	TSYNC	Transmit SYNC data It is valid when MCR.SBSEL = 1 and MCR.SYNSEL = 1 in Manchester mode. When this bit is not used, write the initial value. 0: The Start Bit is transmitted as DATA SYNC. 1: The Start Bit is transmitted as COMMAND SYNC.	R/W
31:13	—	These bits are read as 1. The write value should be 1.	R/W

In FIFO mode (CCR3.FM = 1), this register is 16-stage FIFO buffer configuration.

TDAT[8:0] bit (Serial transmit data)

The TDR is a 9-bit register for storing transmit data.

When empty space is detected in the TSR register, the transmit data stored in the TDR registers is transferred to TSR, and transmitting is started.

The TSR and TDR registers have a double-buffered construction to realize continuous transmission. When the next data to be transmitted is stored in TDR after one frame of data has been transmitted, the transmitting operation is continued by transfer to the TSR register.

When the SCI detects that the transmit shift register (TSR) is empty, it transmits data written in the transmit-FIFO (TDR) into TSR and starts serial transmission. Continuous serial transmission is executed until there is no transmit data left in the transmit-FIFO (TDR).

For non-FIFO mode, when a transmit data empty interrupt (SCIn_TXI) request is issued and CCR0.TE is 1, write transmit data to the TDR only once.

For FIFO mode, when transmit-FIFO is full of transmit data 16 frames, no more data can be written. If writing of new data is attempted, the data is ignored.

The TDR register can always be read / written from the CPU. And when byte access, write TDR[15:8] and then write TDR[7:0].

MPBT bit (Multi-processor transfer bit flag)

Selects the multi processor bit of transmit frame.

TSYNC bit (Transmit SYNC data)

When Manchester mode and MCR.SBSEL = 1 and MCR.SYNSEL = 1, the type of SYNC selected according to this bit becomes the Start Bit of the transmission frame.

26.2.4 TSR : Transmit Shift Register

TSR is a shift register that transmits serial data. To perform serial data transmission, the SCI first automatically transfers transmit data from TDR to TSR, then sends the data to the TXDn pin. The CPU cannot directly access the TSR.

26.2.5 CCR0 : Common Control Register 0

Base address: SCI_Bn = 0x4011_8000 + 0x0100 × n (n = 0 to 4, 9)

Offset address: 0x08

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

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

Bit	Symbol	Function	R/W
0	RE	Receive Enable 0: Serial reception is disabled 1: Serial reception is enabled	R/W*1 *3
3:1	—	These bits are read as 0. The write value should be 0.	R/W
4	TE	Transmit Enable 0: Serial transmission is disabled 1: Serial transmission is enabled	R/W*1
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
8	MPIE	Multi-Processor Interrupt Enable Valid in Asynchronous mode and Manchester mode when CCR3.MP is 1. This bit should set 0 in smart card interface mode. 0: Non-Multi-Processor reception 1: Multi -Processor reception When the data with the multi-processor bit set to 0 is received, the data is not read, and setting the status flags to 1 is disabled. When the data with the multiprocessor bit set to 1 is received, the MPIE bit is automatically cleared to 0, and non-multi-processor reception is resumed. If you want to continue receiving operation using the multiprocessor function, set this bit to 1 sufficiently earlier than receiving the STOP bit of the next received frame. (Consider the synchronization delay time.)	R/W*2
9	DCME	Data Compare Match Enable Valid only in Asynchronous mode 0: Address match function is disabled 1: Address match function is enabled	R/W*2
10	IDSEL	ID frame select Valid only in Asynchronous mode with multi-processor 0: It's always compared data in spite of the value of the MPB bit. 1: It's compared data when the MPB bit is 1 (ID frame) only.	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W
16	RIE	Receive Interrupt Enable 0: SCIn_RXI and SCIn_ERI interrupt requests are disabled 1: SCIn_RXI and SCIn_ERI interrupt requests are enabled	R/W
19:17	—	These bits are read as 0. The write value should be 0.	R/W
20	TIE	Transmit Interrupt Enable 0: SCIn_TXI interrupt request is disabled 1: SCIn_TXI interrupt request is enabled	R/W
21	TEIE	Transmit End Interrupt Enable This bit should set 0 in smart card interface mode. 0: SCIn_TEI interrupt request is disabled 1: SCIn_TEI interrupt request is enabled	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W
24	SSE	SSn Pin Function Enable Valid in Simple SPI mode. In slave mode (CCR3.CKE[1:0] = 1x), set 1 to this bit. 0: SSn pin function is disabled. 1: SSn pin function is enabled.	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. In clock-synchronous mode (CCR3.MOD[2:0] = 010b), Simple SPI mode (CCR3.MOD[2:0] = 011b), and Simple IIC mode (CCR3.MOD[2:0] = 100b), 1 can be written only when TE = 0 and RE = 0. After setting TE or RE to 1, only 0 can be written in TE and RE. In other mode, writing is enabled under any condition.

Note 2. This bit is a bit that is cleared by hardware. Note that writing to a bit other than this bit with a bit manipulation instruction may cause this bit to be unintentionally set to 1 by a read-modify-write operation.

Note 3. In clock synchronous mode and Simple SPI mode, receive only setting is prohibited in the internal clock (master mode) (TE = 0 and RE = 1 setting prohibited).

RE bit (Receive Enable)

Enables or disables serial receive operation.

When this bit is set to 1, serial reception becomes possible after the synchronization delay time has elapsed in asynchronous mode or the synchronous clock input in clock synchronous mode or the neg-edge of RXDn in manchester mode or start bit in smart-card-interface-mode.

Note that CCR3 should be set prior to setting the RE bit to 1 in order to designate the reception format.

Except smart-card-interface-mode, even if reception is halted by setting the RE bit to 0, the CSR. RDRF, FER, PER, ORER, MSR.MER, SBER, SYER, PFER, FRSR. DR flags are not affected, and the previous values is retained. In smart-card-interface-mode, even if reception is halted by setting the RE bit to 0, the CSR.FER, PER, ORER flags are not affected and the previous value is retained. Also, to stop the reception operation, synchronization delay time will be required from when the RE bit is set to 0 until the reception operation is stopped.

TE bit (Transmit Enable)

Enables or disables serial transmission.

When this bit is set to 1, serial transmission becomes possible after the synchronization delay time has elapsed. After the synchronization delay time, transmission is started by writing transmit data to TDR. Note that CCR3 should be set prior to setting the TE bit to 1 in order to designate the transmission format. In addition, the synchronization delay time is required until the transmission control circuit is stopped after the TE bit is set to 0.

MPIE bit (Multi-Processor Interrupt Enable)

When this bit is set to 1 and the data with the multi-processor bit set to 0 is received, the data is not read and setting the status flags (CSR.RDRF, ORER, FER, FRSR, DR, MSR.MER, SYER, PFER, SBER) are disabled.

When the data with the multi-processor bit set to 1 is received, the MPIE is automatically cleared to 0, and normal reception is resumed. For details, see section 26.4. Multi-Processor Communication Function. If you want to continue receiving operation using the multiprocessor function, set this bit to 1 sufficiently earlier than receiving the STOP bit of the next received frame.

When the receive data includes the MPB bit set to 0, the receive data is not transferred from the RSR to the RDR, a receive error is not detected, and setting the flags ORER, FER, MER, SYER, PFER, and SBER to 1 is disabled.

When the receive data includes the MPB bit set to 1, the MPB bit is set to 1, the MPIE bit is automatically cleared to 0, the SCIn_RXI and SCIn_ERI interrupt requests are enabled (if CCR0.RIE is set to 1), and setting the flags ORER, FER, MER, SYER, PFER, and SBER to 1 is enabled.

MPIE should be set to 0 if multi-processor communications function is not to be used.

DCME bit (Data Compare Match Enable)

It can select whether the Address match function (data compare match function) uses or not.

When DCME is 1, if SCI detects the match to the comparison data (CCR4.CMPD) with receive data, DCME is cleared automatically, and after that, SCI operation mode will be receive mode without data compare match function.

See section 26.3.6. Address Match (Receive Data Match Detection) Function.

The write value should be 0 other than asynchronous mode.

IDSEL bit (ID frame select)

It can select whether it is compared in spite of the value of MPB bit or it is compared only the data of MPB bit = 1 (ID frame) when the Address match function is valid. Set at the same time as DCME.

RIE bit (Receive Interrupt Enable)

Enables or disables SCIn_RXI and SCIn_ERI interrupt requests.

SCIn_RXI and SCIn_ERI interrupt request is disabled by setting the RIE bit to 0.

An SCIn_ERI interrupt request can be canceled by reading 1 from the CSR.ORER, FER, or PER and then setting the flag to 0 or setting the RIE bit to 0.

In the case of Manchester mode, the MER, SYER, PFER and SBER flags are also the cause of SCIn_ERI interrupt request, so the same processing is necessary. For details of these flags, see section 26.2.12. MCR : Manchester Control Register and section 26.2.21. MSR : Manchester Status Register.

TIE bit (Transmit Interrupt Enable)

Enables or disables SCIn_TXI interrupt request.

An SCIn_TXI interrupt request is disabled by setting the TIE bit to 0. At the beginning of transmission, set 1 to CCR0.TE and CCR0.TIE simultaneously. Then the SCIn_TXI interrupt request is generated.

TEIE bit (Transmit End Interrupt Enable)

Enables or disables a SCIn_TEI interrupt request. A SCIn_TEI interrupt request is disabled by setting the TEIE bit to 0.

In simple IIC mode, the SCIn_TEI is allocated to the interrupt on completion of issuing a start, restart, or stop condition (STIn). In this case, the TEIE bit can be used to enable or disable the STIn.

SSE bit (SSn Pin Function Enable)

Set this bit to 1 if the SSn pin is to be used in control of transmission and reception (in simple SPI mode).

Set this bit to 0 in any other mode. Do not set both the SSE and CTSE bits to enabled (even if this setting is made, operation is the same as that when these bits are set to 0).

In the slave mode (CCR3.CKE[1:0] = 10 or 11), SSE should be set 1.

In the master mode (CCR3.CKE[1:0] = 00 or 01) and single-master, the SSn pin on the master side is not required to control reception and transmission, so SSE should be set 0.

26.2.6 CCR1 : Common Control Register 1

Base address: SCI_Bn = 0x4011_8000 + 0x0100 × n (n = 0 to 4, 9)

Offset address: 0x0C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	NFEN	—	NFCS[2:0]			—	—	—	SHARPS	—	—	—	SPLP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	RINV	TINV	—	—	PM	PE	—	—	SPB2IO	SPB2DT	—	—	CTSPEN	CTSE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit	Symbol	Function	R/W
0	CTSE	CTS Enable 0: CTS function is disabled (RTS output function is enabled). 1: CTS function is enabled.	R/W
1	CTSPEN	CTS external pin Enable 0: Alternate setting to use CTS and RTS functions as either one pin 1: Dedicated setting for separately using CTS and RTS functions with 2 pins	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	SPB2DT	Serial port break data select The output level of TXDn pin is selected when CCR0.TE = 0 and SPB2IO = 1.*1 0: When TINV is 0, Low level is output in TXDn pin. When TINV is 1, High level is output in TXDn pin. 1: When TINV is 0, High level is output in TXDn pin. When TINV is 1, Low level is output in TXDn pin.	R/W
5	SPB2IO	Serial port break I/O It's selected whether the value of SPB2DT is output to TXDn pin when CCR0.TE = 0.*1 0: The value of SPB2DT bit is not output in TXDn pin. 1: The value of SPB2DT bit is output in TXDn pin.	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
8	PE	Parity Enable Valid only in Asynchronous mode and Manchester mode. In Smart Card Interface mode, set 1 to this bit. 0: When transmitting: Do not add parity bit When receiving: Do not check parity bit 1: When transmitting: Add parity bit When receiving: Check parity bit	R/W
9	PM	Parity Mode Valid only when the PE bit is 1 0: Selects even parity 1: Selects odd parity	R/W
11:10	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
12	TINV	TXD invert 0: Transmit data is not inverted and output to TXDn.*2 1: Transmit data is inverted and output to TXDn.	R/W
13	RINV	RXD invert 0: Received data from RXDn is not inverted and input.*2 1: Received data from RXDn is inverted and input.	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W
16	SPLP	Loopback Control It can be used when internal clock operation in asynchronous mode, internal mode operation in Manchester mode, internal clock operation in clock synchronous mode. 0: Normal mode 1: Loopback mode	R/W
19:17	—	These bits are read as 0. The write value should be 0.	R/W
20	SHARPS	Half-duplex communication select In the Simple IIC mode, in the Smart Card Interface Mode or in the Simple SPI mode, this bit should be set 0. 0: TXDn pin, RXDn pin independent 1: TXDn / RXDn pin combination use (Half-duplex communication using TXDn pin)	R/W
23:21	—	These bits are read as 0. The write value should be 0.	R/W
26:24	NFCS[2:0]	Noise Filter Clock Select Valid in Asynchronous mode and Manchester mode, Simple LIN mode, and Simple IIC mode. In Simple IIC mode, 000 setting is prohibited. The on-chip baud rate generator source clock means the clock selected by CCR2.CKS [1:0]. Select for the noise filter's clock source. 0 0 0: The base clock signal divided by 1. 0 0 1: The on-chip baud rate generator source clock divided by 1. 0 1 0: The on-chip baud rate generator source clock divided by 2. 0 1 1: The on-chip baud rate generator source clock divided by 4. 1 0 0: The on-chip baud rate generator source clock divided by 8. others: Setting prohibited.	R/W
27	—	These bits are read as 0. The write value should be 0.	R/W
28	NFEN	Digital Noise Filter Function Enable Valid in Asynchronous mode, manchester mode, Simple LIN mode and Simple IIC mode 0: In Asynchronous, Manchester, Simple LIN mode: Disable noise cancellation function for RXDn input signal In Simple IIC mode: Disable noise cancellation function for SCLn and SDAn input signals 1: In Asynchronous, Manchester, Simple LIN mode: Enable noise cancellation function for RXDn input signal In Simple IIC mode: Enable noise cancellation function for SCLn and SDAn input signals	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Use this bit in asynchronous mode and manchester mode only. Operation by other mode is not guaranteed.
Note 2. RINV/TINV should be set 0 in smart card interface mode and simple IIC mode.

CTSE bit (CTS Enable)

Set this bit to 1 if the SSn pin is to be used for inputting of the CTSn control signal to control of transmission and reception. The RTSn signal is output when this bit is set to 0. Set this bit to 0 in smart card interface mode, simple SPI mode, Simple LIN mode, and simple IIC mode. Do not set both the CTSE and SSE bits to enabled (even if this setting is made, operation is the same as that when these bits are set to 0).

CTSPEN bit (CTS external pin Enable)

When CTSE is 1, the CTSPEN bit selects the pins usage method when using the CTS and RTS functions. Set this bit to 1 when assigning the CTS/RTS function to 2 pins and using them at the same time. Set it to 0 except in Asynchronous and Manchester modes.

Table 26.3 shows the relationship between the CTSE bit and CTSPEN bit settings and the functions of the CTSn_RTSn pin and CTSn pin.

Table 26.3 CTSE bit and CTSPEN bit settings and pin functions

CTSE bit	CTSPEN bit	CTSn_RTSn pin	CTSn pin
0	0	RTSn signal output	Not use
1	0	CTSn signal input	Not use
1	1	RTSn signal output	CTSn signal input

Note: Set CTSPEN bit = 0 when CTSE bit = 0.

SPB2DT bit (Serial port break data select), SPB2IO bit (Serial port break I/O)

The TXDn pin status decided by combination of CCR0.TE bit, CCR1.SPB2IO bit and CCR1.SPB2DT bit is indicated in Table 26.4.

Table 26.4 TXDn pin status

The value of CCR0.TE	The value of CCR1.SPB2IO	The value of CCR1.SPB2DT	TXDn pin status (When TINV is 0.)
0	0	—	Hi-Z (initial value)
0	1	0	Low level output
0	1	1	High level output
1	—	—	Serial transmission data is output.

Note: —: Don't care

PE bit (Parity Enable)

When the PE bit is 1, the parity bit is added to transmit data before transmission, and the parity bit is checked in reception.

In the multiprocessor format, the parity bit is not added or checked regardless of this bit setting.

PM bit (Parity Mode)

Selects the parity mode for transmission and reception (even or odd). In multi-processor mode, this bit is invalid.

For details on the usage of this bit in smart card interface mode, see section 26.7.2. Data Format (Except in Block Transfer Mode).

TINV bit (TXD invert), RINV bit (RXD invert)

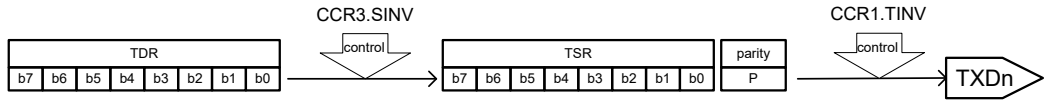
The data of RDR is controlled by RINV and CCR3.SINV. The data from TXDn pin is controlled by TINV and CCR3.SINV. The control by RINV/TINV are done to communication pins (RXDn / TXDn), so they can control not only data-bits but also other bits (start bit, stop bit, parity bit). For details, see Figure 26.3.

During half-duplex communication and slave operation in simple SPI mode, use the TXDn pin for reception, so set the inversion control of the received data with the TINV bit.

Note: Sentences and a timing chart of the IP operation explanation are mentioned by TINV = 0 and RINV = 0 when TINV's value and RINV's value are not specified.

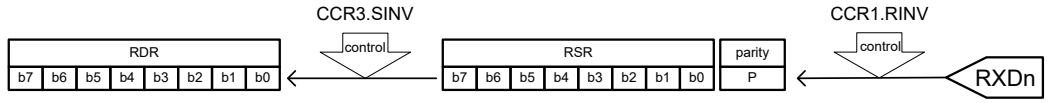
The receive/transmit data control (Data size = 8bits, Even parity, MSB first)

The transmit data is controlled by CCR1.TINV and CCR3.SINV.



CCR3. SINV	CCR1. TINV	TDR	TSR	parity (even)	TXDn waveform
0	0	0xBE	0xBE	0	S b7 b6 b5 b4 b3 b2 b1 b0 P
0	1	0xBE	0xBE	0	S b7 b6 b5 b4 b3 b2 b1 b0 P
1	0	0xBE	0x41	0	S b7 b6 b5 b4 b3 b2 b1 b0 P
1	1	0xBE	0x41	0	S b7 b6 b5 b4 b3 b2 b1 b0 P

The received data is controlled by CCR1.RINV and CCR3.SINV.



CCR3. SINV	CCR1. TINV	RDR	RSR	parity (even)	RXDn waveform
0	0	0xBE	0xBE	0	S b7 b6 b5 b4 b3 b2 b1 b0 P
1	0	0x41	0xBE	0	
0	1	0xBE	0xBE	0	S b7 b6 b5 b4 b3 b2 b1 b0 P
1	1	0x41	0xBE	0	

Figure 26.3 Example of the receive or transmit data control

SPLP bit (Loopback Control)

When this bit is 1, SCI blocks the input path from RXDn and connects the output path to TXDn to the reception data register.

Transmit data can be inverted and received by combining it with TINV bit.

Clock synchronous mode at slave operation, Asynchronous mode when using an external clock, and Simple LIN mode, set this bit to 0.

SHARPS bit (Half-duplex communication select)

Setting this bit to 1 enables half-duplex communication using the TXDn pin. However, it cannot be used in Simple SPI mode, Simple IIC mode and Smart Card Interface mode.

If this bit is set to 1 and CCR0.TE = 1, CCR0.RE = 0, the TXDn pin becomes communication output. If this bit is set to 1 and CCR0.TE = 0, CCR0.RE = 1, the TXDn pin becomes the communication input. For details, see section 26.18. Half-Duplex communication Function.

NFCS[2:0] bits (Noise Filter Clock Select)

These bits select the sampling clock for the digital noise filter.

To use the noise filter in asynchronous mode, manchester mode and Simple LIN mode set these bits from 000b to 100b. In simple IIC mode, set the bits to a value in the range from 001b to 100b.

NFEN bit (Digital Noise Filter Function Enable)

This bit enables or disables the digital noise filter function. When the function is enabled, noise cancellation is applied to the RXDn input signal in asynchronous mode, manchester mode, Simple LIN mode, and noise cancellation is applied to the SDAn and SCLn input signals in simple IIC mode. In any mode other than above, set the NFEN bit to 0 to disable the digital noise filter function. When the function is disabled, input signals are transferred as is, as internal signals.

26.2.7 CCR2 : Common Control Register 2

Base address: SCI_Bn = 0x4011_8000 + 0x0100 × n (n = 0 to 4, 9)

Offset address: 0x10

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	MDDR[7:0]								—	—	CKS[1:0]		—	—	—	BRME
Value after reset:	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
2:0	BCP[2:0]	Base Clock Pulse Selects the number of base clock cycles in smart card interface mode. 0 0 0: 93 clock cycles (S = 93)*1 0 0 1: 128 clock cycles (S = 128)*1 0 1 0: 186 clock cycles (S = 186)*1 0 1 1: 512 clock cycles (S = 512)*1 1 0 0: 32 clock cycles (S = 32)*1 (Initial value) 1 0 1: 64 clock cycles (S = 64)*1 1 1 0: 372 clock cycles (S = 372)*1 1 1 1: 256 clock cycles (S = 256)*1	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	BGDM	Baud Rate Generator Double-Speed Mode Select Valid in asynchronous/Manchester/clock-synchronous/Simple SPI mode and CCR3.CKE[1] = 0. 0: Baud rate generator outputs the clock with single frequency. 1: Baud rate generator outputs the clock with doubled frequency.	R/W
5	ABCS	Asynchronous Mode Base Clock Select Valid only in Asynchronous mode, Manchester mode and Simple LIN mode 0: Selects 16 base clock cycles for 1-bit period. 1: Selects 8 base clock cycles for 1-bit period.	R/W
6	ABCSE	Asynchronous Mode Extended Base Clock Select Valid only in Asynchronous mode and CCR3.CKE[1] = 0 0: Clock cycles for 1-bit period is decided with combination be-tween CCR2.BGDM and CCR2.ABCS. 1: Baud rate is 6 base clock cycles for 1-bit period and the clock of a double frequency is output from the baud rate generator.	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
15:8	BRR[7:0]	Bit rate setting BRR is an 8-bit register that adjusts the bit rate.	R/W

Bit	Symbol	Function	R/W
16	BRME	Bit Modulation Enable 0: Bit rate modulation function is disabled. 1: Bit rate modulation function is enabled.	R/W
19:17	—	These bits are read as 0. The write value should be 0.	R/W
21:20	CKS[1:0]	Clock Select 0 0: TCLK clock (n = 0)*2 0 1: TCLK/4 clock (n = 1)*2 1 0: TCLK/16 clock (n = 2)*2 1 1: TCLK/64 clock (n = 3)*2	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W
31:24	MDDR[7:0]	Modulation Duty Setting MDDR corrects the bit rate adjusted by the BRR[7:0] bits.	R/W

Note 1. S is the value of S in BRR[7:0] bits explanation.
Note 2. n is the decimal notation of the value of n in BRR[7:0] bits explanation.

BCP[2:0] bit (Base Clock Pulse)

These bits select the number of base clock cycles in a 1-bit data transfer time in smart card interface mode.

For details, see section 26.7.4. Receive Data Sampling Timing and Reception Margin.

BGDM bit (Baud Rate Generator Double-Speed Mode Select)

This bit is valid when the on-chip baud rate generator is selected as the clock source (CCR3.CKE[1] = 0) in asynchronous mode, Manchester mode, clock synchronous mode, Simple SPI mode. When external clock is selected (CCR3.CKE[1] = 1), set it to 0. For the clock output from the baud rate generator, either single or doubled frequency can be selected. The base clock is generated by the clock output from the baud rate generator. When the BGDM bit is set to 1, the base clock cycle is halved, and the bit rate is doubled.

Set this bit to 0 in modes other than asynchronous mode or Manchester mode or clock synchronous mode or Simple SPI mode.

ABCS bit (Asynchronous Mode Base Clock Select)

Selects the clock cycles for 1-bit period.

Set it to 0 in modes other than Asynchronous mode, Manchester mode and Simple LIN mode.

ABCSE bit (Asynchronous Mode Extended Base Clock Select)

The pulse number for a base clock at 1-bit period is 6 and the clock of a double frequency is output from baud rate generator. Only when the bit rate is set to 6 dividing frequency of the bus clock, use this bit and set CCR2.CKS [1:0] = 00b and BRR[7:0] = 0x00.

Set it to 0 in modes other than asynchronous mode. Even in asynchronous mode, set it to 0 when using external clock.

Table 26.5 Base clock cycle number per 1-bit

ABCSE	ABCS	BGDM	The base clock cycles /1bit	The frequency of the baud rate generator
0	0	0	16	×1
0	0	1	16	×2
0	1	0	8	×1
0	1	1	8	×2
1	— (don't care)	— (don't care)	6	×2

BRR[7:0] bit (Bit rate setting)

BRR is an 8-bit register that adjusts the bit rate.

SCI has independent baud rate generator control, different bit rates can be set for each. Table 26.6 shows the relation-ship between the setting (N) in the BRR and the bit rate (B) for asynchronous mode, multiprocessor transfer, Manchester mode, clock synchronous mode, smart card interface mode, simple SPI mode, and simple IIC mode.

Table 26.6 Relationship between N Setting in BRR and Bit Rate B

Mode	CCR2 settings			BRR[7:0] setting	Error
	BGDM bit	ABCS bit	ABCS E bit		
Asynchronous, multi-processor, Manchester, Simple-LIN*3	0	0	0	$N = \frac{TCLK \times 10^6}{64 \times 2^{2n-1} \times B} - 1$	$Error\ (\%) = \left\{\frac{TCLK \times 10^6}{B \times 64 \times 2^{2n-1} \times (N+1)} - 1\right\} \times 100$
	1	0	0	$N = \frac{TCLK \times 10^6}{32 \times 2^{2n-1} \times B} - 1$	$Error\ (\%) = \left\{\frac{TCLK \times 10^6}{B \times 32 \times 2^{2n-1} \times (N+1)} - 1\right\} \times 100$
	0	1	0		
	1	1	0	$N = \frac{TCLK \times 10^6}{16 \times 2^{2n-1} \times B} - 1$	$Error\ (\%) = \left\{\frac{TCLK \times 10^6}{B \times 16 \times 2^{2n-1} \times (N+1)} - 1\right\} \times 100$
	Don't care	Don't care	1*2	$N = \frac{TCLK \times 10^6}{12 \times 2^{2n-1} \times B} - 1$	$Error\ (\%) = \left\{\frac{TCLK \times 10^6}{B \times 12 \times 2^{2n-1} \times (N+1)} - 1\right\} \times 100$
Clock synchronous, simple SPI				$N = \frac{TCLK \times 10^6}{8 \times 2^{2n-1} \times B} - 1$	—
Smart card interface				$N = \frac{TCLK \times 10^6}{S \times 2^{2n+1} \times B} - 1$	$Error\ (\%) = \left\{\frac{TCLK \times 10^6}{B \times S \times 2^{2n+1} \times (N+1)} - 1\right\} \times 100$
Simple IIC*1				$N = \frac{TCLK \times 10^6}{64 \times 2^{2n-1} \times B} - 1$	—

Note: B: Bit rate (bps)
N: BRR setting for on-chip baud rate generator (0 ≤ N ≤ 255)
TCLK: Operating frequency (MHz)
n and S: Determined by the settings of the CCR2 registers as listed in the Table 26.8 and Table 26.9. Please be careful about $2^{(2n+1)}$ is used in the expression for Smart card interface, $2^{(2n-1)}$ is used in other mode.
Note 1. Adjust the bit rate so that the widths at high and low level of the SCL output in simple IIC mode satisfy the IIC standard.
Note 2. In Manchester mode, only ABCSE = 0 can be selected.
Note 3. In Simple LIN mode, BGDM = 0 and ABCSE = 0 can be selected.

Table 26.7 Calculating Widths at High and Low Level for SCL

Mode	SCLn	Formula (result in seconds)
IIC	Width at high level (minimum value)	$(N+1) \times 4 \times 2^{2n-1} \times 7 \times \frac{1}{TCLK \times 10^6}$
	Width at low level (minimum value)	$(N+1) \times 4 \times 2^{2n-1} \times 8 \times \frac{1}{TCLK \times 10^6}$

Table 26.8 Clock Source Settings

CCR2 setting		
CKS[1:0] bit	Clock source	n
00	TCLK clock	0
01	TCLK/4 clock	1
10	TCLK/16 clock	2
11	TCLK/64 clock	3

Table 26.9 Base Clock Settings in Smart Card Interface Mode (1 of 2)

CCR2 setting		
BCP[2:0] setting	Base clock cycles for 1-bit period	S
0 0 0	93 clock cycles	93
0 0 1	128 clock cycles	128
0 1 0	186 clock cycles	186

Table 26.9 Base Clock Settings in Smart Card Interface Mode (2 of 2)

CCR2 setting BCP[2:0] setting	Base clock cycles for 1-bit period	S
0 1 1	512 clock cycles	512
1 0 0	32 clock cycles	32
1 0 1	64 clock cycles	64
1 1 0	372 clock cycles	372
1 1 1	256 clock cycles	256

Table 26.10 and Table 26.11 list examples of N settings in BRR in asynchronous mode and Manchester mode. Table 26.12 lists the maximum bit rate settable for each operating frequency. Examples of BRR (N) settings in clock synchronous mode and simple SPI mode are listed in Table 26.15. Examples of BRR (N) settings in smart card interface mode are listed in Table 26.17. Examples of BRR (N) settings in simple IIC mode are listed in Table 26.19. In smart card interface mode, the number of base clock cycles S in a 1-bit data transfer time can be selected. For details, see section 26.7.4. Receive Data Sampling Timing and Reception Margin. Table 26.14 and Table 26.16 list the maximum bit rates with external clock input.

When either the asynchronous mode base clock select bit (ABCS) or the baud rate generator double-speed mode select bit (BGDM) is set to 1 in asynchronous mode and manchester mode, the bit rate becomes twice that listed in Table 26.10 and Table 26.11. When both of those registers are set to 1, the bit rate becomes four times the listed value.

Table 26.10 Examples of BRR Settings for various Bit Rates (Asynchronous Mode and Manchester Mode) (1) (1 of 3)

Bit rate (bps)	Operating Frequency TCLK (MHz)														
	8			9.8304			10			12			12.288		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	141	0.03	2	174	–0.26	2	177	–0.25	2	212	0.03	2	217	0.08
150	2	103	0.16	2	127	0	2	129	0.16	2	155	0.16	2	159	0
300	1	207	0.16	1	255	0	2	64	0.16	2	77	0.16	2	79	0
600	1	103	0.16	1	127	0	1	129	0.16	1	155	0.16	1	159	0
1200	0	207	0.16	0	255	0	1	64	0.16	1	77	0.16	1	79	0
2400	0	103	0.16	0	127	0	0	129	0.16	0	155	0.16	0	159	0
4800	0	51	0.16	0	63	0	0	64	0.16	0	77	0.16	0	79	0
9600	0	25	0.16	0	31	0	0	32	–1.36	0	38	0.16	0	39	0
19200	0	12	0.16	0	15	0	0	15	1.73	0	19	–2.34	0	19	0
31250	0	7	0	0	9	–1.7	0	9	0	0	11	0	0	11	2.4
38400	—	—	—	0	7	0	0	7	1.73	0	9	–2.34	0	9	0

Table 26.10 Examples of BRR Settings for various Bit Rates (Asynchronous Mode and Manchester Mode) (1) (2 of 3)

Bit rate (bps)	Operating Frequency TCLK (MHz)														
	14			16			17.2032			18			19.6608		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	248	–0.17	3	70	0.03	3	75	0.48	3	79	–0.12	3	86	0.31
150	2	181	0.16	2	207	0.16	2	223	0	2	233	0.16	2	255	0
300	2	90	0.16	2	103	0.16	2	111	0	2	116	0.16	2	127	0
600	1	181	0.16	1	207	0.16	1	223	0	1	233	0.16	1	255	0
1200	1	90	0.16	1	103	0.16	1	111	0	1	116	0.16	1	127	0
2400	0	181	0.16	0	207	0.16	0	223	0	0	233	0.16	0	255	0
4800	0	90	0.16	0	103	0.16	0	111	0	0	116	0.16	0	127	0

Table 26.10 Examples of BRR Settings for various Bit Rates (Asynchronous Mode and Manchester Mode) (1) (3 of 3)

Bit rate (bps)	Operating Frequency TCLK (MHz)														
	14			16			17.2032			18			19.6608		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
9600	0	45	–0.93	0	51	0.16	0	55	0	0	58	–0.69	0	63	0
19200	0	22	–0.93	0	25	0.16	0	27	0	0	28	1.02	0	31	0
31250	0	13	0	0	15	0	0	16	1.2	0	17	0	0	19	–1.7
38400	—	—	—	0	12	0.16	0	13	0	0	14	–2.34	0	15	0

Note: This is an example when the CCR2.ABCS = 0, CCR2.BGDM = 0 and CCR2.ABCSE = 0.
When either ABCS bit or BGDM bit is set to 1, the bit rate doubles.
When both ABCS = 1 and BGDM = 1, the bit rate increases four times.

Table 26.11 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode and Manchester Mode) (2) (1 of 2)

Bit rate (bps)	Operating frequency TCLK (MHz)														
	20			25			30			33			40		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	88	–0.25	3	110	–0.02	3	132	0.13	3	145	0.33	3	177	–0.25
150	3	64	0.16	3	80	0.47	3	97	–0.35	3	106	0.39	3	129	0.16
300	2	129	0.16	2	162	–0.15	2	194	0.16	2	214	–0.07	3	64	0.16
600	2	64	0.16	2	80	0.47	2	97	–0.35	2	106	0.39	2	129	0.16
1200	1	129	0.16	1	162	–0.15	1	194	0.16	1	214	–0.07	2	64	0.16
2400	1	64	0.16	1	80	0.47	1	97	–0.35	1	106	0.39	1	129	0.16
4800	0	129	0.16	0	162	–0.15	0	194	0.16	0	214	–0.07	1	64	0.16
9600	0	64	0.16	0	80	0.47	0	97	–0.35	0	106	0.39	0	129	0.16
19200	0	32	–1.36	0	40	–0.76	0	48	–0.35	0	53	–0.54	0	64	0.16
31250	0	19	0	0	24	0	0	29	0	0	32	0	0	39	0
38400	0	15	1.73	0	19	1.73	0	23	1.73	0	26	–0.54	0	32	–1.36

Table 26.11 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode and Manchester Mode) (2) (2 of 2)

Bit rate (bps)	Operating frequency TCLK (MHz)														
	50			60			100			120			160		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	221	–0.02	—	—	—	—	—	—	—	—	—	—	—	—
150	3	162	–0.15	3	194	0.16	—	—	—	—	—	—	—	—	—
300	3	80	0.47	3	97	–0.35	3	162	–0.15	3	194	0.16	—	—	—
600	2	162	–0.15	2	194	0.16	3	80	0.47	3	97	–0.35	3	129	0.16
1200	2	80	0.47	2	97	–0.35	2	162	–0.15	2	194	0.16	3	64	0.16
2400	1	162	–0.15	1	194	0.16	2	80	0.47	2	97	–0.35	2	129	0.16
4800	1	80	0.47	1	97	–0.35	1	162	–0.15	1	194	0.16	2	64	0.16
9600	0	162	–0.15	0	194	0.16	1	80	0.47	1	97	–0.35	1	129	0.16
19200	0	80	0.47	0	97	–0.35	0	162	–0.15	0	194	0.16	1	64	0.16
31250	0	49	0	0	59	0	1	24	0	0	119	0	0	159	0
38400	0	40	–0.76	0	48	–0.35	0	80	0.47	0	97	–0.35	0	129	0.16

Note: This is an example when the CCR2.ABCS = 0, CCR2.BGDM = 0 and CCR2.ABCSE = 0.

When either ABCS bit or BGDM bit is set to 1, the bit rate doubles.
When both ABCS = 1 and BGDM = 1, the bit rate increases four times.

Table 26.12 Maximum Bit Rate for Each Operating Frequency (Asynchronous Mode and Manchester mode) (1)

TCLK (MHz)	CCR2 settings					Maximum bit rate (bps)	TCLK (MHz)	CCR2 settings					Maximum bit rate (bps)
	BGDM	ABCS	ABCSE	n	N			BGDM	ABCS	ABCSE	n	N	
8	0	0	0	0	0	250000	16	0	0	0	0	0	500000
		1	0	0	0	500000			1	0	0	0	1000000
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1000000			1	0	0	0	2000000
	Don't care	Don't care	1	0	0	1333333		Don't care	Don't care	1	0	0	2666666
9.8304	0	0	0	0	0	307200	17.2032	0	0	0	0	0	537600
		1	0	0	0	614400			1	0	0	0	1075200
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1228800			1	0	0	0	2150400
	Don't care	Don't care	1	0	0	1638400		Don't care	Don't care	1	0	0	2867200
10	0	0	0	0	0	312500	18	0	0	0	0	0	562500
		1	0	0	0	625000			1	0	0	0	1125000
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1250000			1	0	0	0	2250000
	Don't care	Don't care	1	0	0	1666666		Don't care	Don't care	1	0	0	3000000
12	0	0	0	0	0	375000	19.6608	0	0	0	0	0	614400
		1	0	0	0	750000			1	0	0	0	1228800
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1500000			1	0	0	0	2457600
	Don't care	Don't care	1	0	0	2000000		Don't care	Don't care	1	0	0	3276800
12.288	0	0	0	0	0	384000	20	0	0	0	0	0	625000
		1	0	0	0	768000			1	0	0	0	1250000
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1536000			1	0	0	0	2500000
	Don't care	Don't care	1	0	0	2048000		Don't care	Don't care	1	0	0	3333333
14	0	0	0	0	0	437500	25	0	0	0	0	0	781250
		1	0	0	0	875000			1	0	0	0	1562500
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1750000			1	0	0	0	3125000
	Don't care	Don't care	1	0	0	2333333		Don't care	Don't care	1	0	0	4166666

Table 26.13 Maximum Bit Rate for Each Operating Frequency (Asynchronous Mode and Manchester mode) (2)

TCLK (MHz)	CCR2 settings					Maximum bit rate (bps)	TCLK (MHz)	CCR2 settings					Maximum bit rate (bps)
	BGDM	ABCS	ABCSE	n	N			BGDM	ABCS	ABCSE	n	N	
30	0	0	0	0	0	937500	50	0	0	0	0	0	1562500
		1	0	0	0	1875000			1	0	0	0	3125000
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	3750000			1	0	0	0	6250000
	Don't care	Don't care	1	0	0	5000000		Don't care	Don't care	1	0	0	8333333
33	0	0	0	0	0	1031250	60	0	0	0	0	0	1875000
		1	0	0	0	2062500			1	0	0	0	3750000
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	4125000			1	0	0	0	7500000
	Don't care	Don't care	1	0	0	5500000		Don't care	Don't care	1	0	0	10000000
40	0	0	0	0	0	1250000	120	0	0	0	0	0	3750000
		1	0	0	0	2500000			1	0	0	0	7500000
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	5000000			1	0	0	0	15000000
	Don't care	Don't care	1	0	0	6666666		Don't care	Don't care	1	0	0	20000000
							160	0	0	0	0	0	5000000
									1	0	0	0	10000000
								1	0	0	0	0	
									1	0	0	0	20000000
								Don't care	Don't care	1	0	0	26666666

Table 26.14 Maximum Bit Rate with External Clock Input (Asynchronous Mode)

TCLK(MHz)	External clock (MHz)	Maximum bit rate (bps)		TCLK(MHz)	External clock (MHz)	Maximum bit rate (bps)	
		CCR2.ABCS bit = 0	CCR2.ABCS bit = 1			CCR2.ABCS bit = 0	CCR2.ABCS bit = 1
8	2	125000	250000	25	6.25	390625	781250
9.8304	2.4576	153600	307200	30	7.5	468750	937500
10	2.5	156250	312500	33	8.25	515625	1031250
12	3	187500	375000	40	10	625000	1250000
12.288	3.072	192000	384000	50	12.5	781250	1562500
14	3.5	218750	437500	60	15	937500	1875000
16	4	250000	500000	120	30	1875000	3750000
17.2032	4.3008	268800	537600	160	40	2500000	5000000
18	4.5	281250	562500				
19.6608	4.9152	307200	614400				
20	5	312500	625000				

Table 26.15 BRR Settings for Various Bit Rates (Clock Synchronous Mode, Simple SPI Mode)

Bit rate (bps)	Operating frequency TCLK (MHz)																	
	8			10			30			60			120			160		
	BGDM	n	N	BGDM	n	N	BGDM	n	N	BGDM	n	N	BGDM	n	N	BGDM	n	N
250	0	3	124	0	3	177	—	—	—	—	—	—	—	—	—	—	—	—
500	0	2	249	0	3	77	0	3	233	—	—	—	—	—	—	—	—	—
1k	0	2	124	0	3	38	0	3	116	0	3	233	—	—	—	—	—	—
2.5k	0	2	49	0	1	249	0	3	46	0	3	93	0	3	187	0	3	249
5k	0	2	24	0	1	124	0	2	93	0	3	46	0	3	93	0	3	124
10k	0	1	49	0	0	249	0	2	46	0	2	93	0	3	46	0	2	249
25k	0	2	4	0	1	24	0	1	74	0	1	149	0	2	74	0	3	24
50k	0	1	9	0	0	49	0	0	149	0	1	74	0	1	149	0	2	49
100k	0	1	4	0	0	24	0	0	74	0	0	149	0	1	74	0	2	24
250k	0	1	1	0	0	9	0	0	29	0	1	14	0	1	29	0	2	9
500k	0	1	0	0	0	4	0	0	14	0	0	29	0	1	14	0	2	4
1M	0	0	1	1	0	4	1	0	14	0	0	14	0	0	29	0	1	9
2.5M	—	—	—	0	0	0	0	0	2	0	0	5	0	1	2	0	2	0
5M	—	—	—	1	0	0	1	0	2	0	0	2	0	0	5	0	1	1
7.5M	—	—	—	—	—	—	0	0	0	0	0	1	1	1	0	1	0	10
60M	—	—	—	—	—	—	—	—	—	—	—	—	1	0	0	—	—	—

Note: —: Can be set, but an error over 10% will occur.

Table 26.16 Maximum Bit Rate with External Clock Input (Clock Synchronous Mode, Simple SPI Mode)

TCLK (MHz)	External clock (MHz)	MAX Bit rate (Mbps)	TCLK (MHz)	External clock (MHz)	MAX Bit rate (Mbps)
8	4	4	25	12.5	12.5
10	5	5	30	15	15
12	6	6	33	16.5	16.5
14	7	7	40	20	20
16	8	8	50	25	25
18	9	9	60	30	30
20	10	10	120	60	60

Table 26.17 BRR Settings for Various Bit Rates (Smart Card Interface Mode, n = 0, S = 372) (1 of 4)

Bit rate (bps)	Operating frequency TCLK (MHz)											
	7.1424			10.00			10.7136			13.00		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
9600	0	0	0.00	0	1	–30	0	1	–25	0	1	–8.99

Table 26.17 BRR Settings for Various Bit Rates (Smart Card Interface Mode, n = 0, S = 372) (2 of 4)

Bit rate (bps)	Operating frequency TCLK (MHz)											
	14.2848			16.00			18.00			20.00		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
9600	0	1	0.00	0	1	12.01	0	2	–15.99	0	2	–6.66

Table 26.17 BRR Settings for Various Bit Rates (Smart Card Interface Mode, n = 0, S = 372) (3 of 4)

Bit rate (bps)	Operating frequency TCLK (MHz)											
	25.00			30.00			33.00			40.00		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
9600	0	3	–12.49	0	3	5.01	0	4	–7.59	0	5	–6.66

Table 26.17 BRR Settings for Various Bit Rates (Smart Card Interface Mode, n = 0, S = 372) (4 of 4)

Bit rate (bps)	Operating frequency TCLK (MHz)											
	50.00			60.00			120.00			160.00		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
9600	0	6	0.01	0	7	5.01	0	16	–1.17	0	21	1.82

Table 26.18 Maximum Bit Rate for Each Operating Frequency (Smart Card Interface Mode, S = 32)

TCLK (MHz)	MAX Bit rate (bps)	n	N	TCLK (MHz)	MAX Bit rate (bps)	n	N
10	156250	0	0	30	468750	0	0
10.7136	167400	0	0	33	515625	0	0
13	203125	0	0	40	625000	0	0
16	250000	0	0	50	781250	0	0
18	281250	0	0	60	937500	0	0
20	312500	0	0	120	1875000	0	0
25	390625	0	0	160	2500000	0	0

Table 26.19 BRR Settings for Various Bit Rates (Simple IIC Mode) (1 of 4)

Bit rate (bps)	Operating frequency TCLK (MHz)											
	8			10			16			20		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
10k	0	24	0	0	31	–2.3	1	12	–3.8	1	15	–2.3
25k	0	9	0	0	12	–3.8	1	4	0	1	6	–10.7
50k	0	4	0	0	6	–10.7	1	2	–16.7	1	3	–21.9
100k	0	2	–16.7	0	3	–21.9	0	4	0	0	6	–10.7
250k	0	0	0	0	1	–37.5	0	1	0	0	2	–16.7
350k										0	1	–10.7
400k										0	1	–21.9

Table 26.19 BRR Settings for Various Bit Rates (Simple IIC Mode) (2 of 4)

Bit rate (bps)	Operating frequency TCLK (MHz)											
	25			30			33			40		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
10k	1	19	–2.3	1	23	–2.3	1	25	–0.8	0	124	0
25k	1	7	–2.3	1	9	–6.3	1	10	–6.3	0	40	0
50k	1	3	–2.3	1	4	–6.3	1	5	–14.1	0	24	0
100k	1	1	–2.3	1	2	–21.9	1	2	–14.1	0	12	–3.85

Table 26.19 BRR Settings for Various Bit Rates (Simple IIC Mode) (3 of 4)

Bit rate (bps)	Operating frequency TCLK (MHz)											
	25			30			33			40		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
250k	0	3	–21.9	0	3	–6.3	0	4	–17.5	0	4	0
350k	0	2	–25.6	0	2	–10.7	0	2	–1.8	0	3	–10.71
400k	0	1	–2.3	0	1	17.2	0	2	–14.1	0	2	4.17

Table 26.19 BRR Settings for Various Bit Rates (Simple IIC Mode) (4 of 4)

Bit rate (bps)	Operating frequency TCLK (MHz)											
	50			60			120			160		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
10k	2	9	–2.3	1	46	–0.27	1	93	–0.27	1	124	0
25k	2	3	–2.3	0	74	0	0	149	0	0	199	0
50k	2	1	–2.3	0	37	–1.32	0	74	0	0	99	0
100k	1	3	–2.3	0	18	–1.32	0	37	–1.31	0	49	0
250k	0	6	–10.7	0	7	–6.25	0	14	0	0	19	0
350k	0	4	–10.7	0	4	7.14	0	10	–2.6	0	13	2.04
400k	0	3	–2.34	0	4	–6.25	0	8	4.17	0	12	–3.85

Table 26.20 Minimum Widths at High and Low Level for SCL at Various Bit Rates (Simple IIC Mode) (1 of 3)

Bit rate (bps)	Operating frequency TCLK (MHz)											
	8			10			16			20		
	n	N	Minimum width of High/Low Level (µs)	n	N	Minimum width of High/Low Level (µs)	n	N	Minimum width of High/Low Level (µs)	n	N	Minimum width of High/Low Level (µs)
10k	0	24	43.75 / 50.00	0	31	44.80 / 51.20	1	12	45.50 / 52.00	1	15	44.80 / 51.20
25k	0	9	17.50 / 20.00	0	12	18.20 / 20.80	1	4	17.50 / 20.00	1	6	19.60 / 22.40
50k	0	4	8.75 / 10.00	0	6	9.80 / 11.20	1	2	10.50 / 12.00	1	3	11.20 / 12.80
100k	0	2	5.25 / 6.00	0	3	5.60 / 6.40	0	4	4.37 / 5.00	0	6	4.90 / 5.60
250k	0	0	1.75 / 2.00	0	1	2.80 / 3.20	0	1	1.75 / 2.00	0	2	2.10 / 2.40
350k										0	1	1.40 / 1.60
400k										0	1	1.40 / 1.60

Table 26.20 Minimum Widths at High and Low Level for SCL at Various Bit Rates (Simple IIC Mode) (2 of 3)

Bit rate (bps)	Operating frequency TCLK (MHz)											
	25			30			33			40		
	n	N	Minimum width of High/Low Level (µs)	n	N	Minimum width of High/Low Level (µs)	n	N	Minimum width of High/Low Level (µs)	n	N	Minimum width of High/Low Level (µs)
10k	1	19	44.80 / 51.20	1	23	44.80 / 51.20	1	25	44.12 / 50.42	1	32	46.20 / 52.80
25k	1	7	17.92 / 20.48	1	9	18.66 / 21.33	1	10	18.66 / 21.33	1	12	18.20 / 20.80
50k	1	3	8.96 / 10.24	1	4	9.33 / 10.66	1	5	10.18 / 11.63	1	6	9.80 / 11.20
100k	1	1	4.48 / 5.12	1	2	5.60 / 6.40	1	2	5.09 / 5.81	0	13	4.90 / 5.60
250k	0	3	2.24 / 2.56	0	3	1.86 / 2.13	0	4	2.12 / 2.42	0	4	1.75 / 2.00
350k	0	2	1.68 / 1.92	0	2	1.40 / 1.60	0	2	1.27 / 1.45	0	3	1.40 / 1.60
400k	0	1	1.12 / 1.28	0	1	0.93 / 1.07	0	2	1.27 / 1.45	0	2	1.05 / 1.20

Table 26.20 Minimum Widths at High and Low Level for SCL at Various Bit Rates (Simple IIC Mode) (3 of 3)

Bit rate (bps)	Operating frequency TCLK (MHz)											
	50			60			120			160		
	n	N	Minimum width of High/Low Level (µs)	n	N	Minimum width of High/Low Level (µs)	n	N	Minimum width of High/Low Level (µs)	n	N	Minimum width of High/Low Level (µs)
10k	2	9	44.80 / 51.20	1	47	44.80 / 51.20	1	93	43.87 / 50.13	1	124	43.75 / 50
25k	2	3	17.92 / 20.48	0	74	17.50 / 20.00	0	149	17.50 / 20.00	0	199	17.50 / 20.00
50k	2	1	8.96 / 10.24	0	37	8.87 / 10.13	0	74	8.75 / 10.00	0	99	8.75 / 10.00
100k	1	3	4.48 / 5.12	0	18	4.43 / 5.07	0	37	4.43 / 5.07	0	49	4.38 / 5.00
250k	0	6	1.96 / 2.24	0	7	1.87 / 2.13	0	15	1.87 / 2.13	0	49	1.75 / 2.00
350k	0	4	1.40 / 1.60	0	5	1.40 / 1.60	0	10	1.28 / 1.47	0	19	1.23 / 1.40
400k	0	3	1.12 / 1.28	0	4	1.17 / 1.33	0	9	1.17 / 1.33	0	12	1.14 / 1.30

BRME bit (Bit Modulation Enable)

Enables and disables the bit rate modulation function. The bit rate generated by on-chip baud rate generator is evenly corrected when this function is enabled.

Set to 0 in Clock-synchronous mode, Simple SPI mode, Smart Card Interface mode, Manchester mode and Simple LIN mode.

CKS[1:0] bit (Clock Select)

These bits select the clock source for the on-chip baud rate generator.

For the relationship between the settings of these bits and the baud rate, see BRR[7:0] bits explanation.

MDDR[7:0] bit (Modulation Duty Setting)

When the BRME bit is set to 1, the bit rate generated by the on-chip baud rate generator is evenly corrected according to the settings of MDDR (M/256). The relationship between the MDDR setting (M) and the bit rate (B) is given in Table 26.21.

The initial value of MDDR is FFh. Bit 7 in this register is fixed to 1.

Table 26.21 Relationship between MDDR Setting (M) and Bit Rate (B) When Bit Rate Modulation Function is Used

Mode*1	CCR2 settings			BRR setting	Error
	BGDM bit	ABCS bit	ABCSE bit		
Asynchronous, Multiprocessor transger	0	0	0	$N = \frac{TCLK \times 10^6}{64 \times 2^{2n-1} \times (256/M) \times B} - 1$	$Error\ (\%) = \left\{\frac{TCLK \times 10^6}{B \times 64 \times 2^{2n-1} \times (256/M) \times (N+1)} - 1\right\} \times 100$
	1	0	0	$N = \frac{TCLK \times 10^6}{32 \times 2^{2n-1} \times (256/M) \times B} - 1$	$Error\ (\%) = \left\{\frac{TCLK \times 10^6}{B \times 32 \times 2^{2n-1} \times (256/M) \times (N+1)} - 1\right\} \times 100$
	0	1	0		
	1	1	0	$N = \frac{TCLK \times 10^6}{16 \times 2^{2n-1} \times (256/M) \times B} - 1$	$Error\ (\%) = \left\{\frac{TCLK \times 10^6}{B \times 16 \times 2^{2n-1} \times (256/M) \times (N+1)} - 1\right\} \times 100$
	x (Arbitrarily)	x (Arbitrarily)	1*2	$N = \frac{TCLK \times 10^6}{12 \times 2^{2n-1} \times (256/M) \times B} - 1$	$Error\ (\%) = \left\{\frac{TCLK \times 10^6}{B \times 12 \times 2^{2n-1} \times (256/M) \times (N+1)} - 1\right\} \times 100$
Simple IIC*2				$N = \frac{TCLK \times 10^6}{64 \times 2^{2n-1} \times (256/M) \times B} - 1$	

Note: B: Bit rate (bps)
M: MDDR setting (128 ≤ M ≤ 255)
N: BRR setting for bound rate generator (0 ≤ N ≤ 255)
TCLK: Operating frequency (MHz)
n: Determined by the settings of the CKS[1:0] as listed in Table 26.8.

Note 1. Do not use this function in Clock-synchronous mode, Simple SPI mode, Smart card Interface mode, Manchester mode and Simple LIN mode.

Note 2. Adjust the bit rate so that the widths at high and low level of the SCL output in simple IIC mode satisfy the I2C standard.

Table 26.22 and Table 26.23 list examples of N settings in BRR and M settings in MDDR in asynchronous mode.

Table 26.22 Examples of BRR and MDDR Settings for Various Bit Rates (Asynchronous Mode) (1) (1 of 3)

Bit rate (bps)	Operating frequency TCLK (MHz)														
	8					9.8304					10				
	n	N	M	BGDM bit	Error (%)	n	N	M	BGDM bit	Error (%)	n	N	M	BGDM bit	Error (%)
38400	0	5	236	0	0.03	0	7	(256) *1	0	0	0	10	173	1	–0.01
57600	0	3	236	0	0.03	0	4	240	0	0	0	4	236	0	0.03
115200	0	1	236	0	0.03	0	1	192	0	0	0	4	236	1	0.03
230400	0	0	236	0	0.03	0	0	192	0	0	0	1	189	1	0.14
460800	0	0	236	1	0.03	0	0	192	1	0	0	0	189	1	0.14

Table 26.22 Examples of BRR and MDDR Settings for Various Bit Rates (Asynchronous Mode) (1) (2 of 3)

Bit rate (bps)	Operating frequency TCLK (MHz)														
	12					12.288					14				
	n	N	M	BGDM bit	Error (%)	n	N	M	BGDM bit	Error (%)	n	N	M	BGDM bit	Error (%)
38400	0	8	236	0	0.03	0	9	(256) *1	0	0	0	16	191	1	0
57600	0	5	236	0	0.03	0	4	192	0	0	0	13	236	1	0.03
115200	0	2	236	0	0.03	0	4	192	1	0	0	6	236	1	0.03
230400	0	2	236	1	0.03	0	2	230	1	–0.17	0	2	202	1	–0.11
460800	0	0	157	1	–0.18	0	0	154	1	0.26	0	0	135	1	0.14

Table 26.22 Examples of BRR and MDDR Settings for Various Bit Rates (Asynchronous Mode) (1) (3 of 3)

Bit rate (bps)	Operating frequency TCLK (MHz)													
	16					17.2032					18				
	n	N	M	BGD M bit	Error (%)	n	N	M	BGD M bit	Error (%)	n	N	M	BGD M bit	Error (%)
38400	0	11	236	0	0.03	0	13	(256) *1	0	0	0	18	166	1	–0.01
57600	0	7	236	0	0.03	0	6	192	0	0	0	18	249	1	–0.01
115200	0	3	236	0	0.03	0	6	192	1	0	0	8	236	1	0.03
230400	0	1	236	0	0.03	0	3	219	1	–0.2	0	1	210	0	0.14
460800	0	1	236	1	0.03	0	1	219	1	–0.2	0	0	210	0	0.14

Note: This is an example when the CCR2.ABCS = 0 and CCR2.ABCSE = 0.
Note 1. It means that the bit rate modulation function is disable. (CCR2.BRME = 0, M = 256)

Table 26.23 Examples of BRR and MDDR Settings for Various Bit Rates (Asynchronous Mode) (2) (1 of 4)

Bit rate (bps)	Operating frequency TCLK (MHz)													
	19.6608					20					25				
	n	N	M	BGD M bit	Error (%)	n	N	M	BGD M bit	Error (%)	n	N	M	BGD M bit	Error (%)
38400	0	15	(256) *1	0	0	0	10	173	0	–0.01	0	11	151	0	0
57600	0	9	240	0	0	0	9	236	0	0.03	0	7	151	0	0
115200	0	4	240	0	0	0	4	236	0	0.03	0	3	151	0	0
230400	0	1	192	0	0	0	4	236	1	0.03	0	1	151	0	0
460800	0	0	192	0	0	0	0	189	0	0.14	0	0	151	0	0

Table 26.23 Examples of BRR and MDDR Settings for Various Bit Rates (Asynchronous Mode) (2) (2 of 4)

Bit rate (bps)	Operating frequency TCLK (MHz)													
	30					33					40				
	n	N	M	BGD M bit	Error (%)	n	N	M	BGD M bit	Error (%)	n	N	M	BGD M bit	Error (%)
38400	0	36	194	1	0.01	0	14	143	0	0.01	0	21	173	0	–0.01
57600	0	10	173	0	–0.01	0	9	143	0	0.01	0	38	230	1	–0.01
115200	0	10	173	1	–0.01	0	4	143	0	0.01	0	9	236	0	0.03
230400	0	6	220	1	–0.09	0	4	143	1	0.01	0	4	236	0	0.03
460800	0	3	252	1	0.14	0	1	229	0	0.1	0	4	236	1	0.03

Table 26.23 Examples of BRR and MDDR Settings for Various Bit Rates (Asynchronous Mode) (2) (3 of 4)

Bit rate (bps)	Operating frequency TCLK (MHz)													
	50					60					120				
	n	N	M	BGD M bit	Error (%)	n	N	M	BGD M bit	Error (%)	n	N	M	BGD M bit	Error (%)
38400	0	23	151	0	0	0	36	194	0	0.01	0	73	194	0	0.01
57600	0	15	151	0	0	0	21	173	0	–0.01	0	58	232	0	0
115200	0	7	151	0	0	0	10	173	0	–0.01	0	21	173	0	–0.01
230400	0	3	151	0	0	0	10	173	1	–0.01	0	10	173	0	–0.01
460800	0	1	151	0	0	0	6	220	1	–0.09	0	10	173	1	–0.01

Table 26.23 Examples of BRR and MDDR Settings for Various Bit Rates (Asynchronous Mode) (2) (4 of 4)

Bit rate (bps)	Operating frequency TCLK (MHz)				
	160				
	n	N	M	BGDM bit	Error (%)
38400	0	117	232	0	0
57600	0	58	174	0	0
115200	1	58	174	0	0
230400	1	38	230	0	–0.01
460800	0	9	236	0	0.03

Note: This is an example when the CCR2.ABCS = 0 and CCR2.ABCSE = 0.
Note 1. It means that the bit rate modulation function is disable. (CCR2.BRME = 0, M = 256)

26.2.8 CCR3 : Common Control Register 3

Base address: SCI_Bn = 0x4011_8000 + 0x0100 × n (n = 0 to 4, 9)

Offset address: 0x14

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	BLK	GM	—	—	CKE[1:0]		—	—	DEN	FM	MP	MOD[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RXDESEL	STP	SINV	LSBF	—	—	CHR[1:0]		BPEN	—	—	—	—	—	CPOL	CPHA
Value after reset:	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1	1

Bit	Symbol	Function	R/W
0	CPHA	Clock Phase Select Valid in Cock-synchronous mode and Simple SPI mode. Set this bit only when CCR0.TE = 0 and RE = 0. 0: Data is sampled at an odd edge and changes at an even edge. (Clock is delayed.) 1: Data changes at an odd edge and is sampled at an even edge. (Clock is not delayed)	R/W
1	CPOL	Clock Polarity Select Valid in Cock-synchronous mode and Simple SPI mode. Set this bit only when CCR0.TE = 0 and RE = 0. 0: SCKn in idle state is 0. 1: SCKn in idle state is 1.	R/W
6:2	—	These bits are read as 0. The write value should be 0.	R/W
7	BPEN	Synchronizer bypass enable This bit controls whether to bypass the synchronizer circuit between the bus clock and operation clock. 0: Synchronizer circuit is not bypassed. 1: Synchronizer circuit is bypassed.	R/W
9:8	CHR[1:0]	Character Length Valid in Asynchronous mode and Manchester mode*1 Select the data length for transmission and reception. 0 0: Transmit/receive in 9-bit data length 0 1: Transmit/receive in 9-bit data length 1 0: Transmit/receive in 8-bit data length (initial value) 1 1: Transmit/receive in 7-bit data length*2	R/W
11:10	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
12	LSBF	LSB First select Set this bit to 0 in simple IIC mode. Set this bit to 1 in simple LIN mode. 0: MSB first 1: LSB first	R/W
13	SINV	Transmitted/Received Data Invert Set this bit to 0 in Simple IIC mode. The level of communication pins (RXDn/TXDn) are controlled by combination of this bit and CCR1.TINV/RINV. For details, see Figure 26.3. 0: TDR contents are transmitted to TSR as they are. RSR contents are stored to RDR as they are. 1: TDR contents are inverted before being transmitted to TSR. RSR contents are inverted and stored to RDR.	R/W
14	STP	Stop Bit Length Valid in Asynchronous mode, Manchester mode, Simple LIN mode 0: 1 stop bit / Break Delimiter length is 1bit 1: 2 stop bits / Break Delimiter length is 2bits	R/W
15	RXDESEL	Asynchronous Start Bit Edge Detection Select Valid only in asynchronous mode Set this bit to 1 in simple LIN mode. 0: The low level on the RXDn pin is detected as the start bit. 1: A falling edge on the RXDn pin is detected as the start bit.	R/W
18:16	MOD[2:0]	Communication mode select Select the SCI communication mode. 0 0 0: Asynchronous mode (Multi-processor mode) 0 0 1: Smart card interface mode 0 1 0: Clock synchronous mode 0 1 1: Simple SPI mode 1 0 0: Simple IIC mode 1 0 1: Manchester mode 1 1 0: Simple LIN mode 1 1 1: Setting prohibited	R/W
19	MP	Multi-Processor Mode Valid in Asynchronous mode, Manchester mode 0: Multi-processor communications function is disabled 1: Multi-processor communications function is enabled	R/W
20	FM	FIFO Mode select Valid in Asynchronous mode (including multi-processor mode), Clock synchronous mode, Simple SPI mode 0: TDR register, RDR register is non-FIFO buffer configuration 1: TDR register, RDR register is FIFO buffer configuration	R/W
21	DEN	Driver enable 0: RS-485 Driver control function disable. 1: RS-485 Driver control function enable.	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
25:24	CKE[1:0]	Clock enable 0 0: In the case of Asynchronous mode On-chip baud rate generator The SCKn pin is available for use as an I/O port in accord with the I/O port settings. In the case of Manchester mode and Simple LIN mode On-chip baud rate generator The SCKn pin functions as I/O port. In the case of Clock synchronous mode, Simple SPI mode Internal clock (Master operation) The SCKn pin functions as the clock output pin. In the case of Smart card interface mode and CCR3.GM = 0 Output disabled (The SCKn pin is available for use as an I/O port in accord with the I/O port settings.) In the case of Smart card interface mode and CCR3.GM = 1 Output fixed low 0 1: In the case of Asynchronous mode On-chip baud rate generator The clock with the same frequency as the bit rate is output from the SCKn pin. In the case of Manchester mode and Simple LIN mode Prohibited In the case of Clock synchronous mode, Simple SPI mode Internal clock (Master operation) The SCKn pin functions as the clock output pin. In the case of Smart card interface mode and CCR3.GM = 0 Clock output In the case of Smart card interface mode and CCR3.GM = 1 Clock output 1 0: In the case of Asynchronous mode External clock ● When using the external clock 16 times the bit rate should be input from the SCKn pin when CCR2.ABCS bit is 0. Input a clock signal with a frequency 8 times the bit rate when the CCR2.ABCS bit is 1. In the case of Manchester mode and Simple LIN mode Prohibited In the case of Clock synchronous mode, Simple SPI mode External clock (Slave operation) The SCKn pin functions as the clock input pin. In the case of Smart card interface mode and CCR3.GM = 0 Prohibited In the case of Smart card interface mode and CCR3.GM = 1 Output fixed high 1 1: In the case of Asynchronous mode External clock ● When using the external clock 16 times the bit rate should be input from the SCKn pin when CCR2.ABCS bit is 0. Input a clock signal with a frequency 8 times the bit rate when the CCR2.ABCS bit is 1. In the case of Manchester mode and Simple LIN mode Prohibited In the case of Clock synchronous mode, Simple SPI mode External clock (Slave operation) The SCKn pin functions as the clock input pin. In the case of Smart card interface mode and CCR3.GM = 0 Prohibited In the case of Smart card interface mode and CCR3.GM = 1 Clock output	R/W
27:26	—	These bits are read as 0. The write value should be 0.	R/W
28	GM	GSM Mode Valid only in Smart card interface mode 0: Non-GSM mode operation 1: GSM mode operation	R/W

Bit	Symbol	Function	R/W
29	BLK	Block Transfer Mode Valid only in Smart card interface mode 0: Non-block transfer mode operation 1: Block transfer mode operation	R/W
31:30	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. In other than asynchronous mode and manchester mode, this bit setting is invalid and a fixed data length of 8 bits is used. In the Simple LIN mode, only the data length of 8 bits can be used, so set it to the initial value.
Note 2. LSB first is fixed and the MSB (bit 7) in TDR is not transmitted in transmission.

CPHA bit (Clock Phase Select)

This bit selects the phase of the clock signal output through the SCKn pin. For details, see Figure 26.97.

Set the bit to 1 in other than simple SPI mode and clock synchronous mode.

CPOL bit (Clock Polarity Select)

This bit selects the polarity of the clock signal output through the SCKn pin. For details, see Figure 26.97.

Set the bit to 1 in other than simple SPI mode and clock synchronous mode.

BPEN bit (Synchronizer bypass enable)

The synchronization circuit can be bypassed by this bit only when the same clock is input to the bus clock and the operation clock (TCLK). For details, see section 26.19. Synchronizer Bypass Function.

Note: For details of this bit setting, see section 26.20.17. Notes on CCR3.BPEN bit setting.

CHR[1:0] bit (Character Length)

Selects the data length for transmission and reception.

Except of asynchronous mode and manchester mode, a fixed data length of 8 bits is used.

LSBF bit (LSB First select)

Select whether to transmit/receive data in MSB first or LSB first.

SINV bit (Transmitted/Received Data Invert)

SINV can invert the transmit data-bit from TDR to TSR, and can invert the received data from RSR to RDR. This bit does not affect the logic level of the parity bit. To invert the parity bit, invert the CCR1.PM.

STP bit (Stop Bit Length)

Selects the stop bit length in transmission.

In reception, only the first stop bit is checked regardless of this bit setting. If the second stop bit is 0, it is treated as the start bit of the next transmit frame.

In addition, it is used as Break Delimiter length setting when sending Start Frame in simple LIN mode.

RXDESEL bit (Asynchronous Start Bit Edge Detection Select)

Selects the detection method of the start bit for reception in asynchronous mode. When a break occurs, data receiving operation depends on the settings of this bit. Set this bit to 1 when reception should be stopped while a break occurs or when reception should be started without retaining the RXDn pin input at high level for the period of one data frame or longer after completion of the break.

Set this bit to 1 in Simple LIN mode. Set this bit to 0 in modes except of asynchronous mode and Simple LIN mode.

MOD[2:0] bit (Communication mode select)

Selects the SCI communication mode.

Table 26.24 Relationship between communication mode selection bits (MOD[2:0]), other operation mode setting bits

Communication mode	Asynchronous								Smart Card Interface	Clock synchronous	Simple SPI				Simple IIC	Manchester		Simple LIN
CCR3.MOD[2:0]	000b								001b	010b	011b				100b	101b		110b
CCR3.MP	0				1				—	—	—				—	0	1	—
CCR3.FM	0		1		0		1		—	0	1	0		1	—	—		—
CCR3.DEN	0	1	0	1	0	1	0	1	—	—	—				—	—		—
CCR0.SSE	—								—		0	1	0	1	—	—		—

Note: — is Prohibited setting.

MP bit (Multi-Processor Mode)

Disables/enables the multi-processor communications function. The settings of the PE bit and PM bit are invalid in multi-processor mode.

FM bit (FIFO Mode select)

When the FM bit is set to 1, the TDR register / RDR register switches to FIFO configuration, and transmit FIFO (TDR register) / receive FIFO (RDR register) can be used for serial transmission / reception.

DEN bit (Driver enable)

Select RS-485 Driver control function disable or enable.

CKE[1:0] bit (Clock enable)

These bits select the clock source and SCKn pin function.

In smart card interface mode, these bits control the clock output from the SCKn pin.

In GSM mode, clock output can be dynamically switched. For details, see section 26.7.8. Clock Output Control.

GM bit (GSM Mode)

Setting this bit to 1 allows GSM mode operation.

In GSM mode, the CSR.TEND flag set timing is put forward to 11.0 etu (elementary time unit = 1-bit transfer time) from the start and the clock output control function is appended. For details, see section 26.7.6. Serial Data Transmission (Except in Block Transfer Mode), section 26.7.8. Clock Output Control.

BLK bit (Block Transfer Mode)

Setting this bit to 1 allows block transfer mode operation.

For details, see section 26.7.3. Block Transfer Mode.

26.2.9 CCR4 : Common Control Register 4

Base address: SCI_Bn = 0x4011_8000 + 0x0100 × n (n = 0 to 4, 9)

Offset address: 0x18

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	AET	ATT[2:0]			AJD	AST[2:0]			—	—	—	—	—	—	ATEN	ASEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
8:0	CMPD[8:0]	Compare Match Data Valid only in Asynchronous mode Set the compare data pattern for address match wake-up function	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W
16	ASEN	Adjust receive sampling timing enable Valid in Asynchronous mode using internal clock, Simple LIN mode using internal clock, Clock-synchronous mode operating as master, Simple SPI mode operating as master 0: Adjust sampling timing disable. 1: Adjust sampling timing enable.	R/W
17	ATEN	Adjust transmit timing enable Valid only in Asynchronous mode using internal clock 0: Adjust transmit timing disable. 1: Adjust transmit timing enable.	R/W
23:18	—	These bits are read as 0. The write value should be 0.	R/W
26:24	AST[2:0]	Adjustment value for receive Sampling Timing This bit enables only when ASEN is 1. in Asynchronous mode and Simple LIN mode using internal clock The sampling timing of RXDn pin is adjusted from the middle of bit by the following formula. Adjustment sampling timing = base clock × the setting value of AST[2:0]. In Clock-synchronous mode and Simple SPI mode using internal clock The RXDn sampling timing can be adjusted by delaying 1 to 4 TCLK 0 0 0: 1TCLK delay 0 0 1: 2TCLK delay 0 1 0: 3TCLK delay 0 1 1: 4TCLK delay Others: Setting prohibited	R/W
27	AJD	Adjustment Direction for receive sampling timing Valid in Asynchronous mode using internal clock, Simple LIN mode using internal clock This bit enables only when ASEN is 1. Adjustment direction for RXDn receive sampling timing is determined by this bit. For details, see section 26.3.10. The function of adjust receive sampling timing (Asynchronous Mode). 0: The sampling timing is adjusted backward to the middle of bit. 1: The sampling timing is adjusted forward to the middle of bit.	R/W
30:28	ATT[2:0]	Adjustment value for Transmit timing Valid in Asynchronous mode using internal clock, Simple LIN mode using internal clock This bit enables only when ATEN is 1. The selected edge timing of TXDn is adjusted by the following formula. Adjustment edge timing = base clock × the setting value of ATT[2:0] This setting timing is limited by setting the base clock cycles. For details, see section 26.3.11. The function of adjust transmit timing (Asynchronous Mode).	R/W
31	AET	Adjustment edge for transmit timing Valid in Asynchronous mode using internal clock, Simple LIN mode using internal clock The adjustable edge is set by this bit. This bit enables only when ATEN is 1. For details, see section 26.3.11. The function of adjust transmit timing (Asynchronous Mode). 0: When CCR1.TINV is 0, adjust the rising edge timing. When CCR1.TINV is 1, adjust the falling edge timing. 1: When CCR1.TINV is 0, adjust the falling edge timing. When CCR1.TINV is 1, adjust the rising edge timing.	R/W

CMPD[8:0] bit (Compare Match Data)

Set the comparison data for receive data when Address match function is enabled (CCR0.DCME = 1). CCR4.CMPD[8:0] should be written while CCR0.DCME is 0.

For the comparison data, it can select length from 3 types, they are CMPD[6:0] with 7bit length enable, CMPD[7:0] with 8bit, and CMPD[8:0] with 9bit length.

Note: If the description in this document and the timing chart do not specify the ASEN / ATEN setting value, it means that the reception sampling adjustment function / transmission timing adjustment function are OFF (CCR4.ASEN = 0, CCR4.ATEN = 0).

ASEN bit (Adjust receive sampling timing enable)

When this bit is 1, the receive sampling timing adjustment function is enabled. Control is different in Asynchronous mode, Simple LIN mode, Clock-synchronous mode, and Simple SPI mode.

For details of using internal clock in Asynchronous mode, see section 26.6.7. Reception Sampling Timing Adjustment Function in Clock Synchronous Mode with internal clock used. The operation when the Simple LIN mode internal clock is selected is the same as when the Asynchronous mode internal clock is selected.

For details of Clock-synchronous mode operating as master and Simple SPI mode operating as master, see section 26.6.6. Simultaneous Serial Data Transmission and Reception in Clock Synchronous Mode. Only the digital delay of the master mode receive sampling clock (MRCLK) can be controlled by this bit. MRCLK analog delay cannot be controlled.

ATEN bit (Adjust transmit timing enable)

When this bit is 1, the transmission timing adjustment function is enabled. The transmission timing adjustment function can adjust the edge timing of the waveform output from the TXDn pin. For details, see section 26.3.11. The function of adjust transmit timing (Asynchronous Mode).

AST[2:0] bit (Adjustment value for receive Sampling Timing)

When the ASEN bit is 1, the receive sampling timing can be adjusted according to this bit setting value.

In Asynchronous mode and Simple LIN mode using internal clock.

The sampling timing of RXDn pin is adjusted from the middle of bit by the following formula. This setting value is limited by setting the base clock cycles. For details, see section 26.3.10. The function of adjust receive sampling timing (Asynchronous Mode).

Adjustment sampling timing = base clock × the setting value of AST[2:0].

In Clock-synchronous mode and Simple SPI mode using internal clock

The sampling timing of RXDn pin can be adjusted by delaying 1 to 4 TCLK. For details, see section 26.6.7. Reception Sampling Timing Adjustment Function in Clock Synchronous Mode with internal clock used.

000: 1TCLK delay

001: 2TCLK delay

010: 3TCLK delay

011: 4TCLK delay

1xx: Setting prohibited

AJD bit (Adjustment Direction for receive sampling timing)

Set the RXDn pin sampling timing adjustment direction from the bit center to the rear or front. For details, see section 26.3.10. The function of adjust receive sampling timing (Asynchronous Mode).

ATT[2:0] bit (Adjustment value for Transmit timing)

The edge timing of the TXDn pin specified by the AET bit is adjusted by the base clock × ATT[2:0] setting value. The upper limit of the adjustment time that can be set is limited by the number of base clock cycles. For details, see section 26.3.11. The function of adjust transmit timing (Asynchronous Mode).

AET bit (Adjustment edge for transmit timing)

Set the TXDn pin edge for timing adjustment. For details, see section 26.3.11. The function of adjust transmit timing (Asynchronous Mode).

26.2.10 ICR : Simple IIC Control Register

Base address: SCI_Bn = 0x4011_8000 + 0x0100 × n (n = 0 to 4, 9)

Offset address: 0x20

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

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

Bit	Symbol	Function	R/W
4:0	IICDL[4:0]	SDA Delay Output Select Cycles below are of the clock signal from the on-chip baud rate generator. 0x00: No output delay 0x01: 0 to 1 cycle 0x02: 1 to 2 cycles 0x03: 2 to 3 cycles 0x04: 3 to 4 cycles 0x05: 4 to 5 cycles ⋮ 0x1E: 29 to 30 cycles 0x1F: 30 to 31 cycles	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
8	IICINTM	IIC Interrupt Mode Select 0: Use ACK/NACK interrupts. 1: Use reception and transmission interrupts	R/W
9	IICCSC	Clock Synchronization 0: No synchronization with the clock signal 1: Synchronization with the clock signal	R/W
12:10	—	These bits are read as 0. The write value should be 0.	R/W
13	IICACKT	ACK Transmission Data 0: ACK transmission 1: NACK transmission and reception of ACK/NACK	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W
16	IICSTAREQ	Start Condition Generation 0: A start condition is not generated 1: A start condition is generated. *1 *3 *4 *5	R/W
17	IICRSTAREQ	Restart Condition Generation 0: A restart condition is not generated. 1: A restart condition is generated. *2 *3 *4 *5	R/W
18	IICSTPREQ	Stop Condition Generation 0: A stop condition is not generated. 1: A stop condition is generated *2 *3 *4 *5	R/W
19	—	This bit is read as 0. The write value should be 0.	R/W
21:20	IICSDAS[1:0]	SDA Output Select 0 0: Serial data output 0 1: Generate a start, restart, or stop condition. 1 0: Output the low level on the SDAn pin. 1 1: Place the SDAn pin in the high-impedance state.	R/W

Bit	Symbol	Function	R/W
23:22	IICSCLS[1:0]	SCL Output Select 0 0: Serial clock output 0 1: Generate a start, restart, or stop condition. 1 0: Output the low level on the SCLn pin. 1 1: Place the SCLn pin in the high-impedance state.	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. In the bus free state, perform the start condition generation.
Note 2. In the bus busy state, perform restart or stop condition generation when the SCLn pin after acknowledgment described in Figure 26.84 and Figure 26.85 is low level.
Note 3. Do not set more than one from among the IICSTAREQ, IICRSTAREQ, and IICSTPREQ bits to 1 at a given time.
Note 4. Execute the generation of a condition after the value of the IICSTIF flag is 0.
Note 5. Do not write 0 to this bit while it is 1. Generation of a condition is suspended by writing 0 to this bit while it is 1.

IICDL[4:0] bit (SDA Delay Output Select)

These bits are used to set a delay for output on the SDAn pin relative to the falling edge of the output on the SCLn pin. The available delay settings range from no delay to 31 cycles, with the clock signal from the on-chip baud rate generators the base. The signal obtained by frequency-dividing TCLK by the divisor set in CCR2.CKS[1:0] is supplied as the clock signal from the on-chip baud rate generator.

Set these bits to 00000b unless operation is in simple IIC mode. In simple IIC mode, set the bits to a value in the range from 00001b to 11111b.

IICINTM bit (IIC Interrupt Mode Select)

This bit selects the sources of interrupt requests in Simple IIC mode.

IICCSC bit (Clock Synchronization)

Set the IICCSC bit to 1 if the internally generated SCLn clock signal is to be synchronized when the SCLn pin has been placed at the low level in the case of a wait inserted by the other device, etc.

The SCLn clock signal is not synchronized if the IICCSC bit is 0. The SCLn clock signal is generated in accord with the rate selected in the BRR regardless of the level being input on the SCLn pin.

Set the IICCSC bit to 1 except during debugging.

IICACKT bit (ACK Transmission Data)

Transmitted data contains ACK bits. Set this bit to 1 when ACK and NACK bits are received.

IICSTAREQ bit (Start Condition Generation)

When a start condition is to be generated, set both the IICSDAS[1:0] and IICSCLS[1:0] bits to 01b as well as setting the IICSTAREQ bit to 1.

If you want to generate the start condition after generating the stop condition, start the generation of the start condition with a half cycle period of the bit rate from the stop condition generation interrupt (STI) request output.

[Setting condition]

- Writing 1 to the bit

[Clearing condition]

- Completion of generation of the start condition

IICRSTAREQ bit (Restart Condition Generation)

When a restart condition is to be generated, set both the IICSDAS[1:0] and IICSCLS[1:0] bits to 01b as well as setting the IICRSTAREQ bit to 1.

[Setting condition]

- Writing 1 to the bit

[Clearing condition]

- Completion of generation of the restart condition

IICSTPREQ bit (Stop Condition Generation)

When a stop condition is to be generated, set both the IICSDAS[1:0] and IICSCLS[1:0] bits to 01b as well as setting the IICSTPREQ bit to 1.

[Setting condition]

- Writing 1 to the bit

[Clearing condition]

- Completion of generation of the stop condition

IICSDAS[1:0] bit (SDA Output Select)

These bits control output from the SDAn pin.

IICSCLS[1:0] bit (SCL Output Select)

These bits control output from the SCLn pin.

26.2.11 FCR : FIFO Control Register

Base address: SCI_Bn = 0x4011_8000 + 0x0100 × n (n = 0 to 4, 9)

Offset address: 0x24

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	RSTRG[4:0]					RFRST	—	—	RTRG[4:0]				
Value after reset:	0	0	0	1	1	1	1	1	0	0	0	1	1	1	1	1

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

Bit	Symbol	Function	R/W
0	DRES	Receive data ready error select bit Valid in Asynchronous mode This bit select the interrupt request for a reception data ready detection. 0: reception data full interrupt (SCIn_RXI) 1: receive error interrupt (SCIn_ERI)	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
12:8	TTRG[4:0]	Transmit FIFO data trigger number Valid in Asynchronous mode (including multi processor mode), Clock-synchronous mode, Simple SPI mode Trigger number must be set to 15 or less number. 0x00: Trigger number 0 ⋮ 0x1F: Trigger number 31	R/W
14:13	—	These bits are read as 0. The write value should be 0.	R/W
15	TFRST	Transmit FIFO Data Register Reset This bit enables only when CCR3.FM is 1. The read value is always 0. 0: It is invalid. It does not affect the operation. 1: The number of data stored in Transmit-FIFO (TDR register) are made 0	W
20:16	RTRG[4:0]	Receive FIFO data trigger number Valid in Asynchronous mode (including multi processor mode), Clock-synchronous mode, Simple SPI mode Trigger number must be set to 15 or less number. 0x00: Trigger number 0 ⋮ 0x1F: Trigger number 31	R/W

Bit	Symbol	Function	R/W
22:21	—	These bits are read as 0. The write value should be 0.	R/W
23	RFRST	Receive FIFO Data Register Reset This bit enables only when CCR3.FM is 1. The read value is always 0. 0: It is invalid. It does not affect the operation. 1: The number of data stored in Receive-FIFO(RDR register) are made 0	W
28:24	RSTRG[4:0]	RTS Output Active Trigger Number Select Valid in Asynchronous mode (including multi processor mode), Clock-synchronous mode This bit enables only when CCR3.FM = 1 and CCR1.CTSE = 0 and CCR0.SSE = 0. Trigger number must be set to 15 or less number. 0x00: Trigger number 0 ⋮ 0x1F: Trigger number 31	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

DRES bit (Receive data ready error select bit)

Select whether the detection of receive data ready (FRSR.DR flag = 1) is the cause of SCIn_RXI interrupt request or the cause of SCIn_ERI interrupt request.

TTRG[4:0] bit (Transmit FIFO data trigger number)

The TDFE flag is set to 1 when the quantity of transmit data in the transmit-FIFO(TDR register) is equal to or less than the specified transmit triggering number. If SCR.TIE = 1, SCIn_TXI interrupt request is occurred.

Note: Trigger number have to be set 15. In case the trigger number is set 16 or more, unexpected SCIn_TXI interrupt will occur.

TFRST bit (Transmit FIFO Data Register Reset)

When the TFRST bit is set to 1, the number of the transmission data stored in Transmit-FIFO(TDR register) is made 0.

RTRG[4:0] bit (Receive FIFO data trigger number)

The CSR.RDRF flag is set to 1 when the quantity of receive data in the receive-FIFO(RDR register) is equal to or greater than the specified receive triggering number. If CCR0.RIE = 1, SCIn_RXI interrupt request is occurred. When FCR.RTRG is set to 0, RDRF bit is set if the quantity of data in receive-FIFO is greater than or equal to 1.

Note: Trigger number have to be set to 15. In case the trigger number is set to 16 or more, unexpected SCIn_RXI interrupt will occur.

RFRST bit (Receive FIFO Data Register Reset)

When the RFRST bit is set to 1, the number of the reception data stored in Receive-FIFO(RDR register) is made 0.

RSTRG[4:0] bit (RTS Output Active Trigger Number Select)

When the quantity of receive data stored in the receive-FIFO (RDR register) is equal to or greater than this number, the RTSn signal is in the High state. When FCR.RSTRG is set to 0, RTSn is in the high state if the quantity of data in reveive FIFO is greater than or equal to 1.

Note: Trigger number have to be set 15. In case the trigger number is set 16 or more, RTSn will go to High state at unexpected timing.

26.2.12 MCR : Manchester Control Register

Base address: SCI_Bn = 0x4011_8000 + 0x0100 × n (n = 0 to 4, 9)

Offset address: 0x2C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	SBERE N	SYERE N	PFERE N	—	—	RPPAT[1:0]		RPLEN[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	TPPAT[1:0]		TPLEN[3:0]				—	SBSEL	SYNSEL	SYNVAL	—	ERTEN	TMPOL	RMPOL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RMPOL	Polarity of Received Manchester Code Sets the polarity of the received Manchester code 0: Logic 0 is coded as a zero-to-one transition in Manchester code Logic 1 is coded as a one-to-zero transition in Manchester code 1: Logic 0 is coded as a one-to-zero transition in Manchester code Logic 1 is coded as a zero-to-one transition in Manchester code	R/W
1	TMPOL	Polarity of Transmit Manchester Code Sets the polarity of the transmit Manchester code 0: Logic 0 is coded as a zero-to-one transition in Manchester code Logic 1 is coded as a one-to-zero transition in Manchester code 1: Logic 0 is coded as a one-to-zero transition in Manchester code Logic 1 is coded as a zero-to-one transition in Manchester code	R/W
2	ERTEN	Manchester Edge Retiming Enable Sets the receive retiming function 0: Disables the receive retiming function 1: Enables the receive retiming function	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	SYNVAL	SYNC value Setting Sets the SYNC type of the start bit(s) in the Manchester code When the start bit area consists of one bit. (SBSEL = 0) ● when transmitting 0: The start bit is added as a zero-to-one transition. 1: The start bit is added as a one-to-zero transition. ● when receiving 0: Only when the start bit is a zero-to-one transition, the data is received. The other cases are judged as an error. 1: Only when the start bit is a one-to-zero transition, the data is received. The other cases are judged as an error. When the start bit area consists of three bits. (SBSEL = 1) ● when transmitting 0: The start bits are added as a zero-to-one transition. (DATA SYNC) 1: The start bits are coded as a one-to-zero transition. (COMMAND SYNC) ● when receiving When the start bit area consists of three bits, data is received regardless of the value of this bit.	R/W
5	SYNSEL	SYNC Select 0: The start bit pattern is set with the SYNVAL bit 1: The start bit pattern is set with the TSYNC bit.	R/W
6	SBSEL	Start Bit Select 0: The start bit area consists of one bit. 1: The start bit area consists of three bits (COMMAND SYNC or DATA SYNC)	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
11:8	TPLEN[3:0]	Transmit preface length Set the preface length of the transmit data in Manchester mode 0x0: Disables the transmit preface generation Others: Transmit preface length (bit length)	R/W
13:12	TPPAT[1:0]	Transmit preface pattern Set the preface pattern of the transmit data 0 0: ALL ZERO 0 1: ZERO ONE 1 0: ONE ZERO 1 1: ALL ONE	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W
19:16	RPLEN[3:0]	Receive Preface Length Set the preface length in received frames when Manchester mode is enabled 0x0: Disables the receive preface generation Others: Receive preface length (bit length)	R/W
21:20	RPPAT[1:0]	Receive Preface Pattern Set the preface pattern of received frames 0 0: ALL ZERO 0 1: ZERO ONE 1 0: ONE ZERO 1 1: ALL ONE	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W
24	PFEREN	Preface Error Enable Specifies whether to handle a preface error as an interrupt source 0: Does not handle a preface error as an interrupt source 1: Handles a preface error as an interrupt source	R/W
25	SYEREN	Receive SYNC Error Enable Specifies whether to handle a receive SYNC error as an interrupt source 0: Does not handle a receive SYNC error as an interrupt source 1: Handles a receive SYNC error as an interrupt source	R/W
26	SBEREN	Start Bit Error Enable Specifies whether to handle a start bit error as an interrupt source 0: Does not handle a start bit error as an interrupt source 1: Handles a start bit error as an interrupt source	R/W
31:27	—	These bits are read as 0. The write value should be 0.	R/W

RMPOL bit (Polarity of Received Manchester Code)

This bit sets the polarity of the received Manchester code. For details, see section 26.5.7. Serial Data Reception in Manchester Mode.

TMPOL bit (Polarity of Transmit Manchester Code)

This bit sets the polarity of the transmit Manchester code. For details, see section 26.5.6. Serial data transmission in Manchester mode.

ERTEN bit (Manchester Edge Retiming Enable)

This bit sets the receive retiming function in Manchester mode.

For information on the receive retiming function, see section 26.5.9. Receive Retiming.

SYNVAL bit (SYNC value Setting)

This bit is valid when the SYNSEL bit of this register is set to 0.

The SYNC type can be set by combining this bit and the SBSEL bit.

For the start bit area determined by the combination of this bit and the SBSEL bit, see Figure 26.52 and Figure 26.53.

SYNSEL bit (SYNC Select)

This bit is valid when the SBSEL bit of this register is set to 1. This bit determines the destination to be referred to for setting the SYNC type of the start bit area added to Manchester frames.

When this bit is set to 0, the SYNVAL bit of this register is referred to.

When this bit is set to 1, the TSYNC bit in the TDR register is referred to.

SBSEL bit (Start Bit Select)

This bit sets the start bit area in Manchester frames.

When this bit is set to 1, the start bit area added to each frame consists of three bits, and the SYNSEL and SYNVAL bits in this register are valid.

When this bit is set to 0, the start bit area added to each frame consists of one bit.

TPLEN[3:0] bit (Transmit preface length)

These bits set the preface bit length of the transmit data in Manchester mode.

The settable range is 0x0 to 0xF (0d to 15d). 0h disables the transmit preface, which is not added.

TPPAT[1:0] bit (Transmit preface pattern)

These bits set one of the four preface patterns in Manchester mode. For the transmit data when the TPPAT[1:0] bits are set, see Figure 26.51.

When these bits are set to 00b, the preface area is set to all zeros.

When these bits are set to 01b, the preface area is set to the zero-one-zero-one pattern.

When these bits are set to 10b, the preface area is set to the one-zero-one-zero pattern.

When these bits are set to 11b, the preface area is set to all ones.

RPLEN[3:0] bit (Receive Preface Length)

These bits set the preface bit length of the received frames in Manchester mode.

The settable range is 0x0 to 0xF (0d to 15d). 0x0 disables the receive preface, which is not added. When 0x1 to 0xF is set, the set value is handled as the receive preface bit length.

RPPAT[1:0] bit (Receive Preface Pattern)

These bits set one of the four preface patterns in Manchester mode. For the transmit and receive data when the TPPAT[1:0] bits are set, see Figure 26.51.

When these bits are set to 00b, the preface area is handled as all zeros.

When these bits are set to 01b, the preface area is handled as the zero-one-zero-one pattern.

When these bits are set to 10b, the preface area is handled as the one-zero-one-zero pattern.

When these bits are set to 11b, the preface area is handled as all ones.

PFEREN bit (Preface Error Enable)

This bit specifies whether to handle a preface error as an interrupt source.

When it is set to 0, a preface error is not handled as an interrupt source. When it is set to 1, a preface error is handled as an interrupt source.

SYEREN bit (Receive SYNC Error Enable)

This bit specifies whether to handle a receive SYNC error as an interrupt source.

When it is set to 0, a receive SYNC error is not handled as an interrupt source. When it is set to 1, a receive SYNC error is handled as an interrupt source.

SBEREN bit (Start Bit Error Enable)

This bit specifies whether to handle a start bit error as an interrupt source.

When it is set to 0, a start bit error is not handled as an interrupt source. When it is set to 1, a start bit error is handled as an interrupt source.

26.2.13 DCR : Driver Control Register

Base address: SCI_Bn = 0x4011_8000 + 0x0100 × n (n = 0 to 4, 9)

Offset address: 0x30

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

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

Bit	Symbol	Function	R/W
0	DEPOL	Driver effective polarity select Valid only in Asynchronous mode 0: The DEn signal is active high. 1: The DEn signal is active low.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
12:8	DEAST[4:0]	Driver Assertion Time Valid only in Asynchronous mode Set the driver assertion time. When DEN = 1, the driver assertion time is inserted in addition to the normal transmission waiting time. Setting DEAST[4:0] = 5'H00 is prohibited.	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
20:16	DENGT[4:0]	Driver negate time Valid only in Asynchronous mode Set the driver negation time. When DEN = 1, the driver negate time is inserted after STOP bit transmission end. Setting DENGT[4:0] = 5'H00 is prohibited.	R/W
31:21	—	These bits are read as 0. The write value should be 0.	R/W

DEPOL bit (Driver effective polarity select)

Select the active level of the DEn signal.

DEAST[4:0] bit (Driver Assertion Time)

Set the driver assertion time (= time from the activation of the DEn (Driver Enable) signal to the start of the start bit). It is expressed in base clock period.

Driver assertion time

= DEAST [4:0] set value × base clock period + transmission waiting time

DENGT[4:0] bit (Driver negate time)

Set the driver negation time (= time from the end of the last stop bit of the transmitted message until the DEn (Driver Enable) signal is disabled). It is expressed in base clock period.

Driver negate time

= DENGT[4:0] set value × base clock period

If the transmission data is written during the driver negate time, transmit starting operation is different depends on the writing timing. (The DEn signal remains valid, and transmission of the start bit may start after the transmission wait time has elapsed. Also, the DEn signal may become invalid once, and start bit transmission may start after the Driver assertion time has elapsed.)

26.2.14 XCR0 : Simple LIN Control Register 0

Base address: SCI_Bn = 0x4011_8000 + 0x0100 × n (n = 0 to 4, 9)

Offset address: 0x34

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PIBS[2:0]			PIBE	CF1DS[1:0]		CF0RE	BFE	—	—	—	—	—	—	TCSS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	TCSS[1:0]	Timer count clock source selection (Valid in Simple LIN mode) Select the clock source of the timer in the LIN module. 0 1: TCLK/4 1 0: TCLK/16 1 1: TCLK/64	R/W*1
7:2	—	These bits are read as 0. The write value should be 0.	R/W
8	BFE	Break Field enable Set the presence or absence of Break Field of Start Frame. 0: No Break Field 1: With Break Field	R/W*3
9	CF0RE	Control Field 0 enable Set the presence or absence of Control Field 0 of Start Frame 0: No Control Field 0 1: With Control Field 0	R/W*3
11:10	CF1DS[1:0]	Control Field1 compare data select Select the compare data for Control Field 1 0 0: Select XCR1.PCF1D[7:0] as the compare data 0 1: Select XCR1.SCF1D[7:0] as the compare data 1 0: Select both XCR1.PCF1D[7:0] and XCR1.SCF1D[7:0] as the compare data 1 1: Setting prohibited	R/W*3
12	PIBE	Priority interrupt bit enable 0: Priority interrupt bit disable 1: Priority interrupt bit enable	R/W*3
15:13	PIBS[2:0]	Priority interrupt bit select Specify one of bits 0 to 7 of Control Field 1 as the priority interrupt bit. 0 0 0: bit 0 of Control Field 1 0 0 1: bit 1 of Control Field 1 0 1 0: bit 2 of Control Field 1 0 1 1: bit 3 of Control Field 1 1 0 0: bit 4 of Control Field 1 1 0 1: bit 5 of Control Field 1 1 1 0: bit 6 of Control Field 1 1 1 1: bit 7 of Control Field 1	R/W*3
16	BFOIE	Break Field output completion interrupt enable Select whether to include Break Field output completion as a SCIn_TXI interrupt factor. 0: Break Field output completion is not included in SCIn_TXI interrupt factor 1: Break Field output completion is included in SCIn_TXI interrupt factor	R/W
17	BCDIE	Bus conflict detection interrupt enable Select whether to output an SCIn_ERI interrupt when a bus collision is detected. 0: Bus conflict detection is not included in SCIn_ERI interrupt factor 1: Bus conflict detection is included in SCIn_ERI interrupt factor	R/W
19:18	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
20	BFDIE	Break Field detection interrupt enable Select whether to output a SCIn_BFD interrupt when a Break Field is detected. 0: Break Field detection interrupt disable 1: Break Field detection interrupt enable	R/W
21	COFIE	Counter overflow interrupt enable Select whether to include counter overflow as an SCIn_ERI interrupt factor. 0: Counter overflow is not included in SCIn_ERI interrupt factor 1: Counter overflow is included in SCIn_ERI interrupt factor	R/W
22	AEDIE	Active edge detection interrupt enable Select whether to output an SCIn_AED interrupt when a valid edge is detected. 0: Active edge detection interrupt disable 1: Active edge detection interrupt enable	R/W
23	—	This bit is read as 0. The write value should be 0.	R/W
25:24	BCCS[1:0]	Bus conflict detection clock selection Select the sampling clock for the bus conflict detection circuit. When CCR2.ABCS = 1, setting BCCS[1:0] = 1x is prohibited. 0 0: Base clock*2 0 1: Base clock/2 1 0: Base clock/4 1 1: Setting prohibited	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. You can rewrite TCSS[1:0] only when the timer is stopped (TCST = 0 and SDST = 0 and BMEN = 0)
Note 2. Base clock: 1/16 period of 1 bit period when CCR2.ABCS = 0, 1/8 period of 1 bit period when CCR2.ABCS = 1.
Note 3. This bit is a setting bit required for Start Frame reception operation. Rewrite this bit when Start Frame reception or transmission is not in progress (XCR1.SDST = 0 and XCR1.TCST = 0).

TCSS[1:0] bit (Timer count clock source selection)

Select clock source of timer in Simple LIN module.

BFE bit (Break Field enable)

Set the presence or absence of Break Field of Start Frame.

CF0RE bit (Control Field 0 enable)

Set the presence or absence of Control Field 0 of Start Frame.

CF1DS[1:0] bit (Control Field1 compare data select)

Select the compare data for Control Field 1.

PIBE bit (Priority interrupt bit enable)

Select whether to enable priority interrupt bit comparison of Control Field 1. When this bit is 1, regardless of the XCR1.CF1CE [7:0] setting value, the bit specified in PIBS [2:0] is compared with the primary comparison data for Control Field 1 (XCR1.PCF1D [7:0]).

PIBS[2:0] bit (Priority interrupt bit select)

Specify bit N (N = 0-7) of Control Field 1 as the priority interrupt bit.

BFOIE bit (Break Field output completion interrupt enable)

Select whether to include Break Field output completion as a SCIn_TXI interrupt factor. Set CCR0.TIE = 1 and CCR3.MOD [1: 0] = 110b, to output SCIn_TXI upon completion of Break Field output.

BCDIE bit (Bus conflict detection interrupt enable)

Select whether to output an SCIn_ERI interrupt when a bus collision is detected. In Simple LIN mode (CCR3.MOD [1:0] = 110b), SCIn_ERI output control is performed with this bit. When CCR3.MOD [1:0] = 110b and BCDIE = 1, an SCIn_ERI interrupt is issued when a bus collision is detected even if CCR0.RIE = 0.

COFIE bit (Counter overflow interrupt enable)

Select whether to include counter overflow as an SCIn_ERI interrupt factor. Set CCR0.RIE = 1 and CCR3.MOD [1:0] = 110b are required to output SCIn_ERI upon counter overflow.

AEDIE bit (Active edge detection interrupt enable)

Select whether to output an SCIn_AED interrupt when a valid edge is detected. To output SCIn_AED with valid edge detection, XCR1.BMEN = 1 and CCR3.MOD [1:0] = 110b must be set.

BCCS[1:0] bit (Bus conflict detection clock selection)

Select the sampling clock for the bus conflict detection circuit.

26.2.15 XCR1 : Simple LIN Control Register 1

Base address: SCI_Bn = 0x4011_8000 + 0x0100 × n (n = 0 to 4, 9)

Offset address: 0x38

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CF1CE[7:0]								SCF1D[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PCF1D[7:0]								—	—	BMEN	SDST	—	—	—	TCST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TCST	Break Field output timer count start trigger 0: Break Field output timer count stopped 1: Break Field output timer count start	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
4	SDST	Start Frame detection enable 0: Start Frame/Break Field detection disabled 1: Start Frame/Break Field detection enabled Do not set this bit and TCST bit to 1 at the same time.	R/W
5	BMEN	Bit rate measurement enable Set this bit to 1 simultaneously with the SDST bit. When this bit is set to 0, it can be set to 0 at any timing. 0: Bit rate measurement disabled 1: Bit rate measurement enabled	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
15:8	PCF1D[7:0]	Priority compare data for Control Field 1 The priority compare data for Control Field 1	R/W
23:16	SCF1D[7:0]	Secondary compare data for Control Field 1 The secondary compare data for Control Field 1	R/W
31:24	CF1CE[7:0]	Control Field 1 compare bit enable Select whether to compare bit N of Control Field 1. (N = 0 -7) 0: Control Field 1 bit N compare disabled 1: Control Field 1 bit N compare enabled	R/W

TCST bit (Break Field output timer count start trigger)

[Clearing condition]

- When 0 is written to TCST. Break Field output timer count is stopped and TXDn output becomes idle level.
- When Break Field output for the period set in XCR2.BFLW [15: 0] is completed.

[Setting condition]

- When 1 is written to TCST. Start Break Field output from TXDn. Holds 1 during Break Field output.

SDST bit (Start Frame detection enable)

When 1 is written to this bit, Start Frame detection starts. When XCR0.BFE = 1 is set, Break Field can be detected during Start Frame is detected and after Start frame is detected. When XCR0.BFE = 0 is set, Break Field is not detected.

When 0 is written to this bit, Start Frame detection and Break Field detection are stopped. However, if XSR0.RXDSF = 0 at the time of stop, it is not possible to stop data reception of the SCI core with this bit. Write 0 to CCR0.RE to stop the reception operation or perform reception completion processing (CSR.RDRF clear or RDR read) after reception is completed.

BMEN bit (Bit rate measurement enable)

Set this bit to 1 simultaneously with the SDST bit. When this bit is set to 1, the valid edge interval of Control Field 0 and Control Field 1 data is measured.

PCF1D[7:0] bit (Priority compare data for Control Field 1)

Set the priority compare data for Control Field 1.

SCF1D[7:0] bit (Secondary compare data for Control Field 1)

Set the secondary compare data for Control Field 1.

CF1CE[7:0] bit (Control Field 1 compare bit enable)

Select whether to compare bit N of Control Field 1. (N = 0 - 7)

When all of these bits are set to 0 (CF1CE[7:0] = 8'h00), it is always judged that Control Field 1 matches when reception is completed, and XSR0.CF1MF is set. This bit is a comparison enable with PCF1D[7:0] or SCF1D[7:0], and it is not a priority interrupt bit comparison enable.

26.2.16 XCR2 : Simple LIN Control Register 2

Base address: SCI_Bn = 0x4011_8000 + 0x0100 × n (n = 0 to 4, 9)

Offset address: 0x3C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	BFLW[15:0]															
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CF0CE[7:0]								CF0D[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	CF0D[7:0]	Control Field 0 compare data The compare data for Control Field 0	R/W
15:8	CF0CE[7:0]	Control Field 0 compare bit enable Select whether to compare bit N of Control Field 0. (N = 0 - 7) 0: Control Field 0 bit N compare disabled 1: Control Field 0 bit N compare enabled	R/W
31:16	BFLW[15:0]	Break Field length setting This register sets the Break Field length. The Break Field length is (BFLW [15:0] setting value + 1) × clock of the timer. The upper limit for setting this register is 0xFFFE. (Setting prohibited for 0xFFFF)	R/W

CF0D[7:0] bit (Control Field 0 compare data)

The compare data for Control Field 0.

CF0CE[7:0] bit (Control Field 0 compare bit enable)

Select whether to compare bit N of Control Field 0. (N = 0 - 7)

When all of these bits are set to 0 (CF0CE[7:0] = 8'h00), it is always judged that Control Field 0 matches when reception is completed, and XSR0.CF0MF is set.

BFLW[15:0] bit (Break Field length setting)

BFLW[15:0] are 16-bit Break Field length setting bits and the initial value is 0xFFFE.

Set the Break Field length to 1 frame or more. The LIN standard stipulates that the Break Field length is 13 bits or more.

When the Break Field sending. Writing 1 to TCST, SCI starts output the Break Field on TXDn. Up-counting is performed with the clock of the timer selected by XCR0.TCSS[1:0]. When the count value matches the value set in this register, up-counting is stopped and Break Field output from TXDn is also stopped.

When the Break Field receiving. Writing 1 to SDST, Start-Frame detection is enabled. SCI starts counting from the negative edge of RXDn. The clock of the timer is selected by XCR0.TCSS[1:0].

When the count value matches the value set in this register, it is determined that a Break Field has been detected. Up-counting continues until the next valid edge or counter overflow.

26.2.17 CSR : Common Status Register

Base address: SCI_Bn = 0x4011_8000 + 0x0100 × n (n = 0 to 4, 9)

Offset address: 0x48

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	RDRF	TEND	TDRE	FER	PER	MFF	—	ORER	—	—	—	—	—	DFER	DPER	DCMF
Value after reset:	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0.	R
4	ERS	Error Signal Status Flag Valid only in Smart card interface mode 0: Error signal Low not responded 1: Error signal Low responded	R
14:5	—	These bits are read as 0.	R
15	RXDMON	Serial input data monitor bit The state of the RXDn pin without synchronizing by bus clock is shown. 0: When RINV is 0, RXDn pin is the Low level. When RINV is 1, RXDn pin is the High level. 1: When RINV is 0, RXDn pin is the High level. When RINV is 1, RXDn pin is the Low level.	R
16	DCMF	Data Compare Match Flag Valid only in Asynchronous mode 0: No matched 1: Matched	R
17	DPER	Data Compare Match Parity Error Flag Valid only in Asynchronous mode 0: No parity error occurred at address match detection 1: A parity error has occurred at address match detection	R

Bit	Symbol	Function	R/W
18	DFER	Data Compare Match Framing Error Flag Valid only in Asynchronous mode 0: No framing error occurred at address match detection 1: A framing error has occurred at address match detection	R
23:19	—	These bits are read as 0.	R
24	ORER	Overrun Error Flag 0: No overrun error occurred 1: An overrun error has occurred	R
25	—	This bit is read as 0.	R
26	MFF	Mode Fault Flag Valid only in Simple SPI mode. 0: No mode fault error 1: Mode fault error	R
27	PER	Parity Error Flag 0: Non-FIFO selected (CCR3.FM = 0): No parity error occurred FIFO selected (CCR3.FM = 1): No parity error in all received data in receive-FIFO 1: Non-FIFO selected (CCR3.FM = 0): A parity error has occurred FIFO selected (CCR3.FM = 1): One or more parity errors occurred in received data in receive-FIFO	R
28	FER	Framing Error Flag 0: Non-FIFO selected (CCR3.FM = 0): No framing error occurred FIFO selected (CCR3.FM = 1): No framing error in all received data in receive-FIFO 1: Non-FIFO selected (CCR3.FM = 0): A framing error has occurred FIFO selected (CCR3.FM = 1): One or more framing errors occurred in received data in receive-FIFO	R
29	TDRE	Transmit Data Empty Flag 0: Non-FIFO selected (CCR3.FM = 0): Transmit data is in TDR register FIFO selected (CCR3.FM = 1): The quantity of transmit data written in transmit-FIFO exceeds the specified transmit triggering number. 1: Non-FIFO selected (CCR3.FM = 0): No transmit data is in TDR register FIFO selected (CCR3.FM = 1): The quantity of transmit data written in transmit-FIFO is equal to or less than the specified transmit triggering number.	R
30	TEND	Transmit End Flag 0: A character is being transmitted or standing by for transmission. 1: Character transfer has been completed, or sending Break Field.	R
31	RDRF	Receive Data Full Flag 0: Non-FIFO selected (CCR3.FM = 0): No received data is in RDR register FIFO selected (CCR3.FM = 1): The quantity of receive data written in receive-FIFO falls below the specified receive triggering number. 1: Non-FIFO selected (CCR3.FM = 0): Received data is in RDR register FIFO selected (CCR3.FM = 1): The quantity of receive data written in receive-FIFO is equal to or greater than the specified receive triggering number.	R

ERS bit (Error Signal Status Flag)

[Setting condition]

- When an error signal LOW is sampled.

[Clearing condition]

- When write 1 to CFCLR.ERSC.

DCMF bit (Data Compare Match Flag)

The DCMF bit indicates that SCI detects the match to the comparison data (CCR4.CMPD) with receive data.

Clearing the CCR0.RE bit to 0 does not affect the DCMF flag, which retains its previous state.

[Setting condition]

- Matched to the comparison data (CCR4.CMPD) with receive data, while CCR0.DCME = 1.

[Clearing condition]

- When write 1 to CFCLR.DCMFC.

DPER bit (Data Compare Match Parity Error Flag)

The DPER bit indicates that a parity error occurred at Address Match detection (reception data match detection).

Clearing the CCR0.RE bit to 0 does not affect the DPER flag, which retains its previous state.

[Setting condition]

- When a parity error was detected by the frame in which an address match was detected.

[Clearing condition]

- When write 1 to CFCLR.DPERC.

DFER bit (Data Compare Match Framing Error Flag)

The DFER bit indicates that a framing error occurred at Address Match detection (reception data match detection).

Clearing the CCR0.RE bit to 0 does not affect the DFER flag, which retains its previous state.

[Setting condition]

- When a stop bit of the frame in which an Address Match was detected is 0.
  When it is a 2-stop mode, the 1st bit of stop bit judges only whether it's 1 and doesn't check the 2nd bit of stop bit.

[Clearing condition]

- When write 1 to CFCLR.DFERC.

ORER bit (Overrun Error Flag)

The ORER bit indicates that an overrun error has occurred during reception and the reception ends abnormally.

Clearing the CCR0.RE bit to 0 does not affect the ORER flag, which retains its previous state. In Simple IIC mode, this bit is not used.

[Setting condition with Non-FIFO mode (CCR3.FM = 0)]

- When the next data is received before reading out RDR with no-error reception data stored in RDR.
  In RDR, receive data prior to an overrun error occurrence is retained, but data received after the overrun error occurrence is lost. When the ORER flag is set to 1, reception data isn’t forwarded to RDR register.
  Note that, in clock synchronous mode and Simple SPI mode, serial reception will be stop.

[Setting condition with FIFO mode (CCR3.FM = 1)]

- When the next serial reception is completed while the receive FIFO is full of 16receive data.

[Clearing condition]

- When write 1 to CFCLR.ORERC.

MFF bit (Mode Fault Flag)

The MFF bit indicates mode fault errors. In a multi-master configuration, determine the mode fault error occurrence by reading the MFF flag.

[Setting condition]

- Input on the SSn pin being at the low level during master operation (CCR3.CKE[1:0] = 00 or 01) in simple SPI mode.

[Clearing condition]

- When write 1 to CFCLR.MFFC.

PER bit (Parity Error Flag)

The PER bit indicates that a parity error has occurred during reception and the reception ends abnormally.

Clearing the CCR0.RE bit to 0 does not affect the PER flag, which retains its previous state.

In Clock-synchronous mode, Simple SPI mode and Simple IIC mode, this bit is not used.

[Setting condition]

- When a parity error is detected during reception. In FIFO select mode, when one or more parity error is detected in receive-FIFO data.
  In non-FIFO mode, although receive data when the parity error occurs is transferred to RDR, no SCIn_RXI interrupt request occurs. Note that when the PER flag is being set to 1, the subsequent receive data is not transferred to RDR.

[Clearing condition]

- When write 1 to CFCLR.PERC.

FER bit (Framing Error Flag)

The FER bit indicates that a framing error has occurred during reception and the reception ends abnormally.

Clearing the CCR0.RE bit to 0 does not affect the FER flag, which retains its previous state.

In Clock-synchronous mode, Simple SPI mode and Simple IIC mode, this bit is not used.

[Setting condition]

- When 0 is sampled as the stop bit during reception. In FIFO select mode, when one or more framing error is detected in receive-FIFO data. In Manchester mode, when both sampling results (1⁄4 and 3⁄4 sampling points) are not 1 for 1 stop bit. In simple LIN mode, even if a condition that changes to 1 occurs when XCR1.SDST = 1, the FER set timing is delayed up to the Break Field judgment timing at the longest, since it may be a Break Field. If an edge is detected on the RXDn signal before the Break Field judgment timing, FER is detected. If no edge is detected in the RXDn signal before the Break Field judgment timing, Break Field is detected.
  In 2-stop-bit mode, only the first stop bit is checked whether it is 1 but the second stop bit is not checked. In non-FIFO mode, although receive data when the framing error occurs is transferred to RDR, no SCIn_RXI interrupt request occurs. In addition, when the FER flag is being set to 1, the subsequent receive data is not transferred to RDR.

[Clearing condition]

- When write 1 to CFCLR.FERC.

TDRE bit (Transmit Data Empty Flag)

[Non-FIFO selected (CCR3.FM = 0)]

The TDRE bit indicates the presence of transmit data in the TDR register.

The condition of CCR0.TE = 0 has priority over the condition of 0.

If other conditions that become 1 and conditions that become 0 are satisfied at the same time, the TDRE flag is set to 0.

[Setting condition]

- When CCR0.TE is 0.
- When data is transmitted from the TDR register to the TSR register.

[Clearing condition]

- When write 1 to CFCLR.TDREC.
- When the transmission data is written to the TDR register during CCR0.TE = 1.

[FIFO selected (CCR3.FM = 1)]

The TDRE bit indicates that data has been transferred from the transmit-FIFO (TDR) into the transmit shift register (TSR), the quantity of data in transmit-FIFO has fallen below the specified transmit triggering number.

When the condition which becomes 1 and the condition which becomes 0 were formed at the same time, TDRE flag will be 0. After that, when the number of data stored in transmit-FIFO is judged, and if that is same or greater than TTRG value, TDRE is set to 1 after 1 PCLK.

[Setting condition]

- When the quantity of transmit data written in transmit-FIFO is equal to or less than the specified transmit triggering number[*1].

Note 1. The transmit-FIFO is FIFO register of 16 steps, the maximum number of data that can be written when the TDRE flag is 1 is indicated in 0x10 - FTSR.T[5:0]. Even if data any more is written, data is discarded.

[Clearing condition]

- When write 1 to CFCLR.TDREC.
- When the transmission data is written to transmit-FIFO by the DTC or DMAC (the last block transfer when block transfer).

TEND bit (Transmit End Flag)

[Non-FIFO selected (CCR3.FM = 0), and Not Smart card interface mode (CCR3.MOD[2:0] ≠ 001)]

The TEND bit indicates completion of transmission.

[Setting condition]

- When CCR0.TE is 0.
- When the CCR0.TE bit is changed from 0 to 1, the TEND flag is not affected and retains the value 1.
- When the TDR register is not updated at the time of transmission of the tail-end bit of a character being transmitted.
- When the TDR register is not updated at the end of DE negate time with DE control function enable (CCR3.DEN = 1).
- When Break Field is sending.

[Clearing condition]

- After the synchronization delay time has elapsed since the transmission data was written to the TDR register during CCR0.TE = 1.
- When write 1 to CFCLR.TDREC during CCR0.TE = 1.

[Non-FIFO selected (CCR3.FM = 0), and Smart card interface mode (CCR3.MOD [2:0] = 001)]

With no error signal from the receiving side, this bit is set to 1 when further data for transfer is ready to be transferred to the TDR register.

[Setting condition]

- When CCR0.TE is 0.
- When the CCR0.TE bit is changed from 0 to 1, the TEND flag is not affected and retains the value 1.
- When a specified period has elapsed after the latest transmission of 1 byte, the ERS flag is 0, and the TDR register is not updated. The set timing is determined by register settings as listed below.1
  When GM = 0 and BLK = 0, 12.5etu after the start of transmission
  When GM = 0 and BLK = 1, 11.5etu after the start of transmission
  When GM = 1 and BLK = 0, 11.0etu after the start of transmission
  When GM = 1 and BLK = 1, 11.0etu after the start of transmission

[Clearing condition]

- After the synchronization delay time has elapsed since the transmission data was written to the TDR register during CCR0.TE = 1.
- When write 1 to CFCLR.TDREC during CCR0.TE = 1.

[FIFO selected (CCR3.FM = 1)]

The TEND bit indicates that the transmit-FIFO does not contain valid data when transmitting the last bit of a serial character, so the transmission is halted.

[Setting condition]

- TEND is set to 1 when transmit-FIFO does not contain transmit data when the last bit of a one-byte serial character is transmitted.
- When the TDR register is not updated at the end of DE negate time with DE control function enable (CCR3.DEN = 1).

[Clearing condition]

- After the synchronization delay time has elapsed since the transmission data was written to the TDR register during CCR0.TE = 1.

RDRF bit (Receive Data Full Flag)

[Non-FIFO selected (CCR3.FM = 0)]

The RDRF bit indicates the presence of receive data in the RDR register.

[Setting condition]

- When the reception ends normally, and receive data is forwarded from the RSR register to the RDR register.

[Clearing condition]

- When write 1 to CFCLR.RDRFC.
- When the read data is read from the RDR register.

[FIFO selected (CCR3.FM = 1)]

The RDRF bit indicates that receive data has been transferred to the receive FIFO data register (RDR), and the quantity of data in receive-FIFO is equal to or greater than the specified receive triggering number. When FCR.RTRG is set to 0, RDRF is set if the quantity of data in receive-FIFO is greater than or equal to 1.

[Setting condition]

- RDRF is set to 1 when the quantity of receive data in receive-FIFO is equal to or greater than the specified receive triggering number*1.

Note 1. Since the receive-FIFO is 16 stage FIFO register, the maximum quantity of data that can be read when RDF is 1 is equivalent to the specified receive data count number (FDR.R[5:0]). If an attempt is made to read after all the data in receive-FIFO has been read, the data is undefined.

[Clearing condition]

- When write 1 to CFCLR.RDRFC.
- When the reception data is read form receive-FIFO by the DTC or DMAC (the last block transfer when block transfer).

When the condition which becomes 1 and the condition which becomes 0 were formed at the same time, RDRF flag will be 0. After that, when the number of stored data in receive-FIFO is judged, and if that is same or greater than RTRG value, RDRF is set to 1 behind 1 PCLK.

Note: Except when interruption communication, RDRF and TDRE should not be cleared by CFCLR register.

26.2.18 ISR : Simple IIC Status Register

Base address: SCI_Bn = 0x4011_8000 + 0x0100 × n (n = 0 to 4, 9)

Offset address: 0x4C

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

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

Bit	Symbol	Function	R/W
0	IICACKR	ACK Reception Data Flag 0: ACK received 1: NACK received	R

Bit	Symbol	Function	R/W
1	—	This bit is read as 0.	R
2	—	The read value is undefined.	R
3	IICSTIF	Issuing of Start, Restart, or Stop Condition Completed Flag 0: There are no requests for generating conditions or a condition is being generated. 1: A start, restart, or stop condition is completely generated.	R
5:4	—	The read value is undefined.	R
31:6	—	These bits are read as 0.	R

IICACKR bit (ACK Reception Data Flag)

Received ACK and NACK bits can be read from this bit.

The IICACKR flag is updated at the rising of SCLn clock for the ACK/NACK receiving bit.

IICSTIF bit (Issuing of Start, Restart, or Stop Condition Completed Flag)

After generating a condition, this bit indicates that the generation is completed. When using the IICSTAREQ, IICRSTAREQ, or IICSTPREQ bit to cause generation of a condition, do so after setting the IICSTIF flag to 0.

When the IICSTIF flag is 1 while an interrupt request is enabled by setting the CCR0.TEIE bit, an STIn request is output.

[Setting condition]

- Completion of the generation of a start, restart, or stop condition (however, in cases where this conflicts with any of the conditions for the flag becoming 0 listed below, the clearing condition takes precedence)

[Clearing condition]

- Writing 1 to CFCLR.IICSTIFC bit
- When operation is not in Simple IIC mode
- Writing 0 to CCR0.TE bit

26.2.19 FRSR : FIFO Receive Status Register

Base address: SCI_Bn = 0x4011_8000 + 0x0100 × n (n = 0 to 4, 9)

Offset address: 0x50

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	FNUM[5:0]						—	—	PNUM[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
0	DR	Receive Data Ready flag 0: Receiving is in progress, or no received data has remained in receive-FIFO after normally completed receiving.(receive-FIFO is empty) 1: The following receive data does not come for a fixed period after storing data under the threshold in the receive-FIFO	R
1	—	The read value is undefined.	R
7:2	—	These bits are read as 0.	R

Bit	Symbol	Function	R/W
13:8	R[5:0]	Receive-FIFO Data Count Valid in Asynchronous mode (including multi-processor), Clock synchronous mode, Simple SPI mode, when FCR.FM is 1. Indicate the quantity of receive data stored in receive-FIFO	R
15:14	—	These bits are read as 0.	R
21:16	PNUM[5:0]	Parity Error Count Valid only in Asynchronous mode Indicates the quantity of data with a parity error among the receive data stored in the receive FIFO data register.	R
23:22	—	These bits are read as 0.	R
29:24	FNUM[5:0]	Framing Error Count Valid only in Asynchronous mode Indicates the quantity of data with a framing error among the receive data stored in the receive FIFO data register.	R
31:30	—	These bits are read as 0.	R

DR bit (Receive Data Ready flag)

The DR bit indicates that the quantity of data stored in the receive FIFO data register (RDR) falls below the specified receive triggering number, and that no next data has been received yet after the elapse of 15 etu from the last stop bit in asynchronous mode. This bit is valid only Asynchronous mode (including multi-processor mode) and FIFO selected. In other mode, this bit does not set to 1.

[Setting conditions]

- DR is set to 1 when the following conditions are met.
  - After receive-FIFO(RDR) receives less data than the specified receive triggering number, no next data has been received yet after the elapse of 15 etu*1 from the last stop bit
  - CSR.FER, PER flags are 0.

Note 1. This is equivalent to one and a half (1.5) frames in the 8-bit format with one stop bit (etu: elementary time unit).

[Clearing conditions]

- When all receive data in the receive FIFO (RDR register) is read and 1 is written to FFCLR.DRC.
- When CCR3.FM bit is 0.

R[5:0] bit (Receive-FIFO Data Count)

The R[5:0] bits indicate the quantity of receive data stored in receive-FIFO.

0x00 means no receive data. 0x10 means receive-FIFO is full.

PNUM[5:0] bit (Parity Error Count)

The value indicates the quantity of data stored in the receive-FIFO registers with a parity error.

FNUM[5:0] bit (Framing Error Count)

The value indicates the quantity of data stored in the receive-FIFO registers with a framing error.

26.2.20 FTSR : FIFO Transmit Status Register

Base address: SCI_Bn = 0x4011_8000 + 0x0100 × n (n = 0 to 4, 9)

Offset address: 0x54

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

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

Bit	Symbol	Function	R/W
5:0	T[5:0]	Transmit-FIFO Data Count Valid in Asynchronous mode (including multi-processor), Clock synchronous mode, Simple SPI mode, when FCR.FM is 1. Indicate the quantity of non-transmit data stored in transmit-FIFO	R
31:6	—	These bits are read as 0.	R

T[5:0] bit (Transmit-FIFO Data Count)

The T[5:0] bits indicate the quantity of non-transmitted data stored in transmit-FIFO.

0x00 means no un-transmit data. 0x10 means transmit- FIFO is full.

26.2.21 MSR : Manchester Status Register

Base address: SCI_Bn = 0x4011_8000 + 0x0100 × n (n = 0 to 4, 9)

Offset address: 0x58

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

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

Bit	Symbol	Function	R/W
0	PFER	Preface Error flag This bit is set when a preface error (pattern mismatch) is detected 0: No preface error detected 1: Preface error detected	R
1	SYER	SYNC Error flag This bit is set when no edge is detected in the adjustable range during receive retiming 0: No receive SYNC error detected 1: Receive SYNC error detected	R
2	SBER	Start Bit Error flag This bit is set when a pattern mismatch in the start bit area is detected 0: No start bit error detected 1: Start bit error detected	R
3	—	This bit is read as 0.	R

Bit	Symbol	Function	R/W
4	MER	Manchester Error Flag Valid for Manchester mode only 0: No Manchester error occurred 1: Manchester error has occurred	R
5	—	This bit is read as 0.	R
6	RSYNC	Receive SYNC data bit It is valid when MCR.SBSEL = 1 in Manchester mode, 0 is read otherwise. 0: The received the Start Bit is DATA SYNC 1: The received the Start Bit is COMMAND SYNC	R
31:7	—	These bits are read as 0.	R

PFER bit (Preface Error flag)

This bit indicates that a preface error was detected when receiving frames in Manchester mode.

Even when the RE bit in CCR0 is set to 0 (serial reception is disabled), the PFER flag is not affected and retains its previous value.

[Setting condition]

- When detecting a preface error when receiving frames in Manchester mode
  The following operations are performed when a preface error occurs.
  - <When MCR.PFEREN = 1>
    The received data is not transferred to the RDR register and no SCIn_RXI interrupt request occurs. Instead, an SCIn_ERI interrupt request occurs. Note that when the PFER flag is being set to 1, the subsequently received data is not transferred to the RDR register.
  - <When MCR.PFEREN = 0>
    The received data is transferred to the RDR register and an SCIn_RXI interrupt request occurs. An SCIn_ERI interrupt request is not generated. The subsequent receive operations are not affected even with the PFER flag being set to 1.

[Clearing condition]

- Write 1 to MFCLR.PFERC.

SYER bit (SYNC Error flag)

This bit indicates that a receive SYNC error was detected when receiving frames in Manchester mode with MCR.ERTEN = 1 (Manchester edge retiming enabled).

Even when the RE bit in CCR0 is set to 0 (serial reception is disabled), the SYER flag is not affected and retains its previous value.

[Setting condition]

- When detecting a receive SYNC error when receiving frames in Manchester mode
  The following operations are performed when a receive SYNC error occurs.
  - <When MCR.SYEREN = 1>
    Although the received data is transferred to the RDR register, no SCIn_RXI interrupt request occurs. Instead, an SCIn_ERI interrupt request occurs. Note that when the SYER flag is being set to 1, the subsequently received data is not transferred to the RDR register.
  - <When MCR.SYEREN = 0>
    The received data is transferred to the RDR register and an SCIn_RXI interrupt request occurs. An SCIn_ERI interrupt request is not generated. The subsequent receive operations are not affected even with the SYER flag being set to 1.

[Clearing condition]

- Write 1 to MFCLR.SYERC.

SBER bit (Start Bit Error flag)

This bit indicates that a start bit error was detected when receiving frames in Manchester mode.

Even when the RE bit in CCR0 is set to 0 (serial reception is disabled), the SBER flag is not affected and retains its previous value.

[Setting condition]

- When detecting a start bit error when receiving frames in Manchester mode
  The following operations are performed when a start bit error occurs.
  - <When MCR.SBEREN = 1>
    The received data is not transferred to the RDR register and no SCIn_RXI interrupt request occurs. Instead, an SCIn_ERI interrupt request occurs. Note that when the SBER flag is being set to 1, the subsequently received data is not transferred to the RDR register.
  - <When MCR.SBEREN = 0>
    The received data is transferred to the RDR register and an SCIn_RXI interrupt request occurs. An SCIn_ERI interrupt request is not generated. The subsequent receive operations are not affected even with the SBER flag being set to 1.

[Clearing condition]

- Write 1 to MFCLR.SBERC.

MER bit (Manchester Error Flag)

When data is received in Manchester mode, Manchester error is detected, and it is displayed. Even when the RE bit in CCR0 is set to 0 (serial reception is disabled), the MER flag is not affected and retains its previ-ous value.

[Setting conditions]

- When receiving in Manchester mode and detecting Manchester code error in data area of received frame.
  Received data when an error occurs is transferred to the RDR register, but the SCIn_RXI interrupt request is not generated and the SCIn_ERI interrupt request is generated.
  When the Manchester error flag is set to 1, subsequent receive data is not transferred to the RDR register.
  For details on Manchester error, section 26.5.11. Errors in Manchester Mode.

[Clearing condition]

- Write 1 to MFCLR.MERC.

RSYNC bit (Receive SYNC data bit)

When Manchester mode (CCR3.MOD[2:0] = 101b) and MCR.SBSEL = 1, this bit indicates the type of SYNC of the received the Start Bit. For other settings, it is fixed to 0.

26.2.22 XSR0 : Simple LIN Status Register 0

Base address: SCI_Bn = 0x4011_8000 + 0x0100 × n (n = 0 to 4, 9)

Offset address: 0x5C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CF1RD[7:0]								CF0RD[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	AEDF	COF	PIBDF	CF1MF	CF0MF	BFDF	BCDF	BFOF	—	—	—	—	—	—	RXDSF	SFSF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SFSF	Start Frame Status flag 0: Start Frame detection disabled or Start Frame detection complete 1: Before Start Frame detection or during detection	R*1
1	RXDSF	RXDn input status flag 0: RXDn input to SCI is enabled 1: RXDn input to SCI is disabled	R*1
7:2	—	These bits are read as 0.	R
8	BFOF	Break Field Output completion flag 0: When Break Field is not output or during output 1: When Break Field output is completed	R
9	BCDF	Bus Conflict detection flag 0: When no Bus Conflict is detected 1: When Bus Conflict is detected	R
10	BFDF	Break Field detection flag 0: When Break Field is not detected 1: When Break Field is detected	R
11	CF0MF	Control Field 0 compare match flag 0: When Control-Field-0 data and the compare data does not match 1: When Control-Field-0 data and the compare data match	R
12	CF1MF	Control Field 1 compare match flag 0: When Control-Field-1 data and the compare data does not match 1: When Control-Field-1 data and the compare data match	R
13	PIBDF	Priority interrupt bit detection flag 0: When Priority interrupt bit is not detected 1: When Priority interrupt bit is detected	R
14	COF	Counter Overflow flag 0: When the counter for Break Field detection does not overflows 1: When the counter for Break Field detection overflows	R
15	AEDF	Active Edge detection flag 0: When Active edge is not detected 1: When Active edge is detected	R
23:16	CF0RD[7:0]	Control Field 0 received data Control Field 0 received data.	R
31:24	CF1RD[7:0]	Control Field 1 received data Control Field 1 received data.	R

Note 1. When PCLK is faster than TCLK, the flag set timing is delayed from the receive data full interrupt (SCIn_RXI) output. To see this flag under these conditions, wait at least 1 TCLK cycle after the receive data full interrupt (SCIn_RXI) before reading this register.

SFSF bit (Start Frame Status flag)

Indicates whether detect Start Frame is being detected.

[Setting condition]

- When 1 is written to XCR1.SDST.
- When a Break Field is detected in the Control Field0/Control Field 1/Information Field phase and the transition to the Control Field0 or Control Field1 reception state occurs.

[Clearing condition]

- When XCR1.SDST is 0.
- When Start Frame detection is completed.

RXDSF bit (RXDn input status flag)

Indicates the RXDn input status to the SCI core. When this bit is 1, RXDn input is received only by the simple LIN module and the Break Field is detected and is not input to the SCI core.

BFOF bit (Break Field Output completion flag)

Indicates the completion of Break Field output.

This bit can be cleared to 0 by writing 1 to XFCLR.BFOC.

BCDF bit (Bus Conflict detection flag)

Indicates the detection of bus conflict in Simple LIN transmit operation.

This bit can be cleared to 0 by writing 1 to XFCLR.BCDC.

BFDF bit (Break Field detection flag)

Indicates Break Field detection.

This bit can be cleared to 0 by writing 1 to XFCLR.BFDC.

CF0MF bit (Control Field 0 compare match flag)

Indicates compare match of Control Field 0 and compare data.

This bit can be cleared to 0 by writing 1 to XFCLR.CF0MC.

CF1MF bit (Control Field 1 compare match flag)

Indicates compare match detection of Control Field 1 and compare data.

This bit can be cleared to 0 by writing 1 to XFCLR.CF1MC.

PIBDF bit (Priority interrupt bit detection flag)

Indicates compare match detection of Control Field 1 and priority interrupt bit.

This bit can be cleared to 0 by writing 1 to XFCLR.PIBDC.

COF bit (Counter Overflow flag)

Indicates that the 16-bit counter in the simple LIN module has overflowed.

This bit can be cleared to 0 by writing 1 to XFCLR.COFC.

AEDF bit (Active Edge detection flag)

Indicates active edge detection.

This bit can be cleared to 0 by writing 1 to XFCLR.AEDC and when read out XSR1.TCNT[15:0].

CF0RD[7:0] bit (Control Field 0 received data)

Stores the received data with a Control Field 0 match detected.

CF1RD[7:0] bit (Control Field 1 received data)

Stores the received data with a Control Field 1 match detected.

26.2.23 XSR1 : Simple LIN Status Register 1

Base address: SCI_Bn = 0x4011_8000 + 0x0100 × n (n = 0 to 4, 9)

Offset address: 0x60

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

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

Bit	Symbol	Function	R/W
15:0	TCNT[15:0]	Timer Count Capture value Stores the 16-bit counter capture value. The initial value is 0000.	R
31:16	—	These bits are read as 0.	R

TCNT[15:0] bit (Timer Count Capture value)

Stores the capture value of the 16-bit counter of the Simple LIN module.

- When sending Start Frame
  This register holds the previous value.
- When receiving Start frame with Bit rate measurement disabled
  If a Break Field is detected in the Break Field detection state (see Figure 26.79), the Break Field length is captured and held (counter value is captured at the rising edge of RXDn). If a Break Field is detected in a state other than the Break Field detection state, the previous value is retained.
  If the counter overflows, it will not be captured.
- When receiving Start frame with Bit rate measurement enabled
  The count value is captured and held at the valid edge (both RXDn edges). However, in the Break Field detection state, the count value is not captured even if a valid edge occurs. Counter capture value retention is canceled by this register read. Even if a valid edge occurs before reading, the counter value is not captured.

26.2.24 CFCLR : Common Flag Clear Register

Base address: SCI_Bn = 0x4011_8000 + 0x0100 × n (n = 0 to 4, 9)

Offset address: 0x68

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	RDRFC	—	TDREC	FERC	PERC	MFFC	—	ORERC	—	—	—	—	—	DFERC	DPERC	DCMFC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
3:0	—	The write value should be 0.	W
4	ERSC	ERS clear bit Setting this bit to 1 clears the CSR.ERS bit. The read value is always 0.	W
15:5	—	The write value should be 0.	W
16	DCMFC	DCMF clear bit Setting this bit to 1 clears the CSR.DCMF bit. The read value is always 0.	W
17	DPERC	DPER clear bit Setting this bit to 1 clears the CSR.DPER bit. The read value is always 0.	W
18	DFERC	DFER clear bit Setting this bit to 1 clears the CSR.DFER bit. The read value is always 0.	W
23:19	—	The write value should be 0.	W
24	ORERC	ORER clear bit Setting this bit to 1 clears the CSR.ORER bit. The read value is always 0.	W
25	—	The write value should be 0.	W
26	MFFC	MFF clear bit Setting this bit to 1 clears the CSR.MFF bit. The read value is always 0.	W
27	PERC	PER clear bit Setting this bit to 1 clears the CSR.PER bit. The read value is always 0.	W

Bit	Symbol	Function	R/W
28	FERC	FER clear bit Setting this bit to 1 clears the CSR.FER bit. The read value is always	W
29	TDREC	TDRE clear bit Setting this bit to 1 clears the CSR.TDRE bit. The read value is always 0.	W
30	—	The write value should be 0.	W
31	RDRFC	RDRF clear bit Setting this bit to 1 clears the CSR.RDRF bit. The read value is always 0.	W

26.2.25 ICFCLR : Simple IIC Flag Clear Register

Base address: SCI_Bn = 0x4011_8000 + 0x0100 × n (n = 0 to 4, 9)

Offset address: 0x6C

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

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

Bit	Symbol	Function	R/W
2:0	—	The write value should be 0.	W
3	IICSTIFC	IICSTIF clear bit Setting this bit to 1 clears the ISR.IICSTIF bit. The read value is always 0.	W
31:4	—	The write value should be 0.	W

26.2.26 FFCLR : FIFO Flag Clear Register

Base address: SCI_Bn = 0x4011_8000 + 0x0100 × n (n = 0 to 4, 9)

Offset address: 0x70

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

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

Bit	Symbol	Function	R/W
0	DRC	DR clear bit Setting this bit to 1 clears the FRSR.DR bit. The read value is always 0.	W
31:1	—	The write value should be 0.	W

26.2.27 MFCLR : Manchester Flag Clear Register

Base address: SCI_Bn = 0x4011_8000 + 0x0100 × n (n = 0 to 4, 9)

Offset address: 0x74

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

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

Bit	Symbol	Function	R/W
0	PFERC	PFER clear bit Setting this bit to 1 clears the MSR.PFER bit. The read value is always 0.	W
1	SYERC	SYER clear bit Setting this bit to 1 clears the MSR.SYER bit. The read value is always 0.	W
2	SBERC	SBER clear bit Setting this bit to 1 clears the MSR.SBER bit. The read value is always 0.	W
3	—	The write value should be 0.	W
4	MERC	MER clear bit Setting this bit to 1 clears the MSR.MER bit. The read value is always 0.	W
31:5	—	The write value should be 0.	W

26.2.28 XFCLR : Simple LIN Flag Clear Register

Base address: SCI_Bn = 0x4011_8000 + 0x0100 × n (n = 0 to 4, 9)

Offset address: 0x78

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	AEDC	COFC	PIBDC	CF1MC	CF0MC	BFDC	BCDC	BFOC	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	The write value should be 0.	W
8	BFOC	BFOF clear bit Setting this bit to 1 clears the XSR0.BFOF bit. The read value is always 0.	W
9	BCDC	BCDF clear bit Setting this bit to 1 clears the XSR0.BCDF bit. The read value is always 0.	W
10	BFDC	BFDF clear bit Setting this bit to 1 clears the XSR0.BFDF bit. The read value is always 0.	W
11	CF0MC	CF0MF clear bit Setting this bit to 1 clears the XSR0.CF0MF bit. The read value is always 0.	W
12	CF1MC	CF1MF clear bit Setting this bit to 1 clears the XSR0.CF1MF bit. The read value is always 0.	W

Bit	Symbol	Function	R/W
13	PIBDC	PIBDF clear bit Setting this bit to 1 clears the XSR0.PIBDF bit. The read value is always 0.	W
14	COFC	COFF clear bit Setting this bit to 1 clears the XSR0.COF bit. The read value is always 0.	W
15	AEDC	AEDF clear bit Setting this bit to 1 clears the XSR0.AEDF bit and cancels holding the XSR1 register. The read value is always 0.	W
31:16	—	The write value should be 0.	W

26.2.29 CESR : Communication Enable Status Register

Base address: SCI_Bn = 0x4011_8000 + 0x0100 × n (n = 0 to 4, 9)

Offset address: 0x1C

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

Bit	Symbol	Function	R/W
0	RIST	RE Internal status 0: RE signal internal state value 0 1: RE signal internal state value 1	R
3:1	—	These bits are read as 0.	R
4	TIST	TE Internal status 0: TE signal internal state value 0 1: TE signal internal state value 1	R
7:5	—	These bits are read as 0.	R

The operation clocks of the communication module and control register can be used asynchronously. Since some control register values are transmitted internally via the synchronization circuit so that they operate correctly even if they are asynchronous, it takes some time for the state to be reflected internally after rewriting the control register.

Communication enable CCR0.TE and CCR0.RE correspond to this register, and when these control bits are set from 1 to 0 to rewrite the control bits for the next communication, the TE and RE signal It is necessary to rewrite the next control bit after the internal state becomes 0. If a very slow clock is used for the communication module clock, the TE and RE bit states will not be reflected slowly internally. At this time, you can check the internal status using this register.

26.3 Operation in Asynchronous Mode

Figure 26.4 shows the general format for asynchronous serial communications. One frame consists of a start bit (low level), transmit or receive data, a parity bit, and stop bits (high level). In asynchronous serial communications, the communications line is usually held in the mark state (high level).

The SCI monitors the communications line. When the SCI detects a start bit, it starts serial communication. The detection condition of the start bit changes according to the CCR3.RXDESEL setting. SCI regards space (Low level) as a start bit when CCR3.RXDESEL is 0. SCI regards a fall edge as a start bit when RXDESEL is 1.

Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex communications. Both the transmitter and receiver have a double-buffered structure in addition to FIFO mode, so that data can be read or written during transmission or reception, enabling continuous data transmission and reception.

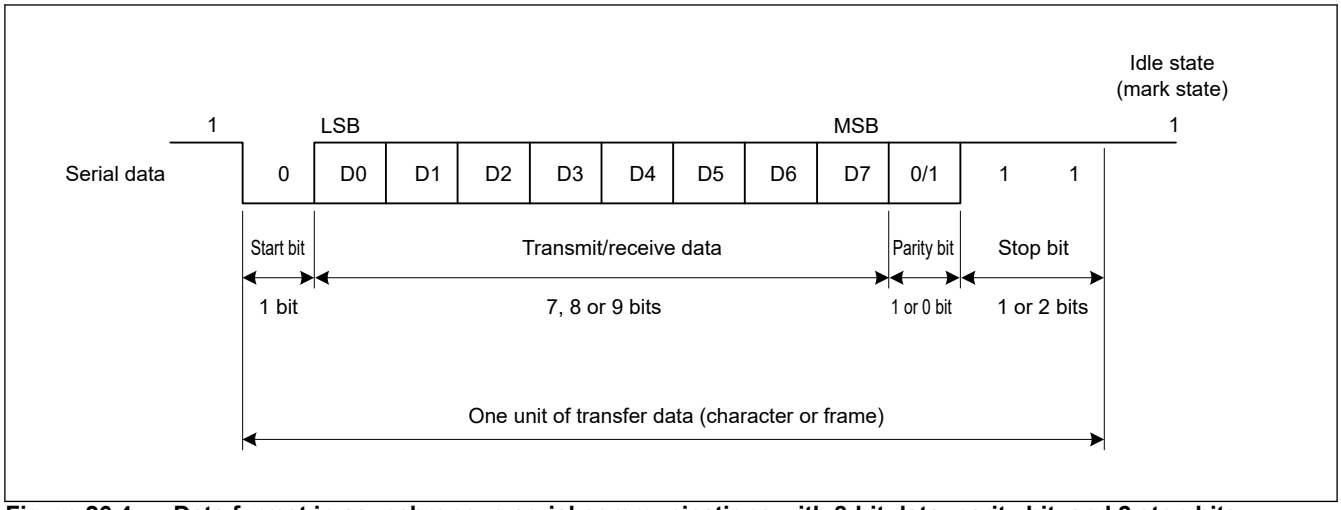


Figure 26.4 Data format in asynchronous serial communications with 8-bit data, parity bit, and 2 stop bits

26.3.1 Serial Data Transfer Format

Table 26.25 lists the serial data transfer formats that can be used in asynchronous mode. Any of 18 transfer formats can be selected with the CCR1 and CCR3 settings. For details on the multi-processor function, see section 26.4. Multi-Processor Communication Function.

Table 26.25 Serial transfer formats in asynchronous mode (1 of 2)

CCR3		CCR1		CCR3	Serial transfer format and frame length
CHR[1:0]		PE	MP	STP	1 2 3 4 5 6 7 8 9 10 11 12 13
0	0	0	0	0	ST 9-bit data SP
0	0	0	0	1	ST 9-bit data SP SP
0	0	1	0	0	ST 9-bit data P SP
0	0	1	0	1	ST 9-bit data P SP SP
1	0	0	0	0	ST 8-bit data SP
1	0	0	0	1	ST 8-bit data SP SP
1	0	1	0	0	ST 8-bit data P SP
1	0	1	0	1	ST 8-bit data P SP SP
1	1	0	0	0	ST 7-bit data SP
1	1	0	0	1	ST 7-bit data SP SP
1	1	1	0	0	ST 7-bit data P SP
1	1	1	0	1	ST 7-bit data P SP SP

Table 26.25 Serial transfer formats in asynchronous mode (2 of 2)

CCR3		CCR1		CCR3	Serial transfer format and frame length
CHR[1:0]		PE	MP	STP	1 2 3 4 5 6 7 8 9 10 11 12 13
0	0	—	1	0	ST | 9-bit data | MPB | SP
0	0	—	1	1	ST | 9-bit data | MPB | SP | SP
1	0	—	1	0	ST | 8-bit data | MPB | SP
1	0	—	1	1	ST | 8-bit data | MPB | SP | SP
1	1	—	1	0	ST | 7-bit data | MPB | SP
1	1	—	1	1	ST | 7-bit data | MPB | SP | SP

ST: Start bit
SP: Stop bit
P: Parity bit
MPB: Multi-processor bit

26.3.2 Receive Data Sampling Timing and Reception Margin in Asynchronous Mode

In asynchronous mode, the SCI operates on a base clock with a frequency of 16 times[*1] the bit rate.

In reception, the SCI samples the falling edge of the start bit using the base clock, and performs internal synchronization.[*2]

Because receive data is sampled on the rising edge of the 8th pulse[*1] of the base clock, data is latched at the middle of each bit (when sampling timing does not adjust (CCR4.ASEN = 0 or CCR4.ASEN = 1 and CCR4.AST[2:0] = 000b)), as shown in Figure 26.5 The reception margin in asynchronous mode is determined by the following formula (1):

$$M = \left|\left(0.5 - \frac{1}{2N}\right) - (L - 0.5)F - \frac{|D - 0.5|}{N}(1 + F)\right| \times 100\ [\%] \quad \ldots \quad \text{Formula (1)}$$

Note: M: Reception margin
N: Ratio of bit rate to clock
(N = 16 when CCR2.ABCSE = 0 and CCR2.ABCS = 0,
N = 8 when CCR2.ABCSE = 0 and CCR2.ABCS = 1,
N = 6 when CCR2.ABCSE = 1)
D: Duty cycle of clock (D = 0.5 to 1.0)
L: Frame length (L = 9 to 13)
F: Absolute value of clock frequency deviation

Assuming values of F = 0 and D = 0.5 in formula (1), the reception margin is determined using the following formula:
$M = \{0.5 - 1/(2 \times 16)\} \times 100\ (\%) = 46.875\ \%$

This represents the computed value. Renesas recommends a margin of 20% to 30% in system design.

Note 1. In this example, the CCR2.ABCS bit is 0 and the CCR2.ABCSE is 0. When the ABCS bit is 1 and the ABCSE bit is 0, a frequency of 8 times the bit rate is used as a base clock, and receive data is sampled on the rising edge of the 4th pulse of the base clock.

When the ABCSE bit is 1, a sextuple frequency of a bit rate is used as a base clock, and receive data is sampled on the rising edge of the 3rd pulse of the base clock.

Note 2. The determination condition of the start bit is as follows.

The function of adjust sampling timing is OFF (ASEN = 0):

The determination condition of a start bit is that Low beyond half bit length continues. It is same as the sampling timing. In Figure 26.5, Low period should be kept over 8-cycles to detect a start bit. If Low period does not keep over 8-cycles, the IP judges this as a noise. So, the IP does not start reception and wait start bit.

The function of adjust sampling timing is ON (ASEN = 1):

The determination condition of a start bit is that Low keeps up until the sampling timing. Adjusting the sampling timing forward (AJD = 1) increases the possibility of erroneously determining a noise as the start bit.

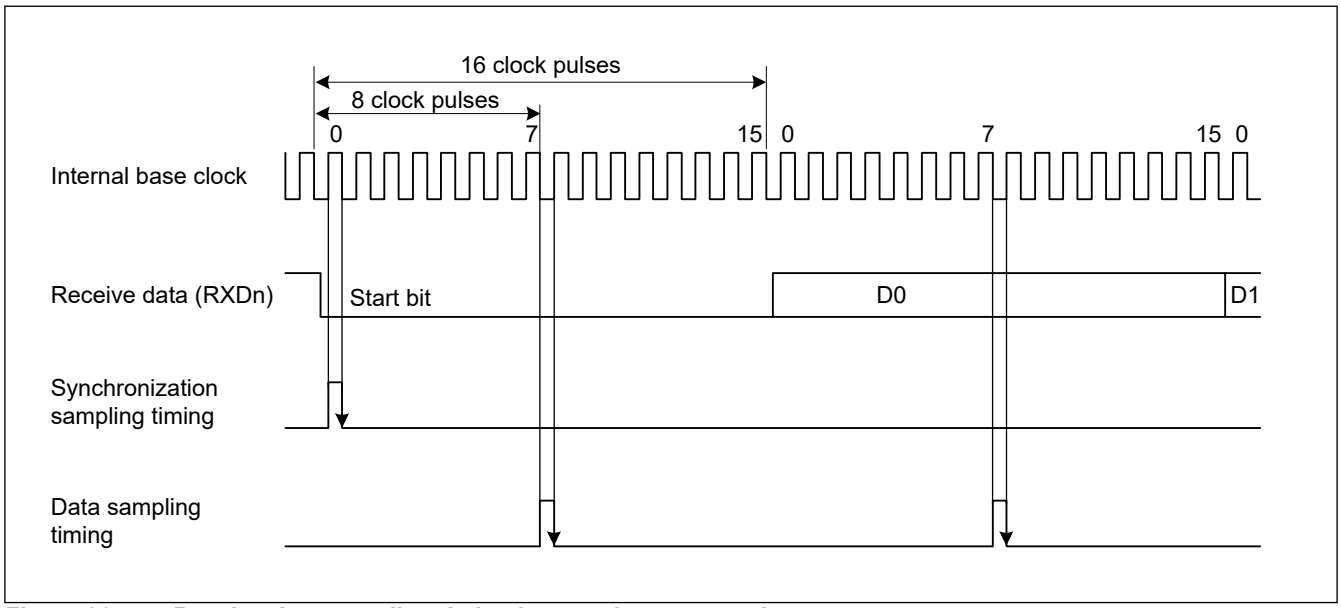


Figure 26.5 Receive data sampling timing in asynchronous mode

26.3.3 Clock

Either an internal clock generated by the on-chip baud rate generator or an external clock input to the SCKn pin can be selected as the transfer clock of the SCI, based on the CCR3.CKE[1:0] settings.

When an external clock is input to the SCKn pin, the clock frequency must be 16 times the bit rate (when CCR2.ABCS = 0) or 8 times the bit rate (when CCR2.ABCS = 1).

When the SCI uses its internal clock, the clock can be output from the SCKn pin. The frequency of the clock output in this case is equal to the bit rate, and the phase is configured so that the rising edge of the clock is in the middle of the transmit data, as shown in Figure 26.6.

When the internal clock is selected, the SCKn pin is outputted after setting the CCR0.TE or CCR0.RE bit to 1.

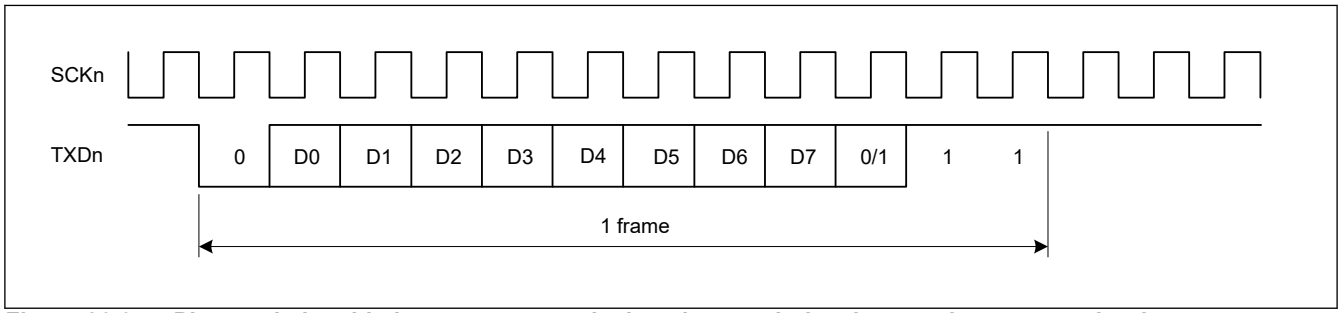


Figure 26.6 Phase relationship between output clock and transmit data in asynchronous mode when CCR1.PE = 1, CCR3.CHR[1:0] = 10b, MP = 0, and STP = 1

26.3.4 Double-Speed Operation and Frequency of 6 Times the Bit Rate

When the CCR2.ABCS bit is set to 1, the SCI operates on the bit rate twice that of when ABCS is set to 0. When the CCR2.BGDM bit is set to 1, the cycle of the base clock is half and the bit rate is double that of when BGDM is set to 0. When the CCR3.CKE[1] bit is set to 0 and the on-chip baud rate generator is selected, setting the ABCS and BGDM bits to 1 allows the SCI to operate at a bit rate four times that when the ABCS and BGDM bits are set to 0.

When the CCR2.ABCSE bit is set to 1, the number of base clock pulses is 6 during a period of 1 bit, and the base clock frequency is half and the SCI operates at a bit rate 16/3 times that when CCR2.ABCS = 0, CCR2.BGDM = 0, and CCR2.ABCSE = 0.

As shown by Formula (1) in section 26.3.2. Receive Data Sampling Timing and Reception Margin in Asynchronous Mode, the reception margin decreases when the CCR2.ABCS or CCR2.ABCSE bit is set to 1. Therefore, if the target bit rate can be obtained with ABCS or ABCSE set to 0, it is recommended that you use the SCI with ABCS and ABCSE set to 0.

26.3.5 CTS and RTS Functions

The CTS function controls transmission using the CTSn pin input. Setting the CCR1.CTSE bit to 1 enables the CTS function. For the functions of CTS and RTS, you can select the alternate setting to set CTSn_RTSn pin as a multiplexed pin that uses either function with one pin or the dedicated setting that uses each function independently with two pins at CTSn pin for CTSn signal and CTSn_RTSn pin for RTSn signal. This setting is done with the CCR1.CTSPEN bit.

When the CTS function is enabled, placing a low level on the CTSn_RTSn pin causes transmission to start.

If FIFO is used and CTSn_RTSn signal is held high before transmission, transmission will not start, so the number of TDR registers written, and the number of data stored are the same (unlike using clock synchronous FIFO).

Driving the CTSn_RTSn pin high while transmission is in progress does not affect transmission of the current frame.

In the RTS function, which uses output on the CTSn_RTSn pin, a low level is output when reception becomes possible. Conditions for output of the low and high levels are shown in this section.

[Conditions for low level output]

Satisfaction of all conditions are listed in this section.

Non-FIFO selected

- The value of the CCR0.RE bit is 1
- When the next reception is possible
  - There are no received data yet to be read and not receiving.
  - CSR.ORER, FER, PER flags are all 0

FIFO selected

- The value of the CCR0.RE bit is 1
- When the next reception is possible
  - When the quantity of receive data written in receive-FIFO(RDR) are less than the setting value of FCR.RSTRG[4:0]
  - CSR.ORER(RDR.ORER) is 0

[Condition for high level output]

- The conditions for low-level output are not satisfied

26.3.6 Address Match (Receive Data Match Detection) Function

The address match function can be used only in asynchronous mode.

If the CCR0.DCME bit is set to 1[*2], when one frame of data is received, the SCI compares that received data with the data set in CCR4.CMPD. If the SCI detects a match to the comparison data (CCR4.CMPD[*1]) with the received data, the SCI can issue the SCIn_RXI interrupt request.

If the CCR3.MP bit is set to 0, comparison occurs only for valid data in receive format. In multi-processor mode (CCR3.MP bit = 1), if the CCR0.IDSEL bit is set to 1, receive data where the MPB bit is 1 is subject to comparison for address match and receive data where the MPB bit is 0 is always treated as a non-match.

If the CCR0.IDSEL bit is set to 0, SCI performs address match detection regardless of the MPB bit value of the received data.

Until SCI detects a match to the comparison data (CCR4.CMPD[*1]) with receive data, received data is skipped (discarded), and the SCI cannot detect a parity error or framing error.

When SCI detects a match, the CCR0.DCME bit is automatically cleared, and the CSR.DCMF flag is set to 1. If the CCR0.IDSEL bit is set to 1, the CCR0.MPIE bit is automatically cleared. If CCR0.IDSEL is set to 0, the value of the CCR0.MPIE bit is retained. If the CCR0.RIE bit is set to 1, the SCI issues an SCIn_RXI interrupt request.

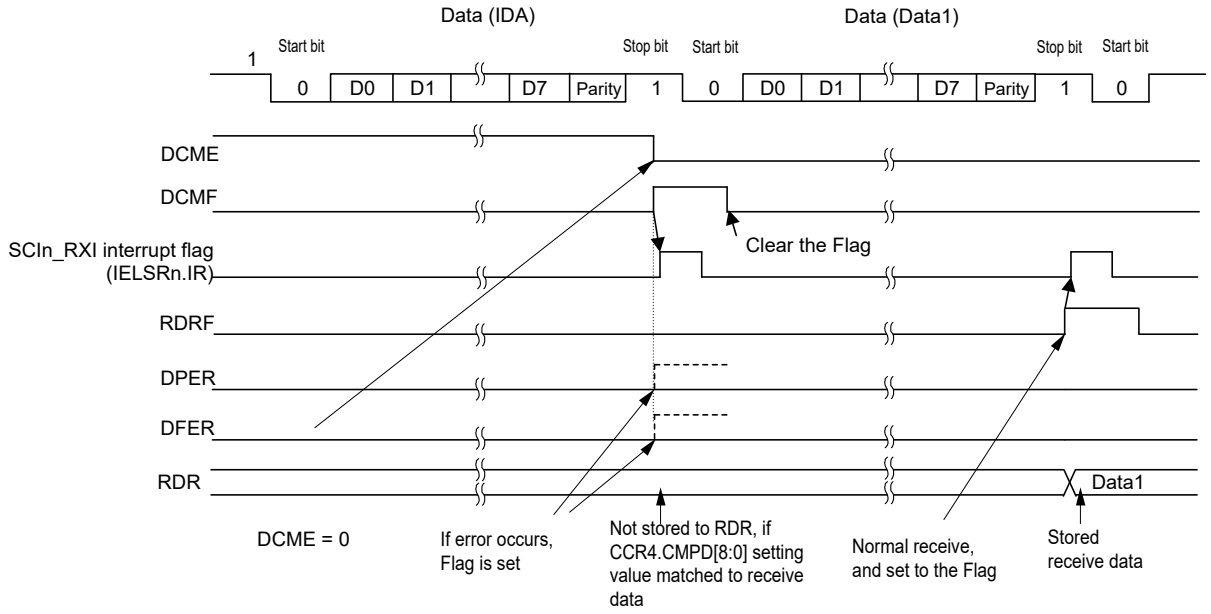
If the SCI detects a framing error in the receive data for which a match is detected, the CSR.DFER flag is set to 1, and if the SCI detects a parity error in that frame, the CSR.DPER flag is set to 1. The compared receive data and MPB bit is not stored in the RDR register, and CSR.RDRF remains 0.

After the SCI detects a match, and CCR0.DCME is automatically cleared, the SCI receives the next data continuously based on the current register setting.

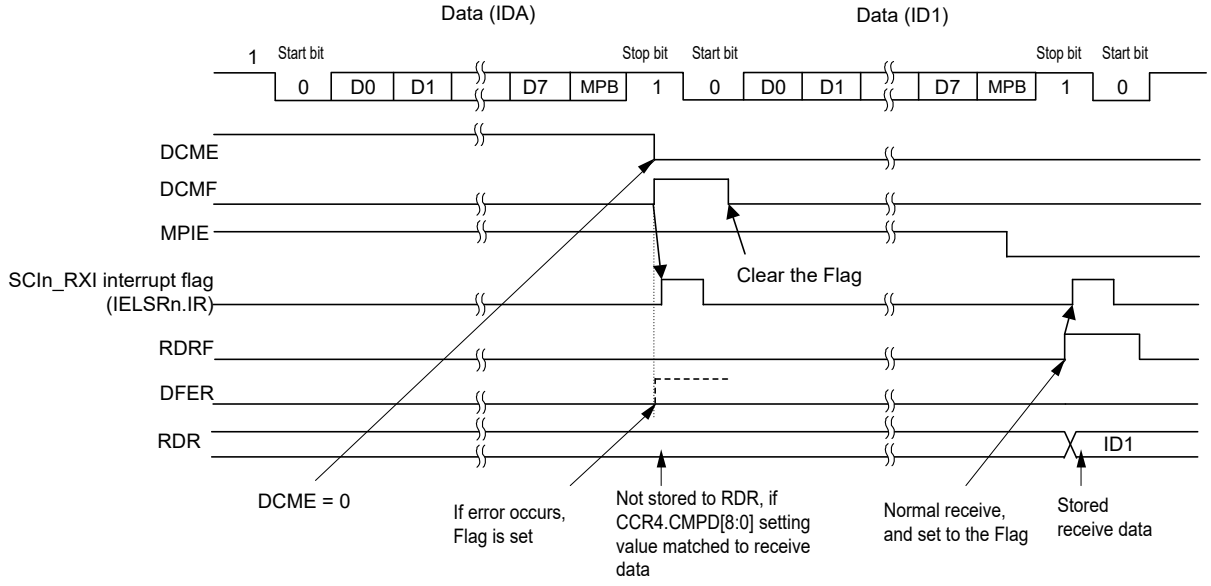
When the CSR.DFER or CSR.DPER flag is set, the address match is not performed. Before enabling the address match function, set the DCCR.DFER and DCCR.DPER flags to 0.

Examples of the address match function are shown in Figure 26.7 and Figure 26.8.

Note 1. This comparative target can select one length of 3 types: CMPD[6:0] with 7-bit length, CMPD[7:0] with 8-bit length, and CMPD[8:0] with 9-bit length.
Note 2. Set the CCR0.DCME bit to 1 before receiving the start bit of the received frame that performs address matching.



(a) Example of Compare matched between receive data and CCR4.CMPD[8:0] (8bit length/parity/Normal mode)



(b) Example of Compare matched between receive data and CCR4.CMPD[8:0] (8bit length/Multi-processor mode)

Figure 26.7 Example of address match (1) normal mode

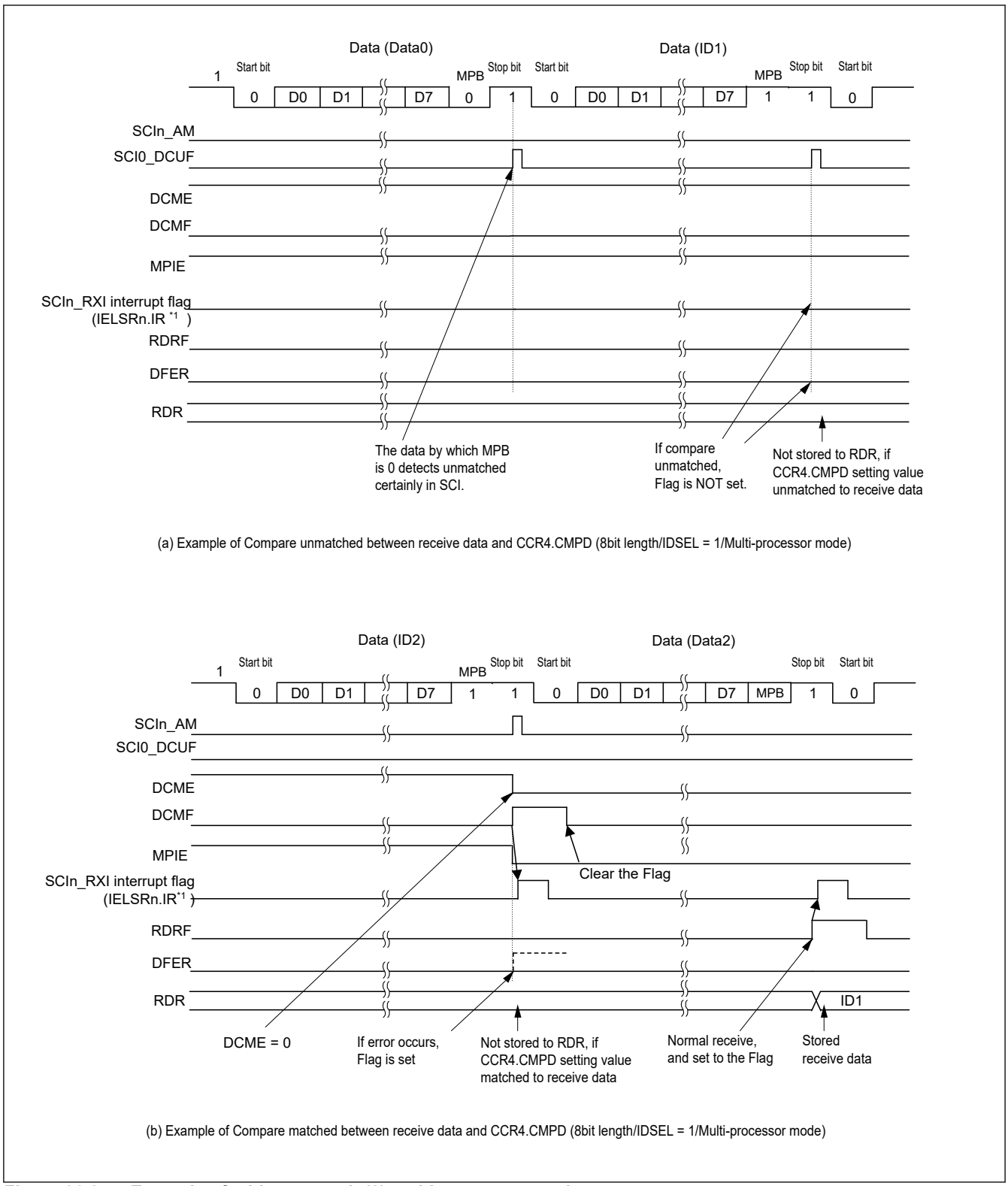


Figure 26.8 Example of address match (2) multi-processor mode

26.3.7 SCI Initialization in Asynchronous Mode

Before transmitting and receiving data, start by writing the initial value 0 to the CCR0.TE and CCR0.RE (or writing the initial value to CCR0), then continue through the SCI initialization procedure (select non-FIFO or FIFO) shown in Table 26.26 and Table 26.27. Whenever the operating mode or transfer format is to be changed, the CCR0.TE and CCR0.RE bits must be initialized before the change is made.

When the external clock is used in asynchronous mode, ensure that the clock signal is supplied during initialization.

Note: Setting the CCR0.RE bit to 0 initializes neither the ORER, FER, RDRF, RDAT, PER, and DR flags in CSR nor RDR. When FIFO selected, even if the TE bit is set as 0, the TEND flag for the selected FIFO buffer is not initialized.

Be also careful at the time of change in the operation mode.

Note: Switching the value of the CCR0.TE bit from 0 to 1 while the CCR0.TIE bit is 1 leads to the generation of an SCIn_TXI interrupt request.

Table 26.26 Example flow of SCI initialization in asynchronous mode with non-FIFO selected

No.	Step Name	Description
1	Start initialization	—
2	Set CCR0	Set CCR0.TEIE, TIE, RIE, TE, RE to 0. If you have not changed from the initial settings, you can skip this step.
3	Set CCR3	Set up following function and communication mode. Driver control function for RS-485, FIFO no-use, Multi-Processor mode, Communication mode (MOD[2:0] = 000b) Transmission / reception format Clock enable (Leave the initial value when outputting the clock) Leave unused bits at their initial values.
4	Set CCR2	Set up the bit-rate-modulation function*1 *2, select clock, set bit rate*2
5	Set CCR1	Set up the Noise-filter function, the loop-back function, communication pin status, the parity check function, and the CTSn_RTSn function.
6	Set CCR4	Set up the adjust sampling timing function and the adjust transmit timing function.
7	Set the I/O port functions	Make I/O port settings to enable input and output functions as required for TXDn, RXDn, and SCKn pins.
8	Set CCR3	Set clock enable bit (CKE[1:0]) at this step when outputting the clock. After this register setting, the clock pin will be in the output state immediately.
9	Set CFCLR	Write 1 to the following bits and clear the corresponding flag. CFCLR.RDRFC,FERC,PERC,MFFC,ORERC,DFERC,DPERC,DCMFC,ERSC If it's initialization flow after a reset, you can skip this step.
10	Set CCR0	Set the TE or RE bit to 1. To enable interrupts, set the TE bit and TIE bit, and the RE bit and RIE bit to 1 with one instruction at the same time. Setting the TE and RE bits allows TXDn and RXDn to be used.
11	Initialization completed	—

Note 1. If you do not use the bit rate modulation function, you do not need to set it.
Note 2. If you use an external clock, you do not need to set it.

Table 26.27 Example flow of SCI initialization in asynchronous mode with FIFO selected (1 of 2)

No.	Step Name	Description
1	Start initialization	—
2	Set CCR0	Set CCR0.TEIE, TIE, RIE, TE, RE to 0. If you have not changed from the initial settings, you can skip this step.
3	Set CCR3	Set up following function and communication mode. Driver control function for RS-485, FIFO use, Multi-Processor mode, Communication mode (MOD[2:0] = 000b) Transmission / reception format Clock enable (Leave the initial value when outputting the clock) Leave unused bits at their initial values.
4	Set CCR2	Set up the bit-rate-modulation function*1 *2, select clock, set bit rate*2
5	Set CCR1	Set up the Noise-filter function, the loop-back function, communication pin status, the parity check function, and the CTSn_RTSn function.
6	Set CCR4	Set up the adjust sampling timing function and the adjust transmit timing function.
7	Set FCR	Set the TFRST and RFRST to 1 to empty FIFO. Set the DRES, TTRG[4:0], RTRG[4:0], and RSTRG[4:0] bits
8	Set the I/O port functions	Make I/O port settings to enable input and output functions as required for TXDn, RXDn, and SCKn pins.

Table 26.27 Example flow of SCI initialization in asynchronous mode with FIFO selected (2 of 2)

No.	Step Name	Description
9	Set CCR3	If you select a clock output in asynchronous mode, set the CKE[1:0] bit here. After setting the register, the clock pin will be in the output state immediately. But the clock operates after setting TE or RE to 1.
10	Set CFCLR	Write 1 to the following bits and clear the corresponding flag. CFCLR.RDRFC, FERC, PERC, MFFC, ORERC, DFERC, DPERC, DCMFC, ERSC If it's initialization flow after a reset, you can skip this step.
11	Set FFCLR	Write 1 to The FFCLR.BRKC,DRC and clear the corresponding flag. If it's initialization flow after a reset, you can skip this step.
12	Set CCR0	Set the TE or RE bit to 1. To enable interrupts, set the TE bit and TIE bit, and the RE bit and RIE bit to 1 with one instruction at the same time. Setting the TE and RE bits allows TXDn and RXDn to be used.
13	Initialization completed	—

Note 1. If you do not use the bit rate modulation function, you do not need to set it.
Note 2. If you use an external clock, you do not need to set it.

Figure 26.9 shows an example of the timing when data is transmitted after reset is released, and SCI is set to asynchronous mode according to Table 26.26 or Table 26.27. As shown in the figure, when the pin function is set to the TXDn pin, the CCR0.TE bit is 0, so the pin is high impedance. When transmit data is written after setting the CCR0.TE bit to 1, data transmission starts. There is a transmission wait time from writing TDR to data transmission starts. In asynchronous mode, TXDn is high during this period.

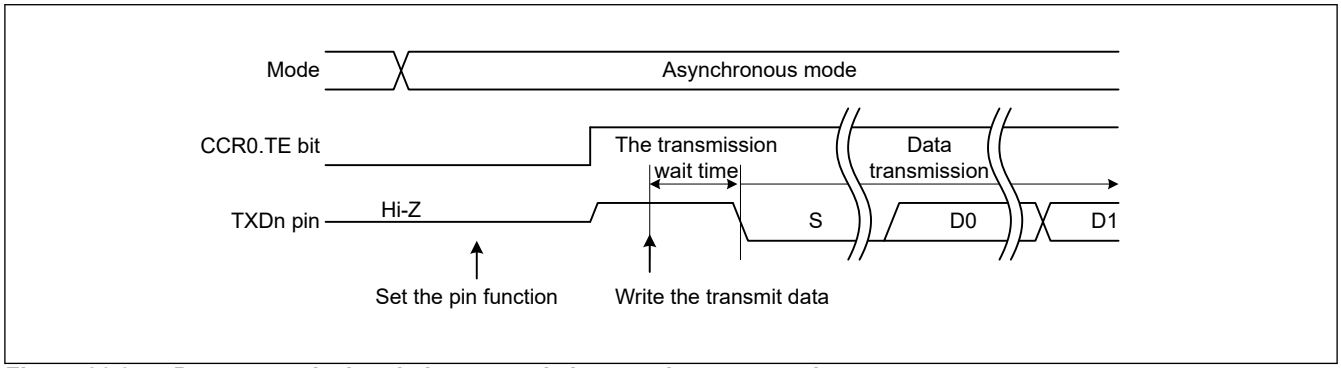


Figure 26.9 Data transmission timing example in asynchronous mode

26.3.8 Serial Data Transmission in Asynchronous Mode

(1) Non-FIFO selected

Figure 26.10, Figure 26.11, Figure 26.12 and Figure 26.13 show examples of serial transmission in asynchronous mode.

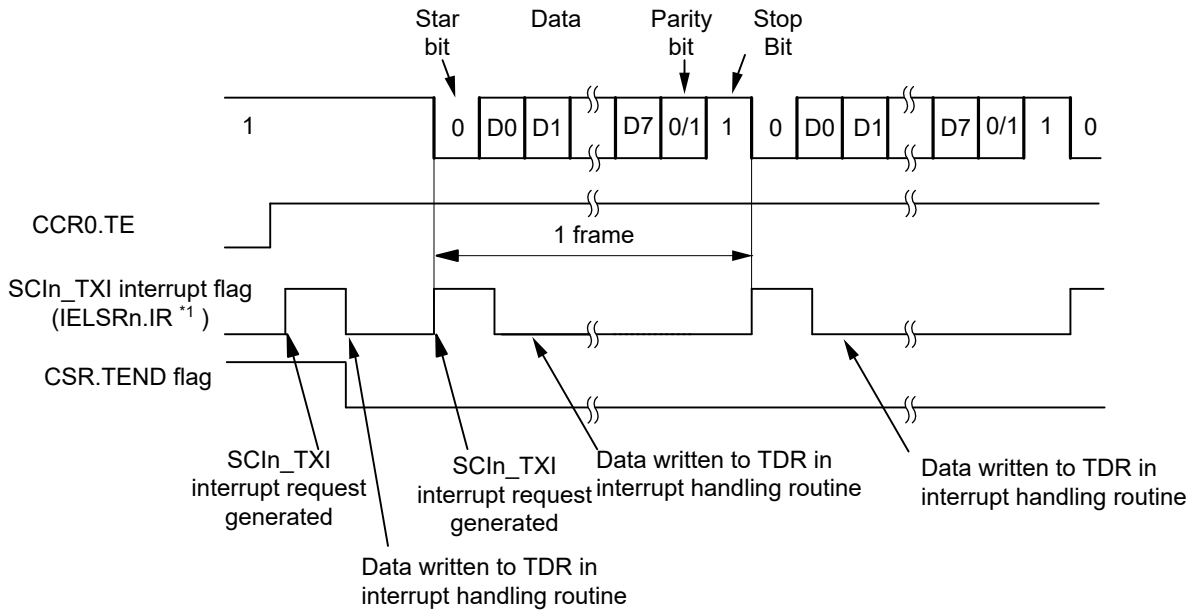
In serial transmission, the SCI operates as described in this section.

1. The SCI transfers data from the TDR register to the TSR register when data is written to TDR in the SCIn_TXI interrupt handling routine.
   The SCIn_TXI interrupt request at the beginning of transmission is generated when the CCR0.TE and CCR0.TIE bits are set to 1 simultaneously by a single instruction.
2. Transmission starts after the CCR1.CTSE bit is set to 0 (CTS function is disabled) or a low level on the CTSn_RTSn pin causes data transfer from the TDR register to the TSR register. If the CCR0.TIE bit is 1, an SCIn_TXI interrupt request is generated. Continuous transmission is possible by writing the next transmit data to the TDR register in the SCIn_TXI interrupt handling routine before transmission of the current transmit data is complete. When SCIn_TEI interrupt requests are in use, set the CCR0.TIE bit to 0 (SCIn_TXI interrupt requests are disabled) and the CCR0.TEIE bit to 1 (an SCIn_TEI interrupt request is enabled) after the last of the data to be transmitted is written to the TDR register from the handling routine for SCIn_TXI requests.
3. Data is sent from the TXDn pin in the following order:
   - Start bit
   - Transmit data

- Parity bit or multi-processor bit (can be omitted depending on the format)
- Stop bit

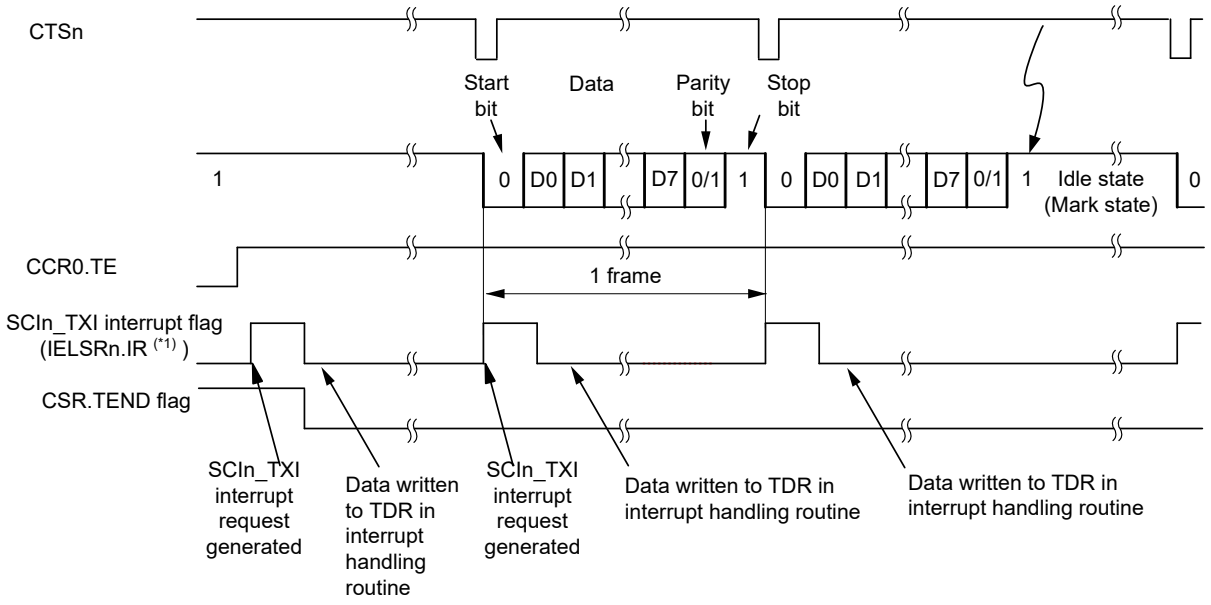
4. The SCI checks for update of the TDR register on output of the stop bit.
5. When the TDR register is updated, setting the CCR1.CTSE bit to 0 (CTS function is disabled) or a low level input on the CTSn_RTSn pin causes transfer of the next transmit data from the TDR register to the TSR register and transmission of the stop bit, after which serial transmission of the next frame starts.
6. If the TDR register is not updated, the CSR.TEND flag is set to 1, the stop bit is sent, and the mark state is entered, in which 1 is output. If the CCR0.TEIE bit is 1, the CSR.TEND flag is set to 1 and an SCIn_TEI interrupt request is generated.

Figure 26.10, Figure 26.11, Figure 26.12 and Figure 26.13 show examples of serial transmission in asynchronous mode.



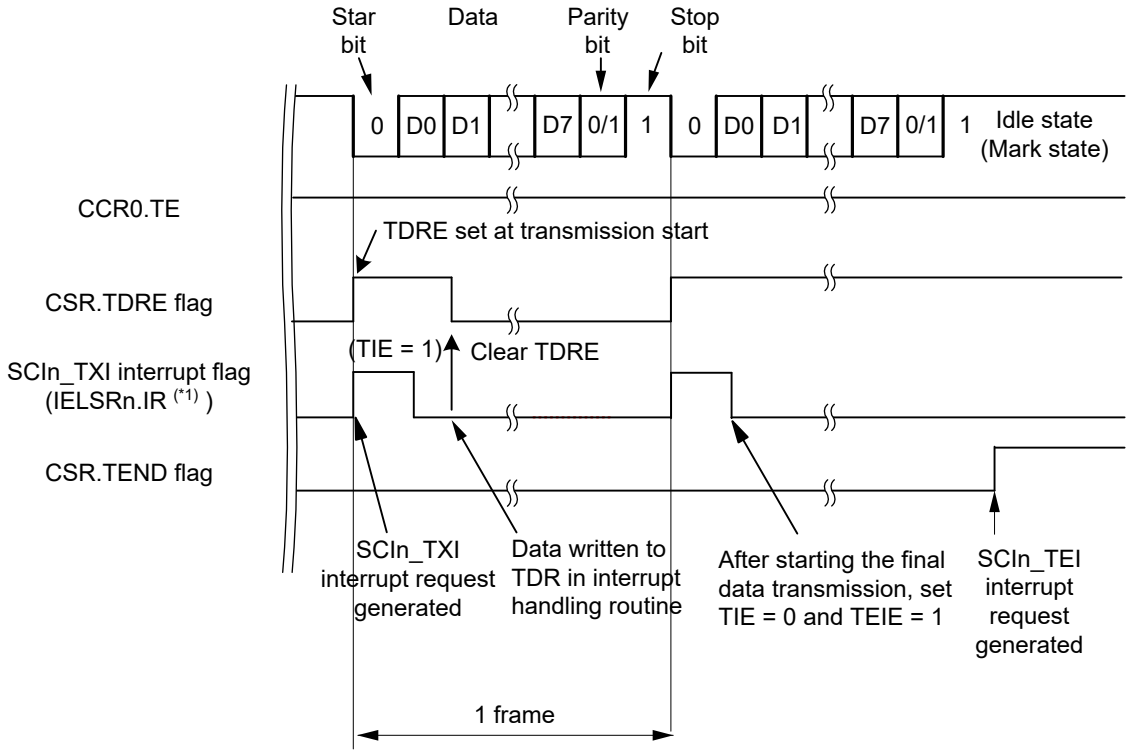
Note 1. See section 12, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 26.10 Example operation for serial transmission in asynchronous mode (1) with 8-bit data, parity bit, 1 stop bit, CTS function not used, and at the beginning of transmission



Note 1. See section 12, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 26.11 Example operation for serial transmission in asynchronous mode (2) with 8-bit data, parity bit, one stop bit, CTS function used, and at the beginning of transmission



Note 1. See section 12, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 26.12 Example operation for serial transmission in asynchronous mode (3) with 8-bit data, parity bit, one stop bit, CTS function not used, and from the middle of transmission until transmission completion

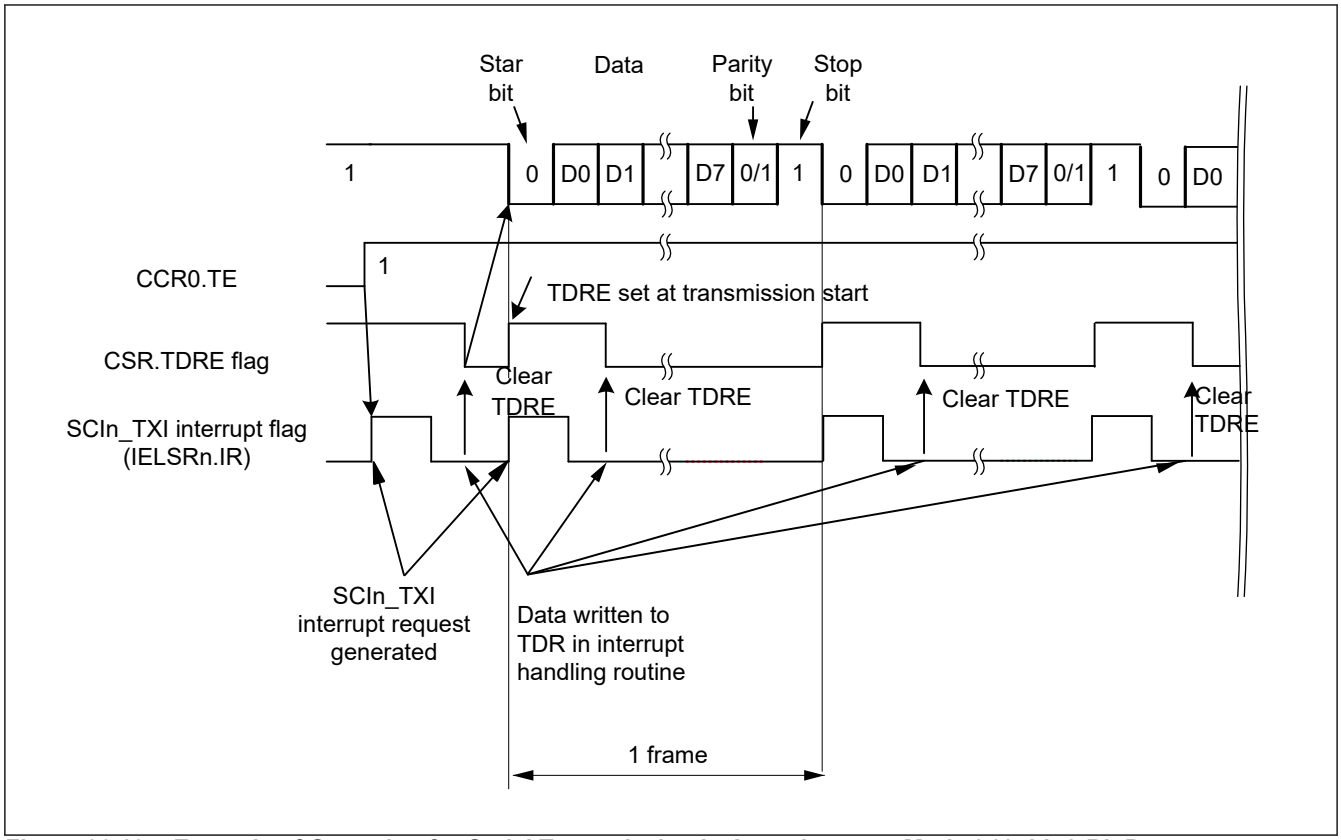
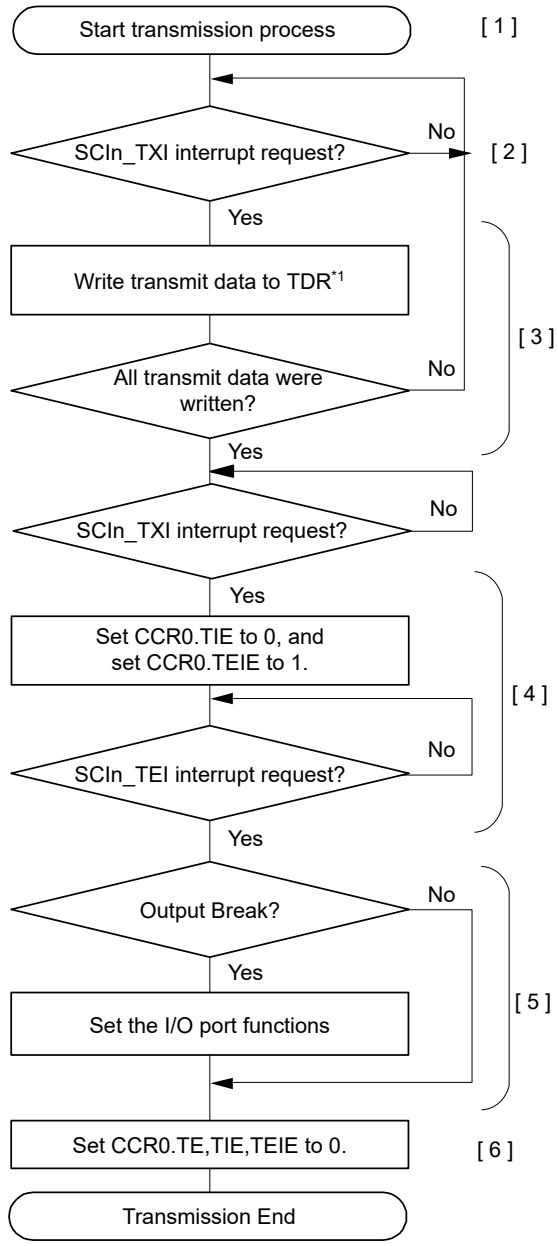


Figure 26.13 Example of Operation for Serial Transmission in Asynchronous Mode (4)(with 8-Bit Data, Parity, 1 Stop Bit, CTS Function Not Used, from the Middle of Transmission until Transmission Completion)



[1] Start transmission process

[2] SCIn_TXI interrupt request:
The transmit-data-empty-interrupt(SCIn_TXI) is generated at following conditions.
When set TE and TIE to 0 at same time.
When transmission-data forwards to TSR from TDR.

[3] Serial transmission continuation procedure:
To continue serial transmission write transmit data to TDR*1 once by using a SCIn_TXI.
Transmit data can also be written by activating the DMAC or DTC.

[4] If you use TEI, set TIE = 0 and TEIE = 1 after writing the last-transmit-data to TDR*1 and transmission is started.

[5] Break output at the end of serial transmission:
To output a break, it is necessary to set SPB2IO and SPB2DT, for output status setting when TE is 0.
(If you set these bits at initialization, you can skip this step.)

[6] Set CCR0.TE,TIE,TEIE to 0.

Note 1. In case of byte access with data length 9 bits, write transmit data in order of TDR[15:8] and TDR[7:0].

Figure 26.14 Example flow of serial transmission in asynchronous mode with non-FIFO selected

(2) FIFO selected

Figure 26.15 shows an example of a data format that is written to TDR register in asynchronous mode with FIFO selected.

Data corresponding to the data length is set to TDR[8:0]. Write 0 for unused bits. Write in order from TDR[15:8] to TDR[7:0].

Data Length	Register setting		Transmit data in TDR[15:0]															
	CCR3.CHR[1:0]		b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
7 bit	1	1	—	—	—	—	—	—	MPBT	—	—	TDAT[6:0]						
8 bit	1	0	—	—	—	—	—	—	MPBT	—	TDAT[7:0]							
9 bit	0	Don't Care	—	—	—	—	—	—	MPBT	TDAT[8:0]								

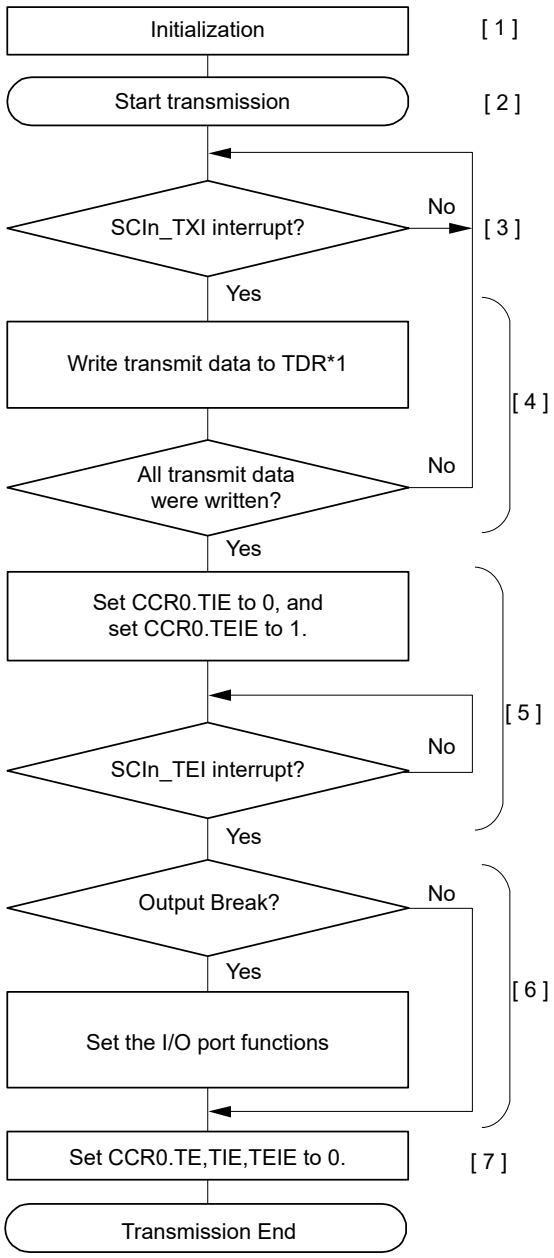
Note: —: Invalid. The write value should be 0.

Figure 26.15 Data format written to transmit-FIFO(TDR) with FIFO selected

In serial transmission, the SCI operates as described in this section.

1. The SCI transfers data from the TDR register to the TSR register when data is written to TDR in the SCIn_TXI interrupt handling routine. The amount of data that can be written to TDR is 16 minus FTSR.T[5:0] bytes. The SCIn_TXI interrupt request at the beginning of transmission is generated when the CCR0.TE and CCR0.TIE bits are set to 1 simultaneously by a single instruction.
2. Transmission starts after the CCR1.CTSE bit is set to 0 (CTS function is disabled) or a low level on the CTSn_RTSn pin causes data transfer from the TDR register to the TSR register. When the amount of transmit data written in TDR is equal to or less than the specified transmit triggering number, CSR.TDFE is set to 1. If the CCR0.TIE bit is 1, an SCIn_TXI interrupt request is generated. Continuous transmission is possible by writing the next transmit data to TDR in the SCIn_TXI interrupt handling routine before transmission of the current transmit data is complete. When SCIn_TEI interrupt requests are in use, set the CCR0.TIE bit to 0 (SCIn_TXI interrupt requests are disabled) and the CCR0.TEIE bit to 1 (an SCIn_TEI interrupt request is enabled) after the last of the data to be transmitted is written to the TDR register from the handling routine for SCIn_TXI requests.
3. Data is sent from the TXDn pin in the following order:
   - Start bit
   - Transmit data
   - Parity bit or multi-processor bit (can be omitted depending on the format)
   - Stop bit
4. On output of the stop bit, the SCI checks whether non-transmitted data remains in the TDR register.
5. When data is set to transmit-FIFO (TDR), setting the CCR1.CTSE bit to 0 (CTS function is disabled) or a low level input on the CTSn_RTSn pin causes transfer of the next transmit data from TDR to TSR and transmission of the stop bit, after which serial transmission of the next frame starts.
6. If data is not set in transmit-FIFO (TDR), the TEND flag in CSR is set to 1, the stop bit is sent, and the mark state is entered in which 1 is output. If the CCR0.TEIE bit is 1, the CSR.TEND flag is set to 1 and an SCIn_TEI interrupt request is generated.

Figure 26.16 shows an example flow of serial transmission in asynchronous mode with FIFO selected.



[1] Initialize SCI.
To enable SCIn_TXI interrupt, set TE and TIE to 1 with one instruction at the same time.

[2] Set transmit-data to the transmit-FIFO(TDR).
The number of transmittable data that can be set is16 .

[3] TXI interrupt request:
The transmit-data-empty-interrupt(SCIn_TXI) is generated
when transmission-data forwards to TSR from the transmit-FIFO(TDR) and the number of transmit-data written in the transmit-FIFO is equal to or less than the transmit-FIFO threshold.

[4] Serial transmission continuation procedure:
To continue serial transmission, write all transmit data to the transmit-FIFO(TDR) by using a SCIn_TXI interrupt and clear TDRE flag to 0. (Transmit data can also be written to the transmit- FIFO(TDR) by activating the DMAC or DTC. In this case, the TDRE flag is cleared automatically, so please don't operate the TDRE flag.)
The number of transmission data it's possible to write in is 16 - (the number of stored transmit FIFO data).

[5] If you use SCIn_TEI, set TIE = 0 and TEIE = 1 after writing the last-transmit-data to TDR*1.

[6] Break output at the end of serial transmission:
To output a break, it is necessary to set SPB2IO and SPB2DT, for output status setting when TE is 0.
(If you set these bits at initialization, you can skip this step.)

[7] Set CCR0.TE,TIE,TEIE to 0.

Note 1. In case of byte access with data length 9 bits, write transmit data in order of TDR[15:8] and TDR[7:0].

Figure 26.16 Example flow of serial transmission in asynchronous mode with FIFO selected

26.3.9 Serial Data Reception in Asynchronous Mode

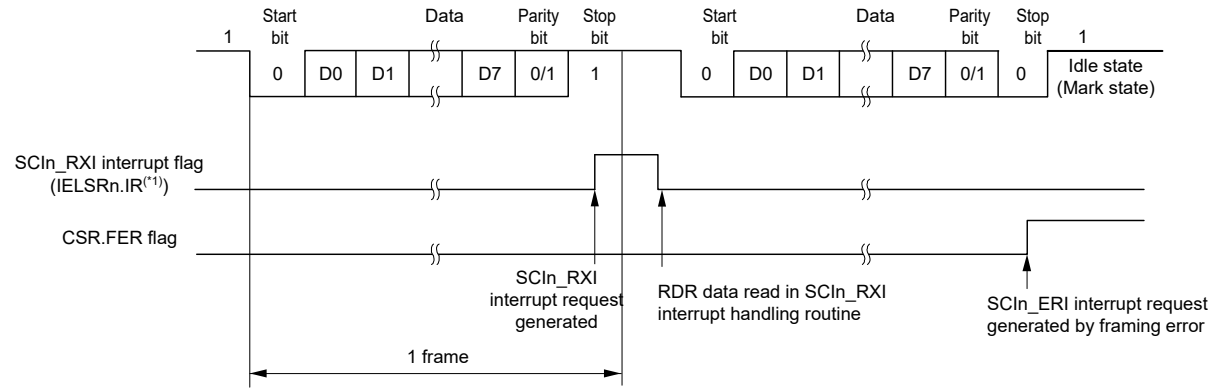
(1) Non-FIFO selected

Figure 26.17 and Figure 26.18 show an example of the operation for serial data reception in asynchronous mode.

In serial data reception, the SCI operates as follows:

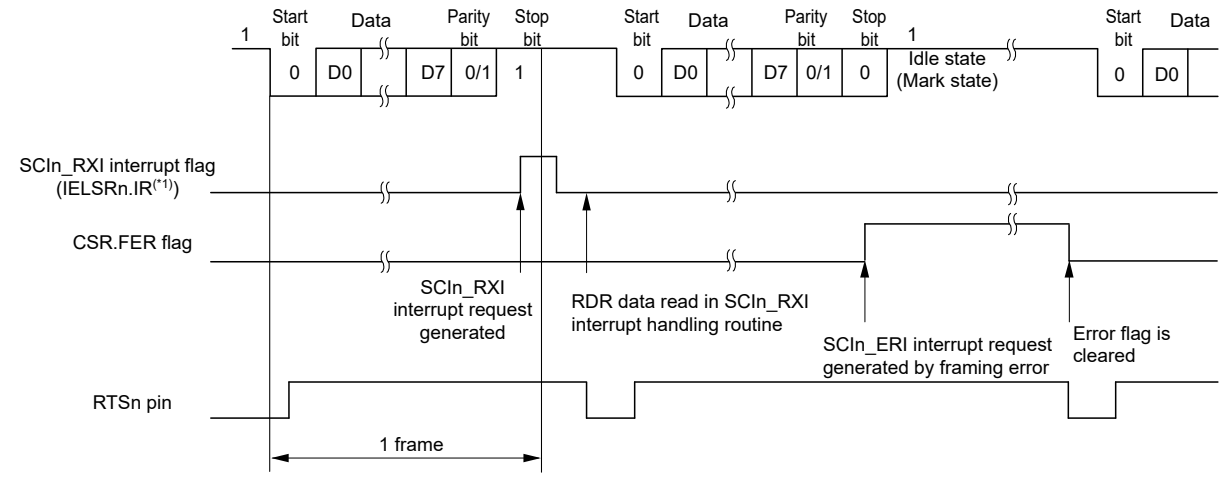
1. When the value of the CCR0.RE bit becomes 1, the output signal on the CTSn_RTSn pin goes low.
2. The SCI monitors the communications line and when it detects a start bit, the SCI performs internal synchronization, stores receive data in RSR, and checks the parity bit and stop bit.
3. If an overrun error occurs, the CSR.ORER flag is set to 1. If the CCR0.RIE bit is 1, an SCIn_ERI interrupt request is generated. Receive data is not transferred to the RDR register.

4. If a parity error is detected, the CSR.PER flag is set to 1 and receive data is transferred to the RDR register. If the SCR.RIE bit is 1, an SCIn_ERI interrupt request is generated.
5. If a framing error is detected, the CSR.FER flag is set to 1 and receive data is transferred to the RDR register. If the CCR0.RIE bit is 1, an SCIn_ERI interrupt request is generated.
6. When reception finishes successfully, receive data is transferred to the RDR register. If the CCR0.RIE bit is 1, an SCIn_RXI interrupt request is generated. Continuous reception is enabled by reading the receive data transferred to the RDR register in the SCIn_RXI interrupt handling routine before reception of the next receive data is complete. Reading the received data that was transferred to the RDR register causes the CTSn_RTSn pin to output low.



Note 1. See section 12, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 26.17 Example of SCI operation for serial reception in asynchronous mode (1) when the RTS function is not used, and with 8-bit data, parity bit, and 1 stop bit



Note 1. See section 12, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 26.18 Example of SCI operation for serial reception in asynchronous mode (2) when RTS function is used, and with 8-bit data, parity bit, and 1 stop bit

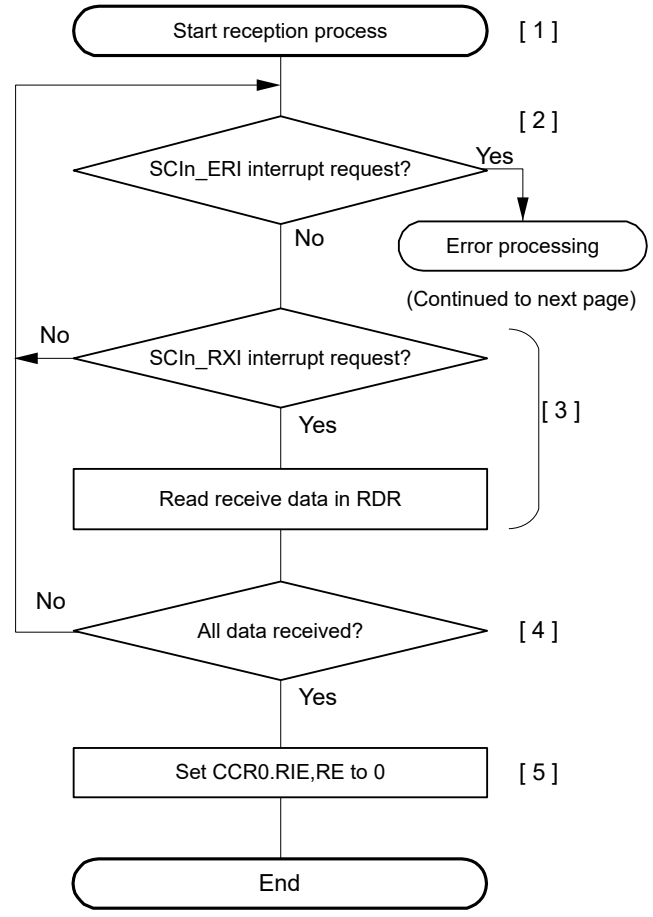
Table 26.28 lists the states of the flags in the CSR status register and receive data handling when a receive error is detected.

If a receive error is detected, an SCIn_ERI interrupt request is generated but an SCIn_RXI interrupt request is not generated. Data reception cannot be resumed while the receive error flag is 1. Accordingly, set the ORER, FER, and PER bits to 0 before resuming reception. In addition, be sure to read the RDR register during overrun error processing. When a reception is forced to terminate by setting the CCR0.RE bit to 0 during operation, read the RDR register because received data that is not yet read might be left in the RDR.

Figure 26.19 and Figure 26.20 show example flows of serial data reception.

Table 26.28 Flags in CSR Status Register and receive data handling

Flags in the CSR Status Register			Receive data	Receive error type
ORER	FER	PER		
1	0	0	Lost	Overrun error
0	1	0	Transferred to RDR	Framing error
0	0	1	Transferred to RDR	Parity error
1	1	0	Lost	Overrun error + framing error
1	0	1	Lost	Overrun error + parity error
0	1	1	Transferred to RDR	Framing error + parity error
1	1	1	Lost	Overrun error + framing error + parity error



[1] Start reception process

[2] Receive error processing and break detection: If a receive error occurs, an SCIn_ERI interrupt is generated. After performing the appropriate error processing, be sure to set the ORER, PER, and FER flags to 0. Reception cannot be resumed if any of these flags is set to 1. In the case of a framing error, a break can be detected by reading the value of the input port corresponding to the RXDn pin.

[3] Read the receive data in RDR once in the SCIn_RXI interrupt handling routine.

[4] Serial reception continuation procedure: To continue serial reception, before the stop bit of the current frame is received, read data from RDR in the SCIn_RXI interrupt handling routine. The RDR data can also be read by activating the DMAC or DTC.

[5] Set CCR0.RIE,RE to 0.

Note 1. Do not set RE to 0 before reading RDR.

Figure 26.19 Example flow of serial reception in asynchronous mode with non-FIFO selected and Address Matching Disabled (1)

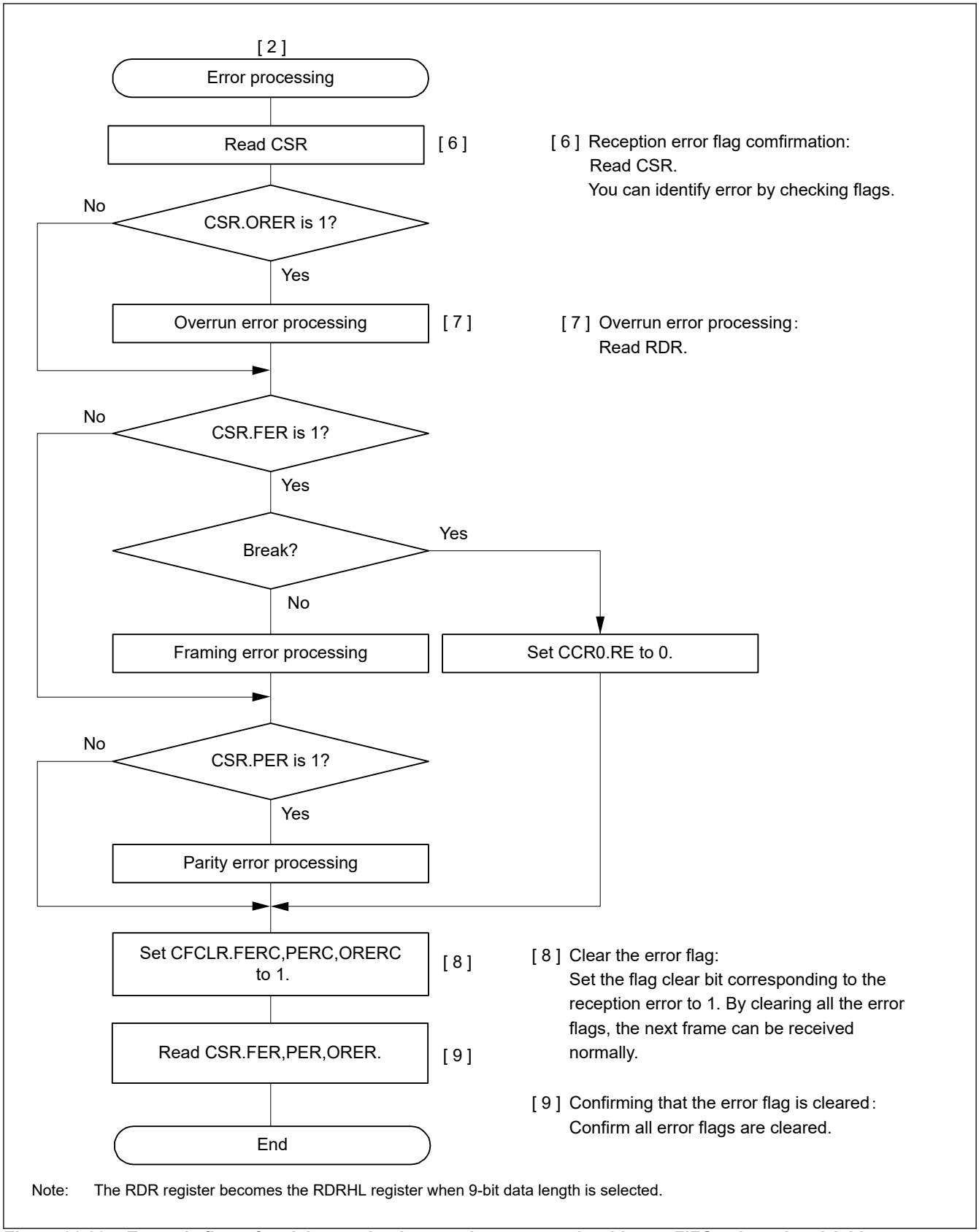


Figure 26.20 Example flow of serial reception in asynchronous mode with non-FIFO selected and Address Matching Disabled (2)

(2) FIFO selected

Figure 26.21 shows an example of a data format that is written to Receive-FIFO(RDR) register in asynchronous mode.

In asynchronous mode, 0 is written to the MPB bit in the RDR register. Data that corresponds to the data length is written to RDR. Unused bits are written as 0. If software reads RDR, the SCI updates FER, PER, and receive data (RDAT[8:0]) in the RDR register with the next data. The flags RDF, ORER, and DR in the RDR register always reflect the associated flags in the CSR register.

Data Length	Register Setting CCR3.CHR[1:0]		Receive flag in RDR[31:0], MPB, RDAT[8:0]															
			b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
7bit	1	1	-	-	-	FFER	FPER	DR	MPB	0	0	RDAT[6:0]						
8bit	1	0	-	-	-	FFER	FPER	DR	MPB	0	RDAT[7:0]							
9bit	0	Don't care	-	-	-	FFER	FPER	DR	MPB	RDAT[8:0]								
	CCR3.CHR[1:0]		b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
7bit	1	1	-	-	-	FER	PER	-	-	ORER	-	-	-	-	-	-	-	-
8bit	1	0	-	-	-	FER	PER	-	-	ORER	-	-	-	-	-	-	-	-
9bit	0	Don't care	-	-	-	FER	PER	-	-	ORER	-	-	-	-	-	-	-	-

Note: 0 is always read from the MPB flag (RDR [9] bit).
When a 7-bit data length is selected, 0 is read from the RDAT [8:7] bits.
When 8-bit data length is selected, 0 is read from the RDAT [8] bit.

Figure 26.21 Data format stored in receive-FIFO(RDR) with FIFO selected

Table 26.29 lists the states of the flags in CSR status register and receive data handling when a receive error is detected with FIFO selected. Figure 26.22 and Figure 26.23 show samples of flowcharts for serial data reception with FIFO selected.
In serial data reception, the SCI operates as follows:

1. When the value of the CCR0.RE bit becomes 1, the output signal on the CTSn_RTSn pin goes low.
2. The SCI monitors the communications line and, when it detects a start bit, the SCI performs internal synchronization, stores receive data in the RSR register.
3. If an overrun error occurs during normal communications, the CSR.ORER flag is set to 1. If the CCR0.RIE bit is 1, an SCIn_ERI interrupt request is generated. Receive data is not transferred to the RDR register.
4. If a parity error is detected, the PER flag and receive data are transferred to the RDR register. If the CCR0.RIE bit is set to 1, an SCIn_ERI interrupt request is generated.
5. If a framing error is detected, the FER flag and receive data are transferred to the RDR register. If the CCR0.RIE bit is set to 1, an SCIn_ERI interrupt request is generated.
6. After a framing error is detected and when SCI detects that the continuous receive data is zero for one frame, reception stops.
7. When the amount of data stored in the RDR register falls below the specified receive triggering number, and the next data is not received after 15 ETUs from the last stop bit in asynchronous mode, the FRSR.DR flag is set to 1. When the CCR0.RIE bit is 1 and the FCR.DRES bit is 0, the SCI generates an SCIn_RXI interrupt request. When the FCR.DRES bit is 1, SCI generates an SCIn_ERI interrupt request.
8. When reception finishes successfully, receive data is transferred to the RDR register. The RDRF bit is set to 1 when the amount of receive data written to RDR is equal to or greater than the specified receive triggering number. If the CCR0.RIE bit is 1, an SCIn_RXI interrupt request is generated. Continuous reception is enabled by reading the receive data transferred to the RDR register in the SCIn_RXI interrupt handling routine, before an overrun error occurs. If the received data that is transferred to RDR is less than the RTS trigger number, the CTSn_RTSn pin outputs low.

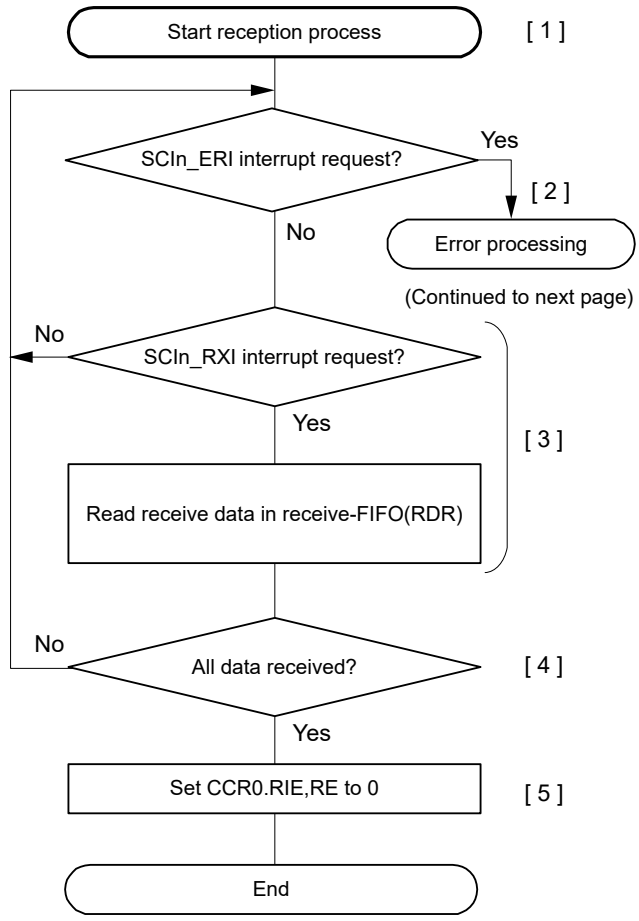
Table 26.29 Flags in the CSR Status Register and Receive Data Handling (FIFO selected) (1 of 2)

CSR value			Receive-FIFO (RDR)	Receive Error Type
ORER	FER*1	PER*1	RDAT[8:0]	
1	0	0	Lost	Overrun error
0	1	0	Transferred RDR	Framing error
0	0	1	Transferred RDR	Parity error

Table 26.29 Flags in the CSR Status Register and Receive Data Handling (FIFO selected) (2 of 2)

CSR value			Receive-FIFO (RDR)	Receive Error Type
ORER	FER*1	PER*1	RDAT[8:0]	
1	1	0	Lost	Overrun error + Framing error
1	0	1	Lost	Overrun error + Parity error
0	1	1	Transferred RDR	Framing error + Parity error
1	0	0	Lost	Overrun error + Framing error + Parity error

Note 1. This flag indicates whether there is an error in received data when reception is completed.



[1] Start reception process

[2] Receive error processing and break detection: If a receive error occurs, an SCIn_ERI interrupt is generated. After performing the appropriate error processing, be sure to set the ORER flags to 0. Reception cannot be resumed if ORER is set to 1. Reception can continue even if PER,FER,DR flags are 1.

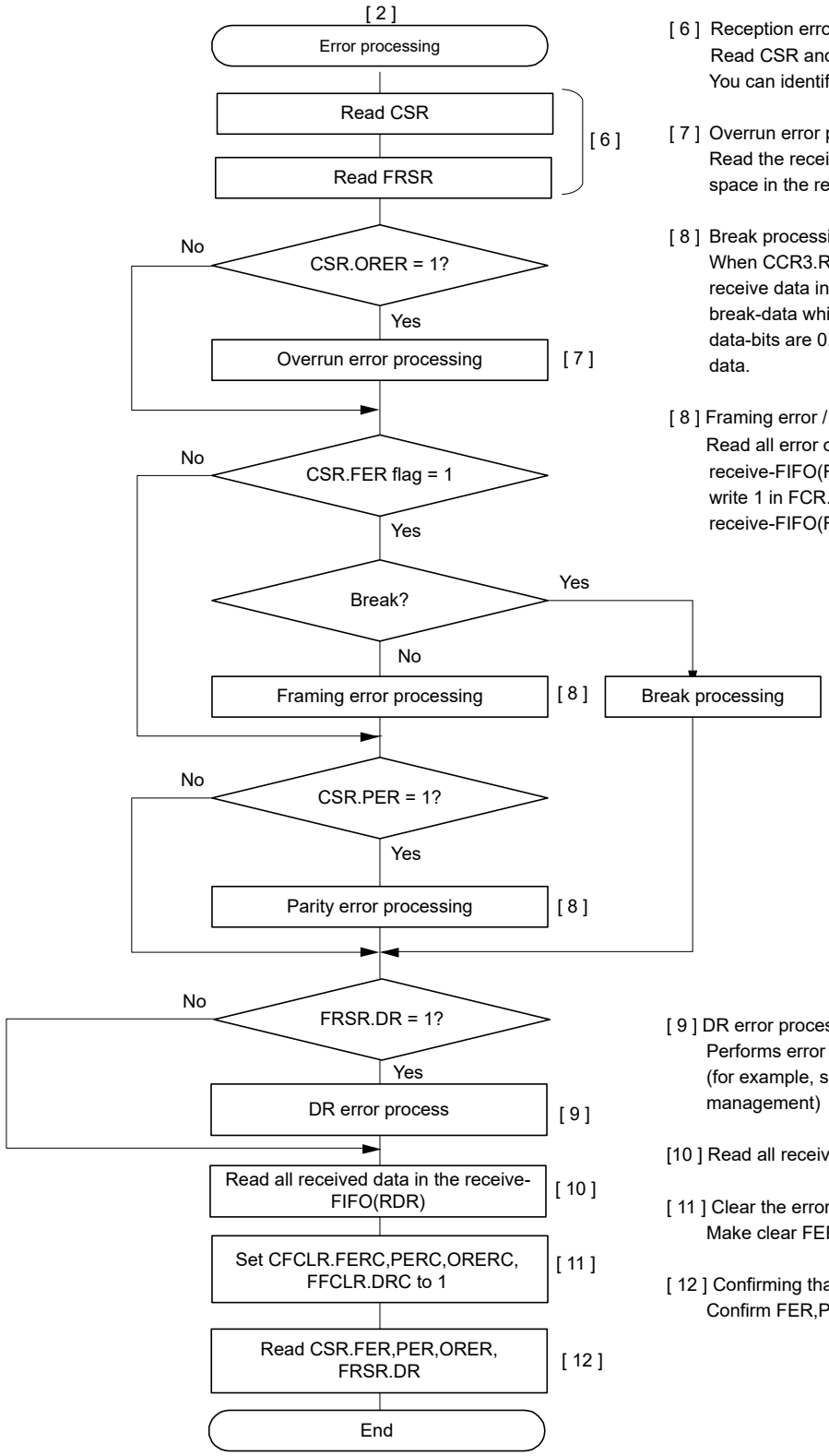
[3] Read the receive data in the receive-FIFO(RDR) in the SCIn_RXI interrupt handling routine. The reception data stored in the receive-FIFO(RDR) is read until the number of stored data will be below the FCR.RTRG value at least. You can confirm the number of the reception data in the FIFO by FRSR.R[5:0].

[4] Serial reception continuation procedure: To continue serial reception, before an overrun error occurs, please read data from the receive-FIFO(RDR) in the SCIn_RXI interrupt handling routine and clear CSR.RDRF flag and FRSR.DR flag to 0. The receive-FIFO data can also be read by activating the DMA or DTC. The RDRF flag is cleared automatically in this case, so please don't operate the RDRF flag.

[5] Set CCR0.RIE,RE to 0.

Note 1. Can be read in the RDR.ORER, PER, FER and DR flags.

Figure 26.22 Example flow of serial reception in asynchronous mode with FIFO selected and Address Matching Enabled (1)



[6] Reception error flag confirmation:
Read CSR and FRSR.
You can identify error by checking flags.

[7] Overrun error processing:
Read the receive-FIFO(RDR) and make space in the receive-FIFO(RDR).

[8] Break processing:
When CCR3.RXDESEL is 0, the latest receive data in the receive-FIFO(RDR) is the break-data which has framing error and all data-bits are 0. So, discard the latest receive data.

[8] Framing error / Parity error processing:
Read all error occurrence data stored in the receive-FIFO(RDR). (You can substitute to write 1 in FCR.RFRST bit and empty the receive-FIFO(RDR).)

[9] DR error process when FCR.DRES is 1.:
Performs error processing required for the system (for example, setting a counter for data number management)

[10] Read all received data in the receive-FIFO(RDR).

[11] Clear the error flag:
Make clear FER, PER, ORER, DR.

[12] Confirming that the error flag is cleared:
Confirm FER,PER,ORER,DR are cleared.

Figure 26.23 Example flow of serial reception in asynchronous mode with FIFO selected Address Matching Disabled (2)

26.3.10 The function of adjust receive sampling timing (Asynchronous Mode)

When there is the difference between the rising transfer time and the falling transfer time through a photo coupler, the receive sampling timing at middle of bit affects the reception margin. In this case, the receive sampling timing is able to adjust from the middle of bit to the optimum timing by using this function.

The receive sampling timing is adjusted from the middle of bit by following formula. The adjustable direction is set by CCR4.AJD. When adjusting backward (CCR4.AJD = 0), substitute AJD = +1 and substitute AJD = −1 when adjusting forward (CCR4.AJD = 1).

Adjusted sampling timing = the middle of bit + AJD × (base clock × the setting value of CCR4.AST[2:0])

The setting timing is limited by base clock cycles per 1 bit. For details, see Table 26.30.

An overview of reception operation of the communication through a photo coupler with this function is shown in Figure 26.24, Figure 26.25 and Figure 26.26, the explanation of operation with this function is shown in Figure 26.27.

Do not use this function when there is no difference between the rising transfer time and the falling transfer time, because there is a possibility of deteriorating the reception margin.

Table 26.30 The acceptable value of CCR4 register (asynchronous mode using internal clock)

CCR2.ABCSE	CCR2.ABCS	The number of base clock cycles/1bit	The acceptable value of CCR4	
			CCR4.AJD	CCR4.AST
1	x	6	0	000b – 010b*1
			1	
0	1	8	0	000b – 011b*1
			1	
0	0	16	0	000b – 111b
			1	

Note: x: Don’t care
Note 1. When the value of CCR4.AST exceeds the acceptable value, sampling is done at default timing. (Adjustment of sampling is not done.)

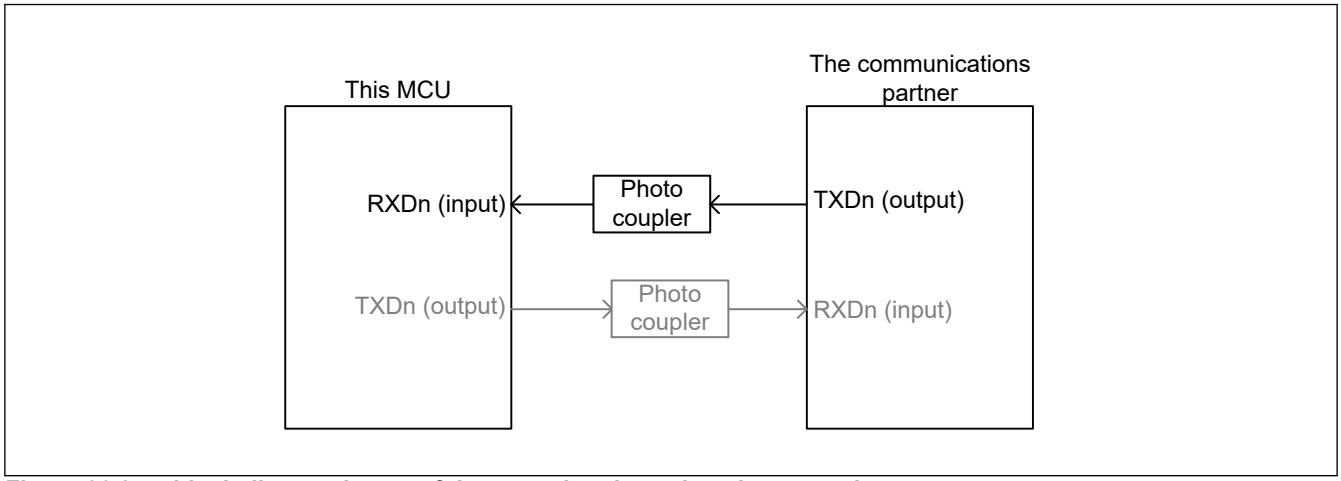
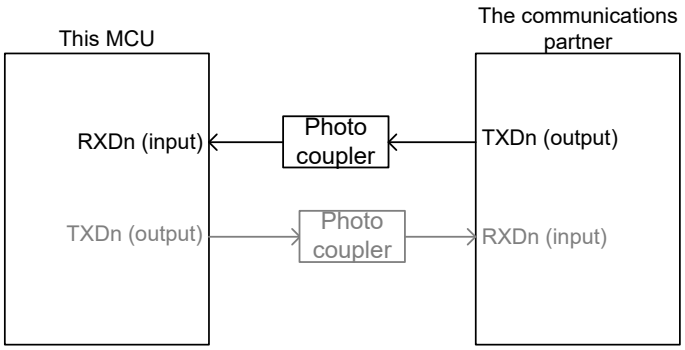
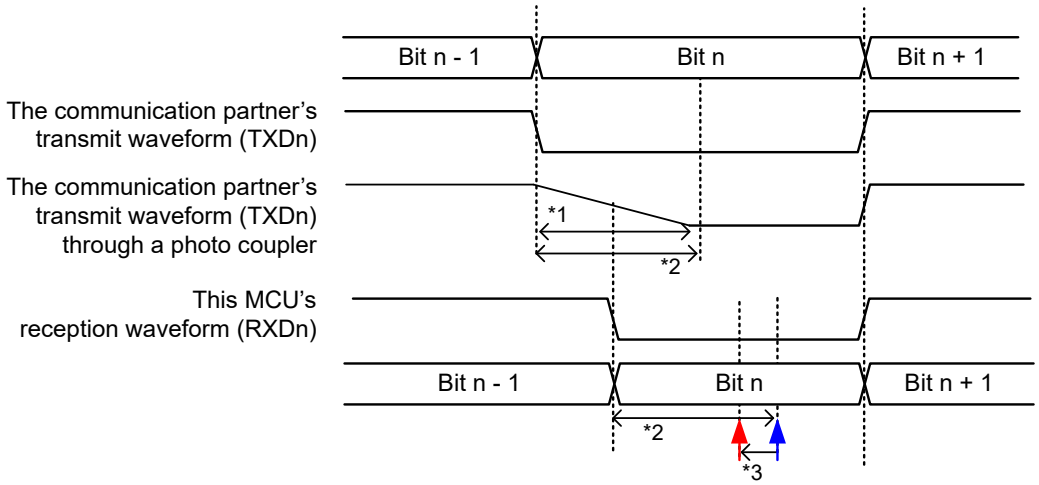


Figure 26.24 block diagram image of the reception through a photo couple



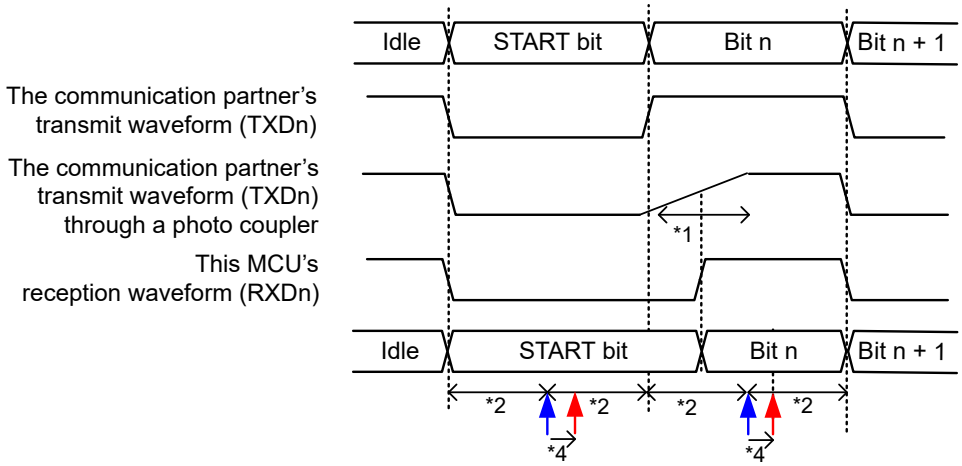
(a) In the case of the falling transfer time >> rising transfer time

The falling edge of reception waveform is made dull like following chart.
In this case, you can sampling at the middle of bit if you adjust the receive sampling timing to forward (AJD = 1).



(b) In the case of the falling transfer time << rising transfer time

The rising edge of reception waveform is made dull like following chart.
Thus, the reception margin of communications partner will be bad. In this case, you can improve the reception margin if you adjust the receive sampling timing to back.



The receive sampling timing when unadjusted (middle of bit)
The adjusted receive sampling timing

Note: This waveform shows the operation image of adjustment in reception sampling timing.
Note 1. The dull period by a photo coupler
Note 2. Bit center timing at set communication rate
Note 3. When CCR4.AJD is 1, the receive sampling timing is shifted to forward by the setting value of CCR4.AST[2:0].
Note 4. When CCR4.AJD is 0, the Receive sampling timing is shifted to backward by the setting value of CCR4.AST[2:0].

Figure 26.25 Overview of reception operation of the communication through a photo coupler

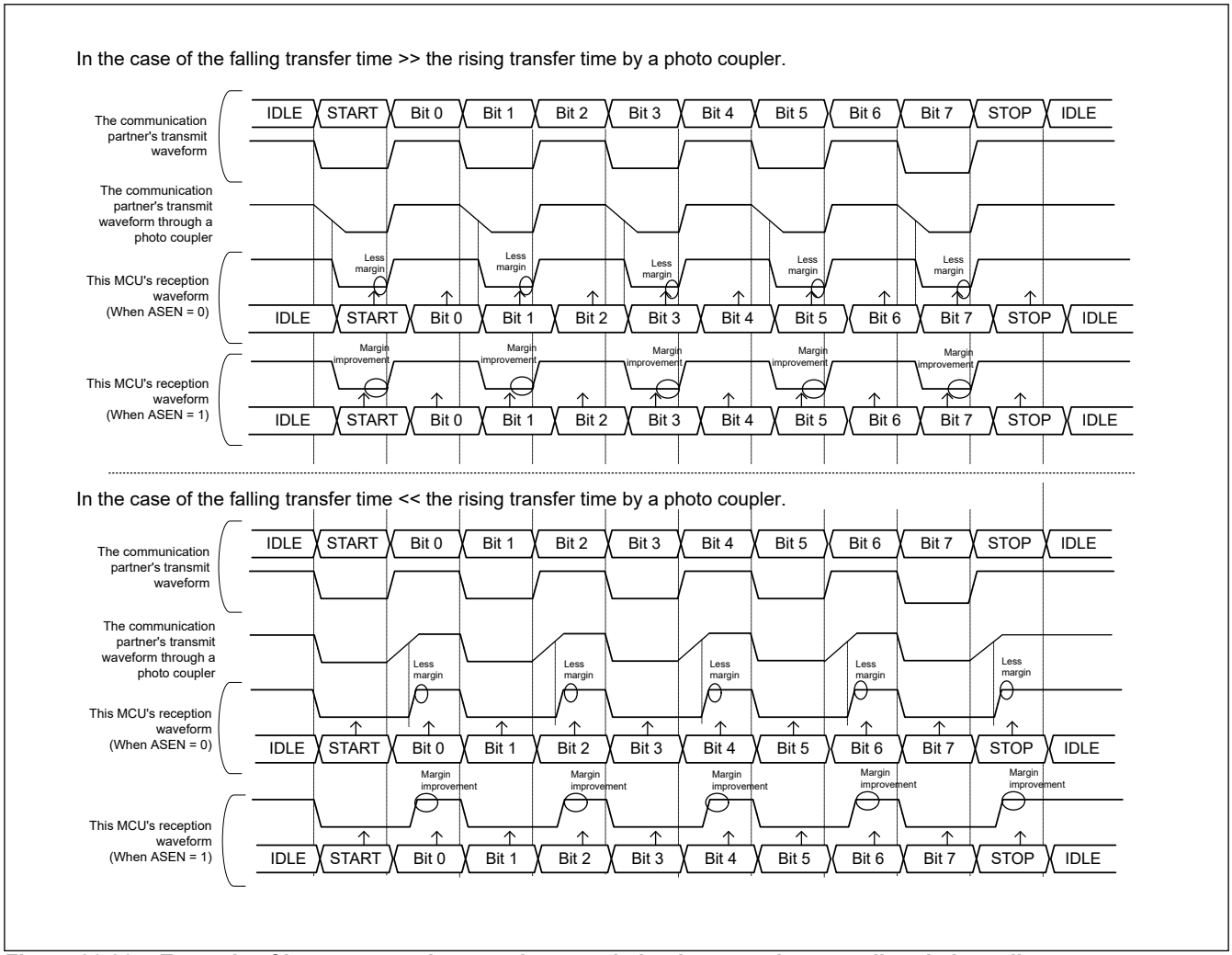


Figure 26.26 Example of improvement in reception margin by the reception sampling timing adjustment function

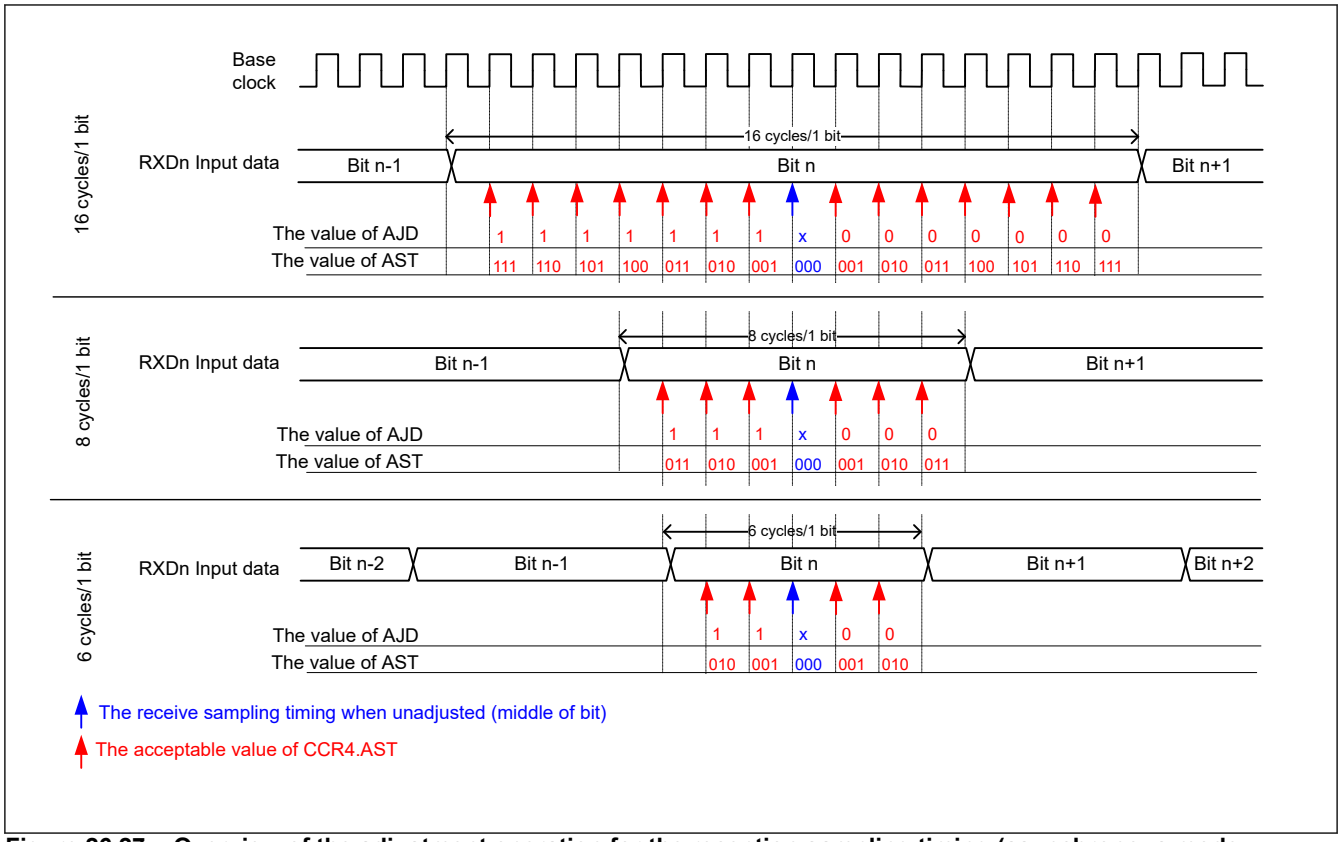


Figure 26.27 Overview of the adjustment operation for the reception sampling timing (asynchronous mode using internal clock)

26.3.11 The function of adjust transmit timing (Asynchronous Mode)

In communication via a photo coupler or the like, when either the rising or falling transition time of the TXDn output signal is long, then a communication partner receive dulled waveform. In this case, the reception margin may be affected.

In these cases, make a communication partner to be sampling at middle of bit using the function of adjust transmit timing.

When CCR4.ATEN is 1, this function can adjust the edge timing at the timing calculated by the following formula for the edge set with CCR4.AET.

The adjustment edge timing = the base clock × CCR4.ATT[2:0]

In addition, the upper limit of the adjustment edge timing is limited by setting the base clock cycles. For details, see Table 26.31.

A transmission movement image figure of the communication through a photo coupler with this function is shown in Figure 26.28, Figure 26.29 and Figure 26.30, the overview of operation with this function is shown in Figure 26.31 and Figure 26.32.

Do not use this function when there is not the difference between the rising transfer time and the falling transfer time, there is a possibility of deteriorating the reception margin of a communication partner.

Table 26.31 The acceptable value of CCR4.AET and CCR4.ATT (asynchronous mode using internal clock) (1 of 2)

ABCSE	ABCS	The number of base clock cycles/1bit	The acceptable value of CCR4	
			AET	ATT[2:0]
1	x	6	0	000b – 101b
			1	
0	1	8	0	000b – 111b
			1	

Table 26.31 The acceptable value of CCR4.AET and CCR4.ATT (asynchronous mode using internal clock) (2 of 2)

ABCSE	ABCS	The number of base clock cycles/1bit	The acceptable value of CCR4	
			AET	ATT[2:0]
0	0	16	0	000b – 111b
			1	

Note: x: Don't care
Note: When the value of ACTR.AET/ATT is out of the acceptable value, this SCI module doesn't adjust transmit timing.

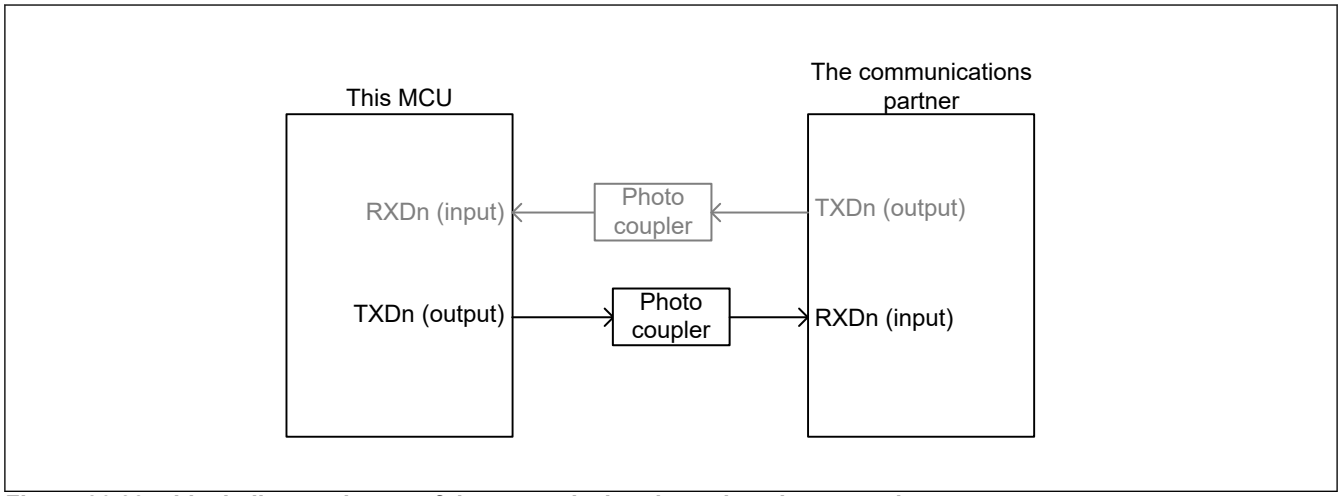
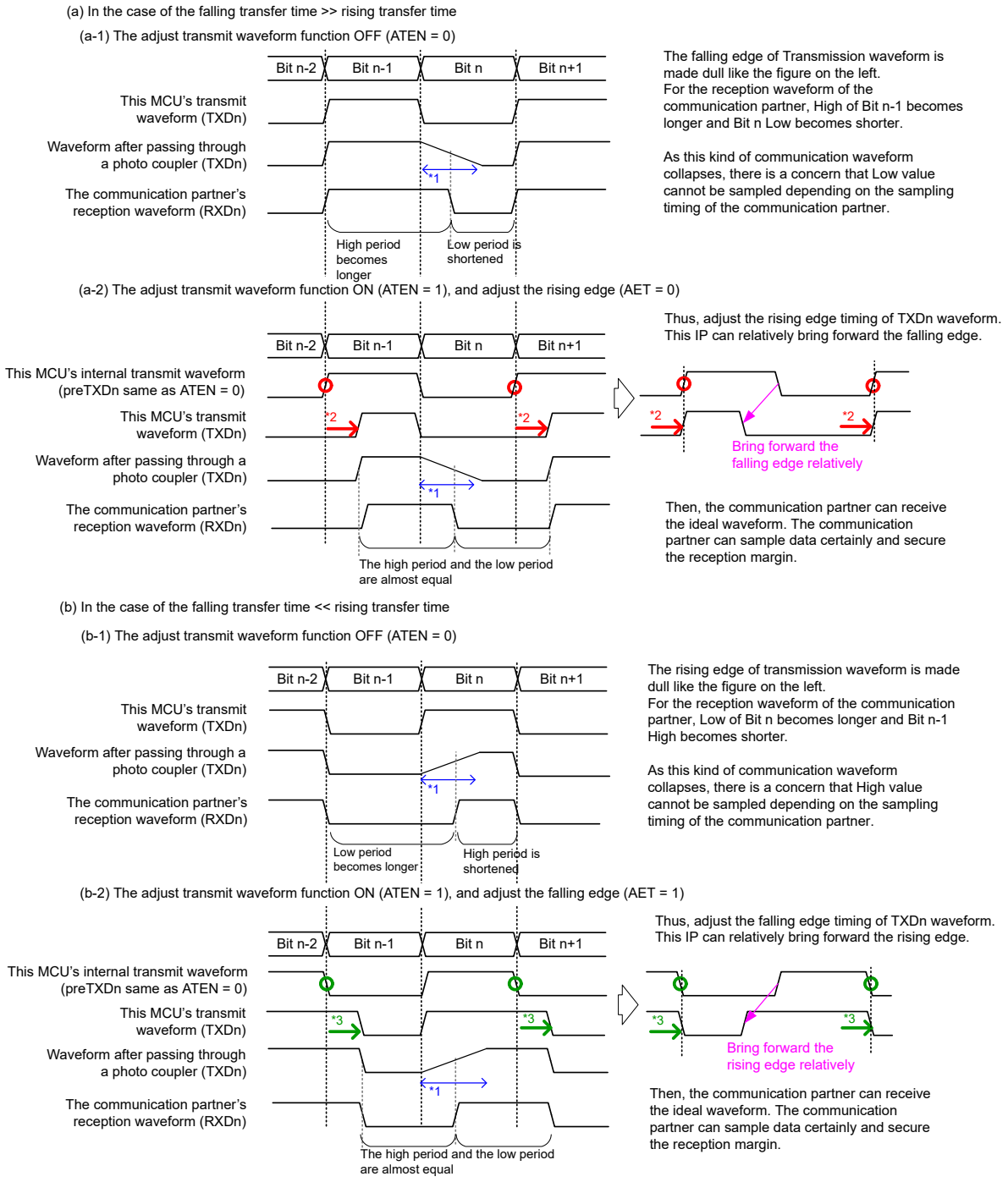


Figure 26.28 block diagram image of the transmission through a photo coupler



Note: This waveform shows the operation image of adjustment transmit waveform.
Note 1. The dull period by a photo coupler
Note 2. When CCR4.AET is 0, the rising edge of transmit waveform timing is shifted to back by the setting value of CCR4.ATT[2:0]. This IP transmit waveform bringing forward the falling edge relatively.
Note 3. When CCR4.AET is 1, the falling edge of transmit waveform timing is shifted to back by the setting value of CCR4.ATT[2:0]. This IP transmit waveform bringing forward the rising edge relatively.

Figure 26.29 The overview of transmission operation in the communication through a photo coupler

The explanation of transmit waveforms of the communication through a photo coupler using adjust transmit timing function

When using the transmission timing adjustment function, adjust the edge timing of the transmission waveform and correct the reception waveform of the communication partner

The following example is 8 bit long data.

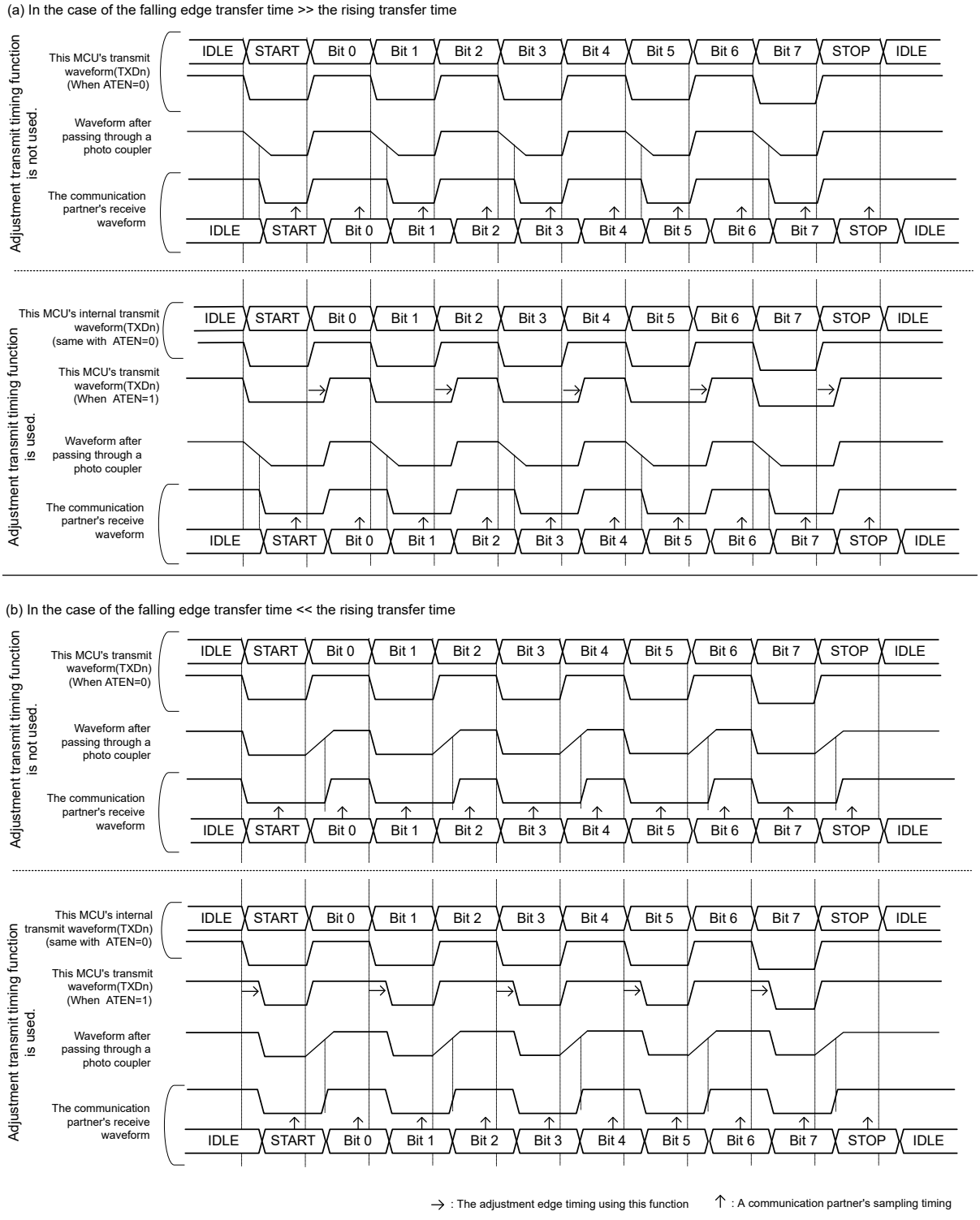
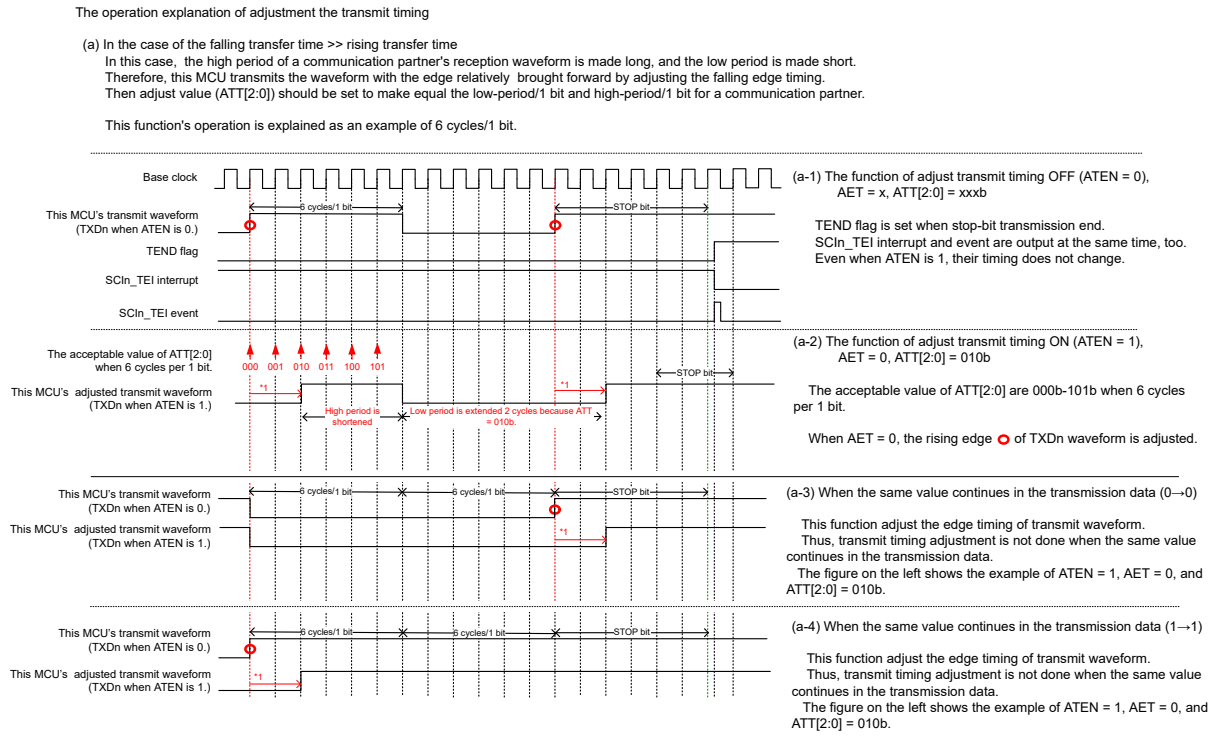
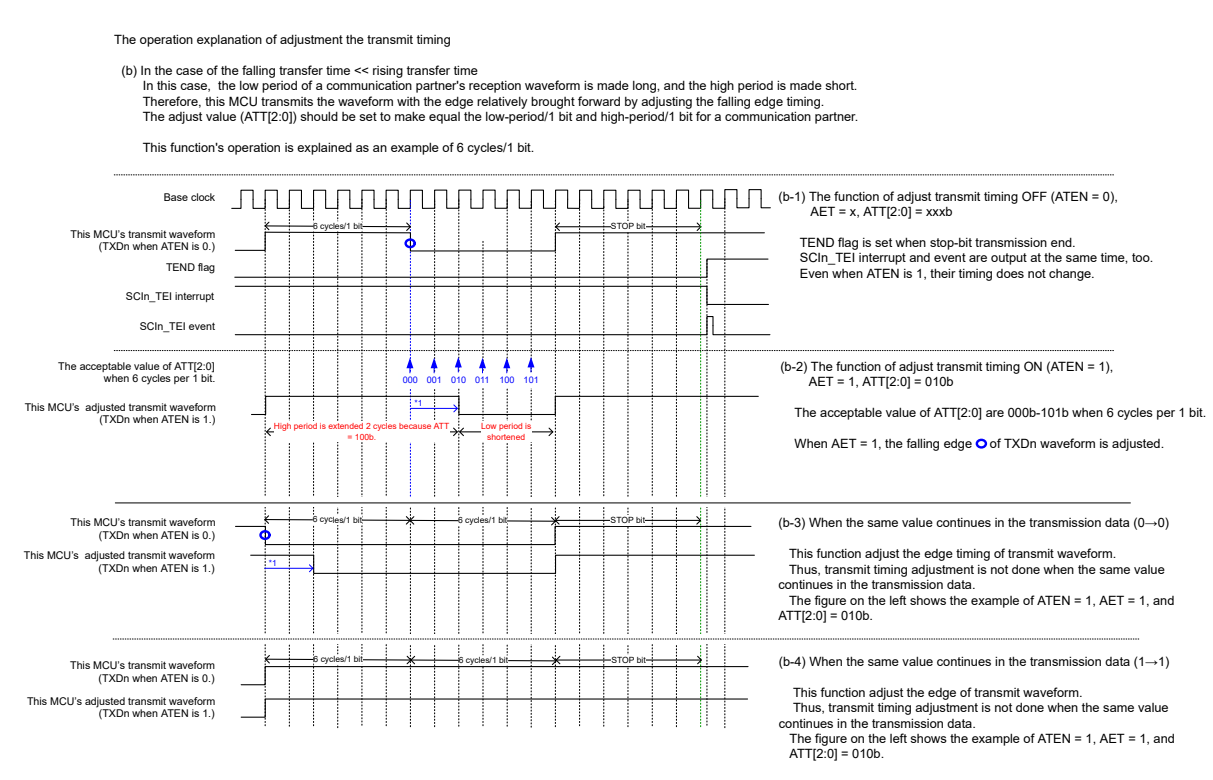


Figure 26.30 The explanation for the transmit waveform through a photo coupler



Note 1. The rising edge of transmit timing is shifted to back by the setting value of CCR4.ATT[2:0].

Figure 26.31 The adjustment operation explanation for the transmit timing when AET is 0



Note 1. The falling edge of transmit timing is shifted to back by the setting value of CCR4.ATT[2:0].

Figure 26.32 The adjustment operation explanation for the transmit timing when AET is 1

26.4 Multi-Processor Communication Function

The multi-processor communication function enables the SCI to transmit and receive data between multiple processors by sharing an asynchronous serial communication line that has an added multi-processor bit. In multi-processor communication, a unique ID code is allocated to each receiving station. Serial communication cycles consist of an ID transmission cycle to specify the receiving station and a data transmission cycle to transmit data to the specified receiving station.

The multi-processor bit is used to distinguish between the ID transmission cycle and the data transmission cycle:

- When the multi-processor bit is set to 1, the transmission cycle is the ID transmission cycle
- When the multi-processor bit is set to 0, the transmission cycle is the data transmission cycle

Figure 26.33 shows an example of communication between processors using a multi-processor format. First, a transmitting station transmits communication data in which the multi-processor bit set to 1 is added to the ID code of the receiving station. Next, the transmitting station transmits communication data in which the multi-processor bit set to 0 is added to the transmit data. After receiving communication data with the multi-processor bit set to 1, the receiving station compares the received ID with the ID of the receiving station itself. If the two match, the receiving station receives communication data that is subsequently transmitted. If the received ID does not match with the ID of the receiving station, the receiving station skips the communication data until it receives data in which the multi-processor bit is set to 1.

RTS control cannot be used at the time of multi-processor communication function use because this is a function corresponding to one-to-many communications.

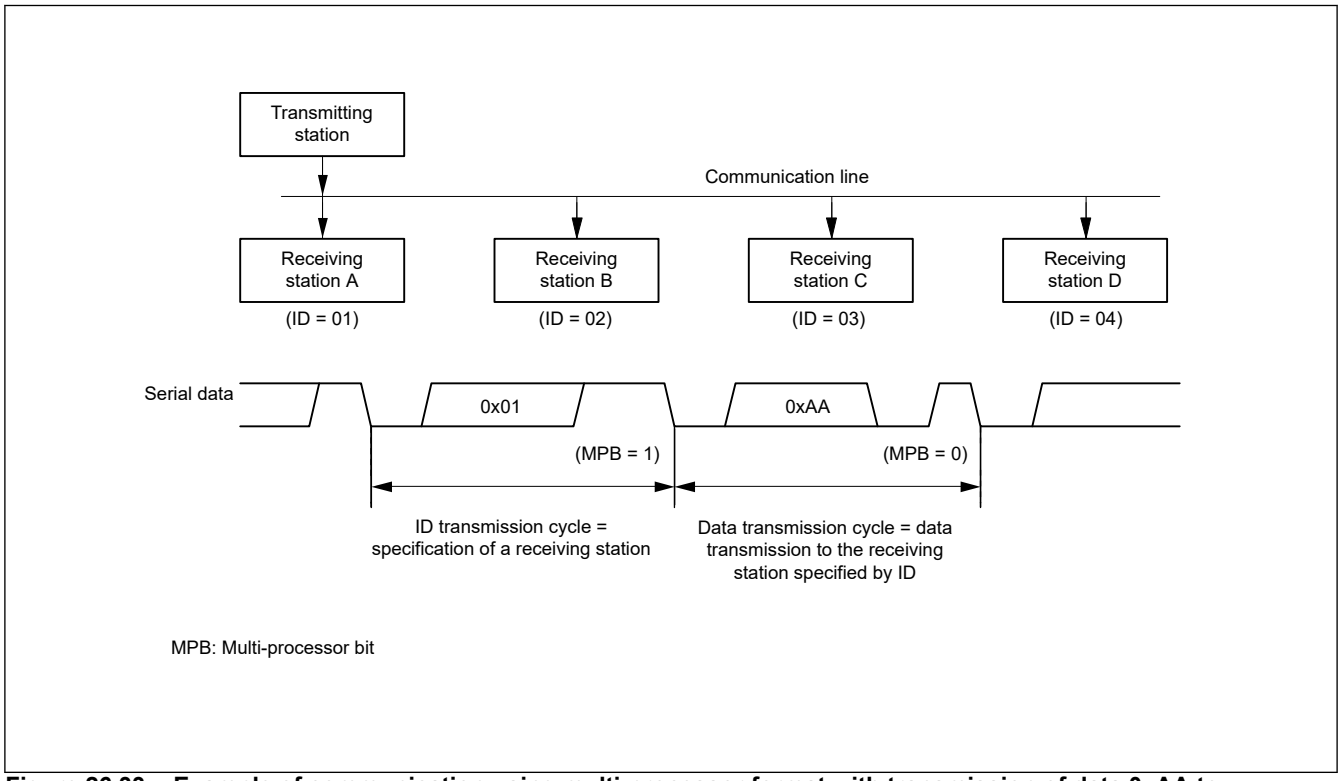


Figure 26.33 Example of communication using multi-processor format with transmission of data 0xAA to receiving station A

(1) Non-FIFO selected

To support this function, the SCI provides the CCR0.MPIE bit. When the MPIE bit is set to 1, the following operations are disabled until the reception of data in which the multi-processor bit is set to 1:

- Transfer of receive data from the RSR register to the RDR register
- Detection of a receive error
- Setting of the respective RDRF, ORER, and FER status flags in the CSR register

When the SCI receives a character in which the multi-processor bit is set to 1, the RDR.MPB bit is set to 1 and the CCR0.MPIE bit is automatically cleared, returning the SCI to non-multi-processor reception operation. If the CCR0.RIE bit is set to 1, an SCIn_RXI interrupt is generated.

When the multi-processor format is specified, the parity bit function is disabled. Apart from this, there is no difference from operation in non-multi-processor asynchronous mode. The clock used for the multi-processor communication is the same as the clock used in non-multi-processor asynchronous mode.

(2) FIFO selected

For data transmission, software must write data to TDR.MPBT (Multi-Processor Bit Transfer) that corresponds to transmit data in TDR.TDAT. For data reception, the multi-processor bit that is part of the receive data is written to RDR.MPB and receive data is written to RDR.RDAT.

When the MPIE bit is set to 1, the following operations are disabled until reception of data in which the multi-processor bit is set to 1:

- Transfer of receive data from the RSR register to the RDR.RDAT register
- Detection of a receive error
- Detection of DR
- Setting of the respective RDRF, ORER, and FER status flags in the CSR register

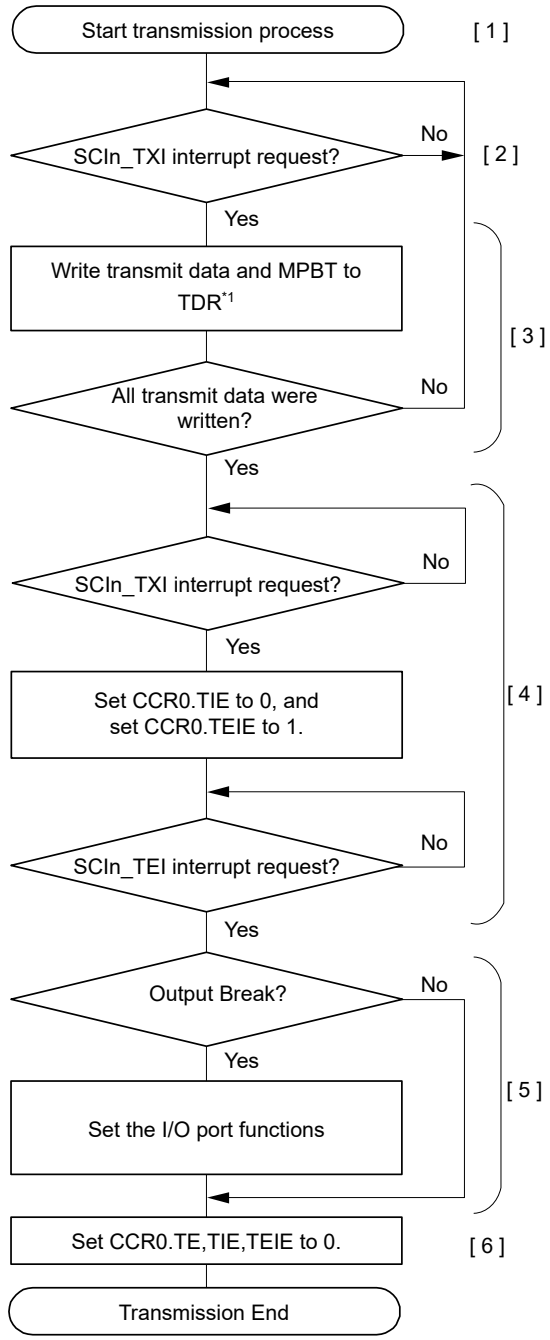
When the SCI receives a character in which the multi-processor bit is set to 1, the RDR.MPB bit is set to 1 and receive data is written to receive-FIFO(RDR.RDAT). The CCR0.MPIE bit is automatically cleared, returning the SCI to normal reception operation. If the CCR0.RIE bit is set to 1, an SCIn_RXI interrupt is generated.

When the multi-processor format is specified, the parity bit function is disabled. Apart from this, there is no difference from operation in normal asynchronous mode with FIFO selected.

26.4.1 Multi-Processor Serial Data Transmission

(1) Non-FIFO selected

Figure 26.34 shows an example flow of multi-processor data transmission. In the ID transmission cycle, the ID must be transmitted with the TDR.MPBT bit set to 1. In the data transmission cycle, the data must be transmitted with the MPBT bit set to 0. The rest of the operations are the same as operations in asynchronous mode.



[1] Start transmission process

[2] SCIn_TXI interrupt request:
The transmit-data-empty-interrupt(SCIn_TXI) is generated at following conditions.
When set TE and TIE to 0 at same time.
When transmission-data forwards to TSR from TDR.

[3] Serial transmission continuation procedure:
To continue serial transmission write transmit data and MPBT setting value to TDR*1 once by using a SCIn_TXI. Transmit data and MPBT setting value can also be written by activating the DMAC or DTC.

[4] If you use SCIn_TEI, set TIE = 0 and TEIE = 1 after writing the last-transmit-data to TDR*1 and transmission is started.

[5] Break output at the end of serial transmission:
To output a break, it is necessary to set SPB2IO and SPB2DT, for output status setting when TE is 0.
(If you set these bits at initialization, you can skip this step.)

[6] Set CCR0.TE,TIE,TEIE to 0.

Note 1. In case of byte access with data length 9 bits or to write MPBT bit, write transmit data in order of TDR[15:8] and TDR[7:0].

Figure 26.34 Example flow of multi-processor serial transmission with non-FIFO selected

(2) FIFO selected

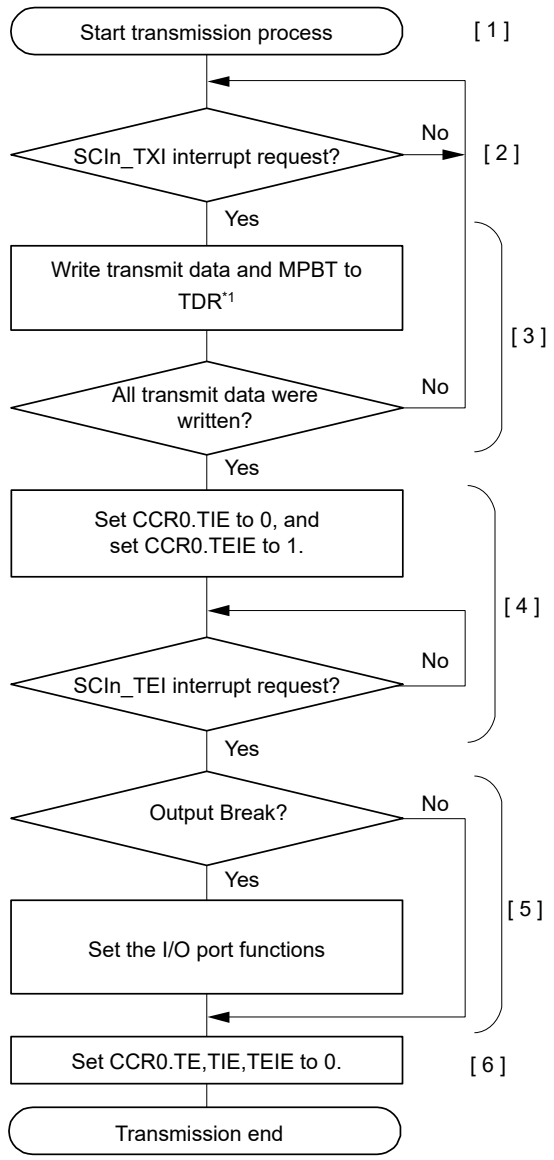
Figure 26.35 shows an example of data format that is written to transmit-FIFO (TDR) in multi-processor mode. The TDR.MPBT bit is set to 1. Data is set to transmit-FIFO (TDR) with the correct data length. Write 0 for unused bits.

Data Length	Register setting		Transmit data in TDR[15:0]															
	CCR3. CHR[1:0]		b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
7 bit	1	1	—	—	—	—	—	—	MPBT	—	—	TDAT[6:0]						
8 bit	1	0	—	—	—	—	—	—	MPBT	—	TDAT[7:0]							
9 bit	0	Don't Care	—	—	—	—	—	—	MPBT	TDAT[8:0]								

Note: —: Invalid. The write value should be 0.

Figure 26.35 Data format written to transmit-FIFO (TDR) in multi-processor mode with FIFO selected

Figure 26.36 shows an example flow of multi-processor serial transmission with FIFO selected. In the ID transmission cycle, the ID must be transmitted with the TDR.MPBT bit set to 1. In the data transmission cycle, the data must be transmitted with the MPBT bit set to 0. The rest of the operations are the same as operations in asynchronous mode with FIFO selected.



[1] Start transmission process
To enable SCIn_TXI interrupt at he start of transmission, set TE and TIE to 1 with one instructions at the same time. And set CCR3.FM,MP to 1.

[2] SCIn_TXI interrupt request:
The transmit-data-empty-interrupt(SCIn_TXI) is generated when transmission-data forwards to TSR from the transmit-FIFO(TDR) and the number of transmit-data written in the transmit-FIFO is equal to or less than the transmit-FIFO threshold.

[3] Serial transmission continuation procedure:
To continue serial transmission, write all transmit data and MPBT setting value to the transmit-FIFO(TDR) by using a SCIn_TXI interrupt. Transmit data and MPBT setting value can also be written to the transmit- FIFO(TDR) by activating the DMAC or DTC.

[4] If you use SCIn_TEI, set TIE = 0 and TEIE = 1 after writing the last-transmit-data to TDR*1.

[5] Break output at the end of serial transmission:
To output a break, it is necessary to set SPB2IO and SPB2DT, for output status setting when TE is 0.
(If you set these bits at initialization, you can skip this step.)

[6] Set CCR0.TE,TIE,TEIE to 0.

Note 1. In case of byte access with data length 9 bits or to write MPBT bit, write transmit data in order of TDR[15:8] and TDR[7:0].

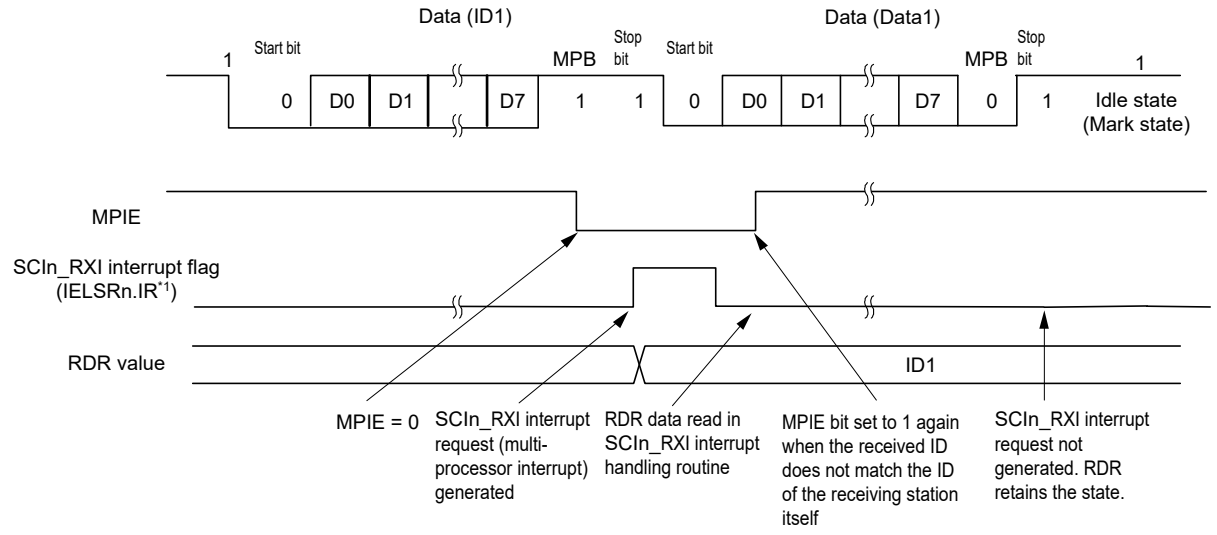
Figure 26.36 Example flow of serial transmission in multi-processor mode with FIFO selected

26.4.2 Multi-Processor Serial Data Reception

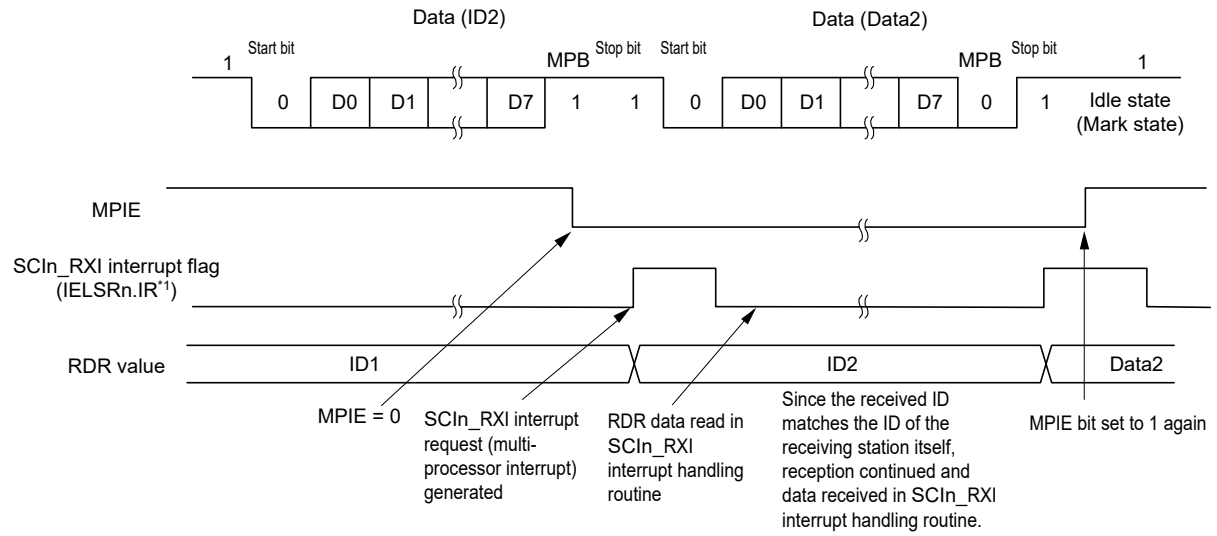
(1) Non-FIFO selected

Figure 26.38 and Figure 26.39 are example flows of multi-processor serial reception. When the CCR0.MPIE bit is set to 1, reading communication data is skipped until reception of communication data in which the multi-processor bit is set to 1. When communication data in which the multi-processor bit is set to 1 is received, the received data is transferred to the RDR register, and the SCIn_RXI interrupt request is generated. The rest of the operations are the same as operations in asynchronous mode.

Figure 26.37 shows an example operation for data reception.



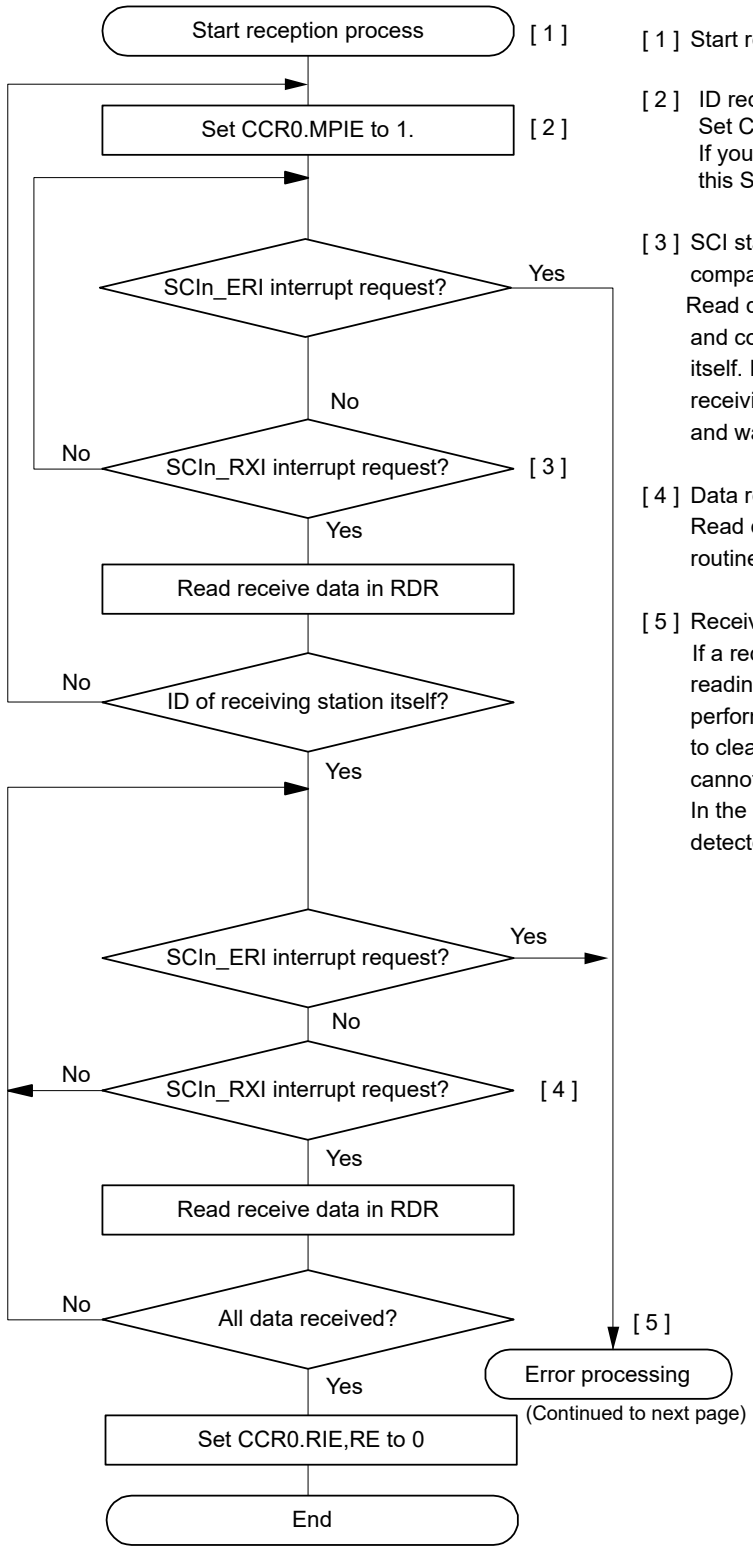
(a) When the received ID does not match the ID of the receiving station itself



(b) When the received ID matches the ID of the receiving station itself

Note 1. See section 12, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 26.37 Example of SCI reception with 8-bit data, multi-processor bit, and 1 stop bit



[1] Start reception process

[2] ID reception cycle:
Set CCR0.MPIE to 1. and wait for ID reception.
If you set this bits at initialization([1]), you can skip this Step.

[3] SCI status confirmation and reception and comparison of ID:
Read data in RDR at the first SCIn_RXI interrupt and compare it with the ID of the receiving station itself. If the ID does not match the ID of the receiving station itself, set the MPIE bit to 1 again, and wait for another SCIn_RXI interrupt request.

[4] Data reception at an SCIn_RXI interrupt:
Read data in RDR once in the SCIn_RXI interrupt routine.

[5] Receive error processing and break detection:
If a receive error occurs, an error is identified by reading ORER and FER flags in CSR. After performing the appropriate error processing be sure to clear the ORER and FER flags. Reception cannot be resumed if any of these flags is set to 1. In the case of a framing error, a break can be detected by reading CSR.RXDMON bit.

Figure 26.38 Example flow of multi-processor serial reception with non-FIFO selected (1)

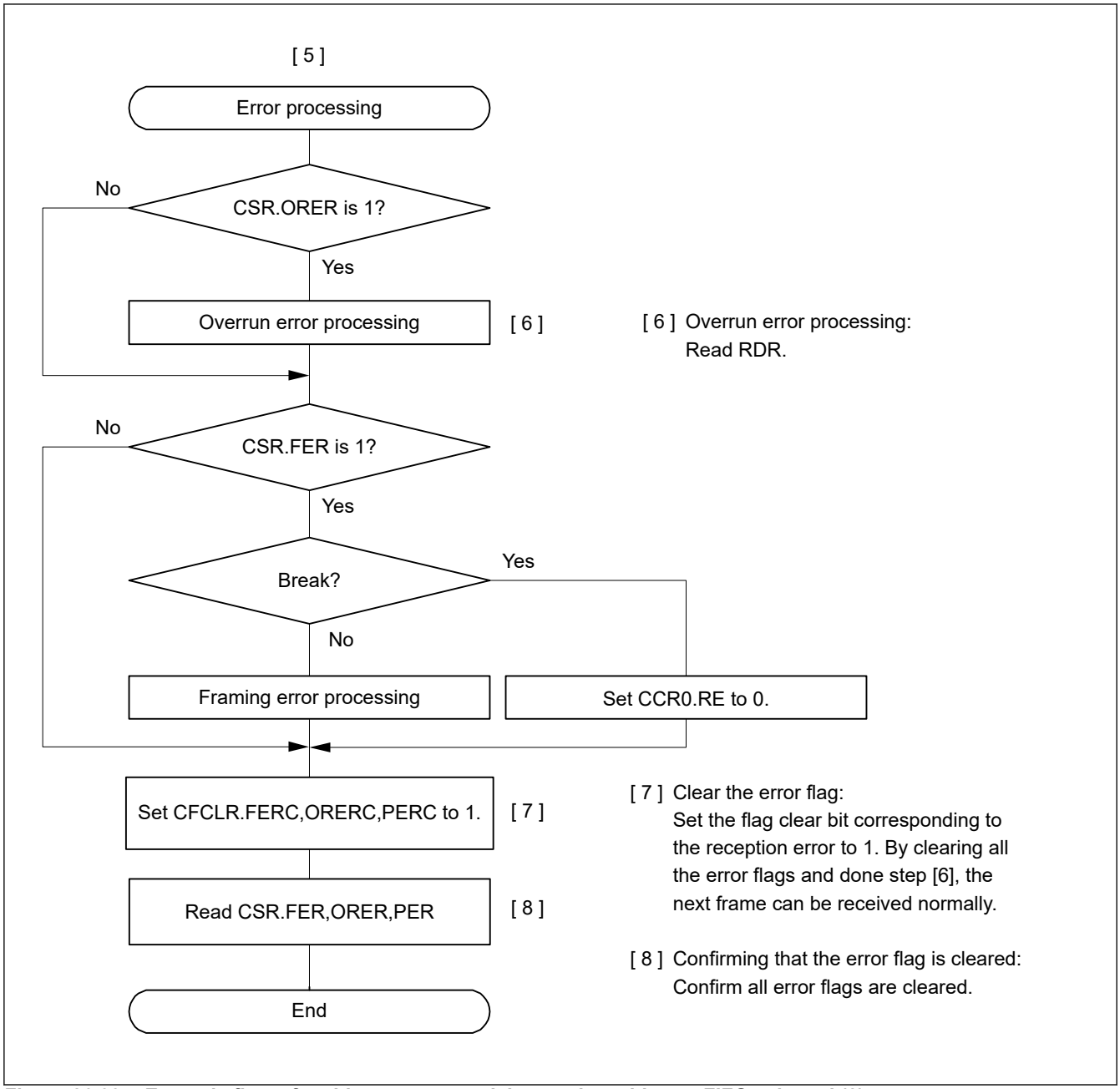


Figure 26.39 Example flow of multi-processor serial reception with non-FIFO selected (2)

(2) FIFO selected

Figure 26.40 shows an example of a data format that is written to receive-FIFO (RDR) in multi-processor mode.

In multi-processor mode, the MPB value that is a part of the receive data is written to the RDR.MPB bit. A value of 0 is written to the RDR.FPER and PER flags. Data is written to receive-FIFO (RDR) with the correct data length. Unused bits are written with 0. When software reads the receive-FIFO (RDR) register, the SCI updates RDR.FFER, FPER, and MPB flags, and receive data (RDAT[8:0]) in receive-FIFO (RDR) with the next data. The FER, PER, and ORER flags in the receive-FIFO (RDR) register always reflect the associated flags in the CSR and FRSR registers.

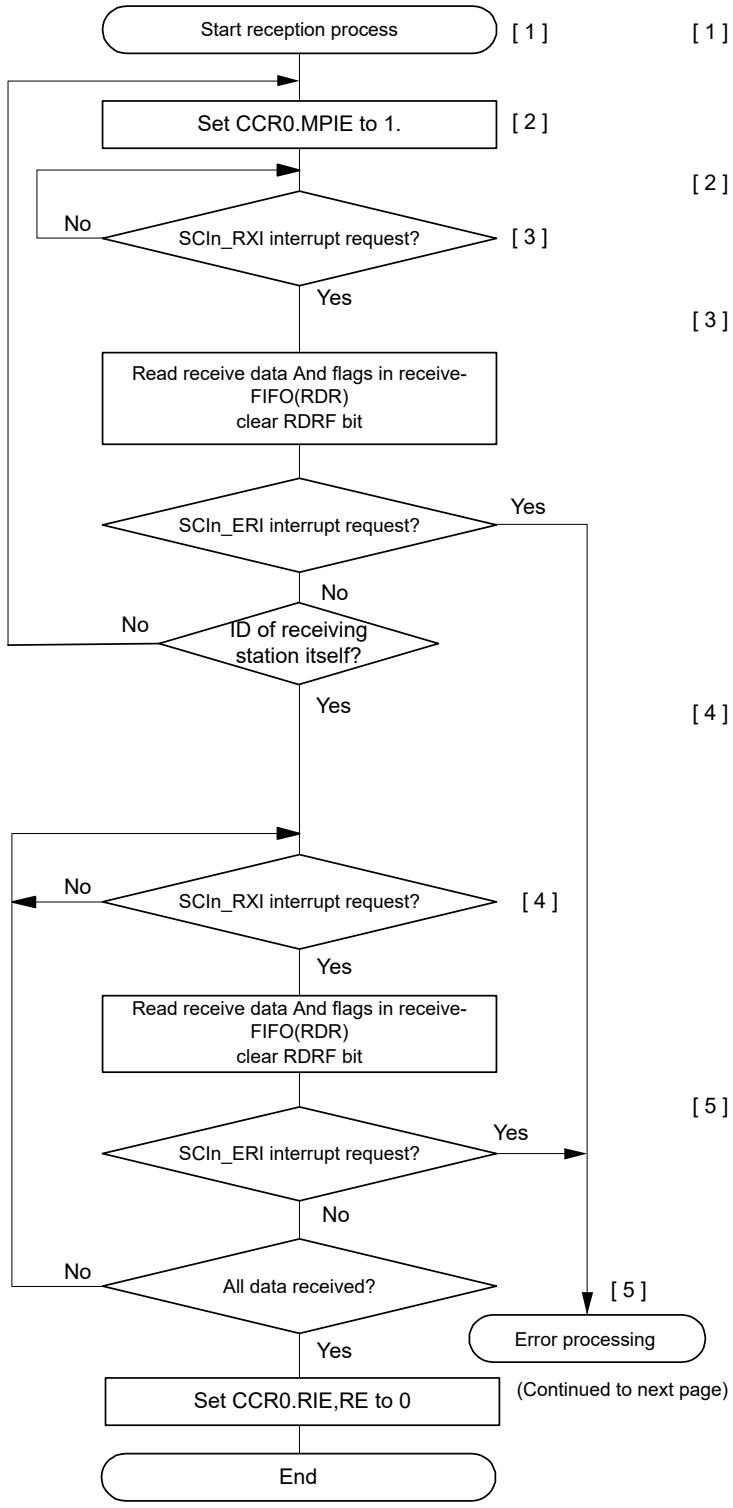
Data Length	Register Setting CCR3.CHR[1:0]		Receive data in RDR[31:0] b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
7bit	1	1	0	0	0	FFER	FPER	DR	MPB	0	0	RDAT[6:0]						
8bit	1	0	0	0	0	FFER	FPER	DR	MPB	0	RDAT[7:0]							
9bit	0	Don't Care	0	0	0	FFER	FPER	DR	MPB	RDAT[8:0]								
Data Length	CCR3.CHR[1:0]		b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
7bit	1	1	0	0	0	FER	PER	0	0	ORER	0	0	0	0	0	0	0	0
8bit	1	0	0	0	0	FER	PER	0	0	ORER	0	0	0	0	0	0	0	0
9bit	0	Don't Care	0	0	0	FER	PER	0	0	ORER	0	0	0	0	0	0	0	0

Note: When data length is 7bit, it can read always 0 in RDAT[8:7].
When data length is 8bit, it can read always 0 in RDAT[8].

Figure 26.40 Data format stored in receive-FIFO (RDR) in multi-processor mode with FIFO selected

Figure 26.41 shows an example flow of multi-processor data reception with FIFO selected. When the CCR0.MPIE is set to 1, reading communication data is skipped until reception of communication data in which the multi-processor bit is set to 1. When communication data in which the multi-processor bit is set to 1 is received, the received data, MPB and associated errors are transferred to the receive-FIFO (RDR) register. The CCR0.MPIE bit is automatically cleared and normal reception continues.

If a framing error occurs and the CSR.FER flag is set to 1, the SCI continues data reception. The rest of the operations are the same as operations in asynchronous mode with FIFO selected.



[1] Start reception process
To generate an SCIn_RXI interrupt request with MP detection, please set FCR.RTRG = 1 and FCR.DRES = 0.

[2] ID reception cycle:
Set CCR0.MPIE to 1. and wait for ID reception. If you set this bits at initialization([1]), you can skip this Step.

[3] SCI status confirmation and reception and comparison of ID:
Skip the communication data until the MPB bit receives data 1. When the MPB bit receives data 1, an SCIn_XRI interrupt request is generated. When an SCIn_RXI interrupt request is received, read the data in the receiving FIFO (RDR register) and compare it to the ID of my station. If it is not my station's ID, set the MPIE bit to 1 again and wait for the SCIn_RXI interrupt request. For my station's ID, set the received FIFO threshold to the necessary value.

[4] Data reception at an SCIn_RXI interrupt:
When the number of received data stored in the received FIFO (RDR register) exceeds the received FIFO threshold, the CSR.RDRF bit becomes 1 and an SCIn_RXI interrupt request is generated. In addition, if the number of received data stored in the received FIFO (RDR register) is 1 or more and less than the received FIFO threshold, the FRSR.DR bit becomes 1 after 15 etus, and an SCIn_RXI interrupt request is generated. The SCIn_RXI interrupt routine reads the data from the received FIFO (RDR register).

[5] Receive error processing and break detection:
If a receive error occurs, an error is identified by reading ORER and FER flags in CSR. After performing the appropriate error processing be sure to clear the ORER and FER flags. In the case of a framing error, a break can be detected by reading the value of the RXDn pin.

Figure 26.41 Example flow of serial reception in multi-processor mode with FIFO selected

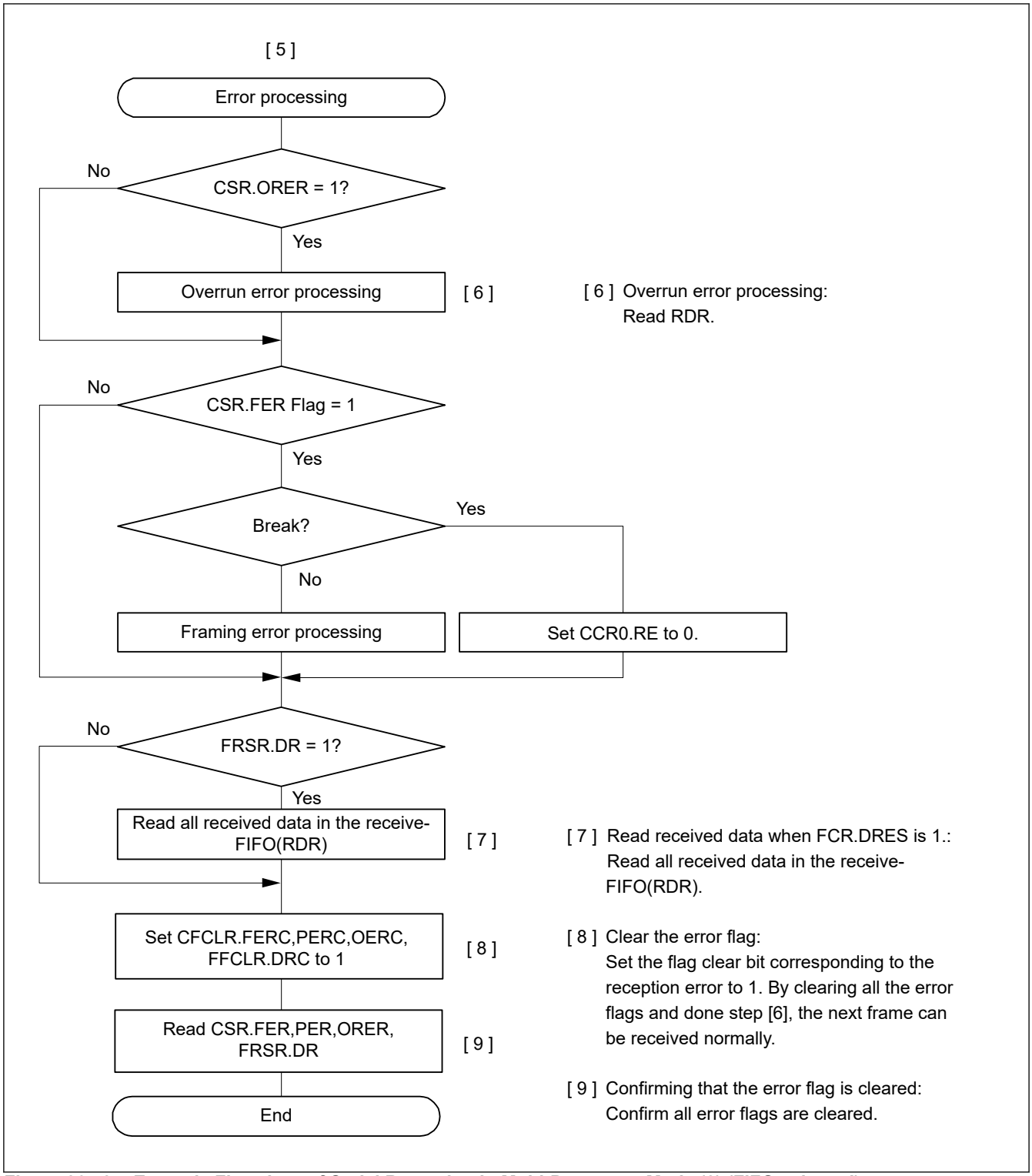


Figure 26.42 Example Flowchart of Serial Reception in Multi-Processor Mode (2) (FIFO selected)

26.5 Operation in Manchester mode

In Manchester mode, the transmit or receive serial data is coded in Manchester encoding.

Figure 26.43 shows the conceptual image of Manchester encoding.

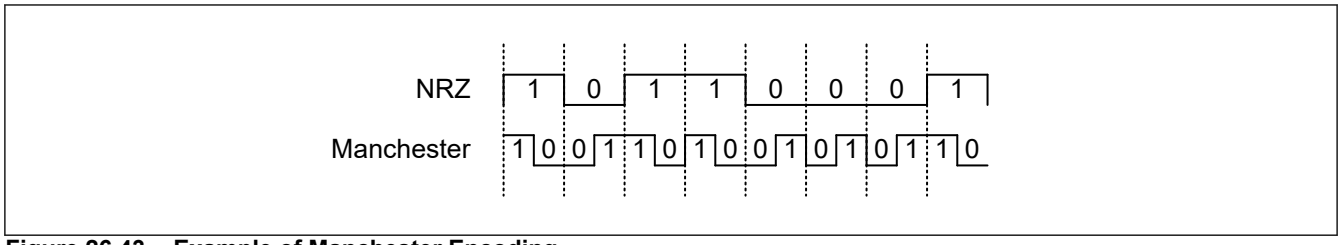


Figure 26.43 Example of Manchester Encoding

In Manchester mode, a preface and a start bit area are added to the transmit data in the register to configure a transmit frame. For transmission, data is encoded in Manchester encoding. When data is received, frames having the same format as transmitted frames are detected and Manchester decoding is performed.

For details on the frame format, see section 26.5.1. Frame Format.

26.5.1 Frame Format

Figure 26.44 shows the frame format in Manchester mode.

In the upper half of the figure, relevant setting registers are shown.

The preface area and the data area are encoded in Manchester encoding.

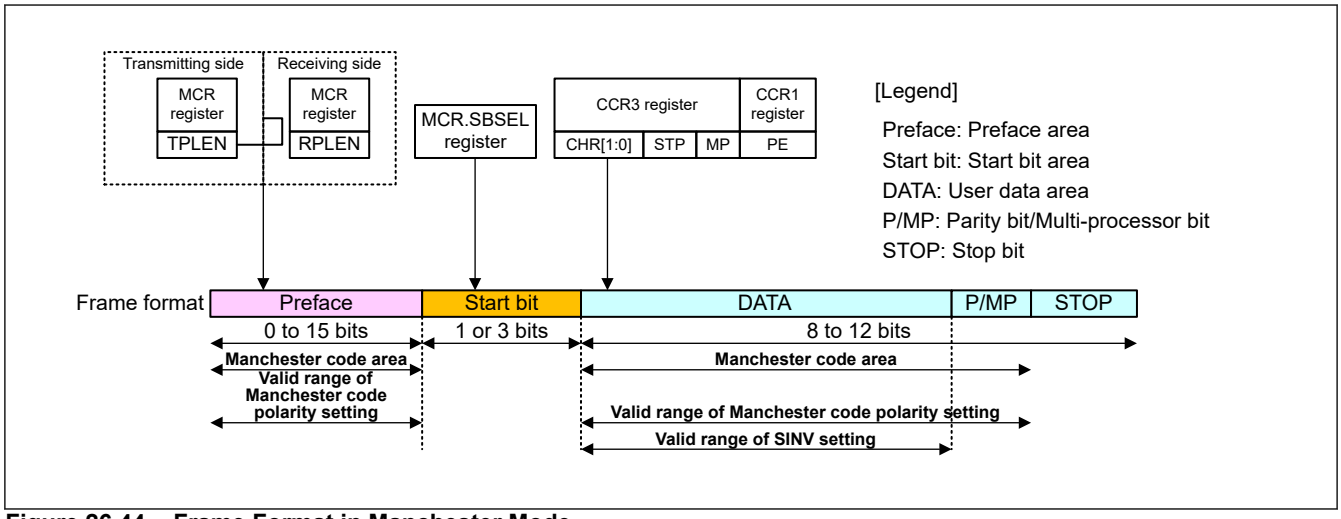


Figure 26.44 Frame Format in Manchester Mode

(1) Preface area

This is a fixed pattern area located at the beginning of each frame.

Different registers are used to set the preface area for transmission and reception. The preface length is determined by setting MCR.TPLEN[3:0] for transmission. It is determined by setting MCR.RPLEN[3:0] for reception.

If it is set to 0, the transmit preface is disabled and is not added.

If it is set to 1d to 15d, a preface whose length is determined by this setting is added.

(For example, if it is set to 1d, a 1-bit preface is added. If it is set to 15d, a 15-bit preface is added.)

The preface pattern is set with MCR.TPPAT[1:0] for transmission and MCR.RPPAT[1:0] for reception, and is selected from four types of patterns.

Figure 26.45 shows how the preface pattern is set. The preface area and the start bit area are added for each communication frame.

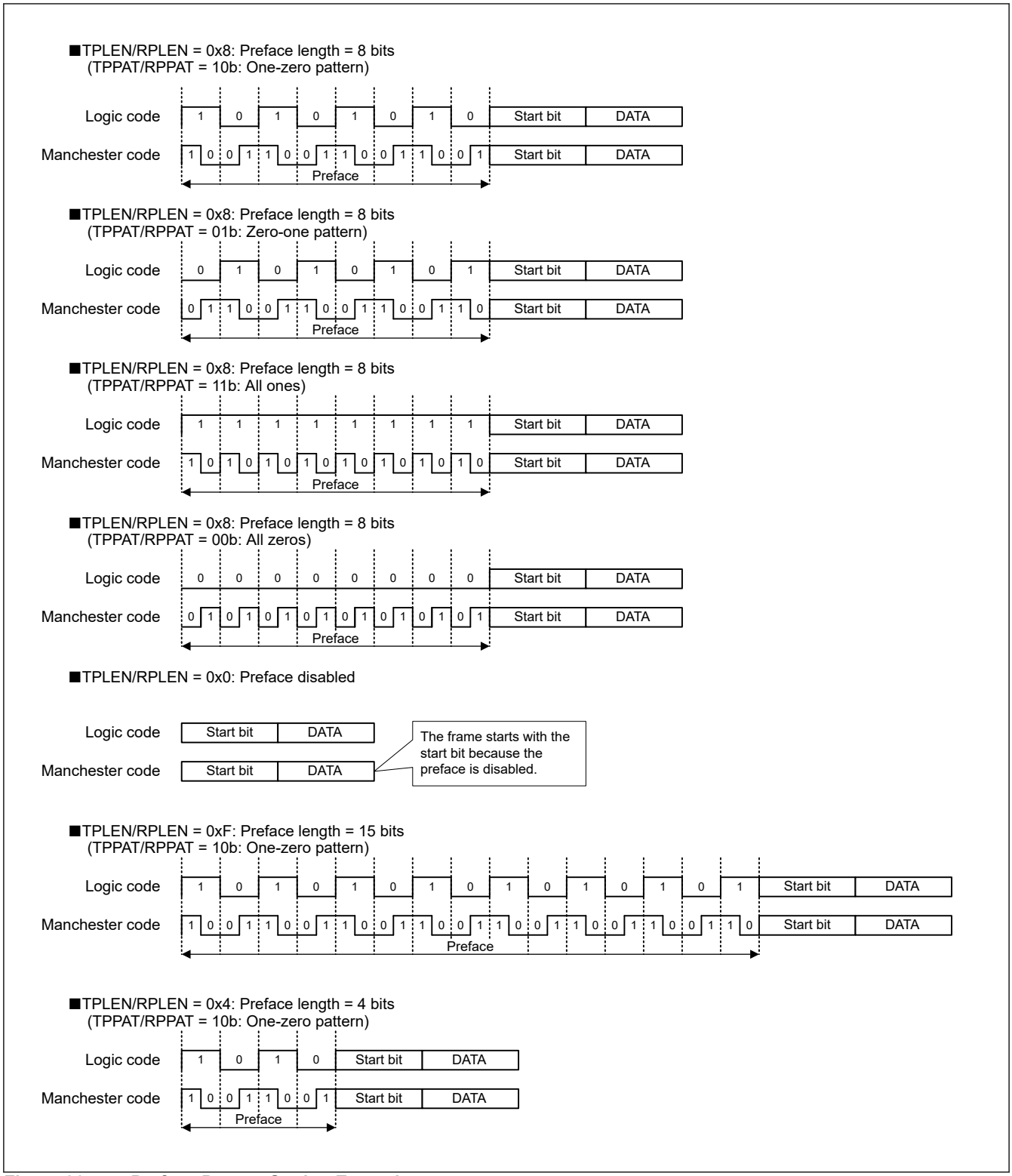


Figure 26.45 Preface Pattern Setting Example

(2) Start bit area

This is an area indicating the start of valid data in a frame. It is added after the preface area.

The start bit length is determined by MCR.SBSEL setting. When MCR.SBSEL = 0, the start bit length is 1 bit.

When MCR.SBSEL = 1, the start bit length is 3 bits.

When MCR.SBSEL = 1, the SYNC type can be selected from command SYNC and data SYNC.

Command SYNC means the three start bits are added as a one-to-zero transition.

Data SYNC means the three start bits are added as a zero-to-one transition.

The SYNC type is determined by the MCR.SYNSEL, MCR.SYNVAL and TDR.TSYNC settings.

(When receiving, the received result is applied to MSR.RSYNC.)

When MCR.SBSEL = 0, the start bit is added as a zero-to-one or one-to-zero transition.

The selection is determined by the MCR.SYNVAL setting.

The MCR.SYNSEL bit specifies the destination to be referred to when setting for transmission.

When the MCR.SYNSEL bit is set to 1, the MCR.SYNVAL setting is referred to. When the MCR.SYNSEL bit is set to 0, the TDR.TSYNC setting is referred to.

Figure 26.46 shows the state of the start bit area according to the settings in the MCR.SYNSEL, MCR.SYNVAL and TDR.TSYNC registers in the case of transmission. Figure 26.47 shows that in the case of reception.

The start bit(s) is not affected by the MCR.TMPOL or MCR.RMPOL setting.

*d.c. = don't care, Preface = preface area, START = start bit area, Data = user data area,
P = parity bit, MP = multi-processor bit, STOP = stop bit

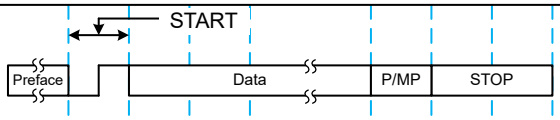
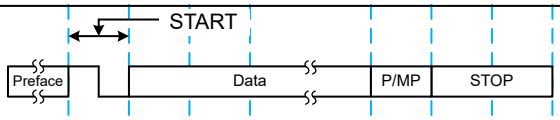
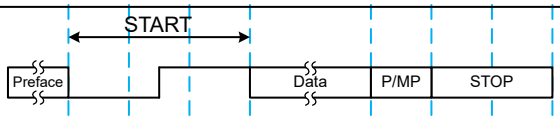
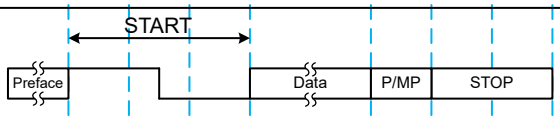
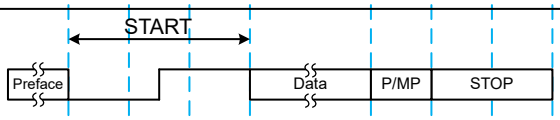
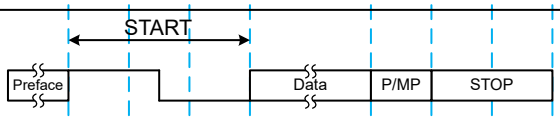
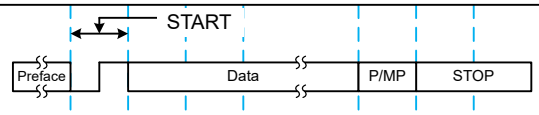
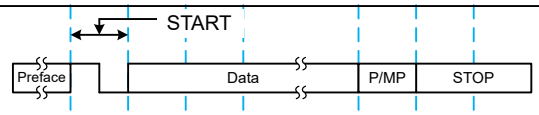
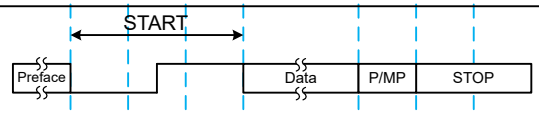
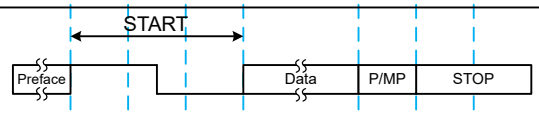
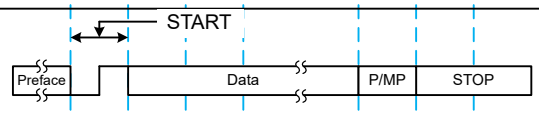
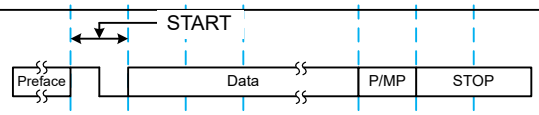
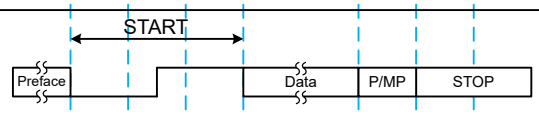
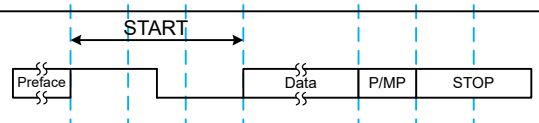
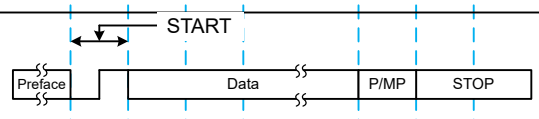
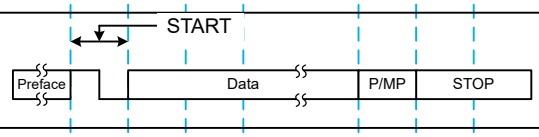
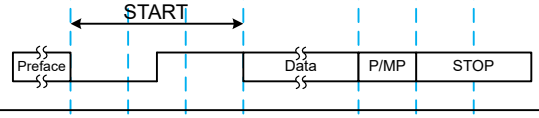
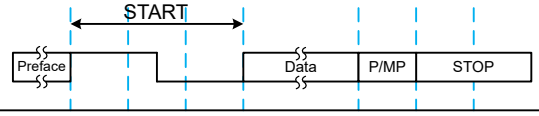
Register setting				Output signal	Output Start bit
MCR			TDR	Output TXDn waveform	
SBSEL	SYNSEL	SYNVAL	TSYNC		
0	d.c.	0	d.c.		1bit 0 to 1 transition
0	d.c.	1	d.c.		1bit 1 to 0 transition
1	0	0	d.c.		3bit 0 to 1 transition (DATA SYNC)
1	0	1	d.c.		3bit 1 to 0 transition (COMMAND SYNC)
1	1	d.c.	0		3bit 0 to 1 transition (DATA SYNC)
1	1	d.c.	1		3bit 1 to 0 transition (COMMAND SYNC)

Figure 26.46 Settings Related to and Format of the Start Bit Area at Transmission

d.c. = don't care, Preface = Preface area, START = Start bit area, Data = Data area
P = Parity bit, MP = Multi-processor bit, STOP = Stop bit

Register setting				Input signal	Start bit detection result*1	Register indication
MCR			TDR	RXDn input waveform		MSR.RSYNC
SBSEL	SYNSEL	SYNVAL	TSYNC			
0	d.c.	0	d.c.		Normal start bit (1 bit: 0-to-1 transition)	0
					Start bit error	0
					Start bit error	0
					Start bit error	0
0	d.c.	1	d.c.		Start bit error	0
					Normal start bit (1 bit: 1-to-0 transition)	0
					Start bit error	0
					Start bit error	0
1	d.c.	d.c.	d.c.		Start bit error	0
					Start bit error	0
					Data SYNC	0
					Command SYNC	1

Note 1. Data other than the start bit is assumed to be normal.

Figure 26.47 Settings Related to and Judgment of the Start Bit Area at Reception

(3) DATA

Since the format of the data area is the same as that of the asynchronous mode, see section 26.3.1. Serial Data Transfer Format.

As shown in Figure 26.43, Frame Format in Manchester Mode, the stop bit is not included in the Manchester encoding range.

26.5.2 Clock

As the transfer clock in Manchester mode, the clock generated by the on-chip baud rate generator is used by setting the CCR2.CKS[1:0] bit.

Also it is possible to set the oversampling (transfer rate of one-bit period) by CCR2.ABCS bit.

When the CCR2.ABCS bit is set to 0, oversampling x16 is selected with the one-bit period being 16 cycles of the base clock. When the CCR2.ABCS bit is set to 1, oversampling x8 is selected with the one-bit period being 8 cycles of the base clock.

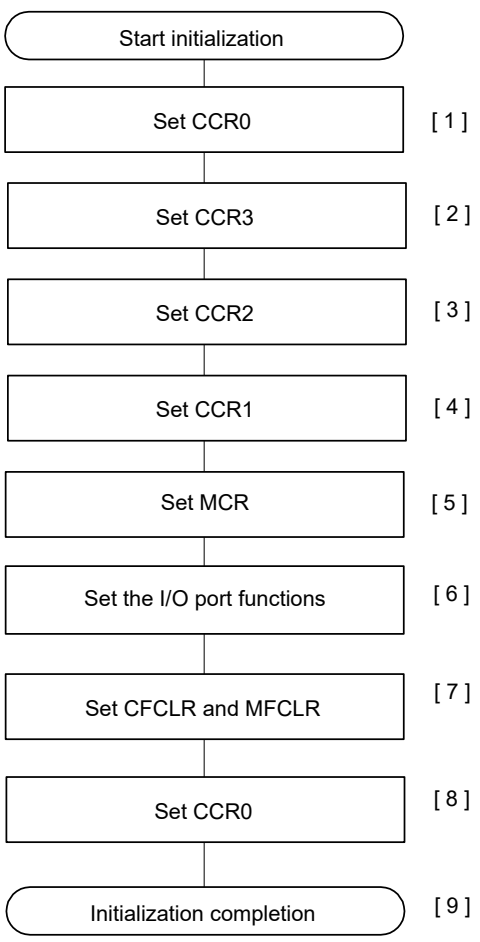
26.5.3 Initialization of the SCI in Manchester Mode

Before transferring data, write 0 to CCR0.TE and CCR0.RE (or write the initial value to CCR0 register) and initialize the SCI following the example of flowchart shown in Figure 26.48.

Whenever the operating mode or transfer format is changed, the CCR0 register must be initialized before the change is made.

Note that setting the CCR0.RE bit to 0 initializes none of the ORER, FER, PER, RDRF, and RDF flags in the CSR register, the SYER, PFER, MER and SBER flags in the MSR register, and the RDR registers.

Note also that switching the value of CCR0.TE from 0 to 1 when CCR0.TIE is 1 generates a SCIn_TXI interrupt request.



[1] Set the following bits of CCR0 to 0.
TEIE, TIE, RIE, TE, RE
If you have not changed from the initial settings, you can skip this step.

[2] Set the driver control, Multi-processor communication, the communication mode, transmission/reception format, and Clock enable.
Do not use function bits such as CPOL and CPHA at the initial value.

[3] Set the clock selection, Bit rate setting.
Leave unused function bits at their initial values.(ex. BRME)

[4] Set the noise filter, loopback control, communication pin status, parity check, and CTS / RTS function.

[5] Set Manchester transmission / reception setting.
Start bit setting, retiming setting, Manchester polarity setting, transmission preface setting, reception preface setting.

[6] Set the I / O port so that the required pin function is valid among the TXDn, RXDn, and SCKn pins.

[7] Set 1 to the following clear bits in the CFCLR and MFCLR registers to clear the target flag.
CFCLR.RDRFC, FERC, PERC, MFFC, ORERC, DFERC, DPERC, DCMFC, ERSC
MFCLR.MERC, SBERC, SYERC, PFERC
This procedure can be omitted for the initial setting after reset.

[9] Set the TE bit or RE bit to 1.
To enable interrupts, set the TE bit and TIE bit, and the RE bit and RIE bit to 1 with one instruction at the same time, and multiprocessor function. The TXDn and RXDn pins can be used by setting the TE and RE bits.

Figure 26.48 SCI Initialization Flow in Manchester Mode

26.5.4 Double-speed operation

When the ABCS bit in CCR2 is set to 1 and eight pulses of the base clock for a 1-bit period is selected, the SCI operates on the bit rate twice that of when ABCS is set to 0.

When the BGDM bit in CCR2 is set to 1, the cycle of the base clock is reduced to half and the SCI operates on the bit rate twice that of when ABCS is set to 0.

When the ABCS and the BGDM bits in CCR2 are set to 1, the SCI operates on the bit rate four times that of when the ABCS and the BGDM bits in CCR2 are set to 0.

26.5.5 CTS and RTS functions

The CTS function uses input on the CTSn pin in transmission control. Setting the CTSE bit in CCR1 to 1 enables the CTS function. The CTSn_RTSn pin can be set as a multiplexed pin which allows one pin to be used for either CTS or RTS function, or as dedicated pins with each pin at CTSn pin for CTS function and CTSn_RTSn pin for RTS function. Use the CTSPEN bit in CCR1 for this setting.

When the CTS function is enabled, reception starts only when the CTSn pin is at the low level.

Applying a high level to the CTSn pin after transmission starts does not affect transmission of the current frame, which continues.

The RTS function uses output on the CTSn_RTSn pin to request transmission. When the SCI is ready to receive, it outputs a low level to the CTSn_RTSn pin, Conditions for output of the low level and high level are as follows:

[Conditions for low-level output]

When all conditions listed below are satisfied:

- The value of the RE bit in CCR0 is 1.
- When The SCI is ready to receive next.
  - When there is no received data yet to be read and not receiving
  - All of the following flags are set to 0: CSR.ORER, FER, and PER, MSR.MER, SYER (when SYEREN = 1), PFER (when PFEREN = 1) and SBER flags (when SBEREN = 1).

[Conditions for high-level output]

- When the conditions for low output are not satisfied

26.5.6 Serial data transmission in Manchester mode

The SCI encodes data in Manchester encoding and sends the resultant data in Manchester mode.

When the polarity setting (MCR.TMPOL) set to 0, logic 0 is coded as a zero-to-one transition in Manchester code and logic 1 is coded as a one-to-zero transition in Manchester code.

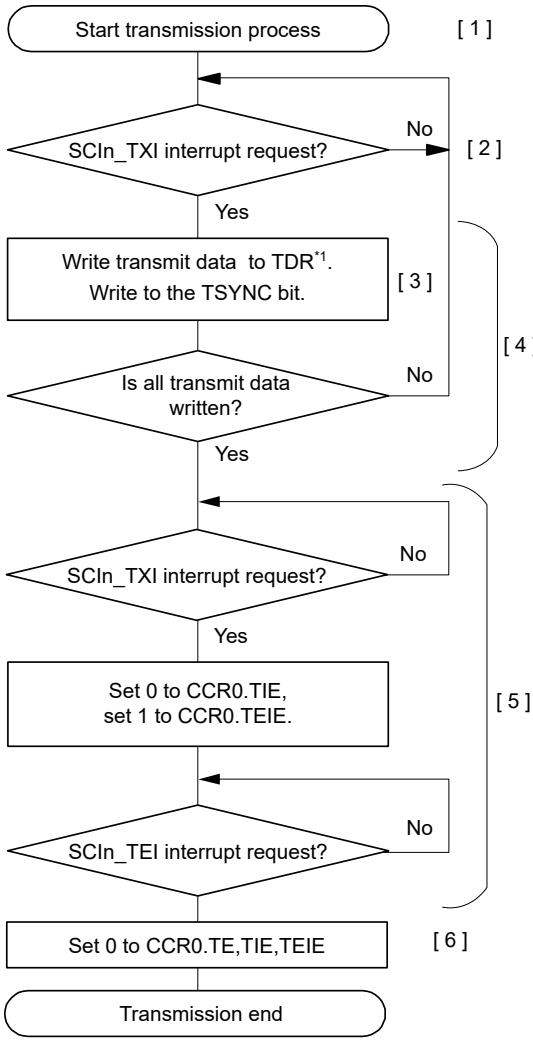
When the polarity setting (MCR.TMPOL) set to 1, logic 0 is coded as a one-to-zero transition in Manchester code and logic 1 is coded as a zero-to-one transition in Manchester code.

For this reason, a level transition occurs with the Manchester encoded data in the middle of individual logic data. (See Figure 26.43).

The transmitter constructs transmit frames in a specific format by adding a preface area to data and setting the start bit(s) according to the polarity setting and sends resultant serial data.

For details on the frame format, see section 26.5.1. Frame Format.

Figure 26.49 shows the flowchart in transmission. At transmission starts, set the CCR0.TIE and CCR0.TE bits to 1 simultaneously with one instruction. Then, a SCIn_TXI interrupt request is generated. Figure 26.50, Figure 26.51, and Figure 26.52 show examples of the operation for serial transmission in Manchester mode.



[1] Start transmission process

[2] SCIn_TXI interrupt:
A transmit data empty interrupt (SCIn_TXI) request is generated when CCR0.TIE = 1 and TE = 0->1, and when transmit data is transferred from TDR to TSR.

[3] TDR setting:
Write the transmit data to TDR.
When TSYNC is used, also set TDR.TSYNC.

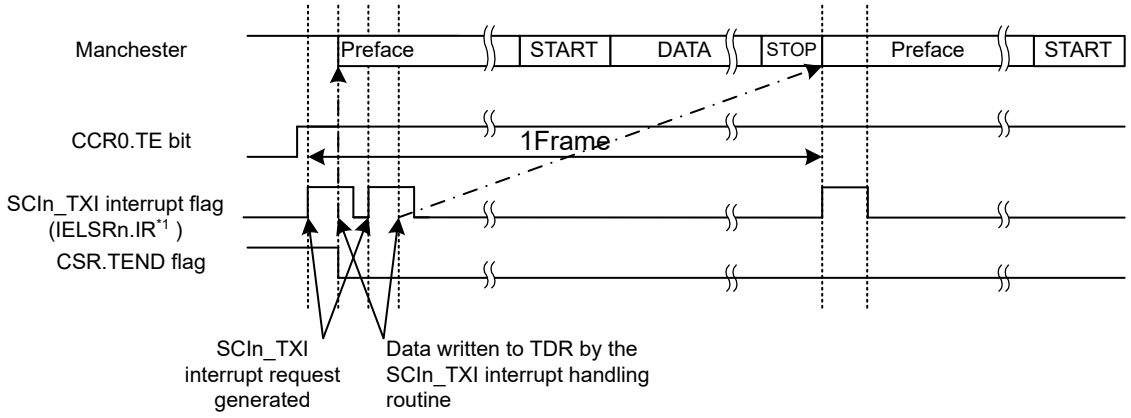
[4] Serial transmission continuation procedure:
To continue serial transmission, write the transmit data to TDR*1 once by using an SCIn_TXI interrupt.
The transmit data can also be written to TDR*1 by activating the DMAC or DTC in the SCIn_TXI interrupt request.

[5] When SCIn_TEI interrupt requests are used, set the CCR0.TIE bit to 0 and the CCR0.TEIE bit to 1 after the last of the data to be transmitted is written to TDR and it's transmission is started.

[6] Set 0 to CCR0.TE, TIE and TEIE.

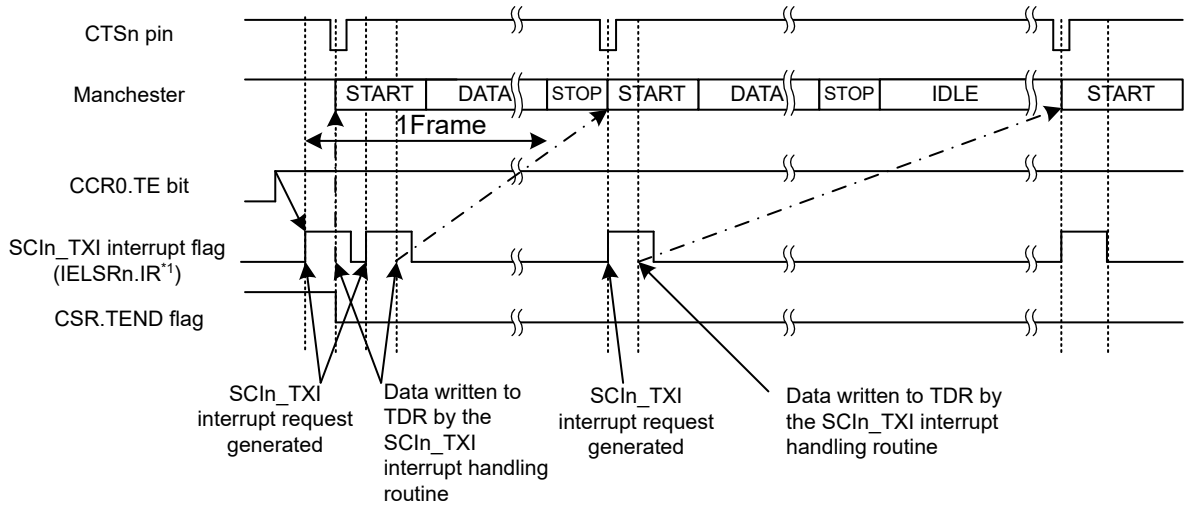
Note 1. If TDR[15:8] requires data write and byte access is required, write the transmit data in the order of TDR [15:8] and TDR [7:0].

Figure 26.49 Example of Serial Transmission Flowchart in Manchester Mode



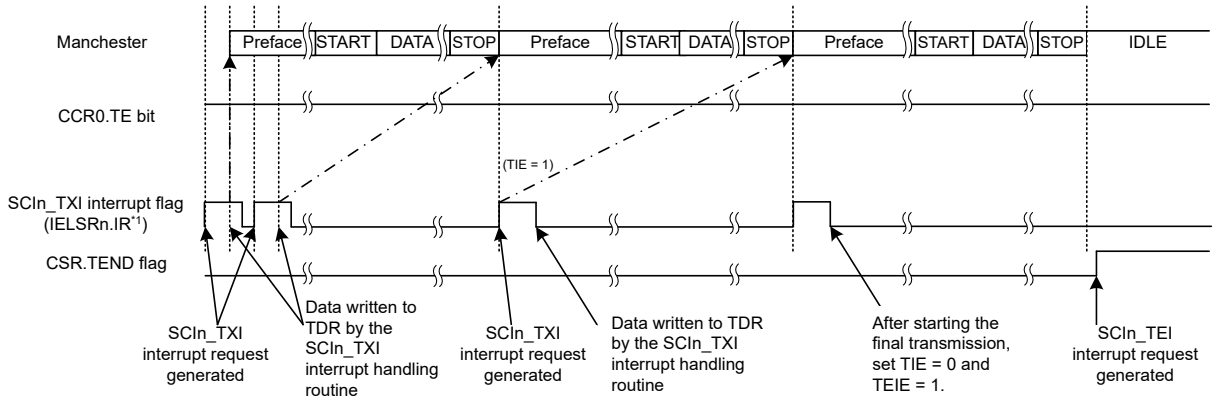
Note 1. For corresponding interrupt numbers, see section 12, Interrupt Controller Unit (ICU)

Figure 26.50 Example of Start-of-Transmission Operation for Serial Transmission in Manchester mode (with Preface but Without the CTS Function)



Note 1. For corresponding interrupt numbers, see section 12, Interrupt Controller Unit (ICU)

Figure 26.51 Example of Start-of-Transmission Operation for Serial Transmission in Manchester Mode (Without Preface but with the CTS Function)



Note 1. For corresponding interrupt numbers, see section 12, Interrupt Controller Unit (ICU)

Figure 26.52 Example of End-of-Transmission Operation for Serial Transmission in Manchester Mode (with Preface but Without the CTS Function)

26.5.7 Serial Data Reception in Manchester Mode

In Manchester mode, the SCI operates on a base clock with a frequency of 16 times*1 the bit rate. Reception starts by sampling the falling edges of received data at the base clock. As shown in Figure 26.53, reception starts at a falling edge of the received data and it continues if the received data keeps low for the duration of 1/4 bit. If the received data goes high within the duration of 1/4 bit, the SCI judges it as an error and waits for a falling edge again.

If a high level is expected in the first half of a bit in the received data, the SCI judges a low level that continues for one base clock cycle as an error and ignores the change to the low level.

Note 1. This is the case when CCR2.ABCS = 0. When CCR2.ABCS = 1, the SCI operates on a base clock with a frequency of 8 times the bit rate.

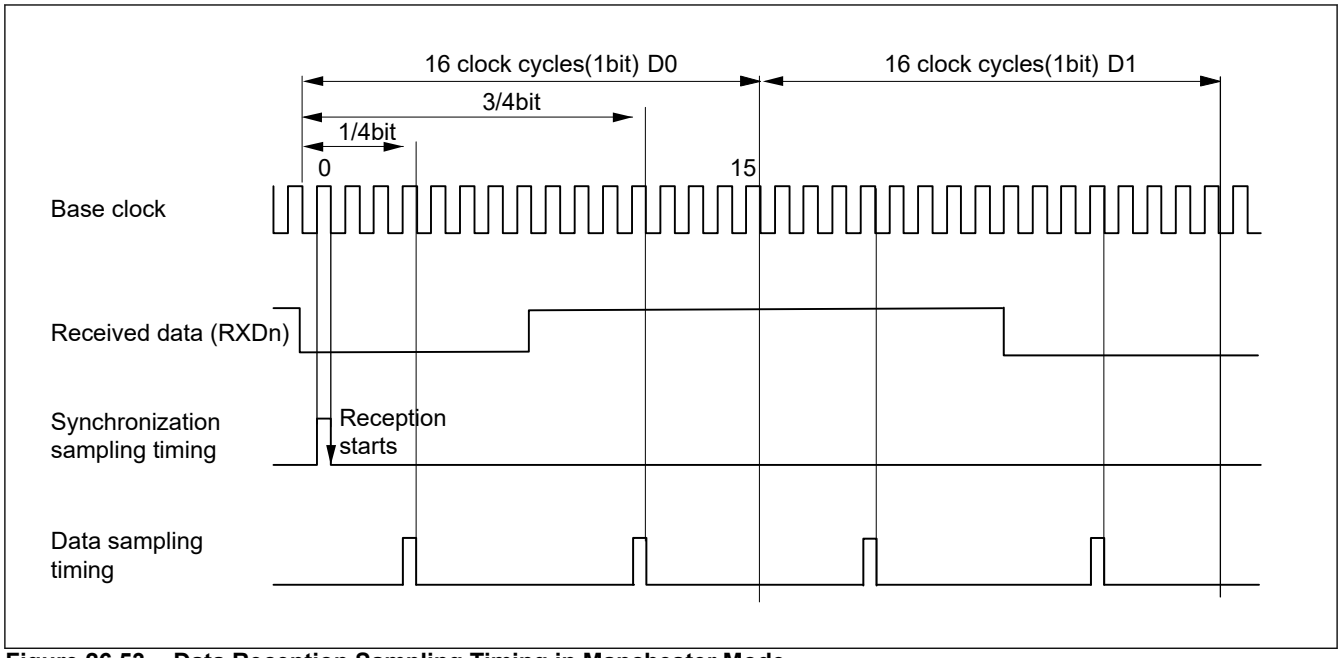


Figure 26.53 Data Reception Sampling Timing in Manchester Mode

In Manchester mode, data reception starts with detection of a preface and start bit area.

The SCI checks the input from the RXDn pin to see whether a preface is added based on the value of MCR.RPLEN.

If the preface is disabled (MCR.RPLEN = 0), it moves on to the detection of a start bit area without detecting a preface.

When a preface is enabled, it identifies a preface pattern setting according to the set value in MCR.RPPAT, and compares it with the RXDn input for a pattern match to detect a preface pattern.

Upon detection of a preface pattern match, it judges it as a normal preface and moves on to the detection of a start bit area.

If detecting a preface pattern mismatch or a Manchester code error in the preface area, it judges it as a preface error and asserts a preface error (PFER).

For start bit detection, the SCI selects an expected value based on the register settings (MCR.SBSEL and SYNVAL), compares it with the RXDn input for a pattern match to detect a start bit area. Upon detection of a start bit pattern match, it judges it as a normal start bit area and moves on to the data processing.

Only when a preface and a start bit area are detected normally, it moves on to the next phase of data reception.

Upon detection of a start bit pattern mismatch, it asserts a start bit error flag (SBER).

In data processing, the SCI shifts the data by the expected received data length based on the register settings (CCR3.CHR[1:0]) through the RSR register. If two sampling points in a bit of the received data are identical, the SCI judges this as a Manchester code error.

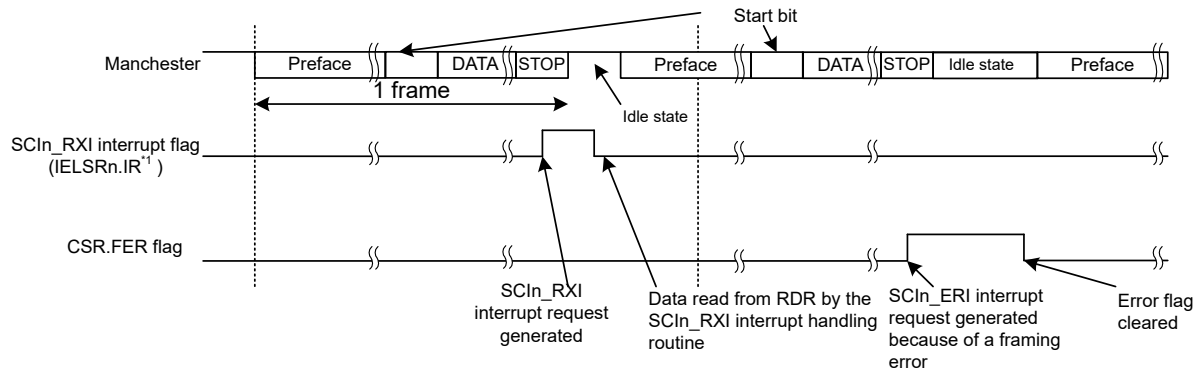
For details, see section 26.5.11. Errors in Manchester Mode (4).

When the parity function is disabled (CCR1.PE = 0), the SCI moves on to the next phase of stop bit detection. When the parity function is enabled (CCR1.PE = 1), the SCI performs parity checking. If detecting a parity error, it asserts a parity error flag (PER), and then moves on to stop bit detection.

In stop bit detection, the SCI checks the following in the stop bit area of the received frame:

It has two sampling points in a bit. If both points are at the high level, the bit is recognized as a normal stop bit and the data is stored in the RDR register. At least one low-level point is judged as an abnormal stop bit, causing a framing error flag (FER) to be set. Even when an error is detected, the received data is stored in the RDR register as abnormal data.

Figure 26.54 shows an example of the operation for serial data reception in Manchester mode.



Note 1. For corresponding interrupt numbers, see section 12, Interrupt Controller Unit (ICU).

Figure 26.54 Example of Operation for Serial Data Reception in Manchester mode (with a Preface)

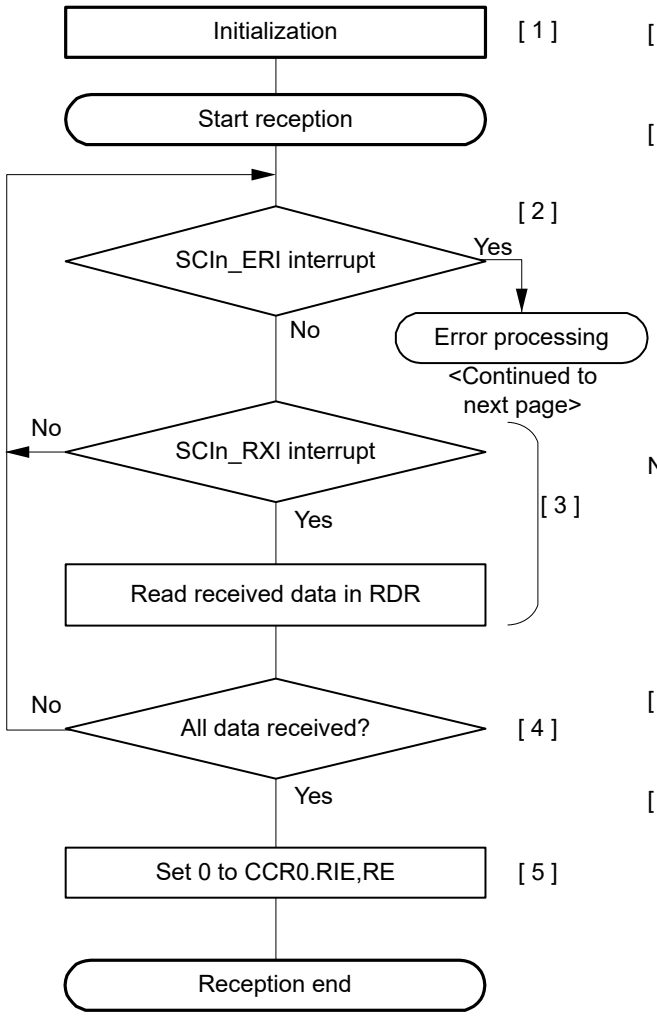
For the state of each status flag in the CCR0 register and RXDn input processing when a receive error is detected, see section 26.5.11. Errors in Manchester Mode.

If a receive error is detected, an SCIn_ERI interrupt request is generated but an SCIn_RXI interrupt request is not generated.

Data reception cannot be resumed while the receive error flag is 1. Accordingly, set the ORER, FER, PER, MER, SYER*1, PFER*1, and SBER*1 flags to 0 before resuming reception. Also, be sure to read the RDR register during overrun error processing. When a reception is forcibly terminated by setting the CCR0.RE bit to 0 during operation, read the RDR register because received data which has not yet been read may be left in the RDR register.

Figure 26.55 and Figure 26.56 show examples of serial data reception flowchart in Manchester mode.

Note 1. Effective when the corresponding bit is enabled.



[1] SCI initialization.
(Set CCR0.RE = 1 and RIE = 1)

[2] Receive error processing:
When a receive error occurs, SCI generates an SCIn_ERI interrupt . An error is identified by reading the ORER, PER, FER in CSR and MER, SBER, SYER, PFER in MSR. After performing the appropriate error processing, always set the ORER, PER, FER, MER, SBER, SYER, PFER flags to 0. Reception cannot be resumed if any of these flags is set to 1.

Note: Each of the PFER, SYER, SBER bits in MSR has an enable bit (PFEREN, SYEREN, and SBEREN in MECR). If the corresponding enable bit is set to 0 (disable), the flag value does not affect the communication. Perform error processing as required.

[3] Read the received data in RDR once in the SCIn_RXI interrupt handling routine.

[4] Serial reception continuation procedure:
To continue serial reception, before the stop bit of a frame is received, read data from RDR in the SCIn_RXI interrupt handling routine. The RDR data can also be read by activating the DMAC or DTC.

[5] Write 0 to CCR0.RIE and RE.

Figure 26.55 Example of Serial Data Reception Flowchart in Manchester Mode (Normal Reception)

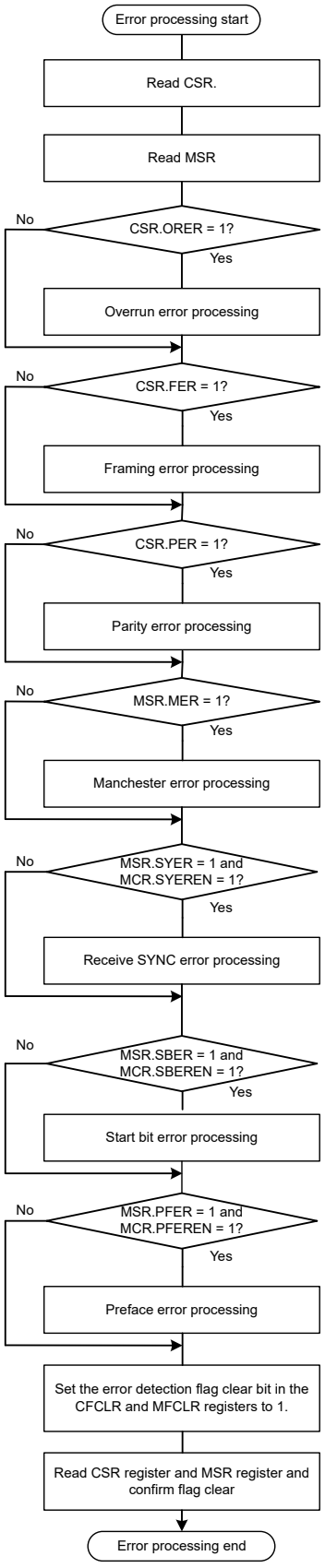


Figure 26.56 Example of Serial Reception Flowchart in Manchester Mode (Error Processing)

26.5.8 Operation When Multi-Processor Bit Is Used

See section 26.4. Multi-Processor Communication Function (1) for the operation in Manchester mode when using multi-processor mode because the operation is the same.

A preface and a start bit area are added to the frame format in Manchester mode. See Figure 26.56 for error processing in Manchester mode for the reception flowchart (Figure 26.39). See Table 26.34 for the operation status when detecting various errors.

26.5.9 Receive Retiming

This function corrects the timing for each central edge of the bit, taking advantage of the fact that each bit has an edge in the center in Manchester code.

The receive retiming function can be turned on or off by setting the ERTEN bit in the MCR register.

When the receive retiming function is turned off (MCR.ERTEN = 0), retiming is not performed, causing misalignment between the internal clock and the RXDn input to be accumulated and the receive margin to be reduced.

When the receive retiming function is turned on (MCR.ERTEN = 1), retiming is performed for the preface area, the start bit area*1, and the data area (excluding the stop bit).

Note 1. Retiming is not performed for the start bit area if the preface length is 0 and the start bit length is 3.

As an example, the receive retiming when oversampling x16 is selected is shown below.

When detecting an RXDn input edge two to four cycles before the expected receive cycle, the receive processing is shortened by one sampling CLK cycle.

When detecting a RXDn input edge two to three cycles after the expected receive cycle, the receive processing is extended by one sampling CLK cycle.

(Even if the clock is misaligned with the data by more than two cycles, one cycle is corrected for each bit.)

Figure 26.57 shows the conceptual image of receive retiming range.

When detecting an edge in the tolerance area in the figure, data is received as is without making correction.

When detecting an edge in the SyncJump area in the figure, data is corrected for reception.

When detecting an edge in the SyncError area in the figure, data is received as abnormal data with no correction made.

For a Manchester code error (data matches at the 1/4-phase and 3/4-phase sampling points), the SCI reports a code error.

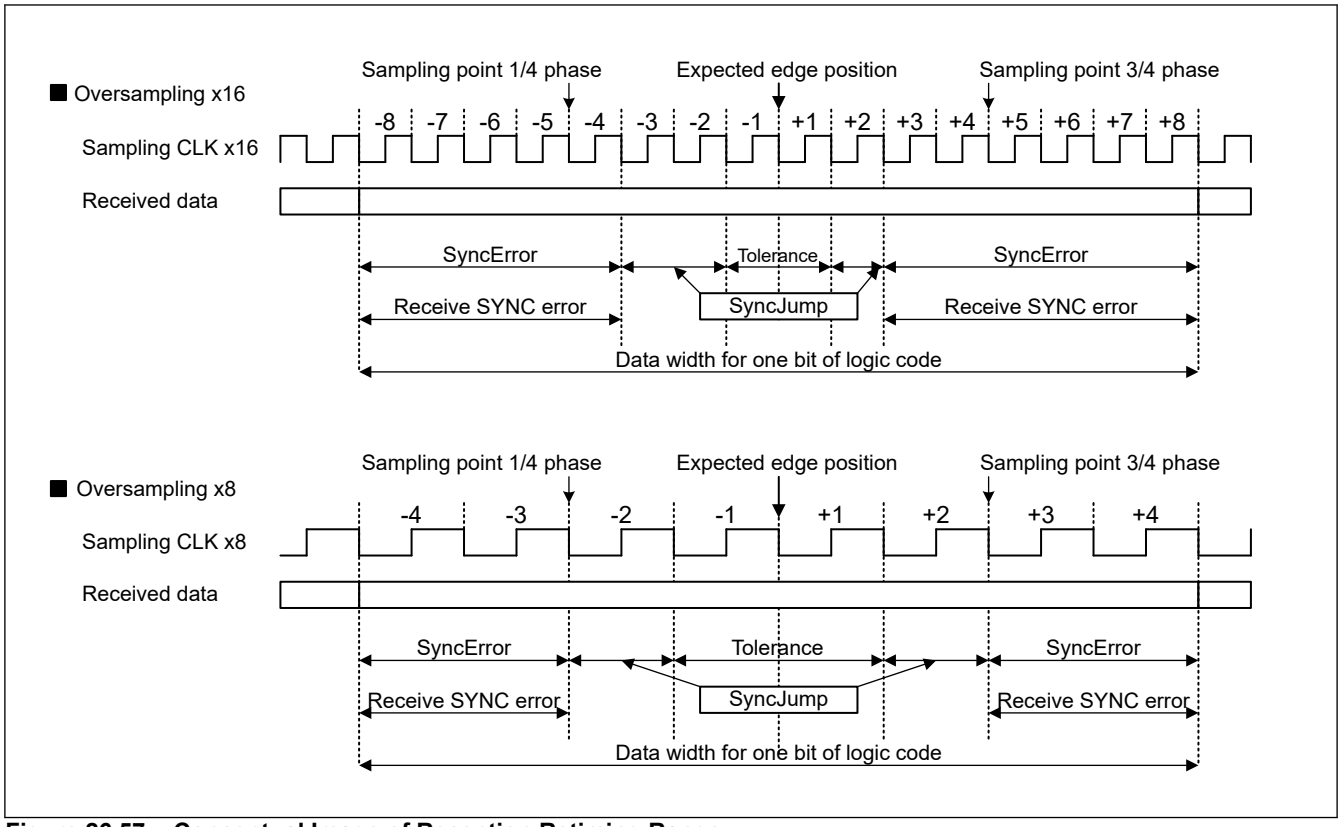


Figure 26.57 Conceptual Image of Reception Retiming Range

26.5.10 Polarity Setting for Manchester Code

The polarity of the Manchester code can be set with the Manchester Control Register (MCR).

It can be set separately for transmission and reception. Use the MCR.TMPOL bit to set the polarity for transmission and the MCR.RMPOL bit to set the polarity for reception.

The Manchester code polarity setting is valid for the preface area, the data area, and the parity or multi-processor area.

When the initial settings (TMPOL/RMPOL = 0) are used for the polarity of Manchester code, logic 0 is encoded as a zero-to-one transition in Manchester code and logic 1 is encoded as a one-to-zero transition in Manchester code. If the settings are changed to TMPOL/RMPOL = 1, logic 0 is encoded as a one-to-zero transition in Manchester code and logic 1 is encoded as a zero-to-one transition in Manchester code. Figure 26.58 shows the conceptual image of the settings and operation.

Separately from the function above, the transmitted and received data in the data area can be inverted by the transmitted/received data inversion function (CCR3.SINV). Since the polarity of Manchester code (MCR.TMPOL/RMPOL) can be set separately from the transmitted/received data invert function (CCR3.SINV), if both are set to inversion (MCR.TMPOL/RMPOL = 1 and CCR3.SINV = 1), the transmitted and received data are set to initial state (inversion + inversion = normal).

The polarity of the start bit area can be set by a register different from the ones mentioned above.

Since a different register is used, the polarity of the start bit area is not affected by the polarity setting for Manchester code mentioned above.

For details on the setting for the start bit area, see section 26.5.1. Frame Format (2).

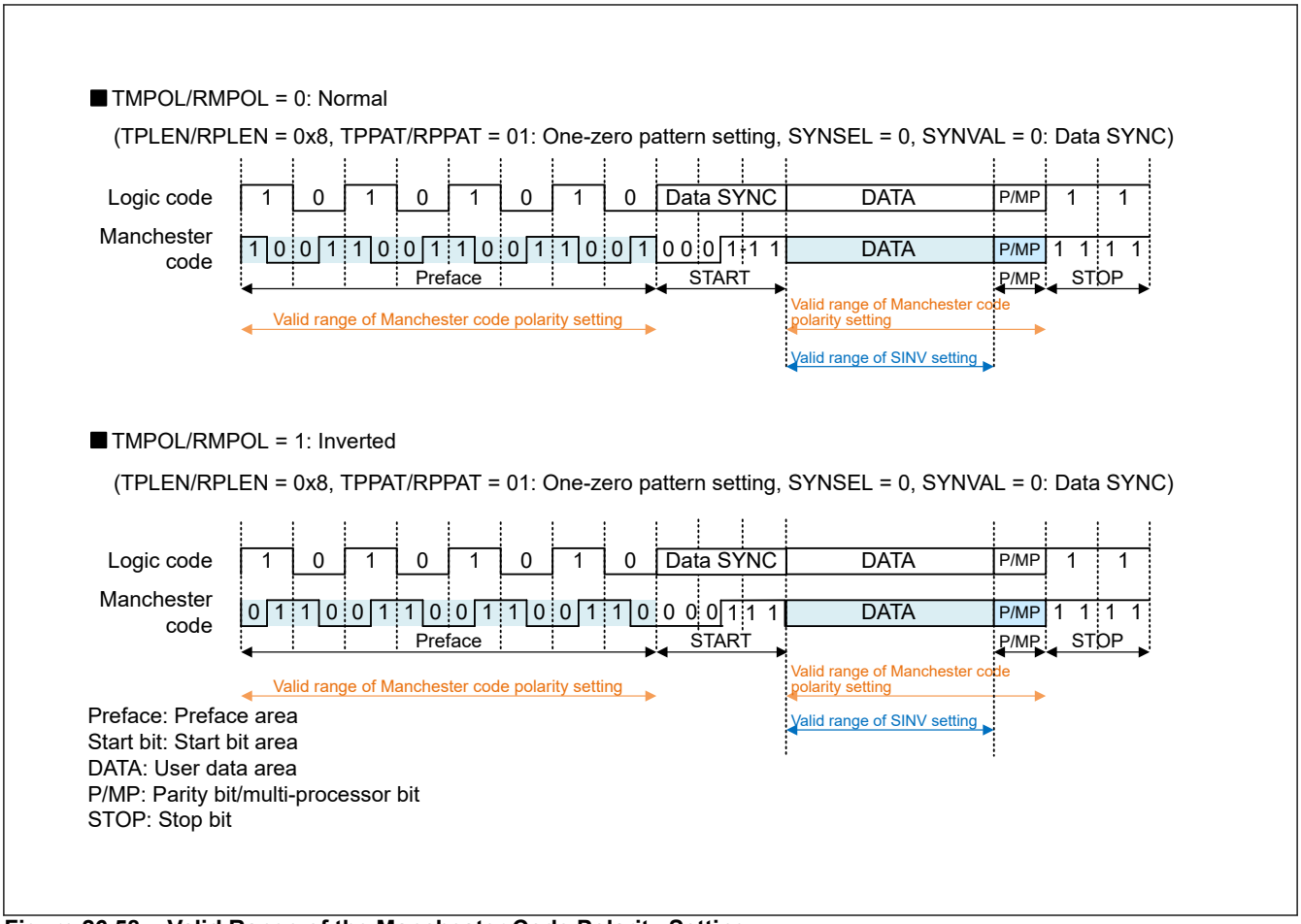


Figure 26.58 Valid Range of the Manchester Code Polarity Setting

26.5.11 Errors in Manchester Mode

There are the following errors in Manchester mode:

1. Parity error
2. Over run error
3. Framing error
4. Manchester error
5. Preface error
6. Start Bit error
7. Receive SYNC error

For errors (1) to (3), see section 26.3.9. Serial Data Reception in Asynchronous Mode (1) because they are the same as in asynchronous mode.

Each errors are judged in each area, but they are reflected on flags and operations at the timing of 3/4-bit sampling of the STOP bit area. If a preface error or start bit error is detected, subsequent data will not be received. Therefore, no other error detection is performed, and the error flag holds the previous information.

Table 26.32 lists the states of the serial status register when detecting errors and judgment about whether to store data in the RDR.

Table 26.33 lists the errors that can be detected in each area of a Manchester frame.

If a Preface error or Start bit error is detected, subsequent data will not be received. Therefore, no other error detection is performed, and the error flag holds the result of the previous frame reception. Also, if an error is detected in the previous

frame, data will not be received, but errors in the pre-face area and start bit area will update that flag. Table 26.34 shows the flags and actions in this case.

(4) Manchester error

A Manchester error is generated when a Manchester code error is detected.

In Manchester code, there must be an edge (transition) in the center of the bit.

In the data area of a received frame (including the parity/multi-processor bit), the values of the 1/4-bit and 3/4-bit sampling points are checked in each received 1-bit data, and a Manchester code error is determined if these two values match.

If a Manchester code error is detected, the Manchester error flag (MSR.MER) is asserted.

If a Manchester error occurs, it is handled as an interrupt source and event source. If a Manchester error is detected, the next reception is not performed until the corresponding error flag is cleared.

(5) Preface error

A preface error is generated when the preface pattern does not match or a Manchester code error is detected in the preface area. If a preface error is detected, the preface error flag (MSR.PFER) is asserted.

It is possible to set whether to use this error flag as an interrupt source with the setting of the MCR register.

When MCR.PFEREN = 1, a preface error is handled as an interrupt source or event source. If a preface error is detected, the next reception is not performed until the corresponding error flag is cleared.

When MCR.PFEREN = 0, a preface error is not handled as an interrupt source or event source, and the next reception is not halted. However, a preface error is notified to MSR.PFER.

(6) Start bit error

A start bit error is generated when a mismatch is detected between the start bit area in the received frame and the preset start bit pattern. Upon detection of a start bit error, a start bit error flag (MSR.SBER) is asserted.

It is possible to set whether to use the start bit error as an interrupt source with the setting of the MCR register.

When MCR.SBEREN = 1, a start bit error is handled as an interrupt source or event source. If a start bit error is detected, the next reception is not performed until the corresponding error flag is cleared.

When MCR.SBEREN = 0, a start bit error is not handled as an interrupt source or event source, and the next reception is not halted. However, a start bit error is notified to MSR.SBER.

(7) Receive SYNC error

When the receive retiming function described in section 26.5.9. Receive Retiming is enabled, the receive retiming operation is performed.

If no edges are detected within the receive retiming range (SyncError area in Figure 26.57) when receive timing operation is being performed, a receive SYNC error is generated. Upon detection of a receive SYNC error, a receive SYNC error flag (MSR.SYER) is asserted. In areas not subject to retiming, receive SYNC errors are not detected.

The preface area[*1], the start bit area[*1],[*2], and the data area (excluding the stop bit) for which receive retiming operation is performed are checked.

It is possible to set whether to use the receive SYNC error as an interrupt source with the setting of the MCR register.

When MCR.SYEREN = 1, a receive SYNC error is handled as an interrupt source or event source. If a receive SYNC error is detected, the next reception is not performed until the corresponding error flag is cleared.

When MCR.SYEREN = 0, a receive SYNC error is not handled as an interrupt source or event source, and the next reception is not halted. However, a receive SYNC error is notified to MSR.SYER.

Note 1. In the case of a frame that starts with a pattern that expects the first half of the bit to be High, it is excluded from retiming.

Note 2. In the start bit area, when there is no preface length and 3 bit start bit is set, it is not subject to retiming.

Also, the 1st bit and the 2nd bit in the start bit area when 3 bit start bit is set are not subject to retiming.

Table 26.32 Flags in the CSR Register and Receive Data Handling in Manchester Mode

Flag in the CSR register			Flag in the MRS register				Received data	Received error status (SCIn_ERI interrupt / event generation)
ORER	FER	PER	MER	SBER*1	PFER*1	SYER		
0	0	0	0	0	0	0	transfer to RDR	No error
0	1	0	0	0	0	0	transfer to RDR	Framing error
0	0	1	0	0	0	0	transfer to RDR	Parity error
0	1	1	0	0	0	0	transfer to RDR	Framing error + Parity error
0	0	0	1	0	0	0	transfer to RDR	Manchester error
0	1	0	1	0	0	0	transfer to RDR	Framing error + Manchester error
0	0	1	1	0	0	0	transfer to RDR	Parity error + Manchester error
0	1	1	1	0	0	0	transfer to RDR	Framing error + Parity error + Manchester error
1	0	0	0	0	0	0	Lost	Overrun error
1	1	0	0	0	0	0	Lost	Overrun error + Framing error
1	0	1	0	0	0	0	Lost	Overrun error + Parity error
1	1	1	0	0	0	0	Lost	Overrun error + Framing error + Parity error
1	0	0	1	0	0	0	Lost	Overrun error + Manchester error
1	1	0	1	0	0	0	Lost	Overrun error + Framing error + Manchester error
1	0	1	1	0	0	0	Lost	Overrun error + Parity error + Manchester error
1	1	1	1	0	0	0	Lost	Overrun error + Framing error + Parity error + Manchester error
0	Combination of above			0	0	1	transfer to RDR	Errors above + Receive SYNC error*2
1				0	0	1	Lost	Errors above + Receive SYNC error*2
hold	hold	hold	hold	0	1	0	Lost	Preface error*3
hold	hold	hold	hold	1	0	0	Lost	Start bit error*3
hold	hold	hold	hold	0	1	1	Lost	Preface error*3 + Receive SYNC error*2
hold	hold	hold	hold	1	0	1	Lost	Start bit error*3 + Receive SYNC error*2

Note 1. Start bit error and Preface error never become 1 at the same time.
Note 2. When MCR.SYEREN = 1, SCIn_ERI interrupt / event is generated by SYER factor.
Note 3. If MCR.PFEREN = 1 or MCR.SBEREN = 1, an SCIn_ERI interrupt / event is generated when the corresponding flag is set.

Table 26.33 Errors Detectable in Each Area

	Preface error (PFER)	Start Bit error (SBER)	Manchester error (MER)	Receive SYNC error (SYER)	Parity error (PER)	Framing error (FER)
Preface area	✓	—	—*1	✓*2	—	—
Start Bit area	—	✓	—	✓*2	—	—
Data area	—	—	✓	✓	—	—
Parity area	—	—	✓	✓	✓	—
Multi-processor area	—	—	✓	✓	—	—
Stop Bit area	—	—	—	—	—	✓

Note: ✓: Detected, —: Not detected
Note 1. When an Manchester code error occurs in the preface area, it is defined as a preface error.
Note 2. It may not be subject to Receive SYNC error detection. For details see the text section 26.5.11. Errors in Manchester Mode (7)

Table 26.34 Operation status due to presence / absence of error in previous frame and operation status list in multiprocessor mode (1 of 2)

Previous frame	Each area of the Frame					PFEREN	SBEREN	SYEREN	received data	Error flag	Interrupt request	Event signal
	preface	start bit	data	parity	stop							
No Error	PFER	No Error	Don't Care	Don't Care	Don't Care	0	Don't Care	Don't Care	Lost	set PFER*1	not output	not output
	No SYER*1					1					output	output
	No Error	SBER	Don't Care	Don't Care	Don't Care	Don't Care	0	Don't Care	Lost	set SBER*1	not output	not output
		No SYER*1					1				output	output
	SYER	No Error	Don't Care	Don't Care	Don't Care	Don't Care	Don't Care	0	transfer to RDR	set SYER	not output	not output
	No PFER							1	Lost		output	output
	No Error	SYER	Don't Care	Don't Care	Don't Care	Don't Care	Don't Care	0	transfer to RDR	set SYER	not output	not output
		No SBER						1	Lost		output	output
	No Error	No Error	SYER		No Error	Don't Care	Don't Care	0	transfer to RDR	set SYER	not output	not output
								1			output	output
	No Error	No Error	MER		No Error	Don't Care	Don't Care	Don't Care	transfer to RDR	set MER	output	output
	No Error	No Error	Don't Care	PER	No Error	Don't Care	Don't Care	Don't Care	transfer to RDR	set PER	output	output
	No Error	No Error	Don't Care	Don't Care	FER	Don't Care	Don't Care	Don't Care	transfer to RDR	set FER	output	output
	There is some error ORER					Don't Care	Don't Care	Don't Care	Lost	set some flags*2	output	output
	No Error	No Error	No Error	No Error	No Error ORER	Don't Care	Don't Care	Don't Care	Lost	set ORER	output	output

Table 26.34 Operation status due to presence / absence of error in previous frame and operation status list in multiprocessor mode (2 of 2)

Previous frame	Each area of the Frame					PFERE N	SBERE N	SYERE N	received data	Error flag	Interrupt request	Event signal
	preface	start bit	data	parity	stop							
some error*3 *6	PFER No SYER*1	No Error	Don't Care	Don't Care	Don't Care	0	Don't Care	Don't Care	Lost	set PFER*1	output*4	not output*5
						1						
	No Error	SBER No SYER*1	Don't Care	Don't Care	Don't Care	Don't Care	0	Don't Care		set SBER*1		
							1					
	SYER No PFER	No Error	Don't Care	Don't Care	Don't Care	Don't Care	Don't Care	0		set SYER		
								1				
	No Error	SYER No SBER	Don't Care	Don't Care	Don't Care	Don't Care	Don't Care	0		set SYER		
								1				
	No Error	No Error	SYER		No Error	Don't Care	Don't Care	0		don't set any flags		
								1				
	No Error	No Error	MER		No Error	Don't Care	Don't Care	Don't Care				
	No Error	No Error	Don't Care	PER	No Error	Don't Care	Don't Care	Don't Care				
	No Error	No Error	Don't Care	Don't Care	FER	Don't Care	Don't Care	Don't Care				
	There is some error ORER					Don't Care	Don't Care	Don't Care				
	No Error	No Error	No Error	No Error	No Error ORER	Don't Care	Don't Care	Don't Care				

Note 1. If SYER is detected, the SYER flag is also set. Other operations are as shown in this table.
Note 2. Other detected error flags including ORER are also set.
Note 3. If all the error flags are cleared before the STOP bit is judged, the operation will be the same as the case where there is no error in the previous frame of this table.
Note 4. Since the SCIn_ERI interrupt request is level output, it remains active due to errors in the previous frame regardless of the presence or absence of error in the relevant frame.
Note 5. Since the error cause is continuously detected, the SCIn_ERI event is not newly output regardless of the presence or absence of errors in the relevant frame.
Note 6. For PFER, SBER, and SYER, when each enable bit is set to disable, it is treated as no error.

Table 26.35 Operation when MPIE = 1 in multi-processor mode (MPIE = 0)

MPB*1	Each area of the frame					PFERE N	SBERE N	SYERE N	received data	Error flag	Interrupt request	Event signal
	preface	start bit	data	parity	stop							
1	No Error	No Error	Don't Care	Don't Care	Don't Care	Don't Care	Don't Care	Don't Care	transfer to RDR	set some flags	output*2	output*2
	No PFER	No SBER	Don't Care	Don't Care	Don't Care	Don't Care	Don't Care	0				
	SYER*3	SYER*3						1	Lost	don't set any flags	not output	not output
	PFER	No Error	Don't Care	Don't Care	Don't Care	Don't Care	Don't Care	Don't Care				
	No Error	SBER	Don't Care	Don't Care	Don't Care	Don't Care	Don't Care					

Note 1. If the received MPB bit is 0, it is not received the frame, and the operation is the same as lost of the reception data of this table.
Note 2. If no error is detected, SCIn_RXI interrupt request or event is output, and if it is detected, SCIn_ERI interrupt request or event is output.

Note 3. When SYER is detected in the preface area or the start bit area, the behavior of handling as an error depending on the SYEREN bit changes.

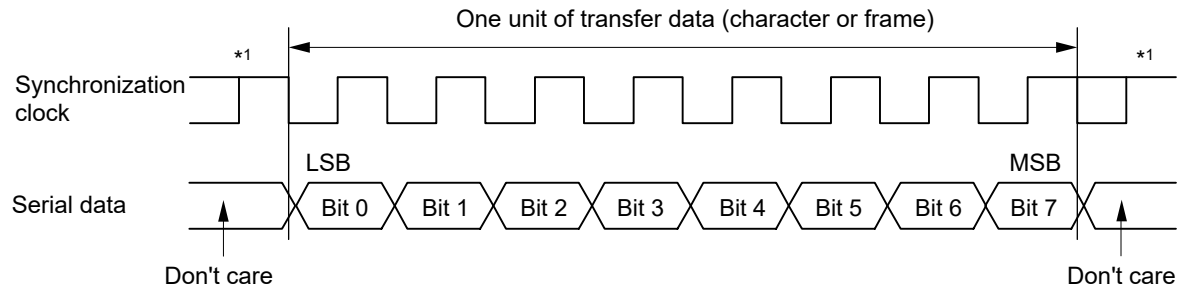
26.6 Operation in Clock Synchronous Mode

Figure 26.59 shows the data format for clock synchronous serial data communications.

In clock synchronous mode, data is transmitted or received in synchronization with clock pulses. For single-character data transfer, data consists of 8-bit. In clock synchronous mode, no parity bit can be added.

In data transmission when CPHA = 1 and CPOL = 1, the SCI outputs data from one falling edge of the synchronization clock to the next falling edge. In data reception, the SCI receives data in synchronization with the rising edge of the synchronization clock. After 8-bit data is output, the transmission line holds the last bit as output state. When the CPHA bit is 0 in slave mode, the transmission line holds the first bit output state.

Within the SCI, the transmitter and receiver are independent units, enabling full-duplex communications by using a shared communication clock of the transmitter and the receiver. Furthermore, because both the transmitter and the receiver also have a double-buffered structure, so that the next transmit data can be written during transmission or the previous receive data can be read during reception, enabling continuous data transfer.



Note 1. Holds a high level except during continuous transfer.

Figure 26.59 Data format in clock synchronous serial communications with LSB-first order

26.6.1 Clock

If the maximum speed of SCK = 1/2TCLK is set in Clock Synchronous and Simple SPI mode, Do not make PCLK less than half the speed of TCLK. If PCLK is made slower than this, malfunction may occur.

1. When the internal clock is selected
   When the SCI operates on an internal clock (CCR3.CKE[1:0] bits are set to 00b or 01b (master mode)), the synchronization clock is output from the SCKn pin. Eight synchronization clock pulses are output during single-character transmission/reception. When no data transfer is performed, the clock is held at high level.*1 In transmission-only or transmission/reception, the synchronization clock is not output unless transmit data is prepared.
   When the internal clock is selected, the clock with a delay from the SCKn signal is used for the master reception sampling clock. This ensures the data setup time and hold time for high-speed communication.

   Note 1. The signal is held high while (CCR3.CPHA = 0 and CCR3.CPOL = 1) or (CCR3.CPHA = 1 and CCR3.CPOL = 1). It is held low while (CCR3.CPHA = 0 and CCR3.CPOL = 0) or (CCR3.CPHA = 1 and CCR3.CPOL = 0).

2. When the external clock is selected
   When the CCR3.CKE[1:0] bits are set to 10b or 11b (slave mode), data is transmitted and received using the external clock that is input from the SCKn pin.

26.6.2 CTS and RTS Functions

In the CTS function, the CTSn_RTSn pin input controls the start of data reception or transmission when the clock source is the internal clock. Setting the CCR1.CTSE bit to 1 enables the CTS function. When the CTS function is enabled, setting the CTSn_RTSn pin low causes data reception or transmission to start.

Setting the CTSn_RTSn pin high while the data transmission or reception is in progress does not affect transmission or reception of the current frame.

In the RTS function, the CTSn_RTSn pin output is used to request the start of data reception or transmission when the clock source is an external synchronizing clock. The CTSn_RTSn output goes low when serial communication is enabled. Conditions for output of the CTSn_RTSn low and high are shown as follows:

[Conditions for low output]

Satisfaction of all the following conditions:

Non-FIFO selected when all of the following conditions are satisfied

- The value of the CCR0.RE bit or the CCR0.TE bit is 1
- Next serial communication is enabled.
  - No receive data is present before reading and not receiving. (when CCR0.RE bit = 1)
  - When the transmission data written in TDR is ready for transmission.*1 (when CCR0.TE bit = 1)
- The CSR.ORER flag is 0

Note 1. The CTSn_RTSn pin will be High after starting transmission.

FIFO selected when all of the following conditions are satisfied

- The value of the CCR0.RE bit or the CCR0.TE bit is 1
- Next serial communication is enabled.
  - The number of receive data stored in the receive FIFO (RDR register) is less than the value set in FCR.RSTRG[4:0] (when CCR0.RE bit = 1)
  - When the transmission data written in the transmission FIFO (TDR register) is ready for transmission.*1 (when CCR0.TE bit = 1)
- The CSR.ORER flag is 0

Note 1. The CTSn_RTSn pin will be High after the last data transmission starts.

[Condition for high output]

- The conditions for low output are not satisfied

26.6.3 SCI Initialization in Clock Synchronous Mode

Before transmitting and receiving data, start by writing the initial value 0x00 to the CCR0 register, then continue through the SCI initialization procedure given in the sections describing non-FIFO and FIFO selection in section 26.6.2. CTS and RTS Functions. Anytime the operating mode or transfer format is to be changed, the CCR0.TE and CCR0.RE must write to 0 before the change can be made.

Note: Setting the CCR0.RE bit to 0 initializes neither the ORER, FER, PER, and RDRF flags in CSR nor the RDR register. When the TE bit is set to 0, the TEND flag for the selected FIFO buffer is not initialized.

Note: Switching the value of the CCR0.TE bit from 1 to 0 when the CCR0.TIE bit is 1 generates an SCIn_TXI interrupt request.

Table 26.36 Example flow of SCI initialization in clock synchronous mode with non-FIFO selected (1 of 2)

No.	Step Name	Description
1	Start initialization	—
2	Set CCR0	Set the CCR0.TEIE, TIE, RIE, TE, and RE bits to 0*1. If you have not changed from the initial settings, you can skip this step.
3	Set FCR	Set the TFRST and RFRST to 1 to empty FIFO. Set the DRES, TTRG[4:0], RTRG[4:0], and RSTRG[4:0] bits.
4	Set CCR3 except MOD[2:0]	Set CCR3 except of communication mode. • FIFO use/no-use • Transmission/reception format • Clock setting • Leave unused bits (CHR[1:0],STP,RXDSEL,MP,DE, ACS0,GM,BLK) at their initial values.

Table 26.36 Example flow of SCI initialization in clock synchronous mode with non-FIFO selected (2 of 2)

No.	Step Name	Description
5	Set CCR3.MOD[2:0]	Set communication mode (MOD[2:0] = 010b)*2.
6	Set CCR2	Select clock, set bit rate *3. Leave unused bits (BCP[2:0],ABCS, ABCSE, BRME,MDDR[7:0]) at their initial values.
7	Set CCR1	Set up the loop-back function, communication pin status and the CTS/RTS function.
8	Set CCR4	Set up the adjust sampling timing function. Leave unused bits (CMPD[8:0]) at their initial values.
9	Set the I/O port functions	Make I/O port settings to enable input and output functions as required for TXDn, RXDn, and SCKn pins.
10	Set CFCLR, FFCLR	Write 1 to the following bits and clear the corresponding flag. CFCLR.RDRFC, FERC, PERC, MFFC, ORERC, DFERC, DPERC, DCMFC, ERSC FFCLR.BRKC, DRC
11	Set CCR0	Set the TE or RE bit to 1.*1 *4 To enable interrupts, set the TE bit and TIE bit, and the RE bit and RIE bit to 1 with one instruction at the same time. Setting the TE and RE bits allows TXDn and RXDn to be used.
12	Initialization completion	—

Note 1. In simultaneous transmit and receive operations, the TE and RE bits in CCR0 should both be to 0 or set to 1 simultaneously.
Note 2. Set CPOL and CPHA before setting the communication mode.
Note 3. If you use an external clock, you do not need to set it.
Note 4. When using the internal clock (master), the setting of reception only is prohibited.

26.6.4 Serial Data Transmission in Clock Synchronous Mode

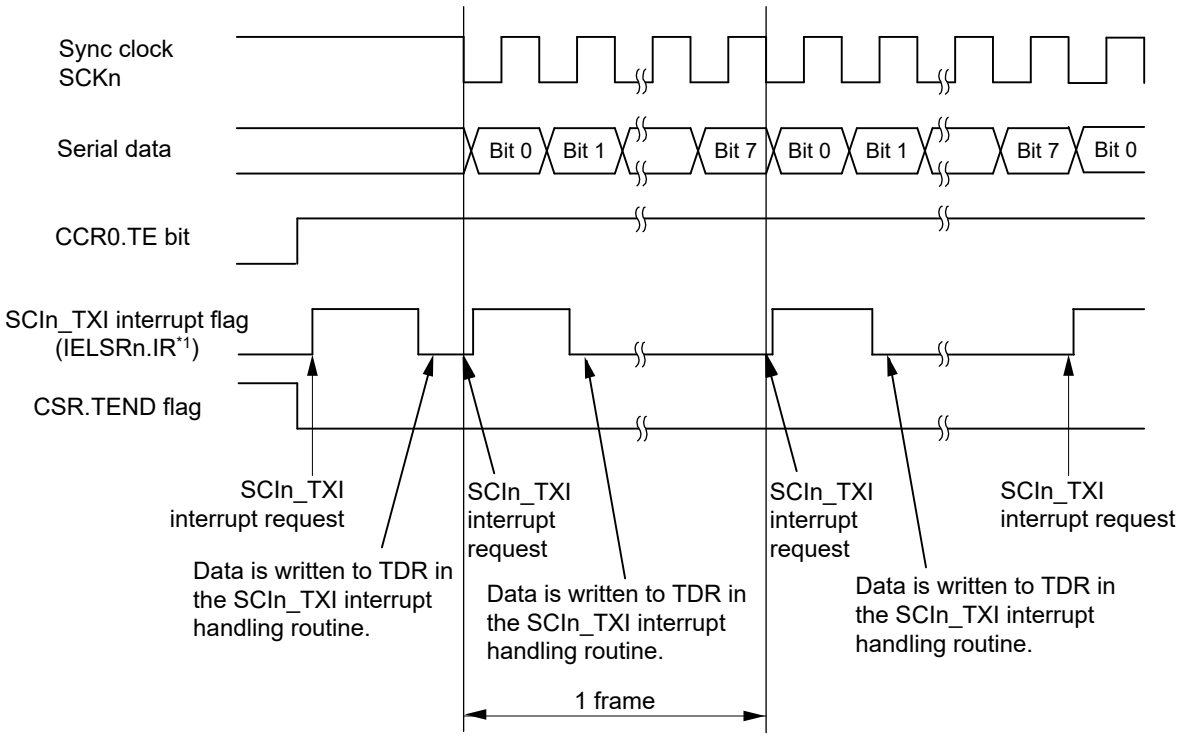
(1) Non-FIFO selected

Figure 26.60, Figure 26.61, and Figure 26.62 show examples of serial transmission in clock synchronous mode.

In serial data transmission, the SCI operates as follows:

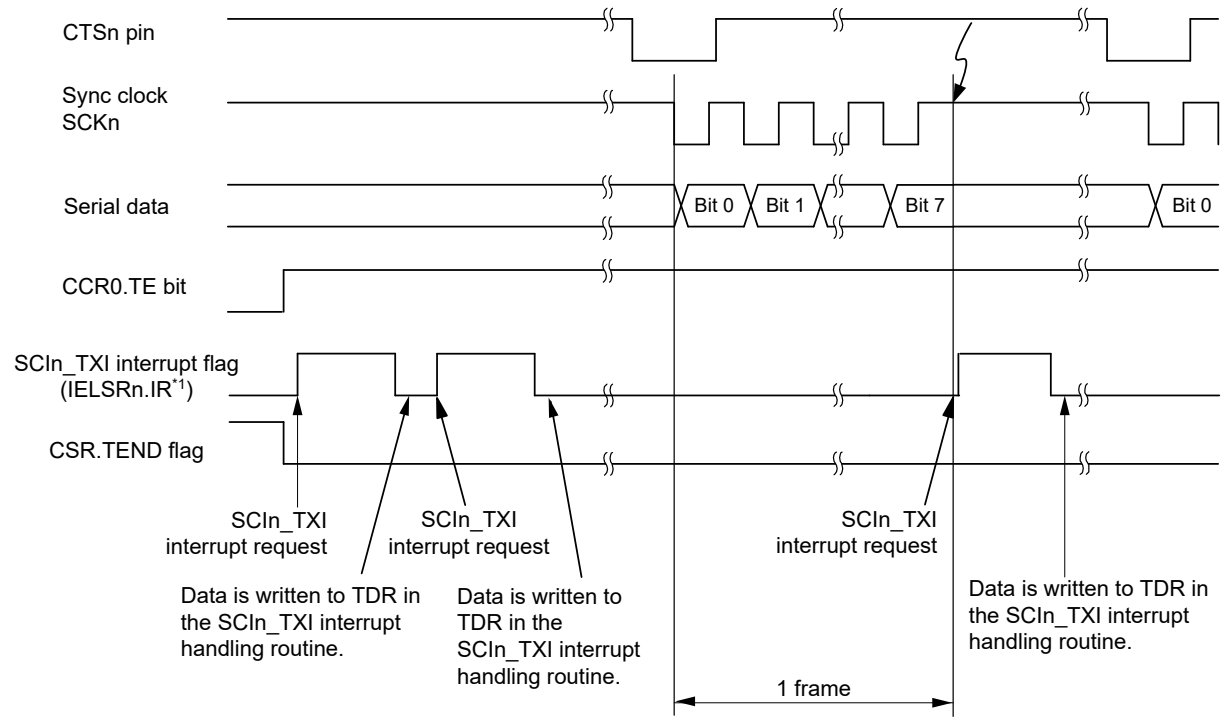
1. The SCI transfers data from the TDR register to the TSR register when data is written to TDR in the SCIn_TXI interrupt handling routine. When starting data transmission, set the CCR0.TIE bit and the CCR0.TE bit to 1 simultaneously by a single instruction. Then a TXI interrupt request is generated.
2. After transferring data from TDR to TSR, the SCI starts transmission. When the CCR0.TIE bit is set to 1, an SCIn_TXI interrupt request is generated. Continuous transmission is enabled by writing the next transmit data to TDR in the SCIn_TXI interrupt handling routine before transmission of the current transmit data finishes. When SCIn_TEI interrupt requests are in use, set the CCR0.TIE bit to 0 and the CCR0.TEIE bit to 1 after the last of the data to be transmitted is written to the TDR register from the handling routine for SCIn_TXI requests.
3. 8-bit data is sent from the TXDn pin in synchronization with the output clock when the clock output mode is specified and in synchronization with the input clock when the use of an external clock is specified. Output of the clock signal is suspended until the input CTS signal is low when the CCR1.CTSE bit is 1 (CTS function enabled).
4. The SCI checks for update to the TDR register on output of the last bit.
5. When the TDR register is updated, the next transmit data is transferred from TDR to TSR, and serial transmission of the next frame starts.
6. If TDR is not updated, the CSR.TEND flag is set to 1. The TXDn pin retains the output state of the last bit. If the CCR0.TEIE bit is 1, an SCIn_TEI interrupt request is generated and the SCKn pin is held high.

Figure 26.60, Figure 26.61, and Figure 26.62 show examples of serial data transmission.



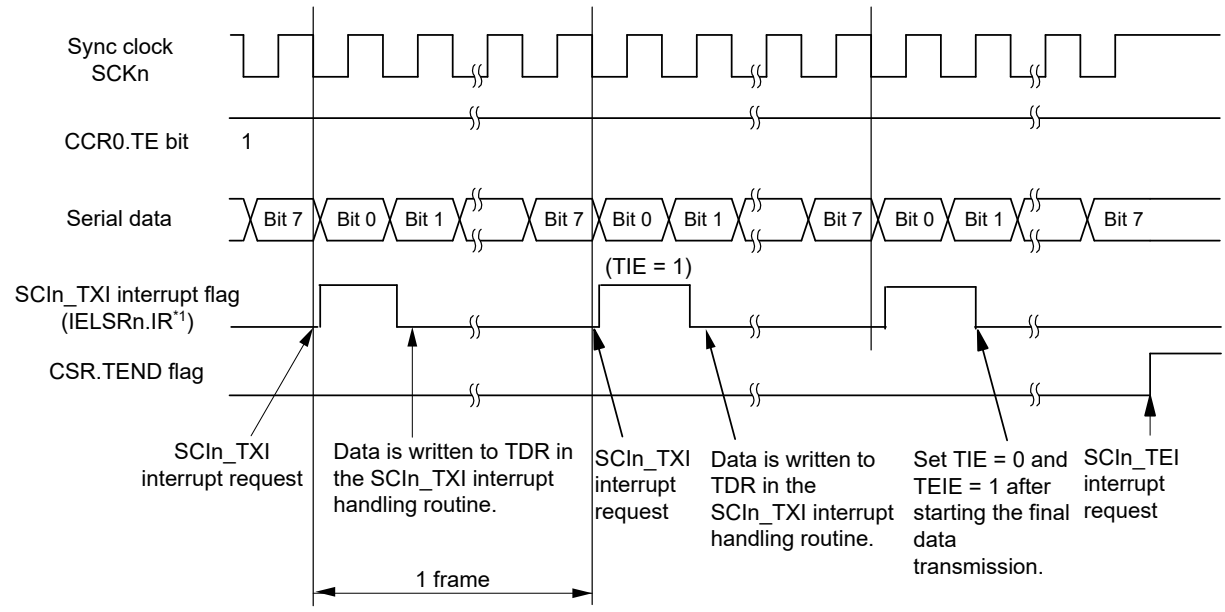
Note 1. See section 12, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 26.60 Example of serial data transmission in clock synchronous mode when the CTS function is not used at the beginning of transmission



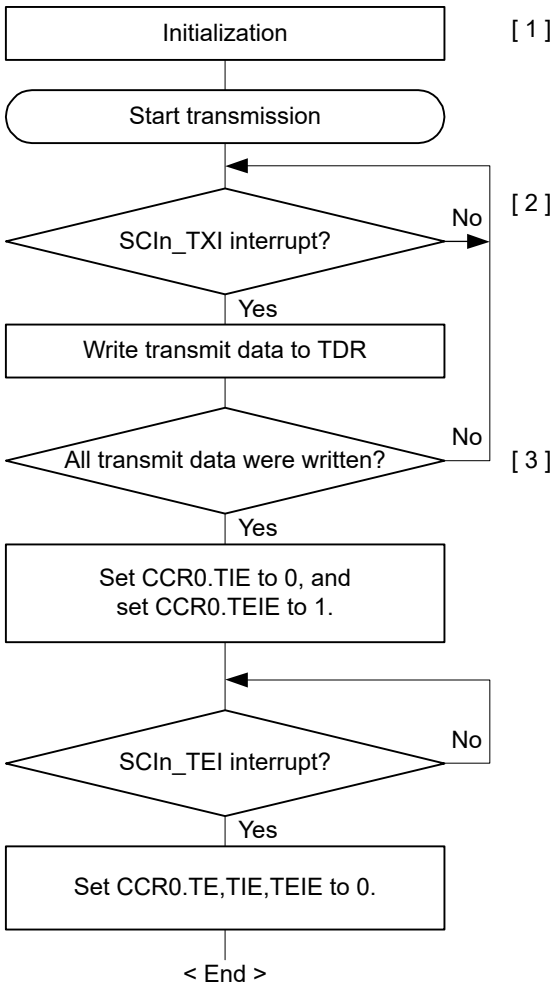
Note 1. See section 12, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 26.61 Example of serial data transmission in clock synchronous mode when the CTS function is used at the beginning of transmission



Note 1. See section 12, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 26.62 Example of serial data transmission in clock synchronous mode from the middle of transmission until transmission completion



[1] Initialize SCI.
(Set CCR0.TE and TIE to 1.)

[2] Write transmit data to TDR by using SCIn_TXI:
The transmit-data-empty-interrupt(SCIn_TXI) is Generated when transmission-data forwards to TSR from TDR. Write transmit data to TDR once by using a SCIn_TXI.

[3] Serial transmission continuation procedure:
To continue serial transmission write transmit data to TDR by using a SCIn_TXI.
Transmit data can also be written by activating the DMAC or DTC.
If you use SCIn_TEI, set CCR0.TIE = 0 and CCR0.TEIE = 1 after writing the last-transmit-data to TDR.

Note: When the external clock is in use (the value of the CCR3.CKE[1:0] is 10b or 11b), the rising edge on the SCK pin for the last bit sets the CSR.TEND flag to 1. Setting TE to 0 immediately after this, then it may lead to insufficient received data hold time on the receiver side.

Figure 26.63 Example flow of serial transmission in clock synchronous mode with non-FIFO selected

(2) FIFO selected

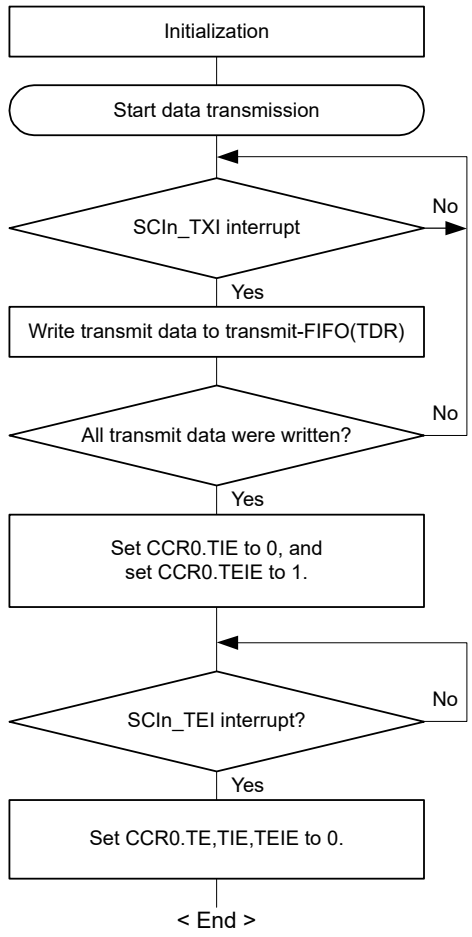
Figure 26.64 shows an example of serial transmission in clock synchronous mode with FIFO selected.

In serial data transmission, the SCI operates as follows:

1. The SCI transfers data from the transmit-FIFO (TDR register) to the TSR register when data is written to transmit-FIFO (TDR register) in the SCIn_TXI interrupt handling routine. The amount of data that can be written to transmit-FIFO (TDR register) is 16 - FTSR.T[5:0] bytes. In addition, when starting data transmission, set the CCR0.TIE bit and the CCR0.TE bit to 1 simultaneously by a single instruction. Then a SCIn_TXI interrupt request is generated.
2. After transferring data from transmit-FIFO (TDR register) to TSR, the SCI starts transmission. When the amount of transmit data written in transmit-FIFO (TDR register) is equal to or less than the specified transmit triggering number, the CSR.TDRE is set to 1. When the CCR0.TIE bit is set to 1, an SCIn_TXI interrupt request is generated. Continuous transmission is enabled by writing the next transmit data to transmit-FIFO (TDR register) in the SCIn_TXI interrupt handling routine before transmission of the current transmit data has finished. When SCIn_TEI interrupt requests are in use, set the CCR0.TIE bit to 0 and the CCR0.TEIE bit to 1 after the last of the data to be transmitted is written to the transmit-FIFO (TDR register) from the handling routine for SCIn_TXI requests.
3. 8-bit data is sent from the TXDn pin in synchronization with the output clock when the clock output mode is specified and in synchronization with the input clock when the use of an external clock is specified. Output of the clock signal is suspended until the input CTS signal is low when the CCR1.CTSE bit is 1 (CTS function enabled).
4. The SCI checks whether non-transmitted data remains in transmit-FIFO (TDR register)*1 on output of the stop bit.

5. When data is remaining in the transmit-FIFO (TDR register), the next transmit data is transferred from transmit-FIFO (TDR register) to TSR and serial transmission of the next frame starts.
6. If no data is remaining in the transmit FIFO (TDR register), the CSR.TEND flag is set to 1. The TXDn pin retains the output state of the last bit. If the CCR0.TEIE bit is 1, an SCIn_TEI interrupt request is generated and the SCKn pin is held high.

Note 1. The number of unsent transmit data stored in the TDR register (transmit FIFO) can be monitored by reading the FTSR.T[5:0] bits.



[1] [1] Initialize SCI.
(Set CCR0.TE and TIE to 1.)

[2] Write transmit data to transmit-FIFO(TDR) by using SCIn_TXI: The transmit-data-empty-interrupt(SCIn_TXI) is generated when transmission-data forwards to TSR from the transmit-FIFO(TDR) and the number of transmit-data written in the transmit-FIFO is equal to or less than the transmit-FIFO threshold.

[3] Serial transmission continuation procedure:
To continue serial transmission, write transmit data to the transmit-FIFO(TDR) by using a SCIn_TXI interrupt and clear CSR.TDRE flag to 0. Transmit data can also be written to the transmit- FIFO(TDR) by activating the DMAC or DTC. In this case, the TDRE flag is cleared automatically, so please don't operate the TDRE flag. If you use SCIn_TEI, set CCR0.TIE = 0 and CCR0.TEIE = 1 after writing the last-transmit-data to TDR.

Note: When the external clock is in use (the value of the CCR3.CKE[1:0] is 10b or 11b), the rising edge on the SCK pin for the last bit sets the CSR.TEND flag to 1. Setting CCR0.TE to 0 immediately after this, then it may lead to insufficient received data hold time on the receiver side.

Figure 26.64 Example flow of serial transmission in clock synchronous mode with FIFO selected

26.6.5 Serial Data Reception in Clock Synchronous Mode

(1) Non-FIFO selected

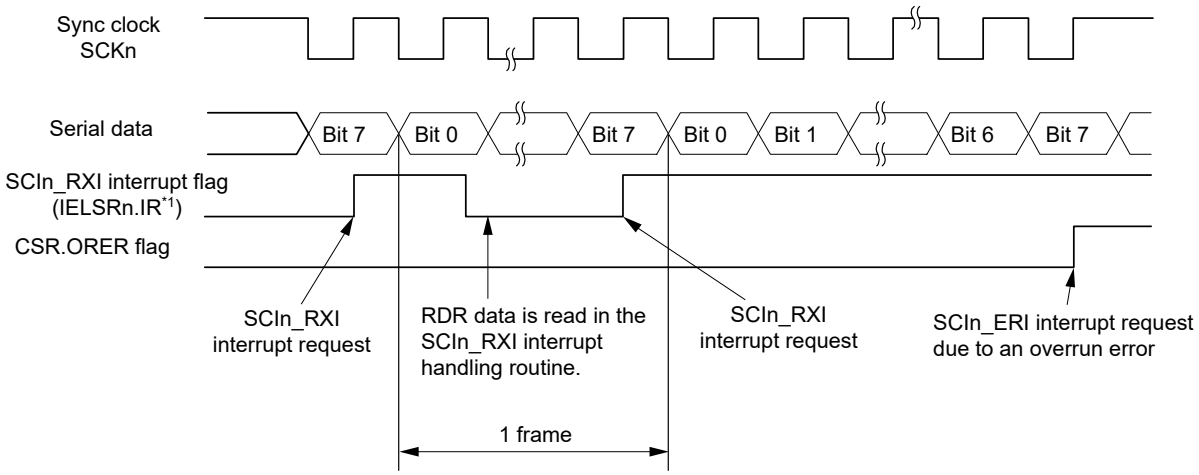
Figure 26.65 and Figure 26.66 show examples of SCI operation for serial reception in clock synchronous mode.

In serial data reception, the SCI operates as follows during serial data reception. Reception-only operation is possible only in slave mode. (In master mode, reception-only operation is prohibited.)

1. When the value of the CCR0.RE bit becomes 1, the CTSn_RTSn pin goes low (when the RTS function is used).
2. The SCI performs internal initialization and starts receiving data in synchronization with a synchronization clock input or output, and stores the receive data in the RSR register.

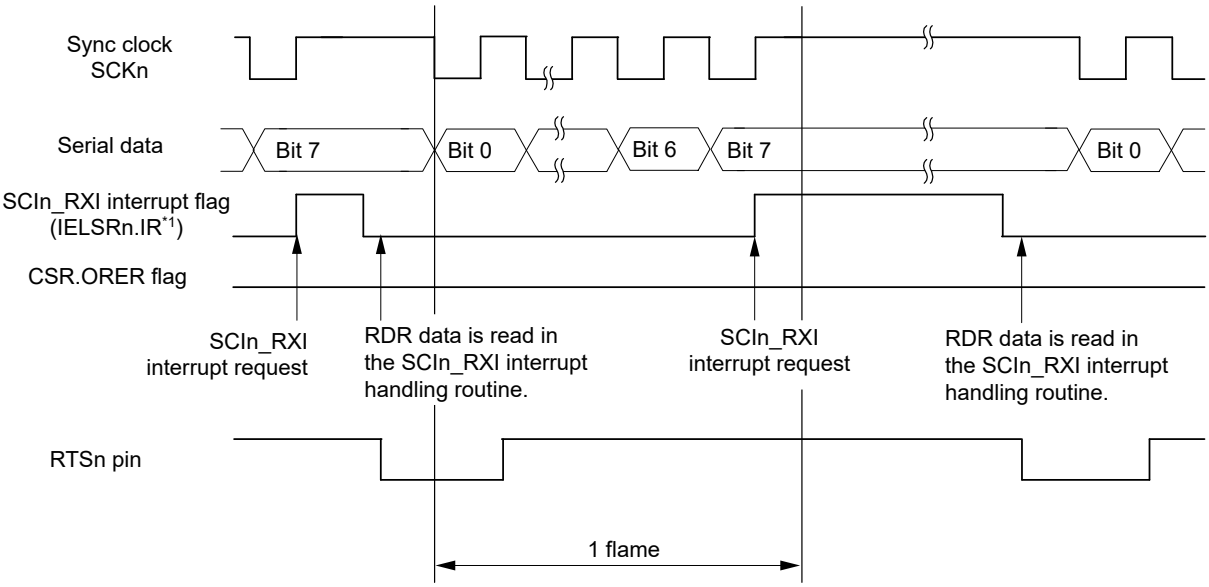
3. If an overrun error occurs, the CSR.ORER flag is set to 1. If the CCR0.RIE bit is 1, an SCIn_ERI interrupt request is generated. Receive data is not transferred to the RDR register.
4. When reception completes successfully, receive data is transferred to the RDR register. If the CCR0.RIE bit is 1, an SCIn_RXI interrupt request is generated. Continuous reception is enabled by reading the received data transferred to the RDR register in the SCIn_RXI interrupt handling routine before reception of the next receive data completes. Reading the received data that is transferred to RDR causes the CTSn_RTSn pin to output low (when the RTS function is used).

If you want to prevent the CTSn_RTSn pin output from turning low level after the final data is received, clear the CCR0.RE bit to 0 and then read the RDR register.



Note 1. See section 12, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 26.65 Example operation for serial reception in clock synchronous mode (1) when the RTS function is not used



Note 1. See section 12, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 26.66 Example operation for serial reception in clock synchronous mode (2) when RTS function is used

Data transfer cannot resume while the receive error flag is 1. Therefore, clear the ORER, FER, and PER flags in the CSR register to 0 before resuming data reception. Additionally, always read the RDR register during overrun error processing. When a data reception is forced to terminate by a 0 write to the CCR0.RE bit during operation, read the RDR register because received data that is not yet read might be left in the RDR register.

Figure 26.67 shows an example flow of serial data reception.

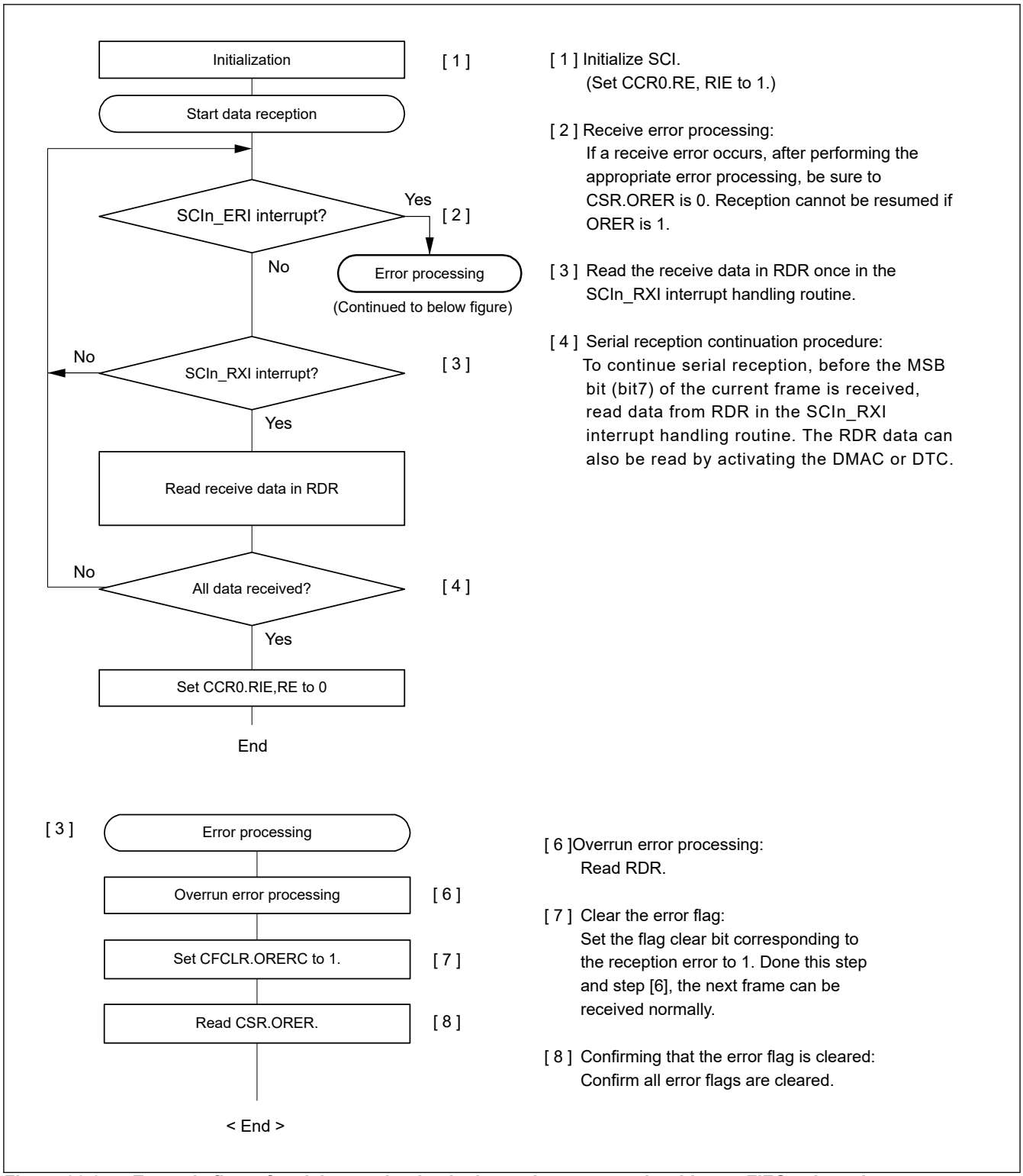


Figure 26.67 Example flow of serial reception in clock synchronous mode with non-FIFO selected

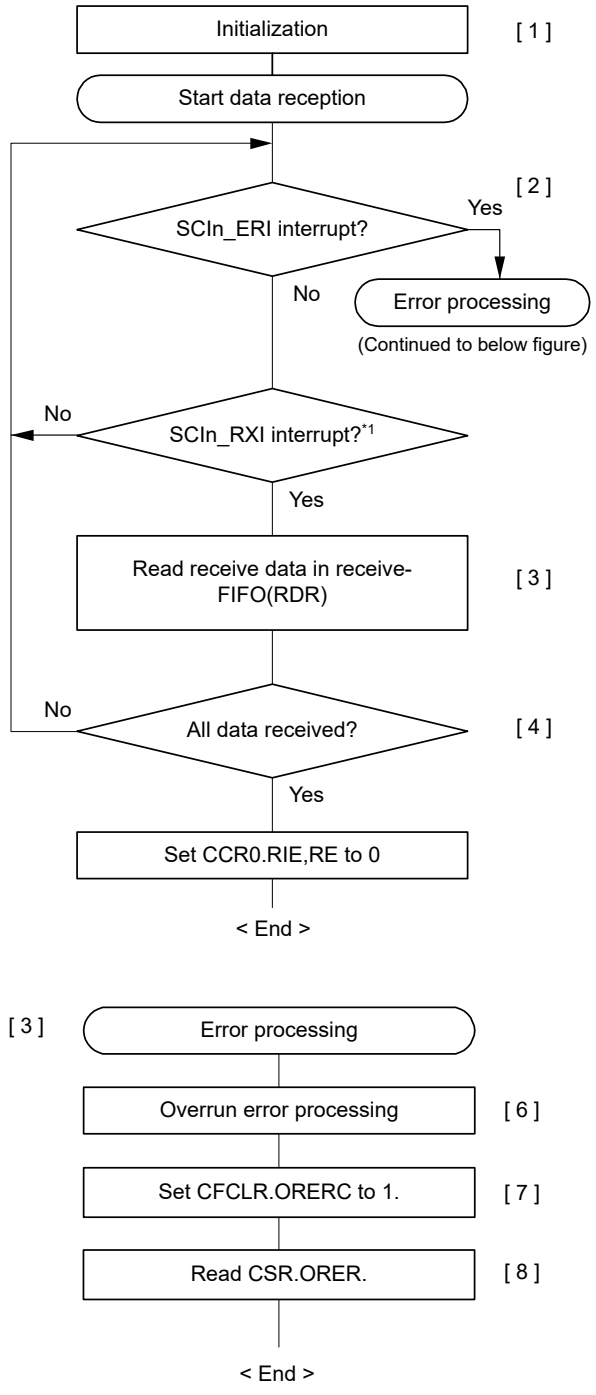
(2) FIFO selected

Figure 26.68 shows an example of serial reception in clock synchronous mode with FIFO selected.

In serial data reception, the SCI operates as follows during serial data reception. Reception-only operation is possible only in slave mode. (In master mode, reception-only operation is prohibited.)

1. When the value of the CCR0.RE bit becomes 1, the CTSn_RTSn pin goes low (when the RTS function is used).
2. The SCI performs internal initialization and starts receiving data in synchronization with a synchronization clock input or output, and transfers the received data to the receive-FIFO (RDR register).
3. If an overrun error occurs, the CSR.ORER flag is set to 1. If the CCR0.RIE bit is 1, an SCIn_ERI interrupt request is generated. Received data is not transferred to the receive-FIFO (RDR register)*1.
4. When data reception completes successfully, the receive data is transferred to the receive-FIFO (RDR register)*1. The FRSR.RDRF flag is set to 1 when the amount of the receive data stored in receive-FIFO (RDR register) is equal to or greater than the specified receive triggering number. If the CCR0.RIE bit is 1, an SCIn_RXI interrupt request is generated. Continuous data reception is enabled by reading the receive data transferred to receive-FIFO (RDR register)*1 in the SCIn_RXI interrupt handling routine before an overrun error occurs. If the amount of received data that is transferred to receive-FIFO (RDR register) is less than the specified receive triggering number, the CTSn_RTSn pin goes low (when the RTS function is used).

Note 1. In clock synchronous mode, RDR.RDAT[8] is not used.



[1] Initialize SCI.
(Set CCR0.RE, RIE to 1.)

[2] Receive error processing:
If a receive error occurs, after performing the appropriate error processing, be sure to CSR.ORER is 0. Reception cannot be resumed if ORER is 1.

[3] Read the receive data in the receive-FIFO(RDR) in the SCIn_RXI interrupt handling routine. The reception data stored in the receive-FIFO(RDR) is read until the number of stored data will be below the FCR.RTRG value at least. You can confirm the number of the reception data in the FIFO by FRSR.R[5:0].

[4] Serial reception continuation procedure:
To continue serial reception, before the overrun error occurs, read data from RDR. The RDR data can also be read by activating the DMAC or DTC.

Note1:
Set all received data to an integer multiple of the receive FIFO threshold. If it is not an integral multiple, return to step [2] and it will become an infinite loop, so set a timeout time separately using a timer etc.

[6]Overrun error processing:
Read RDR.

[7] Clear the error flag:
Set the flag clear bit corresponding to the reception error to 1. Done this step and step [6], the next frame can be received normally.

[8] Confirming that the error flag is cleared:
Confirm all error flags are cleared.

Figure 26.68 Example flow of serial reception in clock synchronous mode with FIFO selected

26.6.6 Simultaneous Serial Data Transmission and Reception in Clock Synchronous Mode

(1) Non-FIFO selected

Figure 26.69 shows an example flow of simultaneous serial transmit and receive operations in clock synchronous mode. After initializing the SCI, use the following procedure for simultaneous serial data transmit and receive operations.

To switch from transmit mode to simultaneous transmit and receive mode:

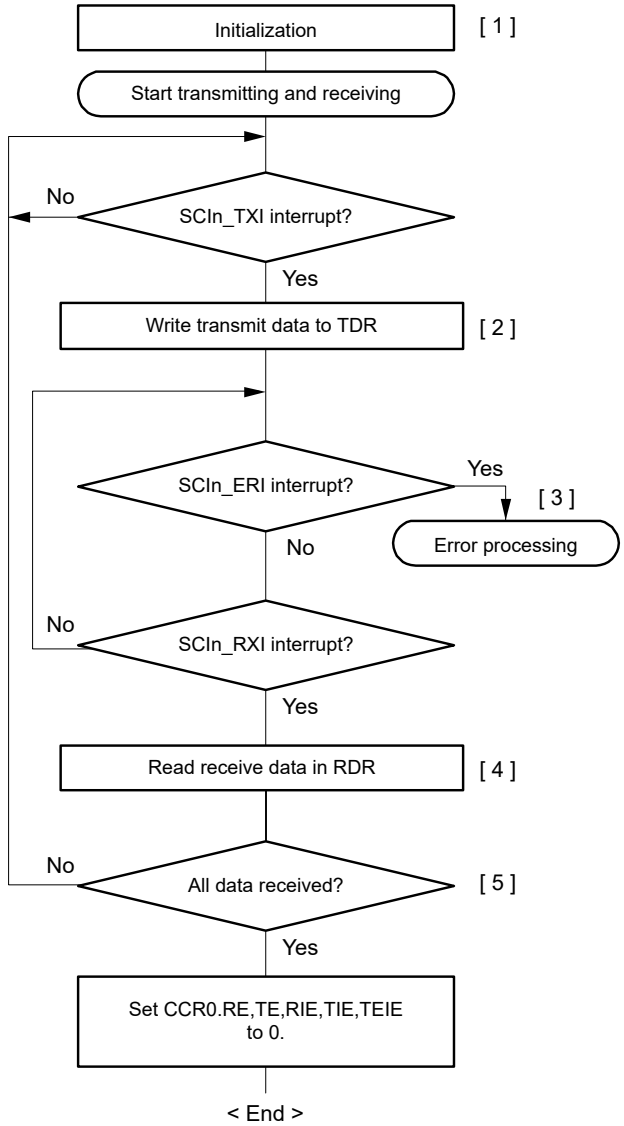
1. Check that the SCI completes the data transmission by verifying that the CSR.TEND flag is set to 1.

2. Initialize the CCR0 register, and then set the TIE, RIE, TE, and RE bits in the CCR0 register to 1 simultaneously by a single instruction.

To switch from receive mode to simultaneous transmit and receive mode:

1. Check that the SCI completes the data reception.
2. Set the CCR0.TE and RE bits to 0, and then check that the receive error flag (ORER, FER, and PER) in the CSR register is 0.
3. Set the TIE, RIE, TE, and RE bits in the CCR0 register to 1 simultaneously by a single instruction.

When the RTS function is used in the concurrent transmission/reception operation, if you want to prevent the CTSn_RTSn pin output from turning to low after the final data is received as in the reception operation, clear the RE and TE bits in CCR0 to 0 simultaneously, and then read the RDR register.



[1] Initialize SCI:
Make the TXDn pin can act as the output pin for transmitted data and the RXDn pin can act as the input pin for received data at the same time.

[2] Transmit data write:
Write transmit data to TDR in the SCIn_TXI Interrupt request handling routine.

[3] Receive error processing:
If a receive error occurs, after performing the appropriate error processing, be sure to CSR.ORER is 0. Reception cannot be resumed if ORER is 1.

[4] Reading receive data:
The reception data stored in RDR once in the SCIn_RXI interrupt request handling routine.

[5] Serial transmission/reception continuation procedure
To continue serial reception, before the MSB bit (bit7) of the current frame is received, read data from RDR in the SCIn_RXI interrupt handling routine. And before the MSB(bit7) of the current frame is transmitted, write transmit data to TDR in the SCIn_TXI interrupt handling. Transmit data can also be written by activating the DMAC or DTC. And the RDR data can also be read by activating the DMAC or DTC.

Figure 26.69 Example flow of simultaneous serial transmission and reception in clock synchronous mode with non-FIFO selected

(2) FIFO selected

Figure 26.70 shows an example flow of simultaneous serial transmit and receive operations in clock synchronous mode with FIFO selected.

After initializing the SCI, use the following procedure for simultaneous serial data transmit and receive operations.

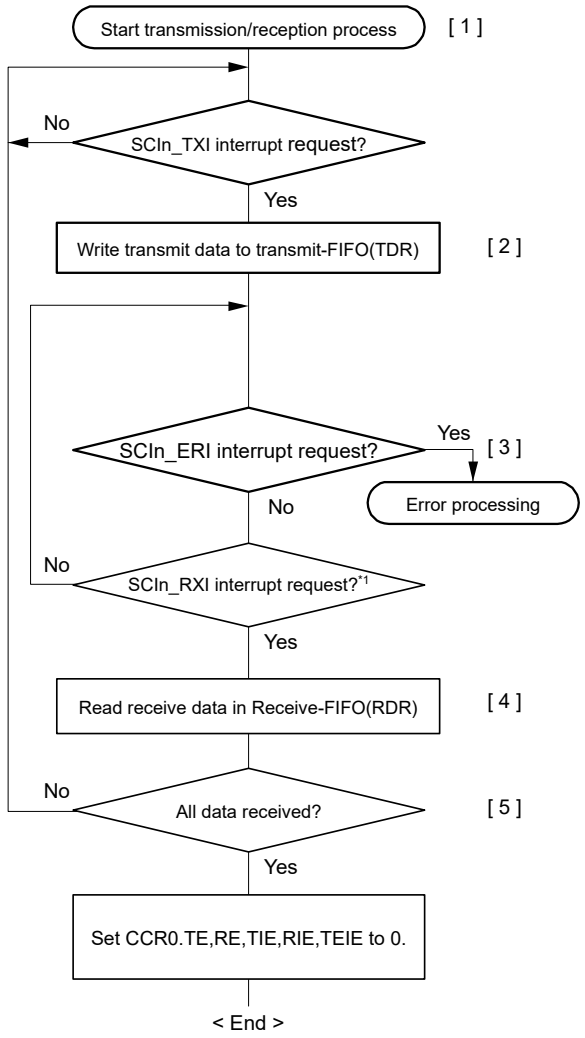
To switch from transmit mode to simultaneous transmit and receive mode:

1. Check that the SCI completes the transmission by verifying that the CSR.TEND flag is set to 1.
2. Initialize the CCR0 register, then set the TIE, RIE, TE, and RE bits in the CCR0 register to 1 simultaneously by a single instruction.

To switch from receive mode to simultaneous transmit and receive mode:

1. Check that the SCI completes the reception.
2. Set the CCR0.TE and RE bits to 0.
3. Check that the receive error flags (ORER, FER, and PER) in the CSR register are 0, and then set the TIE, RIE, TE, and RE bits in the CCR0 register to 1 simultaneously by a single instruction.

Since clock synchronous communication performs transmission and reception at the same time, make sure that the number of data to be transmitted and received is the same.



[1] Start transmission / reception process

[2] Transmit data write:
Write transmit data to Transmit-FIFO(TDR) in the SCIn_TXI Interrupt request handling routine.
Match the number of transmitted data with the FCR.RTRG.

[3] Receive error processing:
If a receive error occurs, after performing the appropriate error processing (Initialization such as stopping communication, reading all received data, clearing CSR.RDRF, etc.), be sure to ORER is 0. Reception cannot be resumed if ORER is 1.

[4] Read receive data:
Read the received data stored in receive-FIFO(RDR) for the number of the FCR.RTRG in the SCIn_RXI interrupt handling routine.

[5] Serial transmission/reception continuation procedure
To continue serial reception, before the overrun error occurs, read data from RDR and clear CSR.RDRF.
To continue serial transmission, write transmit data to the transmit-FIFO(TDR) by using a SCIn_TXI interrupt and clear TDRE flag to 0.
Transmit data can also be written to the transmit- FIFO(TDR) by activating the DMAC or DTC. And The RDR data can also be read by activating the DMAC or DTC. In this case, the TDRE and RDRF are cleared automatically, so please don't operate the TDRE And RDRF.

Note 1. Set all received data to an integer multiple of the receive FIFO threshold. If it is not an integral multiple, return to step [2] and it will become an infinite loop, so set a timeout time separately using a timer etc.

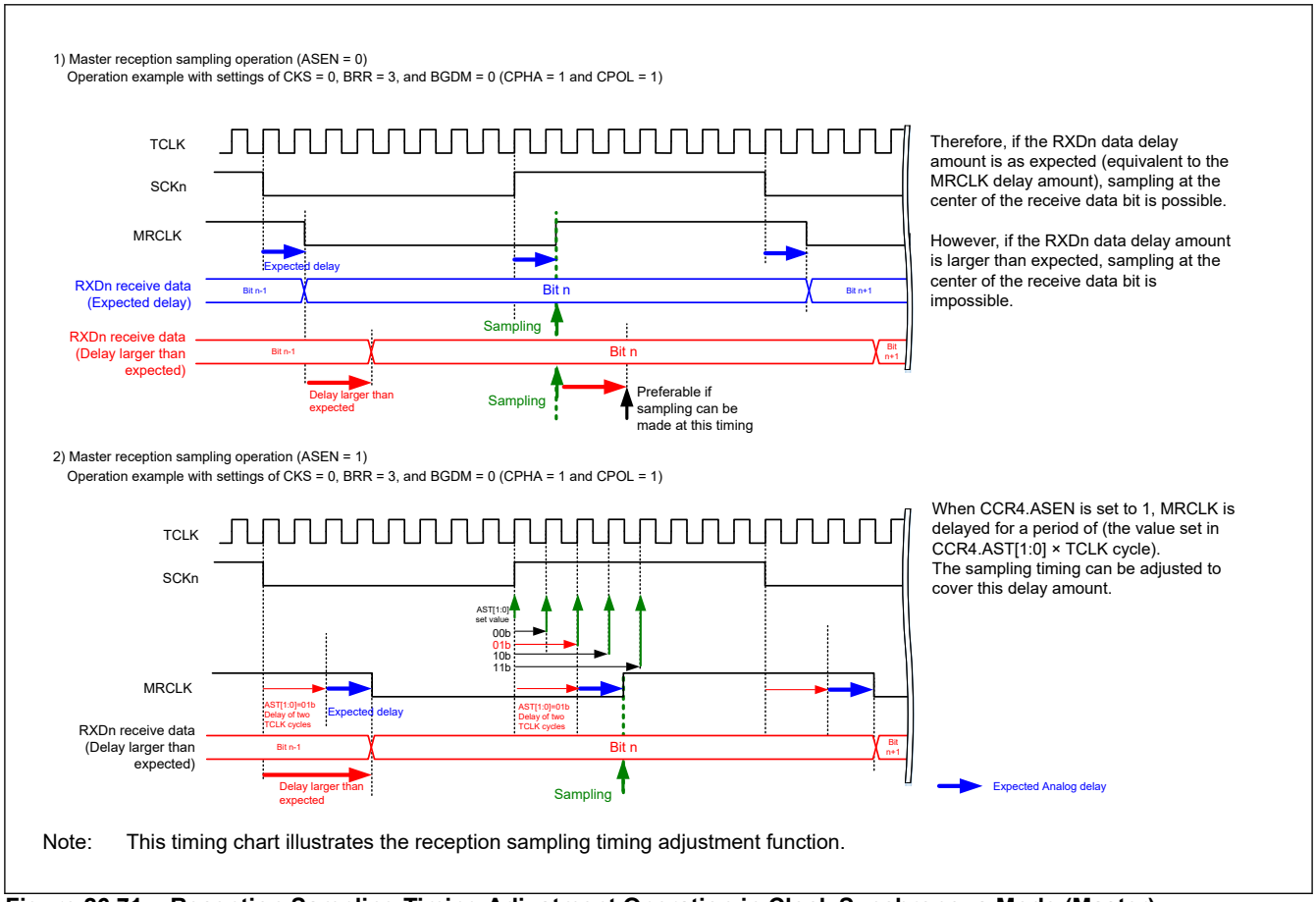
Figure 26.70 Example flow of simultaneous serial transmission and reception in clock synchronous mode with FIFO selected

26.6.7 Reception Sampling Timing Adjustment Function in Clock Synchronous Mode with internal clock used

When the clock synchronous internal clock is used (master mode), MRCLK is used as a reception sampling clock.

This function adjusts the reception sampling timing by delaying MRCLK by 1 to 4 TCLK and adding a digital delay. MRCLK's analog delay cannot be adjusted by this function.

Setting the CCR4.ASEN bit to 1 enables this function. The delay value is set in CCR4.AST[1:0].



Note: This timing chart illustrates the reception sampling timing adjustment function.

Figure 26.71 Reception Sampling Timing Adjustment Operation in Clock Synchronous Mode (Master)

26.7 Operation in Smart Card Interface Mode

The SCI supports smart card (IC card) interfaces conforming to ISO/IEC 7816-3 (standard for Identification Cards), as an extended function of the SCI.

Smart card interface mode can be selected using the appropriate register.

26.7.1 Example Connection

Figure 26.72 shows an example connection between a smart card (IC card) and the MCU. As shown in Figure 26.72, because the MCU communicates with an IC card using a single transmission line, interconnect the TXDn and RXDn pins and pull up the data transmission line to VCC using a resistor.

Setting the CCR0.TE and CCR0.RE bits to 1 with an IC card disconnected enables closed-loop transmission or reception, allowing self-diagnosis. To supply an IC card with the clock pulses generated by the SCI, input the SCKn pin output to the CLK pin of an IC card.

An output port of the MCU can be used to output a reset signal.

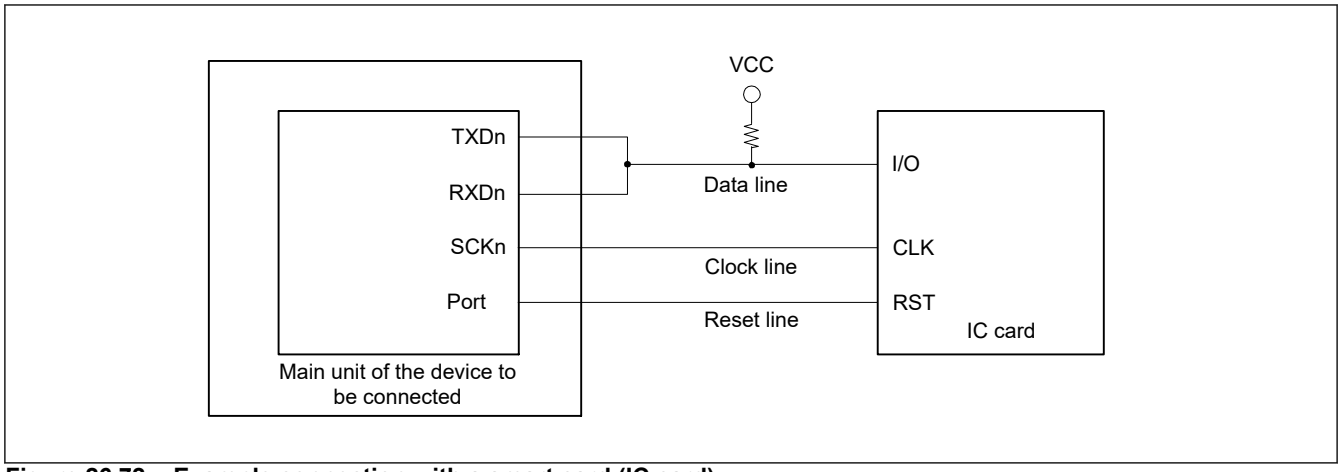


Figure 26.72 Example connection with a smart card (IC card)

26.7.2 Data Format (Except in Block Transfer Mode)

Figure 26.73 shows the data transfer formats in smart card interface mode:

- One frame consists of 8-bit data and a parity bit in asynchronous mode.
- During transmission, at least 2 ETUs (elementary time unit – the time required for transferring 1 bit) is set as a guard time from the end of the parity bit until the start of the next frame.
- If a parity error is detected during reception, a low error signal is output for 1 ETU after 10.5 ETUs elapse from the start bit.
- If an error signal is sampled during transmission, the same data is automatically retransmitted after at least 2 ETUs.

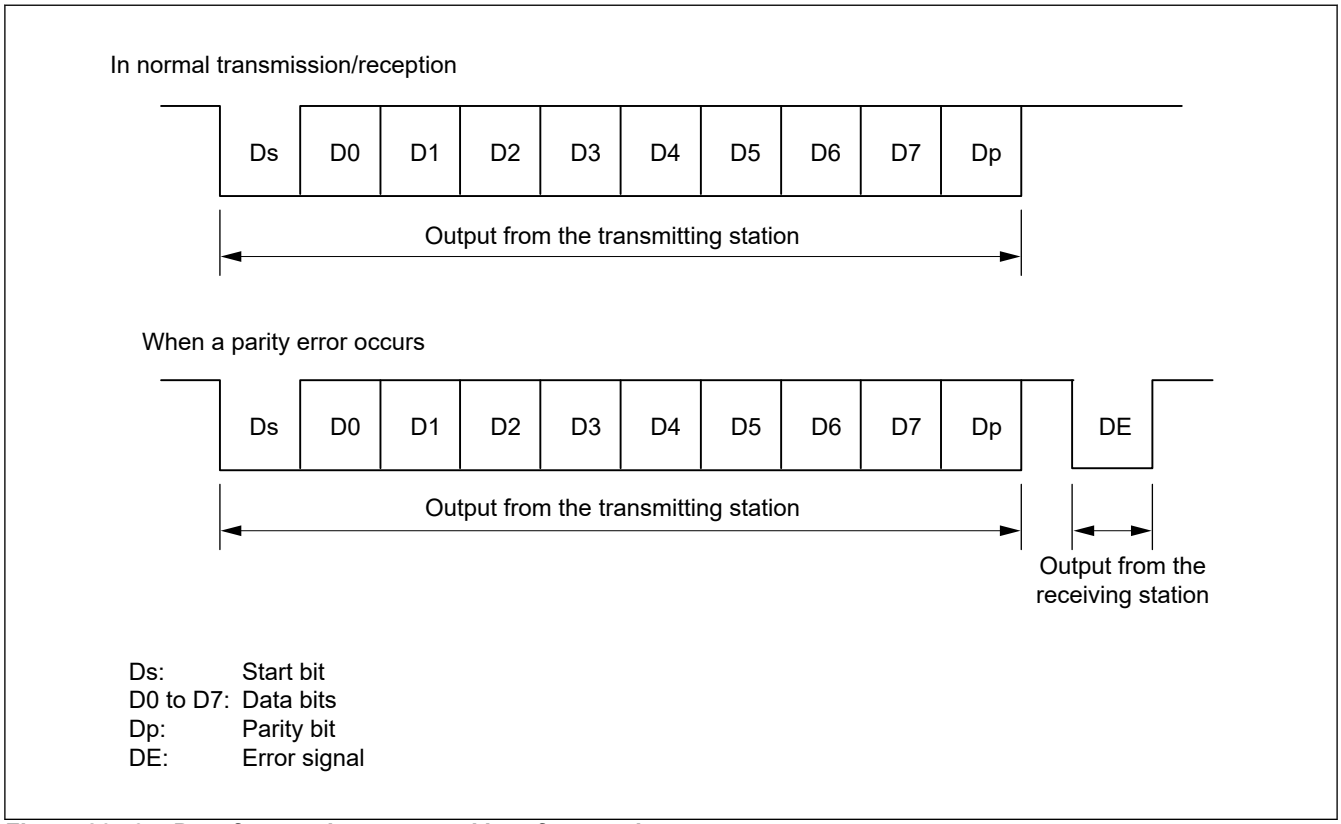


Figure 26.73 Data formats in smart card interface mode

For communications with IC cards of the direct convention type and inverse convention type, follow the procedures in this section.

(1) Direct Convention Type

For the direct convention type, logic levels 1 and 0 indicate the Z and A states, respectively, and data is transferred with LSB-first for the start character, as shown in Figure 26.74. Therefore, data in the start character in the figure is 0x3B.

When using the direct convention type, write 1 to the CCR3.LSBF and write 0 to the CCR3.SINV. Write 0 to the CCR1.PM bit to use even parity, which is prescribed by the smart card standard.

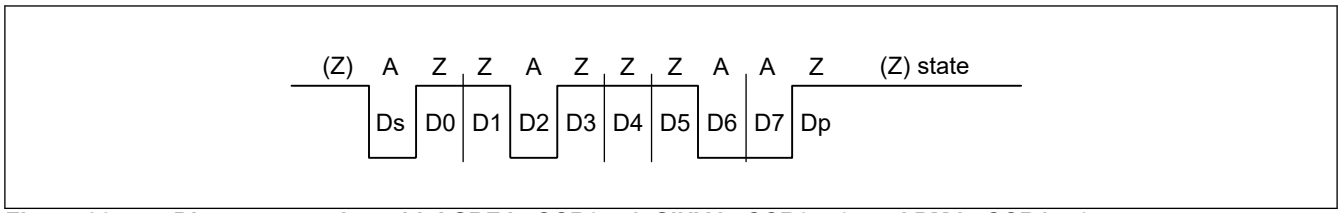


Figure 26.74 Direct convention with LSBF in CCR3 = 1, SINV in CCR3 = 0, and PM in CCR1 = 0

(2) Inverse Convention Type

For the inverse convention type, logic levels 1 and 0 indicate the A and Z states, respectively, and data is transferred with MSB-first for the start character, as shown in Figure 26.75. Therefore, data in the start character in the figure is 0x3F.

When using the inverse convention type, write 0 to the CCR3.LSBF and write 1 to the CCR3.SINV. The parity bit is logic level 0 to produce even parity, which is prescribed by the smart card standard, and corresponds to the Z state. Because the SINV bit of the MCU only inverts data bits D7 to D0, write 1 to the PM bit in CCR1 to invert the parity bit for both transmission and reception.

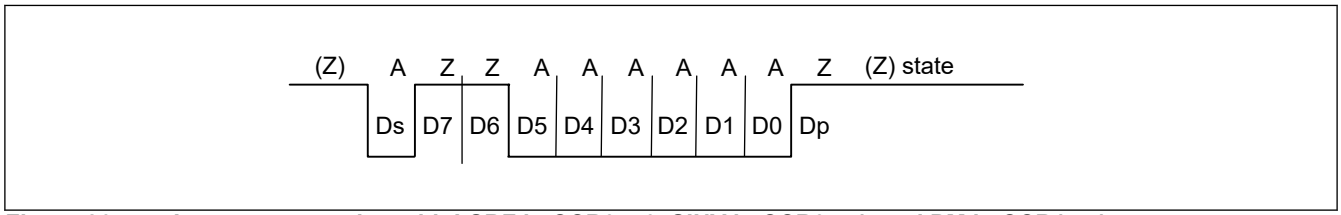


Figure 26.75 Inverse convention with LSBF in CCR3 = 0, SINV in CCR3 = 1, and PM in CCR1 = 1

26.7.3 Block Transfer Mode

Block transfer mode differs from normal smart card interface mode as follows:

- Even if a parity error is detected during reception, no error signal is output. Because the PER flag in CSR is set by error detection, clear the PER flag before receiving the parity bit of the next frame.
- During transmission, at least 1 etu is set as a guard time from the end of the parity bit until the start of the next frame
- Because the same data is not retransmitted, the TEND flag in CSR is set to 11.5 etus after transmission starts
- In block transfer mode, the ERS flag in CSR indicates the error signal status as in normal smart card interface mode, but the flag is read as 0 because no error signal is transferred

26.7.4 Receive Data Sampling Timing and Reception Margin

Only the internal clock generated by the on-chip baud rate generator can be used as a transfer clock in smart card interface mode.

In this mode, the SCI can operate on a base clock with a frequency of 32, 64, 372, 256, 93, 128, 186, or 512 times the bit rate set up in the CCR2.BCP[2:0] bits. The frequency is always 16 times the bit rate in normal asynchronous mode.

For data reception, the falling edge of the start bit is sampled with the base clock to perform internal synchronization.

Receive data is sampled on the 16th, 32nd, 186th, 128th, 46th, 64th, 93rd, and 256th rising edges of the base clock so that it can be latched at the middle of each bit as shown in Figure 26.76. The reception margin is determined by the following formula:

$$M = \left|\left(0.5 - \frac{1}{2N}\right) - (L - 0.5)F - \frac{|D - 0.5|}{N}(1 + F)\right| \times 100\ [\%]$$

M: Reception margin (%)

N: Ratio of bit rate to clock (N = 32, 64, 372, 256)

D: Duty cycle of clock (D = 0 to 1.0)

L: Frame length (L = 10)

F: Absolute value of clock frequency deviation

Assuming values of F = 0, D = 0.5, and N = 372 in the specified formula, the reception margin is determined using the following formula:

$$M = \{0.5 - 1/(2 \times 372)\} \times 100\ [\%] = 49.866\ \%$$

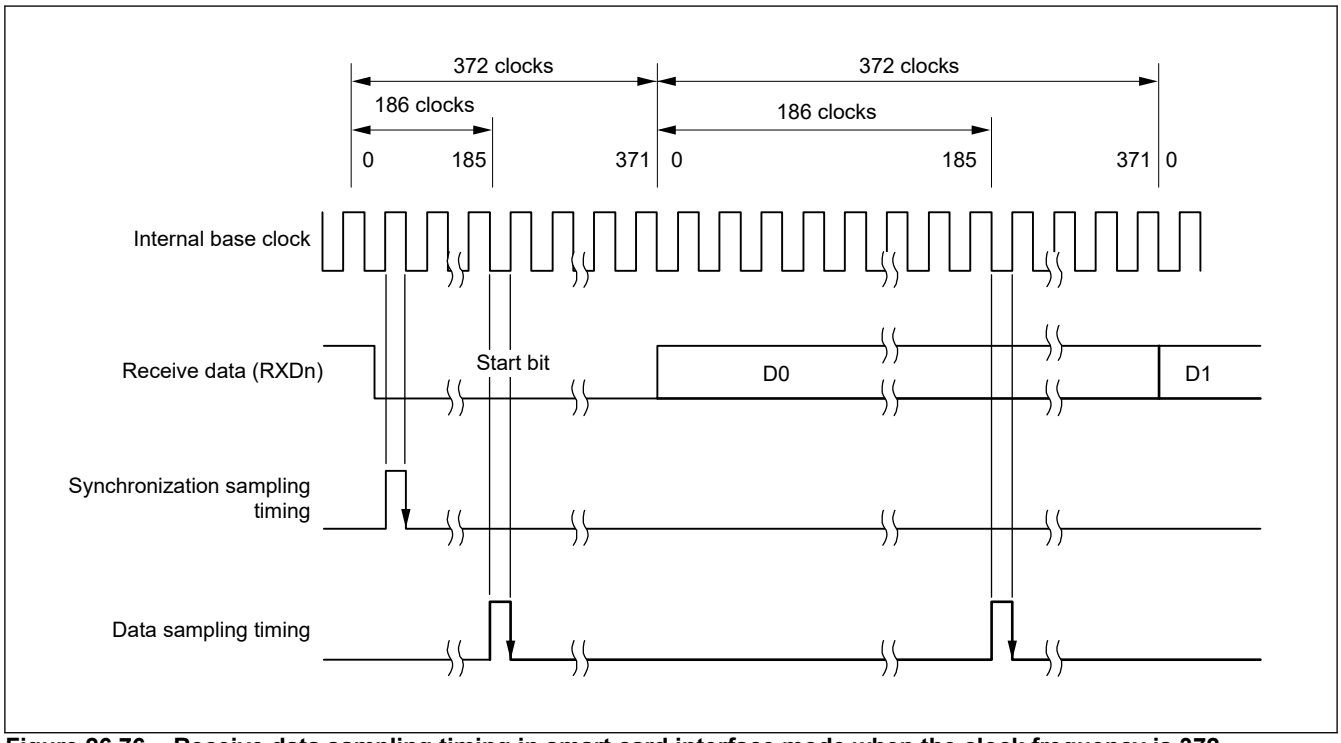


Figure 26.76 Receive data sampling timing in smart card interface mode when the clock frequency is 372 times the bit rate

26.7.5 SCI Initialization (Smart Card Interface Mode)

Before transmitting and receiving data, write the initial value 0x00 in the CCR0 register and initialize the SCI following the example flow shown in Table 26.37.

Always set the initial value in the TIE, RIE, TE, RE, TEIE bits in the CCR0 register before switching from transmission to reception mode or from reception to transmission mode. When CCR0.RE is set to 0, the RDR register is not initialized.

In transmission mode, set 1 to the CCR0.TE bit and CCR0.TIE bit simultaneously, then the SCIn_TXI interrupt request is generated.

To change from reception mode to transmission mode, first check that reception has completed, then initialize the SCI. At the end of initialization, set CCR0.TE = 1 and CCR0.RE = 0. Reception completion can be verified by reading the SCIn_RXI request, ORER, or PER flag in CSR.

To change transmission mode to reception mode, first check that transmission has completed, then initialize the SCI. At the end of initialization, set CCR0.TE = 0 and CCR0.RE = 1. Transmission completion can be verified by reading the TEND flag in CSR.

Table 26.37 Example flow of SCI initialization in smart card interface mode

No.	Step Name	Description
1	Start initialization	—
2	Set CCR0	Set CCR0.TEIE, TIE, RIE, TE, RE to 0. If you have not changed from the initial settings, you can skip this step.
3	Set CCR3	Set communication mode (MOD[2:0] = 001b), BLK, GM, and SINV. Leave other bits at their initial values.
4	Set CCR2	Set clock-select and bit-rate. Set BRME to 0.
5	Set CCR1	Set up the loop-back function, communication pin status. Set NFEN, PE, CTSE to 0 and set PE to 1.
6	Set the I/O port functions	Make I/O port settings to enable input and output functions as required for TXDn, RXDn, and SCKn pins.
7	Set CCR3	Set CKE[1:0]. When the CKE[0] bit is set to “1” due on GM setting value, the clock is output from the SCKn pin.
8	Set CFCLR	Write 1 to the following bits and clear the corresponding flag. CFCLR.RDRFC,FERC,PERC,MFFC,ORERC,DFERC,DPERC,DCMFC,ERSC
9	Set CCR0	Set the TE or RE to 1. And set the TIE and RIE. Do not simultaneously set the TE and RE bits to 1 if self-diagnosis in not used.
10	Initialization completed	—

Figure 26.77 is a timing chart when data transmission is performed by making transition to the Smart Card Interface mode according to the above flow chart. The figure shows the case when CCR3.GM bit is 0. As shown in the figure, when the pin function is set to the SCKn pin, the SCKn pin is high impedance because the CCR3.CKE [0] bit is 0. When the TXDn pin is set, the TXDn pin is high impedance because the CCR0.TE bit is 0. Start clock output to the SCKn pin with the clock output setting CCR3.CKE [0] to 1, start data transmission by writing transmit data after setting CCR0.TE to 1.

In the smart card interface mode, even if not communicating at CCR0.TE = 0 and CCR0.RE = 0, the clock is continuously output if the clock output setting is used.

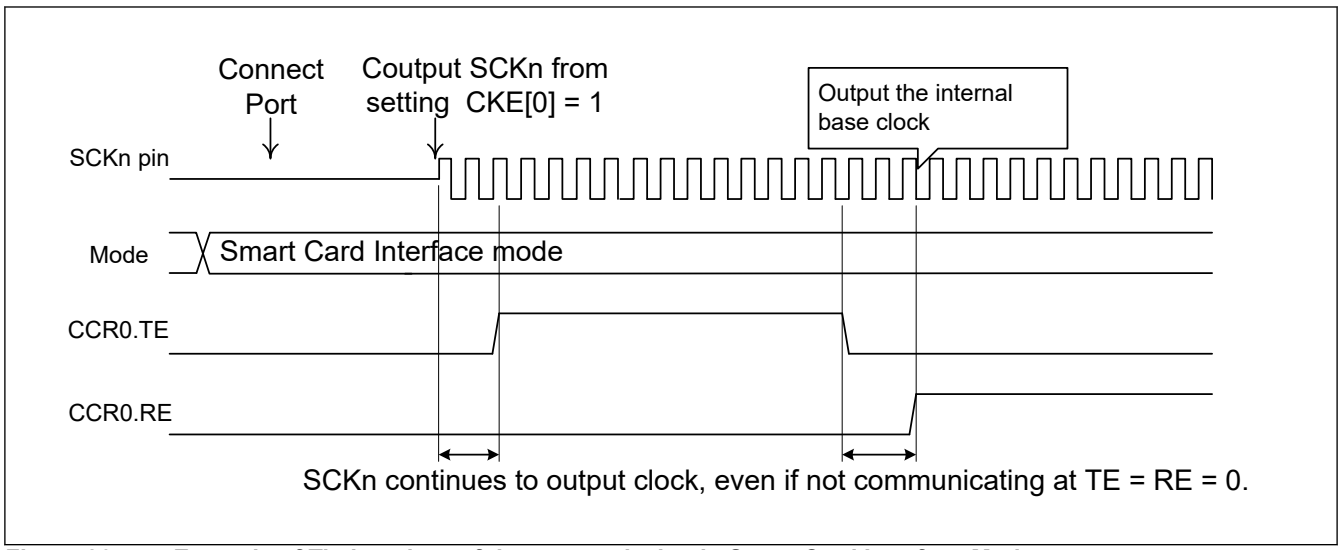


Figure 26.77 Example of Timing chart of data transmission in Smart Card Interface Mode

26.7.6 Serial Data Transmission (Except in Block Transfer Mode)

Serial data transmission in smart card interface mode (except in block transfer mode) is different from that in non-smart card interface mode, in that an error signal is sampled and data can be re-transmitted in smart card mode. Figure 26.78 shows the data re-transfer operation during transmission.

1. When an error signal from the receiver end is sampled after 1-frame data is transmitted, the CSR.ERS flag is set to 1. If the CCR0.RIE bit is 1, an SCIn_ERI interrupt request is generated. Clear the ERS flag to 0 before the next parity bit is sampled.

2. For a frame in which an error signal is received, the CSR.TEND flag is not set. Data is re-transferred from TDR to TSR, allowing automatic data retransmission.
3. If no error signal is returned from the receiver, the ERS flag is not set to 1.
4. In this case, the SCI determines that transmission of 1-frame data, including the re-transfer, is complete, and the TEND flag is set. If the CCR0.TIE bit is 1, an SCIn_TXI interrupt request is generated. Write transmit data to the TDR to start transmission of the next data.

Figure 26.80 shows an example flow of serial transmission. All the processing steps are automatically performed using an SCIn_TXI interrupt request to activate the DTC or DMAC.

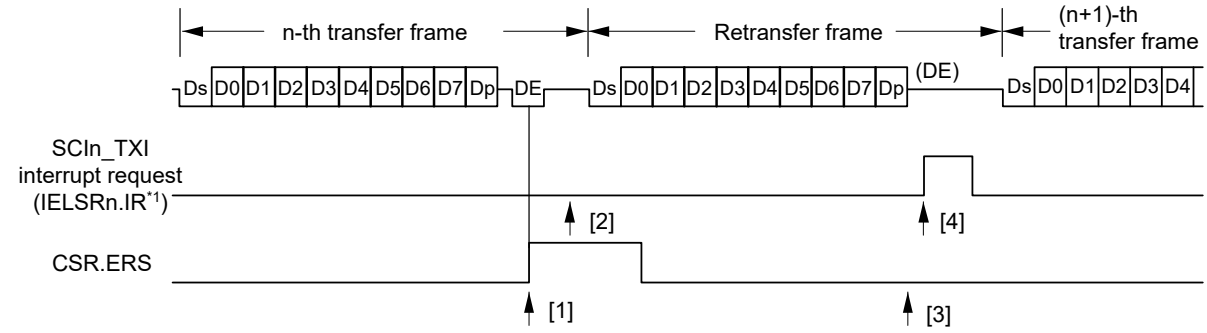
When the CSR.TEND flag is set to 1 in transmission and when the CCR0.TIE bit is 1, an SCIn_TXI interrupt request is generated.

The DTC or DMAC is activated by an SCIn_TXI interrupt request if the SCIn_TXI interrupt request is previously specified as a source of DTC or DMAC activation, allowing the transfer of transmit data. The TEND flag is automatically set to 0 when the DTC or DMAC transfers the data.

If an error occurs, the SCI automatically retransmits the same data. During this retransmission, the TEND flag is kept at 0 and the DTC or DMAC is not activated. Therefore, the SCI and DTC or DMAC automatically transmit the specified number of bytes, including retransmission when an error occurs. Because the ERS flag is not automatically cleared, set the RIE bit to 1 before enabling an SCIn_ERI interrupt request to be generated if an error occurs, and clear the ERS flag to 0.

When transmitting or receiving data using the DTC or DMAC, always enable the DTC or DMAC before making the SCI settings.

For DTC or DMAC settings, see section 16, Data Transfer Controller (DTC), section 15, DMA Controller (DMAC).



Note 1. See section 12, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 26.78 Data re-transfer operation in smart card interface transmission mode

The CSR.TEND flag is set at different timings depending on the CCR3.GM bit setting. Figure 26.79 shows the TEND flag generation timing.

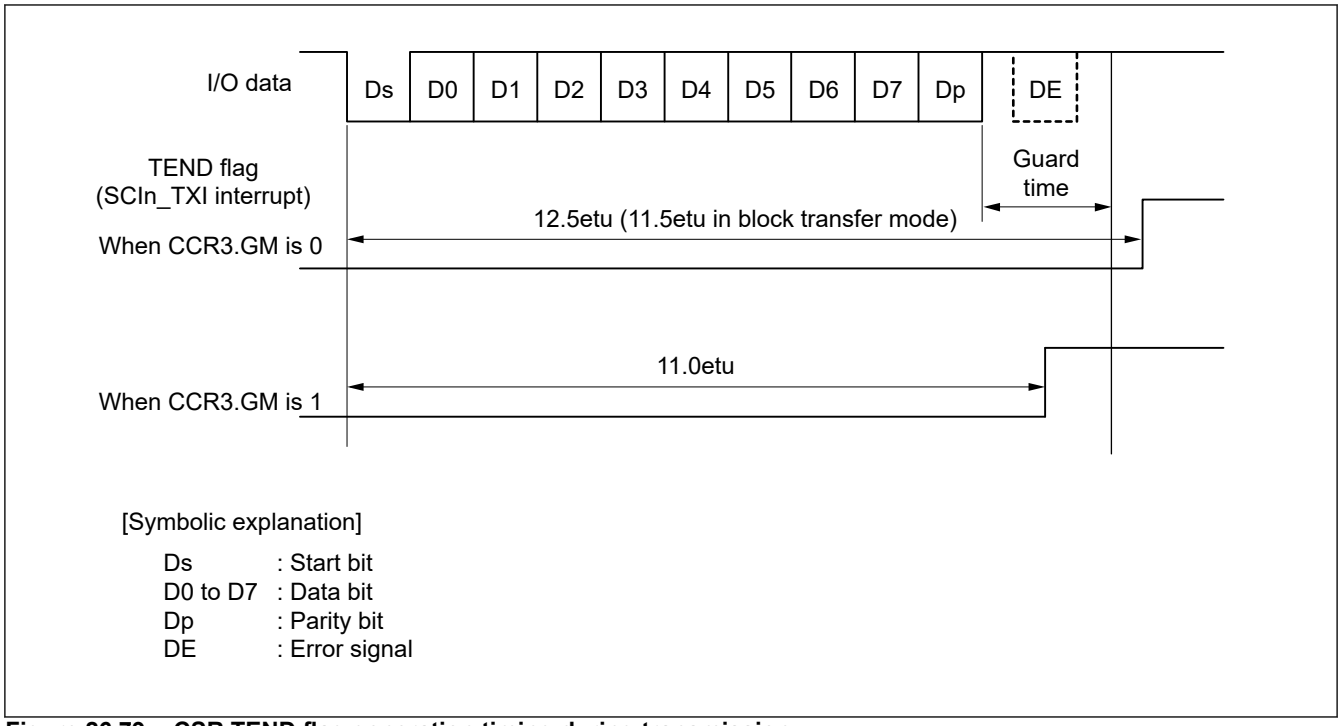


Figure 26.79 CSR.TEND flag generation timing during transmission

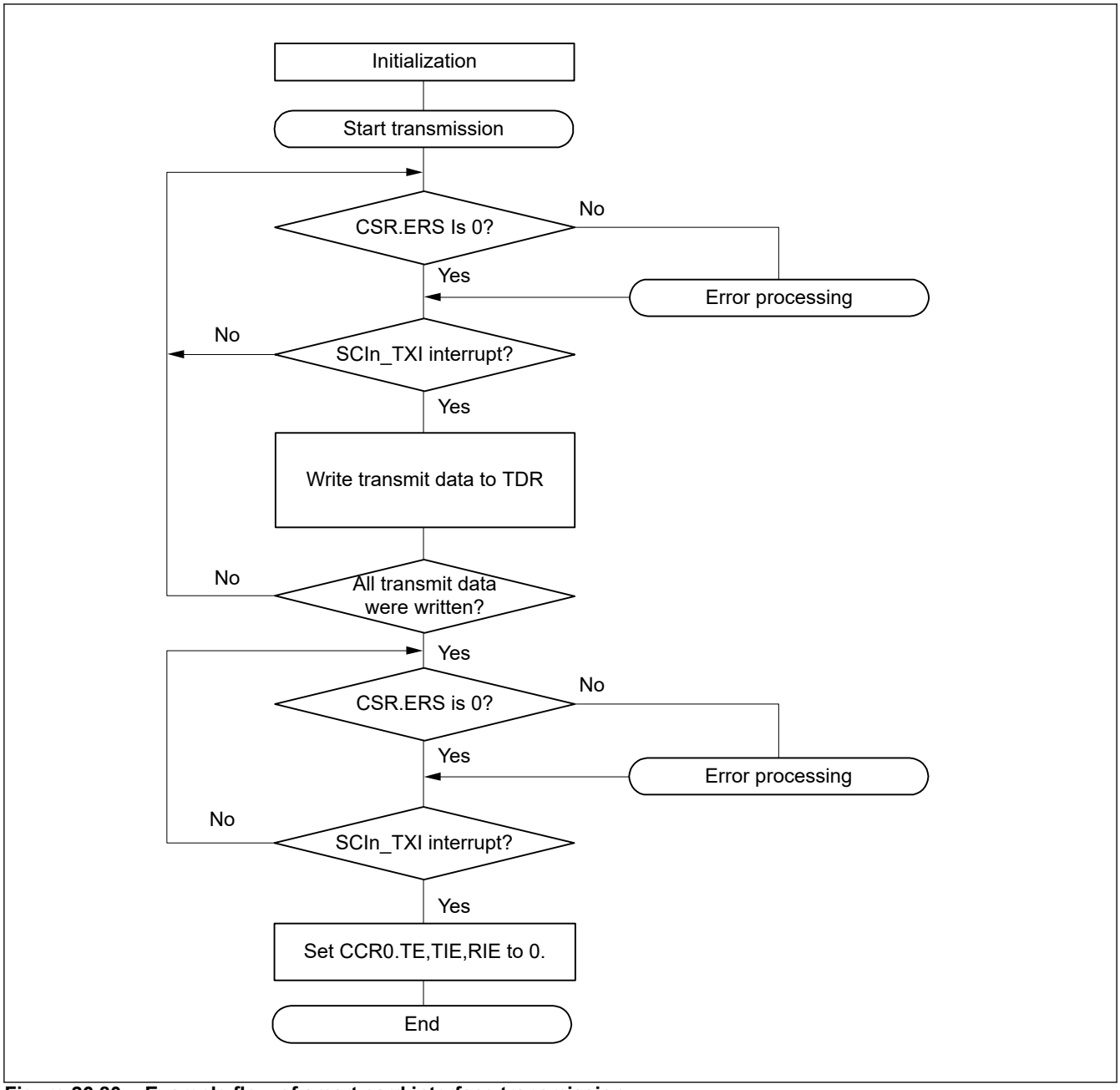


Figure 26.80 Example flow of smart card interface transmission

26.7.7 Serial Data Reception (Except in Block Transfer Mode)

Serial data reception in smart card interface mode is similar to that in non-smart card interface mode. Figure 26.81 shows the data re-transfer operation in reception mode.

1. If a parity error is detected in the receive data, the CSR.PER flag is set to 1. When the CCR0.RIE bit is 1, an SCIn_ERI interrupt request is generated. Clear the PER flag to 0 before the next parity bit is sampled.
2. For a frame in which a parity error is detected, no SCIn_RXI interrupt is generated.
3. When no parity error is detected, the CSR.PER flag is not set to 1.
4. In this case, data is determined to be received successfully. When the CCR0.RIE bit is 1, an SCIn_RXI interrupt request is generated.

Figure 26.82 shows an example flow of serial data reception. All the processing steps are automatically performed using an SCIn_RXI interrupt request to activate the DTC or DMAC.

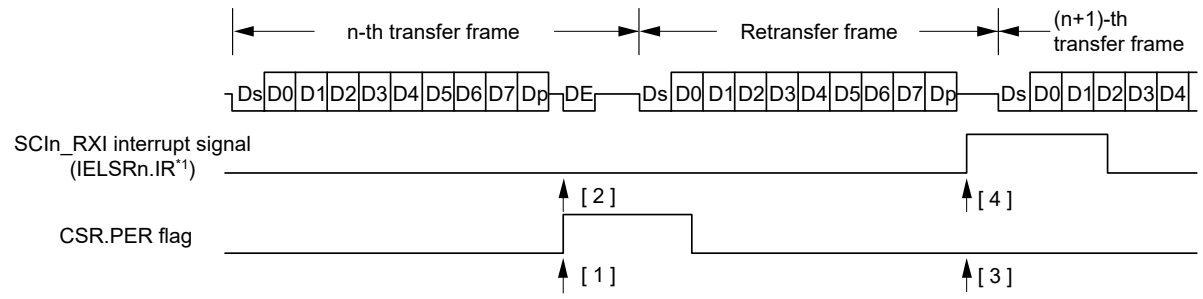
In reception, setting the RIE bit to 1 allows an SCIn_RXI interrupt request to be generated. The DTC or DMAC is activated by an SCIn_RXI interrupt request if the SCIn_RXI interrupt request is previously specified as a source of DTC or DMAC activation, allowing the transfer of receive data.

If an error occurs during reception and either the ORER or PER flag in CSR is set to 1, a receive error interrupt (SCIn_ERI) request is generated. Clear the error flag after the error occurrence. If an error occurs, the DTC or DMAC is not activated and receive data is skipped. Therefore, the number of bytes of receive data specified in the DTC or DMAC is transferred.

If a parity error occurs and the PER flag is set to 1 during reception, the receive data is transferred to RDR, allowing the data to be read.

When a reception is forced to terminate by setting CCR0.RE to 0 during operation, read the RDR register because the received data that is not yet read might be left in the RDR.

Note: For operations in block transfer mode, see section 26.3.9. Serial Data Reception in Asynchronous Mode.



Note 1. See section 12, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 26.81 Data re-transfer operation in smart card interface reception mode

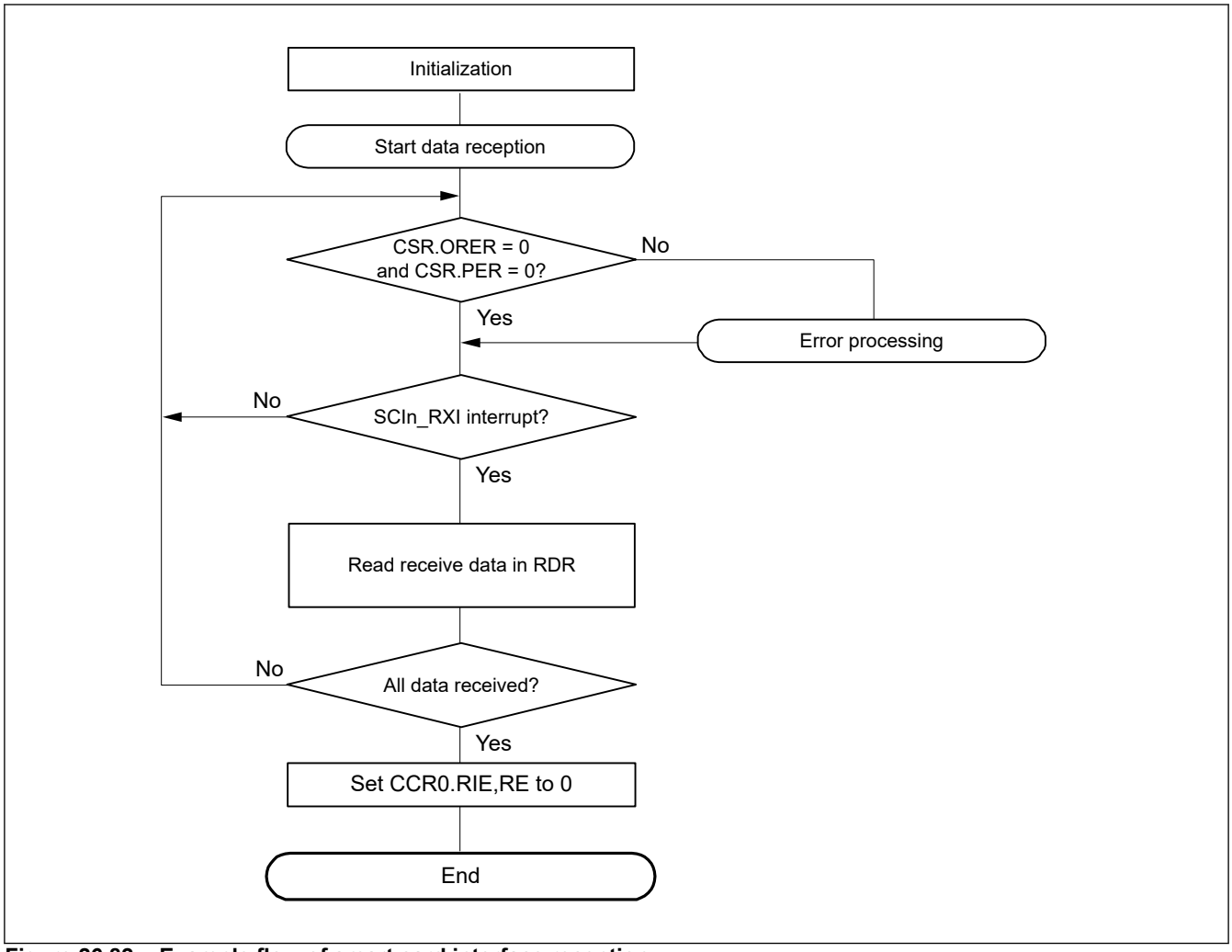


Figure 26.82 Example flow of smart card interface reception

26.7.8 Clock Output Control

When the GM bit in CCR3 is set to 1, the clock output can be controlled by the CKE[1:0] bits in CCR3. For details on the CKE[1:0] bits, see section 26.2.8. CCR3 : Common Control Register 3. When setting the clock output, the base clock described in section 26.7.4. Receive Data Sampling Timing and Reception Margin the bit rate is set by CCR2.CKS, CCR2.BCP[2:0] and BRR[7:0].

Figure 26.83 shows an example timing for the clock output control when the CKE[1] bit in CCR3 is set to 0 and the CKE[0] bit in CCR3 is controlled.

When the GM bit in CCR3 is 0, output control by the CKE[0] bit in CCR3 is immediately reflected on the SCKn pin, so there is a possibility that pulses with an unintended width may be output from the SCKn pin.

When the GM bit in CCR3 is 1, the output pulse control by the CCR3.CKE [0] controls the pulse width set to be based on the state of the base clock.

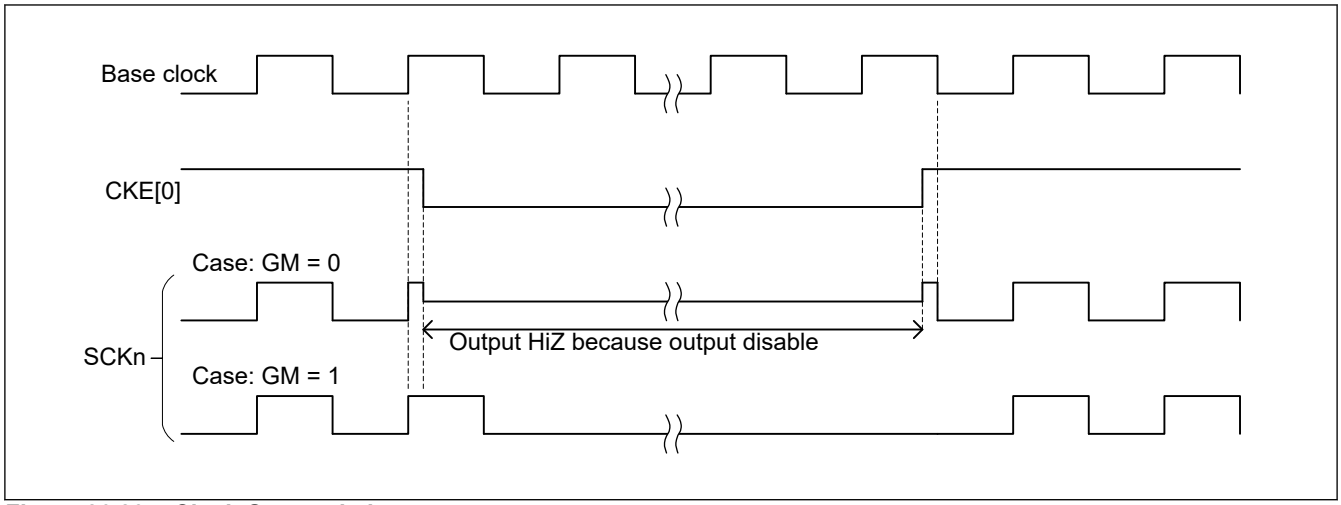


Figure 26.83 Clock Output timing

26.8 Operation in Simple IIC Mode

Simple IIC mode format is composed of 8 data bits and an acknowledge bit. By continuing into a slave-address frame after a start condition or restart condition, a master device can specify a slave device as the partner for communications. The currently specified slave device remains valid until a new slave device is specified or a stop condition is satisfied. The 8 data bits in all frames are transmitted in order from the MSB.

The I^2C bus format and timing of the I^2C bus are shown in Figure 26.84 and Figure 26.85.

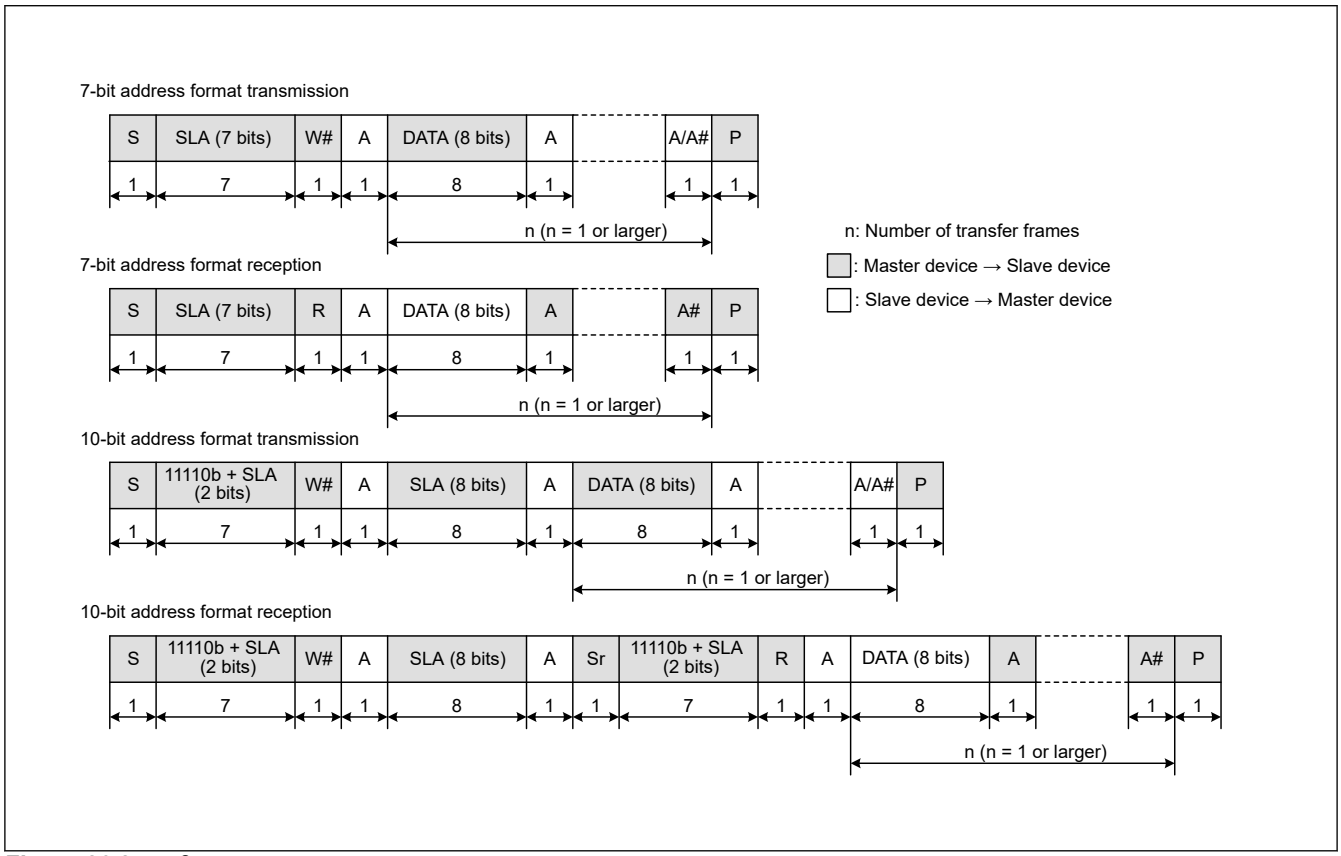


Figure 26.84 I^2C bus format

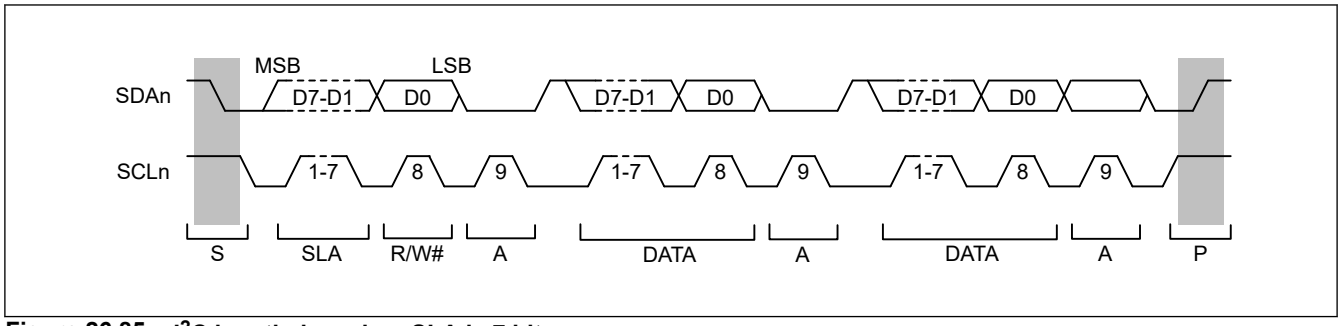


Figure 26.85 I²C bus timing when SLA is 7 bits

- S: Indicates a start condition, when the master device changes the level on the SDAn line from high to low while the SCLn line is high
- SLA: Indicates a slave address, by which the master device selects a slave device
- R/W#: Indicates the direction of transfer (reception or transmission). The value 1 indicates transfer from the slave device to the master device and 0 indicates transfer from the master device to the slave device.
- A/A#: Indicates an acknowledge bit. This is returned by the slave device for master transmission and by the master device for master reception. Return low indicates ACK and return high indicates NACK.
- Sr: Indicates a restart condition, when the master device changes the level on the SDAn line from high to low while the SCLn line is high and after the setup time elapses
- DATA: Indicates the data being received or transmitted
- P: Indicates a stop condition, when the master device changes the level on the SDAn line from low to high while the SCLn line is high

26.8.1 Generation of Start, Restart, and Stop Conditions

Writing 1 to the ICR.IICSTAREQ bit causes the generation of a start condition. The generation of a start condition proceeds through the following operations:

- The level on the SDAn line falls (from the high level to the low level) and the SCLn line is kept in the released state
- The hold time for the start condition is set as half of a bit period at the bit rate determined by the CCR2.BRR setting
- The level on the SCLn line falls (from the high level to the low level), the IICSTAREQ bit in ICR is set to 0, and a start-condition generated interrupt is output

Writing 1 to the IICRSTAREQ bit in ICR causes the generation of a restart condition. The generation of a restart condition proceeds through the following operations:

- The SDAn line is released and the SCLn line is kept at the low level
- The period at low level for the SCLn line is set as half of a bit period at the bit rate determined by the CCR2.BRR setting
- The SCLn line is released (transition from the low to the high level)
- When a high level is detected on the SCLn line, the setup time for the restart condition is set as half of a bit period at the bit rate determined by the CCR2.BRR setting
- The level on the SDAn line falls (from the high level to the low level)
- The hold time for the restart condition is set as half of a bit period at the bit rate determined by the CCR2.BRR setting
- The level on the SCLn line falls (from the high level to the low level), the ICR.IICRSTAREQ bit is set to 0, and a restart-condition generated interrupt is output

Writing 1 to the ICR.IICSTPREQ bit causes the generation of a stop condition. The generation of a stop condition proceeds through the following operations:

- The level on the SDAn line falls (from the high level to the low level) and the SCLn line is kept at the low level
- The period at low level for the SCLn line is set as half of a bit period at the bit rate determined by the CCR2.BRR setting

- The SCLn line is released (transition from the low to the high level)
- When a high level is detected on the SCLn line, the setup time for the stop condition is set as half of a bit period at the bit rate determined by the CCR2.BRR setting
- The SDAn line is released (transition from the low to the high level), the ICR.IICSTPREQ bit is set to 0, and a stop-condition generated interrupt is output

Figure 26.86 shows the timing of operations in the generation of start, restart, and stop conditions.

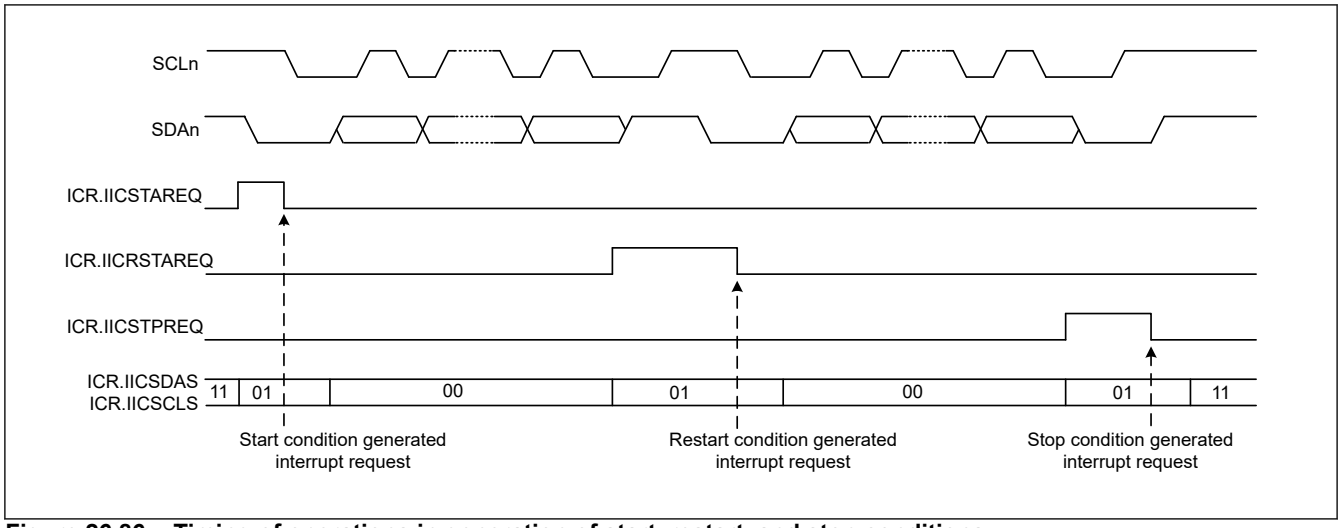


Figure 26.86 Timing of operations in generation of start, restart, and stop conditions

26.8.2 Clock Synchronization

The SCLn line can be driven low if a wait is inserted by a slave device at the other side of the transfer. Setting the ICR.IICCSC bit to 1 applies control to obtain synchronization when a difference arises between the levels of the internal SCLn clock signal and the level being input on the SCLn pin.

When the ICR.IICCSC bit is set to 1, the level of the internal SCLn clock signal changes from low to high. Counting to determine the period at a high level stops while the low level is being input on the SCLn pin. Counting to determine the period at a high level starts after the transition of the input on the SCLn pin to the high level.

The interval from this time until counting to determine the period at high level starts on the transition of the SCLn pin to the high level, is the total time which contains the SCLn input delay, delay for noise filtering of the input on the SCLn pin (2 or 3 cycles of sampling clock for the noise filter), and delay for internal processing (1 or 2 cycles of PCLK). The period at high level of the internal SCLn clock is extended even when other devices do not place the low level on the SCLn line.

If the ICR.IICCSC bit is 1, synchronization is obtained for the transmission and reception of data by taking the logical AND of the input on the SCLn pin and the internal SCLn clock. If the ICR.IICCSC bit is 0, synchronization with the internal SCLn clock is obtained for the transmission and reception of data.

If a slave device inserts a wait period into the interval until the transition of the internal SCLn clock signal from the low to the high level after a request for the generation of a start, restart, or stop condition is issued, the time until generation is prolonged by that period.

If a slave device inserts a wait period after the transition of the internal SCLn clock signal from the low to the high level, although the generation-completed interrupt is issued without stopping the waiting period, generation of the condition itself is not guaranteed.

Figure 26.87 shows an example operation for synchronizing the clocks.

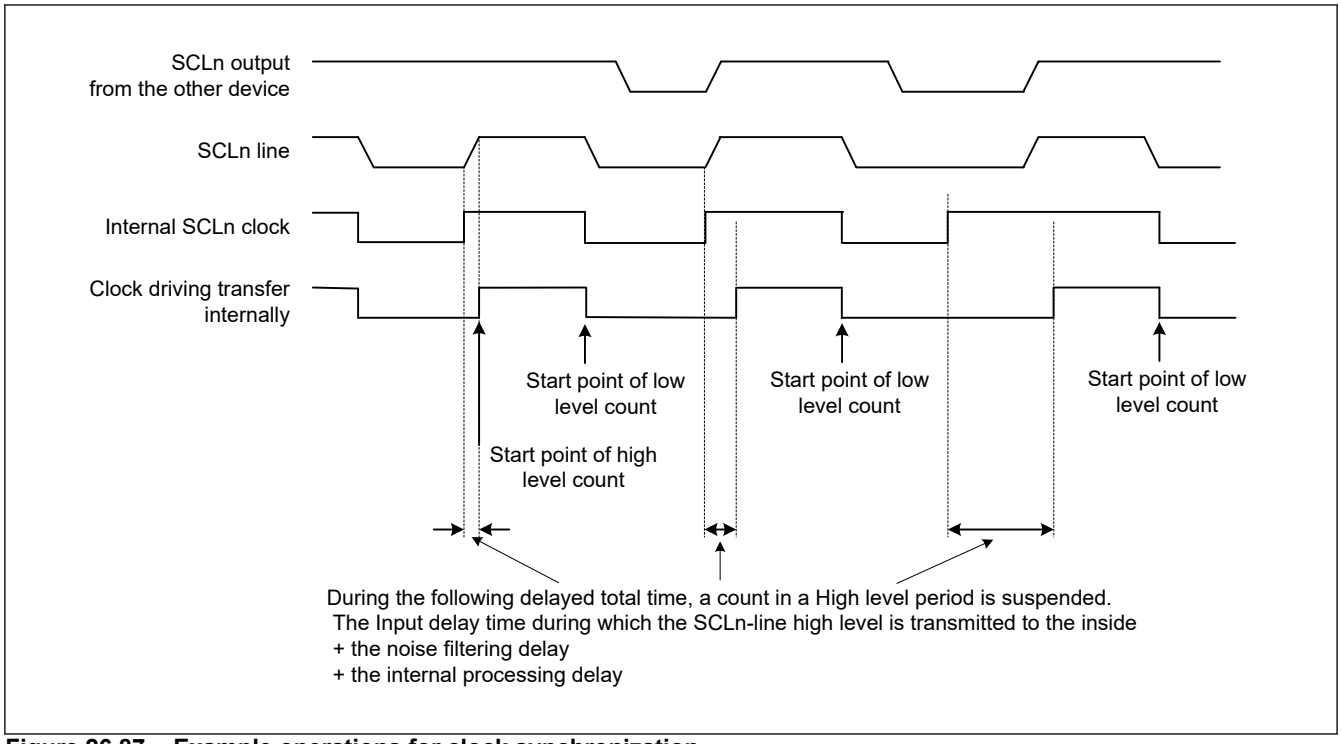


Figure 26.87 Example operations for clock synchronization

26.8.3 SDAn Output Delay

The ICR.IICDL[4:0] bits can be used to set a delay for output on the SDAn pin relative to falling edges of output on the SCLn pin. Delay settings from 0 to 31 are selectable, representing periods of the corresponding numbers of cycles of the clock signal from the on-chip baud rate generator (derived by frequency-dividing the base clock, TCLK, by the divisor selected in the CCR2.CKS[1:0] bits). A delay for output on the SDAn pin applies to the start condition/restart condition/stop condition signal, 8-bit transmit data, and acknowledge bit.

If the SDAn output delay is shorter than the time for the level on the SCLn pin to fall, the change of the output on the SDAn pin starts while the output level on the SCLn pin is falling, creating a possibility of erroneous operation for slave devices. Ensure that settings for the delay of output on the SDAn pin specify times greater than the time output on the SCLn pin takes to fall (300 ns for IIC in normal mode and fast mode).

Figure 26.88 shows the timing of delays in SDAn output.

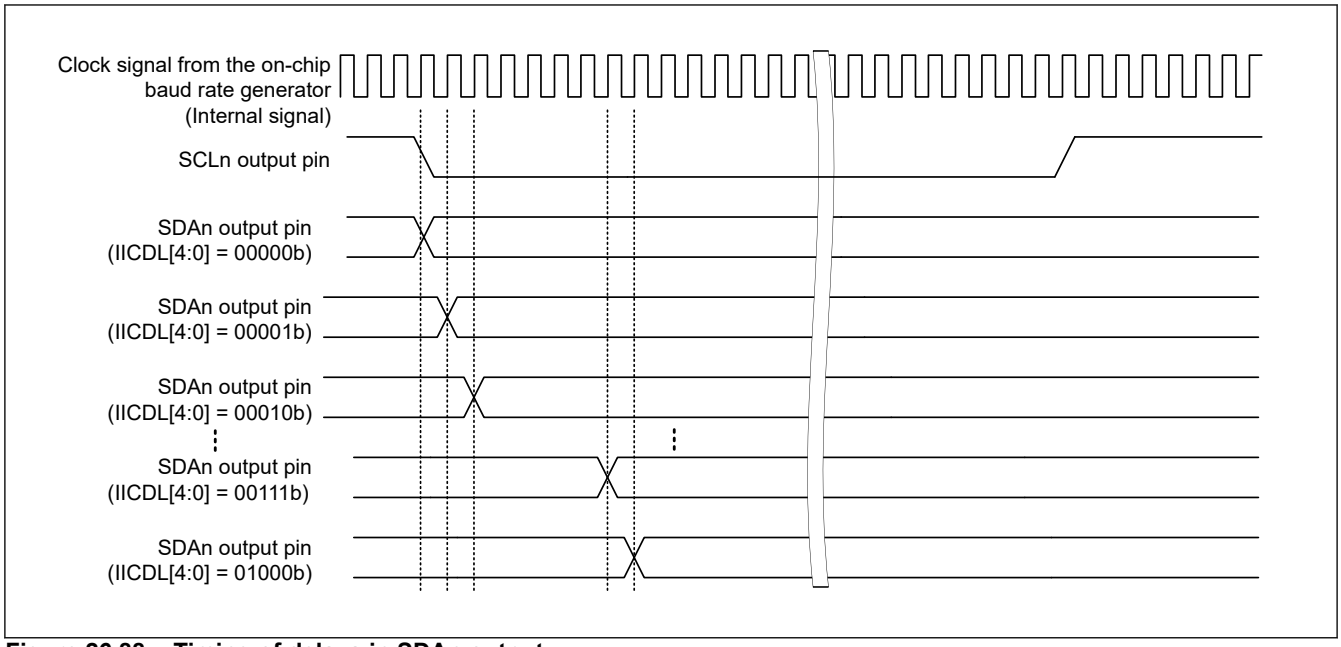


Figure 26.88 Timing of delays in SDAn output

26.8.4 SCI Initialization in Simple IIC Mode

Before transferring data, write the initial value 0x00 to CCR0 and initialize the interface following the example shown in Table 26.38.

Before making any changes to the operating mode or transfer format, be sure to set CCR0 to its initial value. In simple IIC mode, the open-drain setting for the communication ports should be made on the port side.

Table 26.38 Example flow of SCI initialization in simple IIC mode

No.	Step Name	Description
1	Start initialization	—
2	Set CCR0	Set CCR0.TEIE, TIE, RIE, TE, RE to 0. If you have not changed from the initial settings, you can skip this step.
3	Set ICR	Set the IICSDAS[1:0] and IICSCLS[1:0] to 11b. Set the IICDL[4:0] and IICINTM as required. Set the IICACKT and the IICCSC bits to 1.
4	Set CCR3	Set the transmission / reception format as the communication mode (MOD [2:0] = 100b) and CKE [1:0] = 00b.
5	Set CCR2	Set the bit rate modulation function*1, the clock selection, and the bit rate.
6	Set CCR1	Set noise filter, communication pin status, parity check, and CTSn / RTSn function.
7	Set the I/O port functions	Set I/O port settings that allow use (on NMOS open-drain output pins and Hi-Z) of the SCLn and SDAn pin functions.
8	Set CFCLR, ICFCLR	Write 1 to the following bits and clear the corresponding flag. CFCLR.RDRFC, FERC, PERC, MFFC, ORERC, DFERC, DPERC, DCMFC, ERSC ICFCLR.IICSTIFC
9	Set CCR0 (TE, RE, TIE, RIE)	Set the TE and RE bits to 1. To enable interrupts, set the TE, TIE, RE and RIE bits to 1 with one instruction at the same time (for transmission and when the IICINTM bit is 1, clear the RIE bit). Setting the TE and RE bits to 1 makes the SCLn and SDAn pins functions available.
10	Initialization completed	—

Note: Set the CCR0.TE and RE bits to 0 or 1 at the same time.
Note 1. If you do not use the bit rate modulation function, you do not need to set it.

26.8.5 Operation in Master Transmission in Simple IIC Mode

Figure 26.89 and Figure 26.90 show examples of master transmission and Figure 26.91 shows an example flow of data transmission.

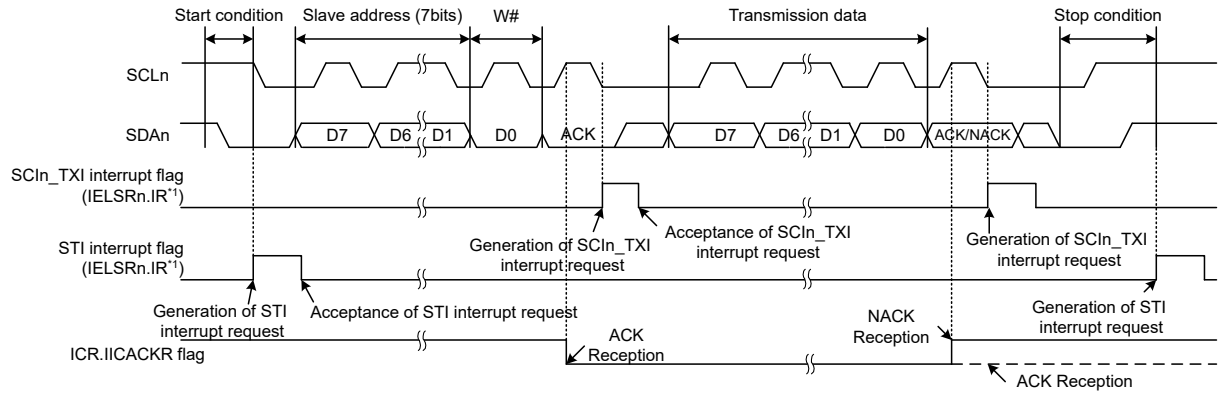
Figure 26.89 shows the operation example when ICR.IICINTM bit is 1 (use reception and transmission interrupts). In this case, you can start DMAC or DTC by SCIn_TXI interrupt. However, if use DMAC or DTC, ACK/NACK cannot be confirmed. If you want to confirm ACK/NACK, prepare the transmit data by CPU.

In simple IIC mode, SCIn_TXI interrupt is generated when communication of one frame is completed. And it is not used SCIn_RXI interrupt in master transmission, so the CCR0.RIE set to 0.

See Table 26.43 for more information on the STI interrupt.

Figure 26.91 shows a flow chart in the case of ICR.IICINTM is 1 and address transmission by CPU and data transmission by DTC or DMAC. When 10-bit slave addresses are in use, steps [3] and [4] are repeated twice.

In simple IIC mode, the transmit data empty interrupt (SCIn_TXI) is generated when communication of one frame is complete, unlike the SCIn_TXI interrupt request generation timing during clock synchronous transmission.



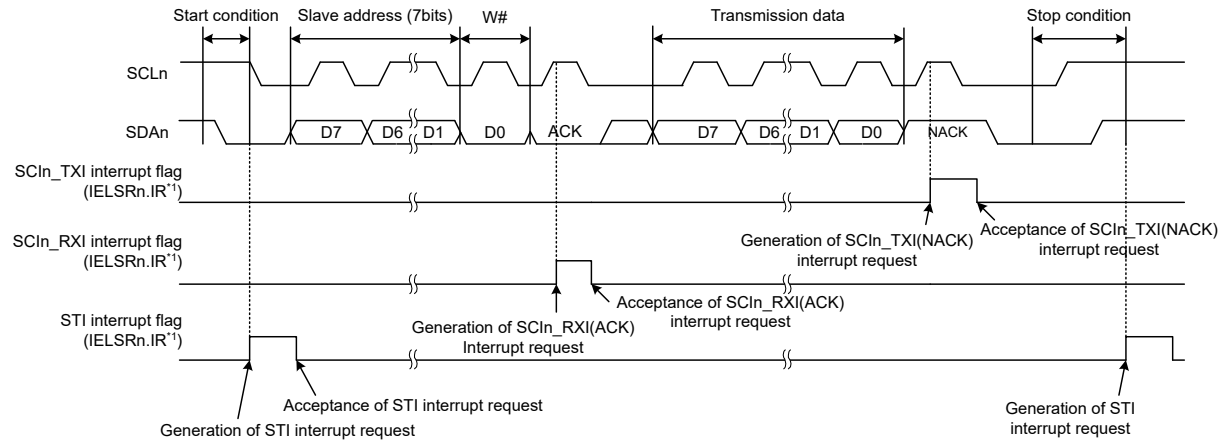
Note 1. See section 12, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 26.89 Example 1 of operations for master transmission in simple IIC mode with 7-bit slave addresses, transmission interrupts, and reception interrupts (ICR.IICINTM = 1)

When the ICR.IICINTM bit is set to 0 (use ACK/NACK interrupts) during master transmission, the DTC or DMAC is activated by the ACK interrupt as the trigger and required number of data bytes are transmitted. When the NACK is received, error processing such as transmission stop and retransmission is performed using the NACK interrupt as the trigger.

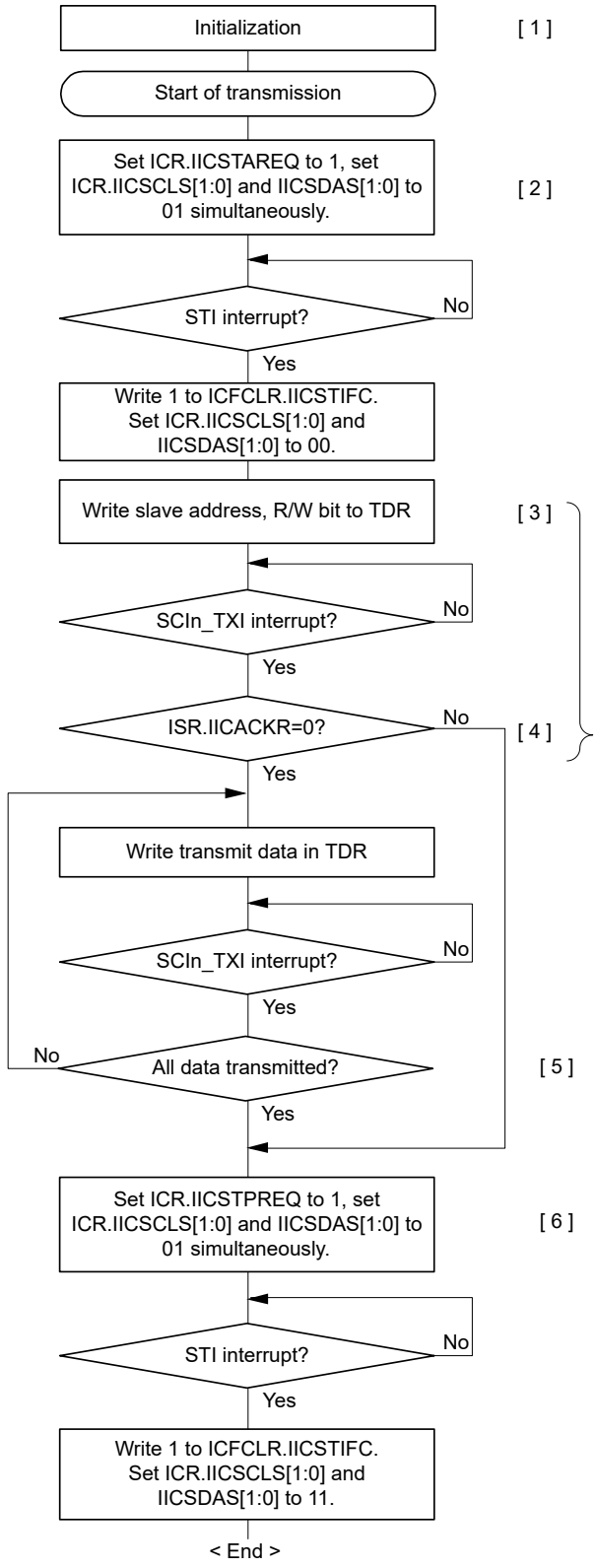
To restart communication for some reason after writing data in the TDR register, use the following procedure:

1. Set the TE and RE bits in the CCR0 register to 0 to stop communication.
2. Set ICR.IICSCLS[1:0] and ICR.IICSDAS[1:0] bits to 11b, release the I^2C bus, and clear the generation of a condition.
3. If the RDRF flag in the CSR register is set to 1, the RDR register is read by dummy and the RDRF bit is set to 0.
4. Set the TE and RE bits in the CCR0 register to 1 and start the next communication.



Note 1. See section 12, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 26.90 Example 2 of operations for master transmission in simple IIC mode with 7-bit slave addresses, ACK interrupts, and NACK interrupts (ICR.IICINTM = 0)



[1] Initialization for simple IIC mode
For transmission, set the CCR0.RIE to 0.
(SCIn_RXI and SCIn_ERI interrupts requests are disabled)

[2] Generate a start condition.

[3] Writing to TDR:
Writing the slave address and value for the R/W bit to TDR.

[4] Confirm ACK response from the slave device:
Check the ISR.IICACKR bit. If its value is 0, it is indicated that the slave device responded with ACK and operations proceed. If its value is 1, it is indicated that there was no response from a slave device so the next transition is to generation of the stop condition.

[5] Steps for continuing with serial transmission:
When transmission is to continue, write further transmit data to TDR. Except for the first data to be transmitted, a SCIn_TXI interrupt request can activate the DMAC or DTC to handle writing of data to TDR.
But it cannot be confirmed status of ACK or NACK in this case.

[6] Generation of a stop condition.

If 10-bits slave address are in use, processing from [3] to [4] is repeated twice.

Note: In simple IIC mode, the SCIn_TXI interrupt request is generated when communication is completed.

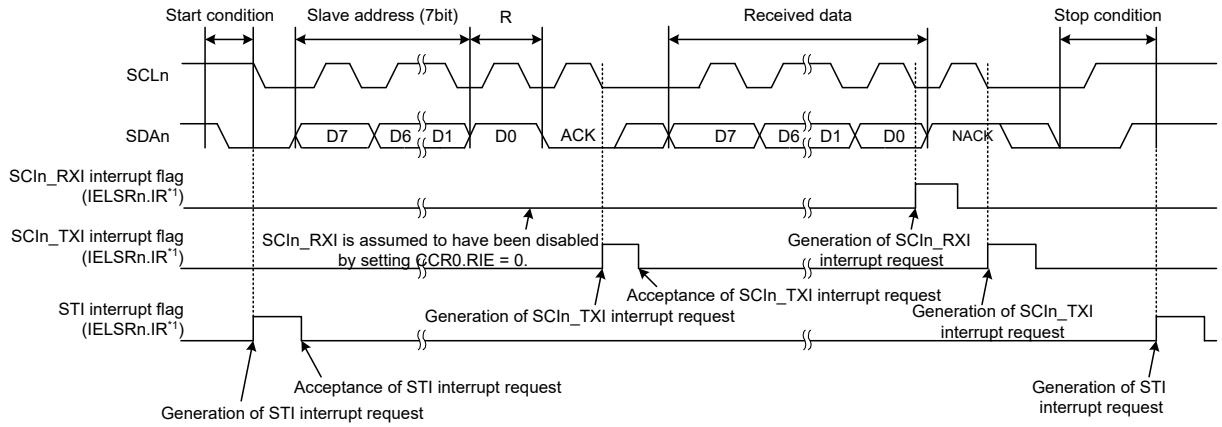
Figure 26.91 Example flow of master transmission in simple IIC mode with transmission interrupts and reception interrupts

26.8.6 Master Reception in Simple IIC Mode

Figure 26.92 shows an example operation in simple IIC mode master reception and Figure 26.94 shows an example flow of master reception.

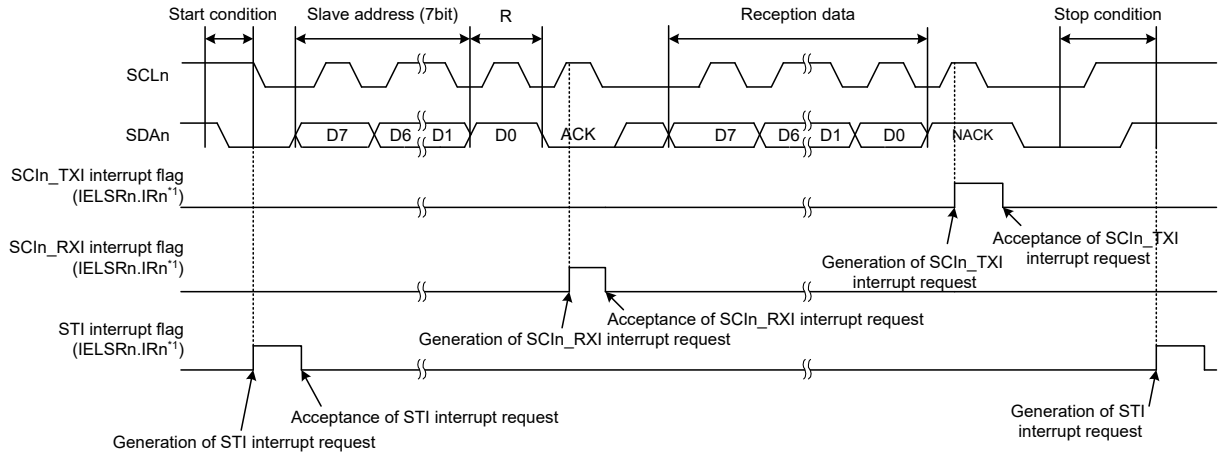
The value of the ICR.IICINTM bit is assumed to be 1 (use reception and transmission interrupts) and 0 (use ACK and NACK interrupts).

In simple IIC mode, the transmit data empty interrupt (SCIn_TXI) is generated when communication of one frame is complete, unlike the SCIn_TXI interrupt request generation timing during clock synchronous transmission.



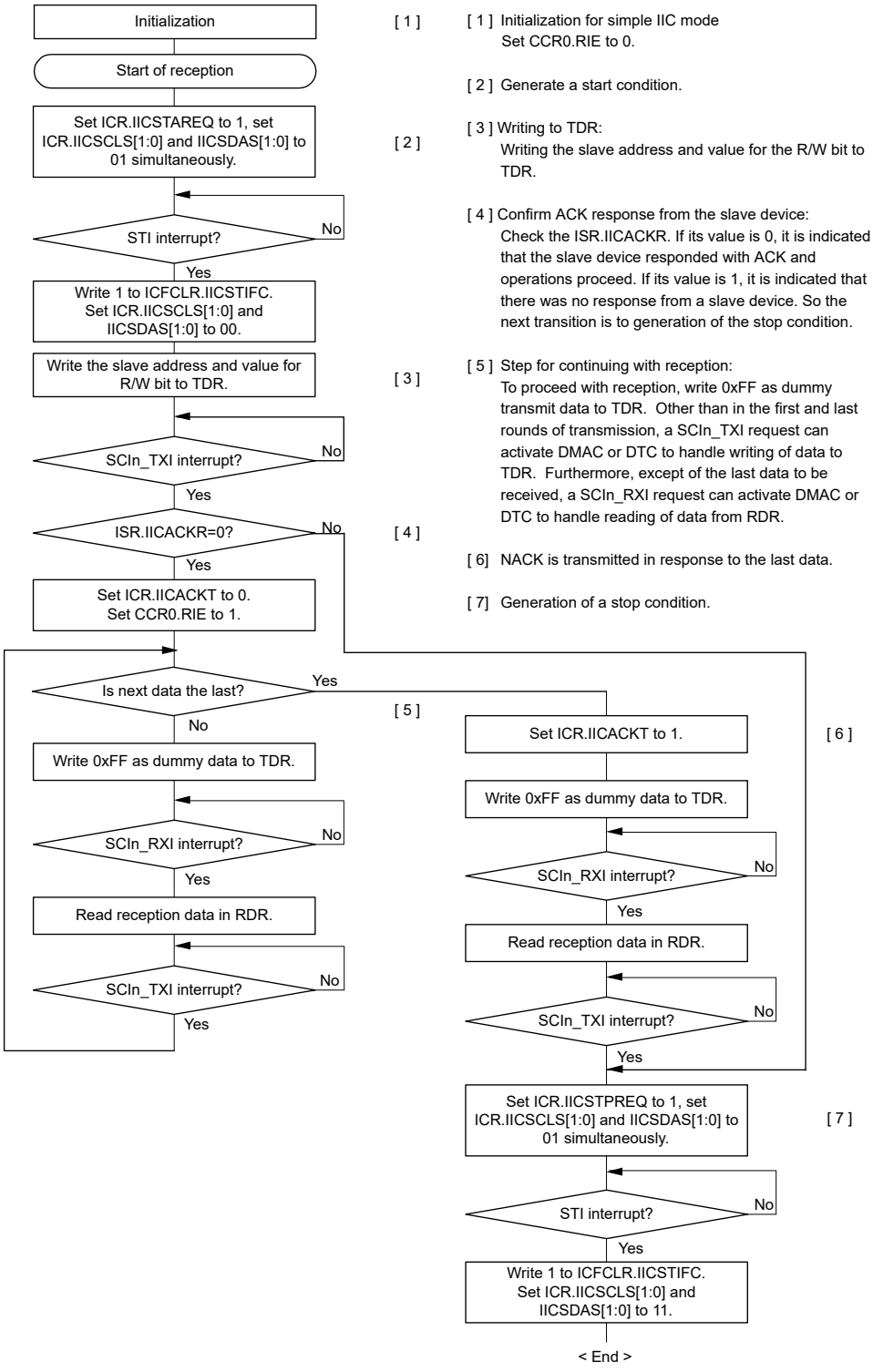
Note 1. See section 12, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 26.92 Example operations for master reception in simple IIC mode with 7-bit slave addresses, transmission interrupts, and reception interrupts (ICR.IICINTM = 1)



Note 1. See section 12, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 26.93 Example of Operations for Master Reception in Simple IIC Mode (7bit Slave address, ACK and NACK interrupt in use (ICR.IICINTM = 0))



[1] Initialization for simple IIC mode
Set CCR0.RIE to 0.

[2] Generate a start condition.

[3] Writing to TDR:
Writing the slave address and value for the R/W bit to TDR.

[4] Confirm ACK response from the slave device:
Check the ISR.IICACKR. If its value is 0, it is indicated that the slave device responded with ACK and operations proceed. If its value is 1, it is indicated that there was no response from a slave device. So the next transition is to generation of the stop condition.

[5] Step for continuing with reception:
To proceed with reception, write 0xFF as dummy transmit data to TDR. Other than in the first and last rounds of transmission, a SCIn_TXI request can activate DMAC or DTC to handle writing of data to TDR. Furthermore, except of the last data to be received, a SCIn_RXI request can activate DMAC or DTC to handle reading of data from RDR.

[6] NACK is transmitted in response to the last data.

[7] Generation of a stop condition.

Note: In simple IIC mode, the SCIn_TXI interrupt request is generated when communication is completed.

Figure 26.94 Example flow of master reception in simple IIC mode with transmission interrupts and reception interrupts (ICR.IICINTM = 1)

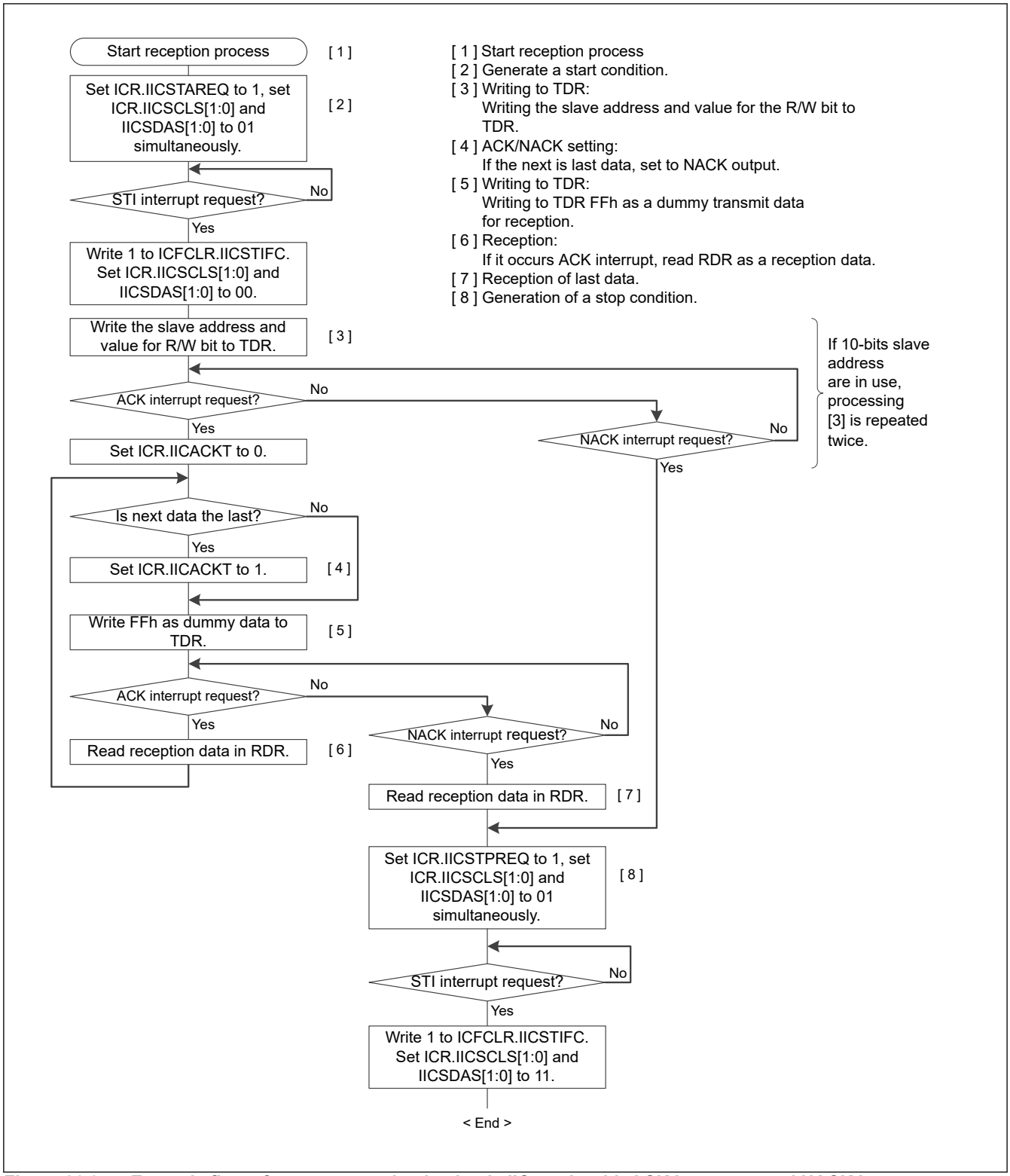


Figure 26.95 Example flow of master reception in simple IIC mode with ACK Interrupts and NACK Interrupts (ICR.IICINTM = 0)

26.9 Operation in Simple SPI Mode

As an extended function, the SCI supports a simple SPI mode that handles transfer among one or multiple master devices and multiple slave devices.

Using the settings for Simple SPI mode setting (CCR3.MOD[2:0] bit = 011b) and setting the CCR0.SSE bit to 1 place the SCI in simple SPI mode. However, the SSn pin function on the master side is not required for connection of the device used

as the master in simple SPI mode when the configuration only has a single master. Therefore, set the CCR0.SSE bit to 0 in such cases.

Figure 26.96 shows an example of connections for simple SPI mode. Control a general port pin to produce the SSn output signal from the master.

In simple SPI mode, data is transferred in synchronization with clock pulses in the same way as in clock synchronous mode. One character of data for transfer consists of 8 bits of data, and parity bits cannot be appended. The data can be inverted by setting the CCR3.SINV bit to 1.

Because the receiver and transmitter are independent of each other within the SCI module, full-duplex communications are possible, with a shared clock signal. Additionally, because both the transmitter and receiver have a buffered structure, writing the next transmit data while transmission is in progress and reading previously received data while reception is in progress are both possible. This enables continuous transfer.

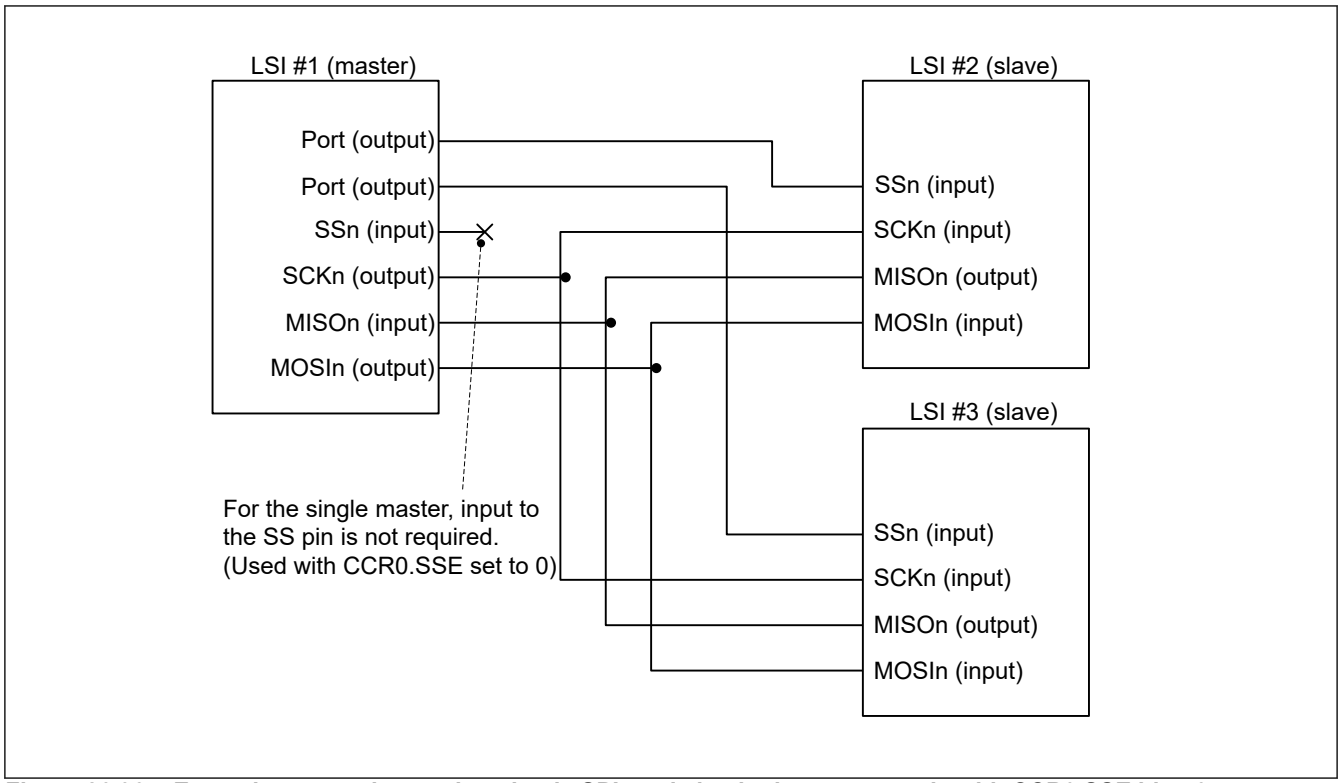


Figure 26.96 Example connections using simple SPI mode in single master mode with CCR0.SSE bit = 0

26.9.1 States of Pins in Master and Slave Modes

The direction (input or output) of pins for the simple SPI mode interface differs according to whether the device is a master (CCR3.CKE[1:0] = 00b or 01b) or slave (CCR3.CKE[1:0] = 10b or 11b).

Table 26.39 lists the relationship between the pin states, mode, and level on the SSn pin.

Table 26.39 States of pins by mode and input level on SSn pin

Mode	Input on SSn pin	State of MOSIn pin	State of MISOn pin	State of SCKn pin
Master mode*1	High level (transfer can proceed)	Output for data transmission*2	Input for received data	Clock output*3
	Low level (transfer cannot proceed)	High-impedance	Input for received data (but disabled)	High-impedance
Slave mode	High level (transfer cannot proceed)	Input for received data (but disabled)	High-impedance	Clock input (but disabled)
	Low level (transfer can proceed)	Input for received data	Output for data transmission	Clock input

Note 1. When there is only a single master (CCR0.SSE = 0), transfer is possible regardless of the input level on the SSn pin. This is equivalent to input of a high level on the SSn pin. The SSn pin is not used and is available for other purposes.

Note 2. The MOSIn pin output is in the high-impedance state when serial transmission is disabled (CCR0.TE bit = 0).
Note 3. The SCKn pin output is in the high-impedance state when serial transmission is disabled (CCR0.TE = 0 and CCR0.RE = 0) in a multi-master configuration (CCR0.SSE = 1).

26.9.2 SS Function in Master Mode

Setting the CKE[1:0] bits in the CCR3 to 00b or 01b selects master mode operation. The SSn pin is not used in single-master configurations (CCR0.SSE = 0), so transmission or reception can proceed regardless of the value of the SSn pin.

When the level on the SSn pin is high in a multi-master configuration (CCR0.SSE = 1), a master device outputs clock signals from the SCKn pin before starting transmission or reception to indicate that there are no other masters or another master is performing reception or transmission.

When the level on the SSn pin is low in a multi-master configuration (CCR0.SSE = 1), there are other masters, and a transmission or reception is in progress. The MOSIn output and SCKn pins are placed in the high-impedance state and starting transmission or reception is not possible. In addition, the value of the CSR.MFF bit is 1, indicating a mode fault error. In a multi-master configuration, start error processing by reading CSR.MFF flag. If a mode fault error occurs while transmission or reception is in progress, transmission or reception does not stop, but the MOSIn and SCKn outputs are in the high-impedance state after completion of the transfer.

When the SSn pin input becomes high level, the SCKn pin outputs a clock signal and the MOSIn outputs data. Even if the SCKn pin and the MOSIn pin are in the high impedance state, internal transmission or reception operation continues, but it stops after transmission or reception of a single character is complete. In this case, any of SCIn_TXI, SCIn_RXI, and SCIn_TEI interrupts occurs.

Use a general port pin to produce the SS output signal from the master.

26.9.3 SS Function in Slave Mode

Setting the CCR3.CKE[1:0] bits to 10b or 11b selects slave operation. When the SSn pin is high, the MISOn output pin is in the high-impedance state and clock input through the SCKn pin is ignored. When the SSn pin is low, clock input through the SCKn pin is valid and transmission or reception can proceed.

If the input on the SSn pin changes from low to high during transmission or reception, the MISOn output pin is placed in the high-impedance state. Transmission / reception operation is immediately suspended. If the transmission is in progress, the CSR.TEND flag will not be set, a transmit end interrupt will not be output, and an abnormal stop status will occur. So, do not negate the SSn pin during slave transmission / reception. If an abnormal stop occurs, set CCR0.RE and CCR0.TE to 0 to stop transmission / reception. To resume transmission / reception, set CCR0.RE and CCR0.TE to 1 after at least TCLK × 3 cycles + PCLK × 3 cycles.

26.9.4 Relationship between Clock and Transmit/Receive Data

The CPOL and CPHA bits in the CCR3 register can be used to set up the clock for use in transmission and reception in four different ways. The relation between the clock signal and the transmission and reception of data is shown in Figure 26.97. The relation is the same for both master and slave operation. This is the same as when the level on the SSn pin is high.

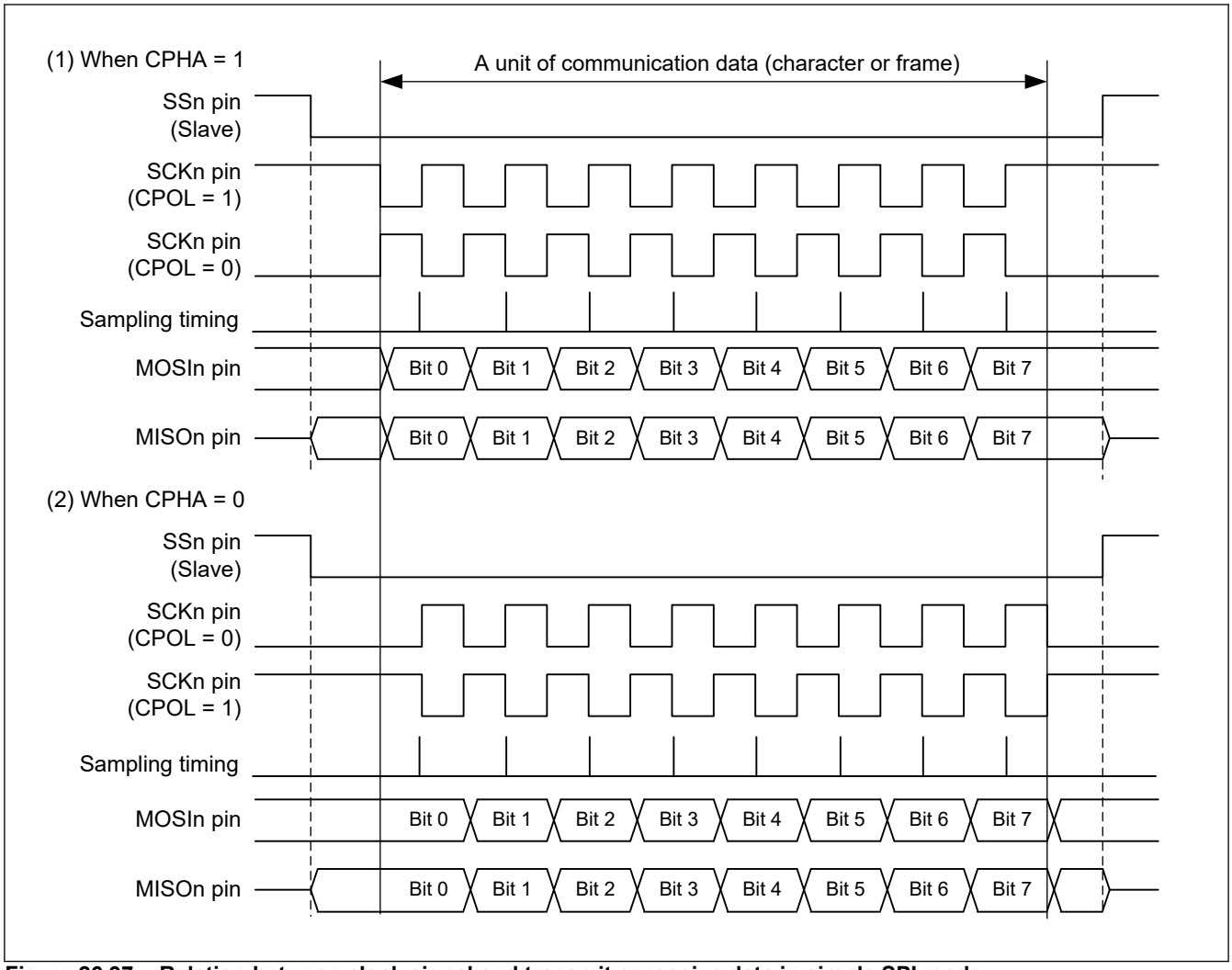


Figure 26.97 Relation between clock signal and transmit or receive data in simple SPI mode

26.9.5 SCI Initialization in Simple SPI Mode

Initialization in simple SPI mode is the same as in clock synchronous mode. See section 26.6.3. SCI Initialization in Clock Synchronous Mode for an example initialization flow. The CPOL and CPHA bits in the CCR3 register must be set to ensure that the clock signal is suitable for both master and slave devices.

Always initialize the CCR0 register before making any changes to the operating mode or transfer format.

Note: Only the RE bit is set to 0. The CSR.ORER, FER, PER, and RDR flags are not initialized.

Changing the value of the TE bit from 1 to 0 or from 0 to 1 when the TIE bit in the CCR0 register is 1 at the same time, leads to the generation of a transmit data empty interrupt (SCIn_TXI).

26.9.6 Transmission and Reception of Serial Data in Simple SPI Mode

In master operation, ensure that the SSn pin of the slave device on the other side of the transfer is at the low level before starting the transfer and at the high level on completion of the transfer. In multiple master operation with CCR0.SSE = 1 even in master mode, a mode fault error will occur if the SSn pin goes low. Therefore, make sure that no mode fault error has occurred before starting communication, and start communication, and make sure that no mode fault error has occurred even after communication ends. If a mode fault error has occurred, communication may be incomplete, so measures such as retransmission are required. Otherwise, the procedures are the same as in clock synchronous mode.

In slave mode, it operates according to the SSn pin input level. Other steps are the same as those of clock synchronous mode.

26.9.7 Reception Sampling Timing Adjustment Function in Simple SPI Mode with internal clock used

The reception sampling timing adjustment function in simple SPI mode is the same as the reception sampling timing adjustment function in clock synchronous mode. For the description of operation, see section 26.6.7. Reception Sampling Timing Adjustment Function in Clock Synchronous Mode with internal clock used.

26.10 Bit Rate Modulation Function

Using the bit rate modulation function, the bit rate can be evenly corrected using the number specified in the MDDR register among 256 clock cycles of internal clocks which is selected by the CKS[1:0] bits in CCR2.

Figure 26.98 shows an example where the PCLK is selected in the CKS[1:0] bits in CCR2, the BRR bit is set to 0, and the MDDR is set to 160 in asynchronous mode. In this example, the cycle of the base clock is evenly corrected (256/160) and the bit rate is also corrected (160/256).

Note: Enabling an internal clock causes bias, and expansion and contraction are generated in the pulse width of the internal base clock.

Do not use this function in clock synchronous mode, simple SPI mode, Smart Card Interface mode, Manchester mode and Simple LIN mode.

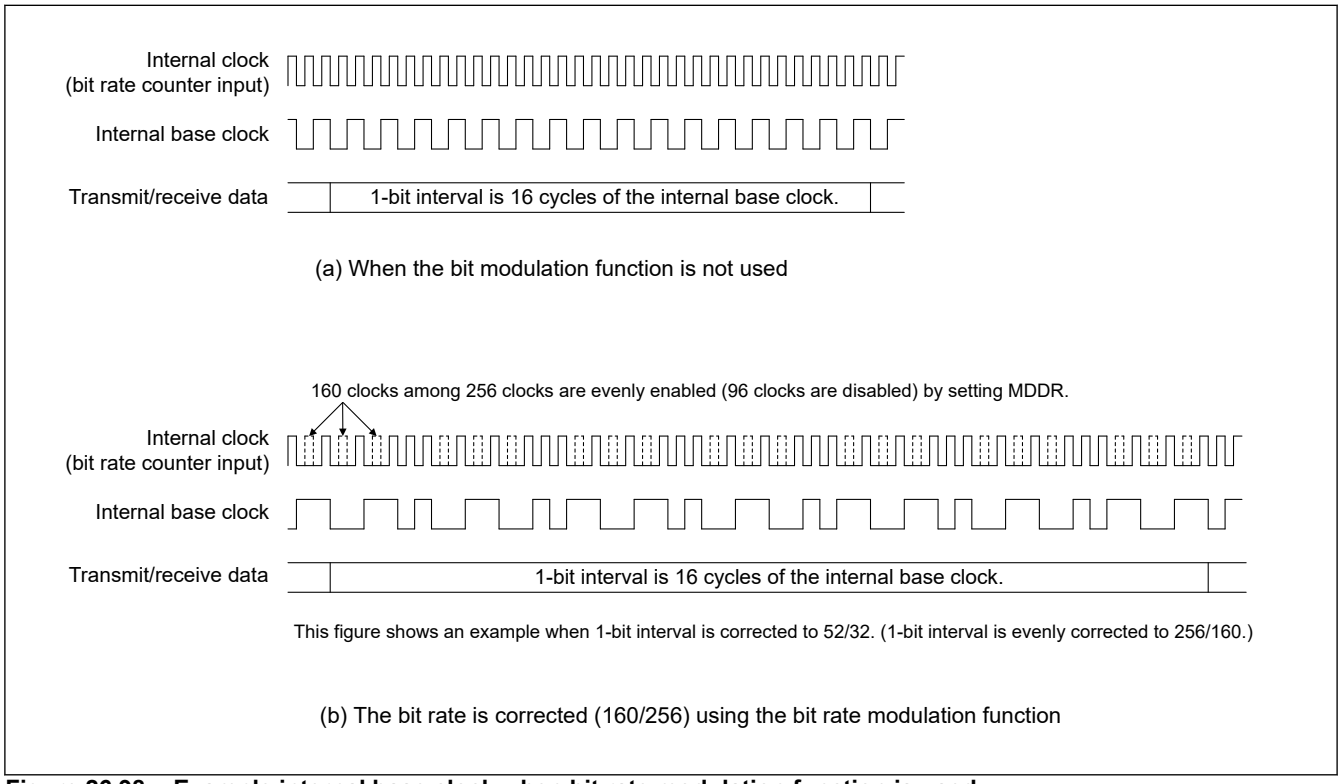


Figure 26.98 Example internal base clock when bit rate modulation function is used

26.11 Simple LIN mode

As an extended function of the SCI, the SCI supports the serial communication protocol (Figure 26.99) consisting of a Start Frame and an Information Frame as Simple LIN. Simple LIN mode is enabled by the CCR3.MOD[2:0] = 110b. Since the simple LIN mode uses the same circuit as the asynchronous mode for transmission / reception control other than Break Field, the basic communication settings required for the asynchronous mode are also required for the simple LIN mode.

(For the setting value when using simple LIN, see the explanation in section 26.2. Register Descriptions. In particular, note that CCR3.RXDESEL needs to be changed from the initial value and set to 1.)

The Start Frame consists of a Break Field, Control Field 0, and Control Field 1. The Information Frame can be configured with some Data Fields, a CRC16 Upper Field, and a CRC16 Lower Field.

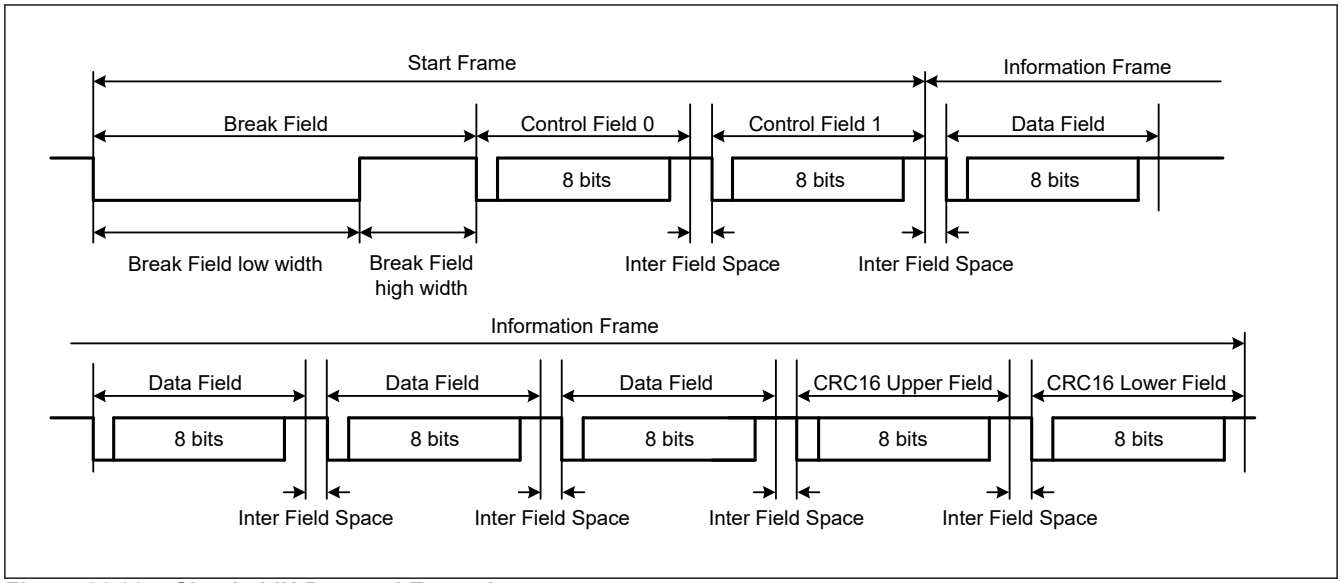


Figure 26.99 Simple LIN Protocol Example

The following describes operations when the Simple LIN is used. In this section, operations are described with the following conditions:

Communication pin (RXDn / TXDn) level inversion function: OFF (RINV = TINV = 0)

When using the simple LIN with the communication pin (RXDn / TXDn) level inversion function enabled, replace the RXDn and TXDn signal levels with their inverted levels.

26.11.1 Simple LIN Start Frame Transmission

Figure 26.100 shows an example of transmission of the Start Frame consisting of a Break Field, Control Field0, and Control Filed1. (Omit Break Field and Control Field0 according to the Start Frame configuration.)

Figure 26.101 shows a flowchart for Start Frame transmission.

The SCI operates as follows during Start Frame transmission.

1. Make the initial settings for the SCI according to the SCI initialization flow (Figure 26.66) in asynchronous mode. In simple LIN mode, do not set CCR0.TE and TIE to 1 at the same time to avoid SCIn_TXI output before the Break Field. Therefore, perform the following two steps sequentially to set the SCI initialization flow (asynchronous mode) procedure [9].
   - Set the bits except CCR0.TIE. (CCR0.TIE = 0, CCR0.TE = 1, and CCR0.RE = 0)
   - Set CCR0.TIE to 1.
2. When 1 is written to TCST, the Break Field output timer starts counting and outputs a low level (Break Field) from the TXDn pin for the period set in XCR2.BFLW[15:0]. A timer count clock source can be selected by XCR0.TCSS[1:0]. Writing 0 to XCR1.TCST suspends output of the Break Field. After the suspension, set CCR0.TE = 0 and turn off the transmission.
3. When the Simple LIN module timer count value matches the set XCR2.BFLW[15:0] value, the timer stops counting and inverts the TXDn pin output level, and the XSR0.BFOF flag is set to 1[*1]. Furthermore, if XCR0.BFOIE has been set to 1 at this time, a SCIn_TXI interrupt is generated.
4. After the SCIn_TXI interrupt and confirming XSR0.BFOF = 1, write the transmitted data then the Control Field 0 data is transmitted using the SCI[*1].

Note: LIN communication requires a Break Delimiter (IDLE period) of 1 bit or more from the end of Break Field output until the next data transmission starts. For this reason, the Break Delimiter length is counted upon completion of Break Field output. If transmit data is written while the Break Delimiter length is being counted, transmission does not start until the Break Delimiter length counting is completed. When transmit data is written after the Break Delimiter length has been counted, transmission starts at the same timing as normal data transmission.
Break Delimiter length count time after Break Field output:
1-bit to 2-bit length (CCR3.STP = 0)

2-bit to 3-bit length (CCR3.STP = 1)

5. After the Control Field 0 data has been transmitted, write the Control Field 1 data to TDR. And it is transmitted.
6. After the Control Field 1 data has been transmitted, the Information Frame data is transmitted.

Note 1. After XSR0.BFOF is set to 1, if 1 is written to XCR1.TCST without clearing it, no SCIn_TXI interrupt is output at the end of Break Field transmission. Clear XSR0.BFOF before writing 1 to XCR1.TCST.

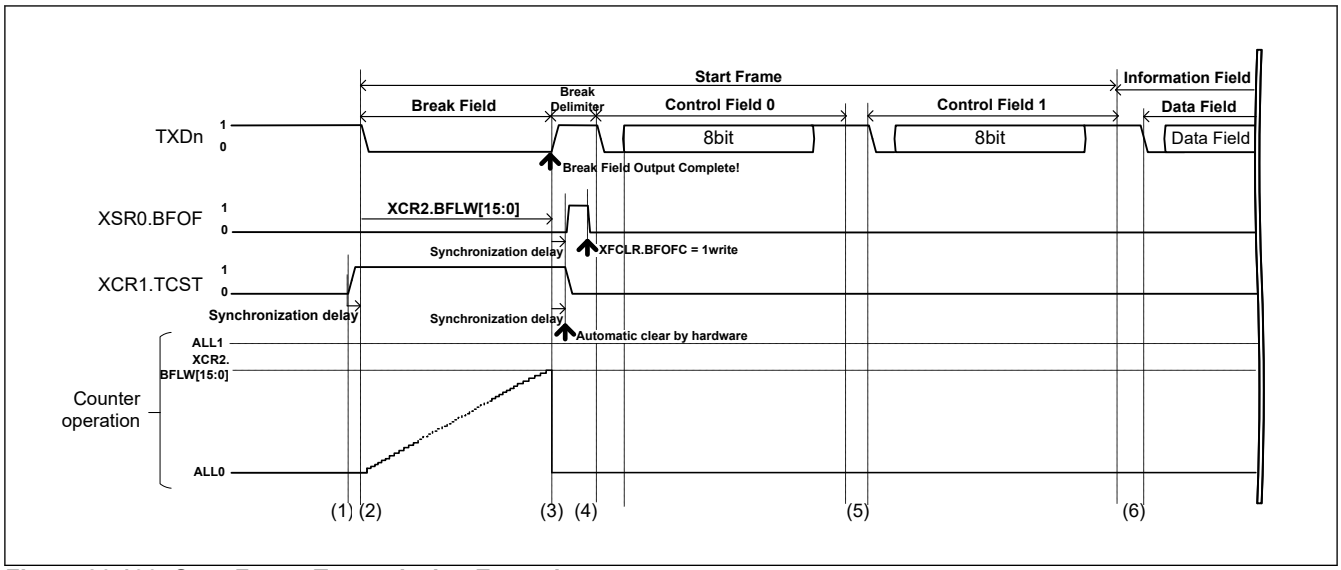


Figure 26.100 Start Frame Transmission Example

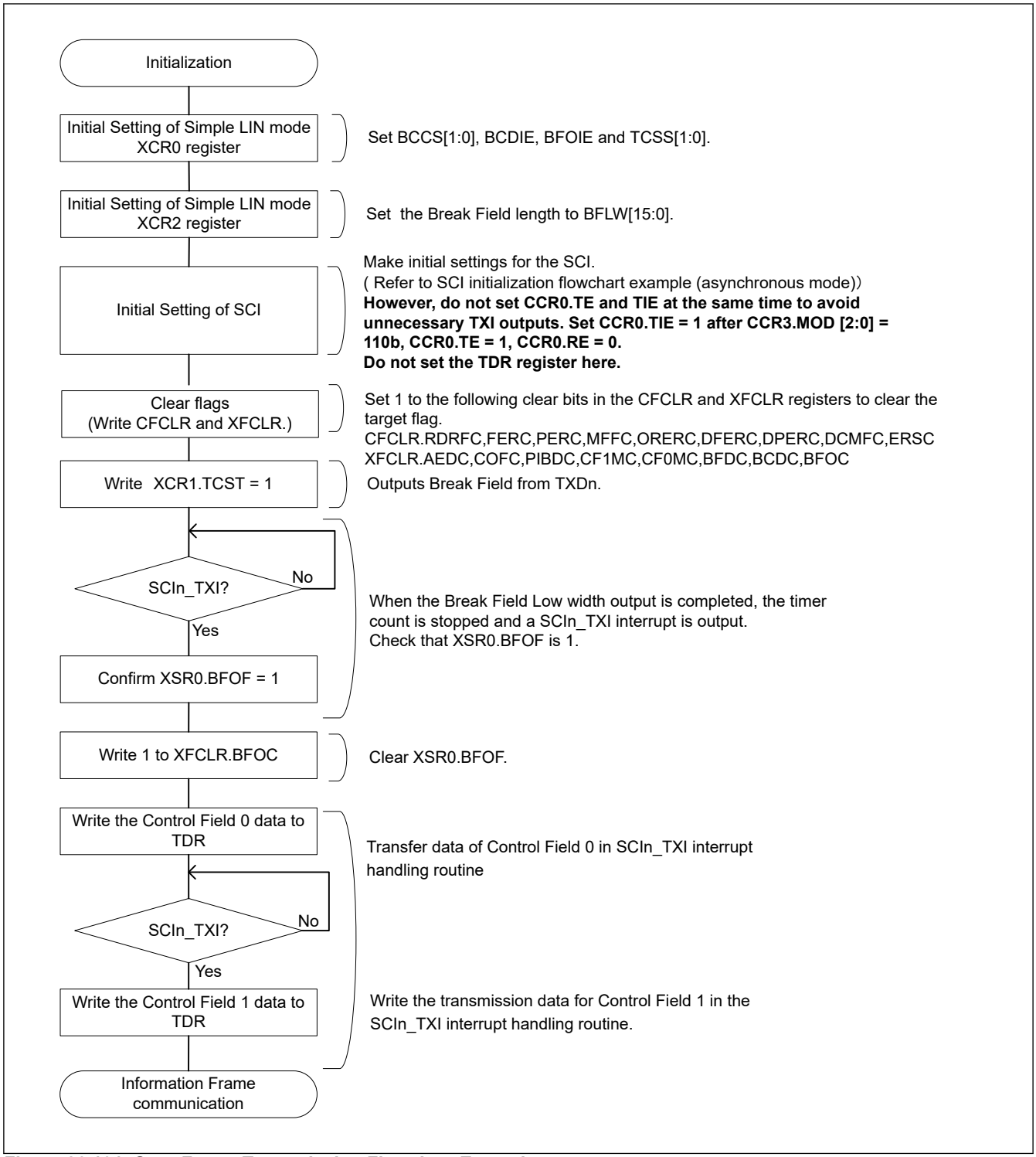


Figure 26.101 Start Frame Transmission Flowchart Example

26.11.2 Simple LIN Start Frame Reception

The SCI can detect Start Frames configured as shown in Figure 26.102.

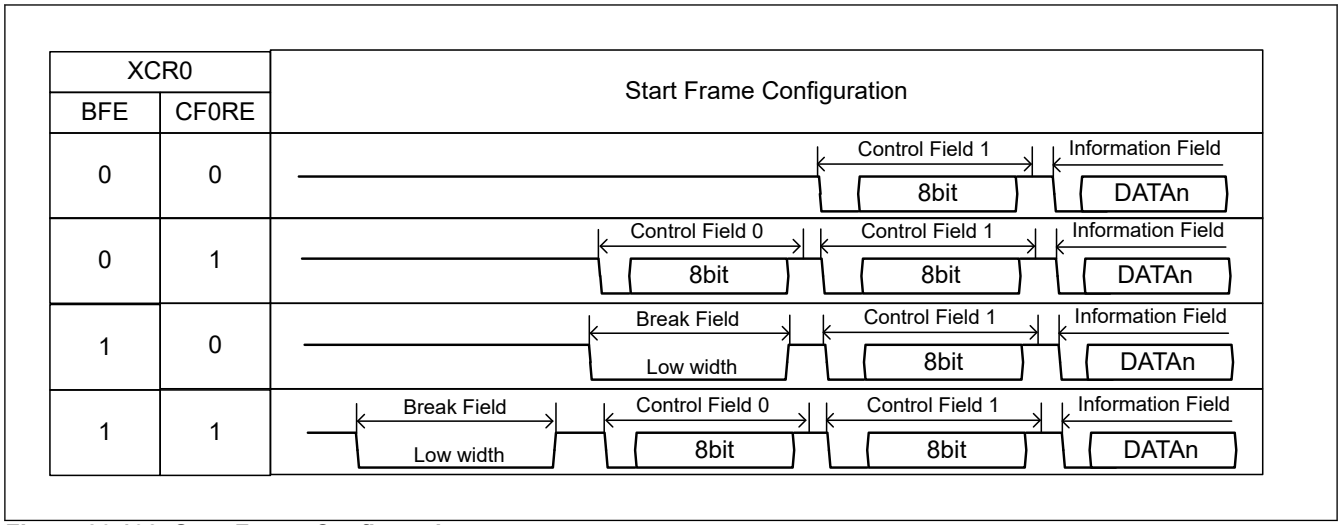


Figure 26.102 Start Frame Configuration

(1) Simple LIN normal reception of Start Frame (PIB not used)

Figure 26.103 shows an example of normal reception of the Start Frame consisting of a Break Field, Control Field0, and Control Field1. Figure 26.104 shows an example of reception to detect the Break Field during Control Field 1. Figure 26.105 shows a flowchart to receive the Start Frame, and Figure 26.106 shows a state transition diagram.

When receiving the Start Frame, the SCI operates as follows. Omit the processing of Break Field and Control Field0 according to the Start Frame configuration.

1. Writing 1 to XCR1.SDST makes it possible to detect the Start Frame. When XCR0.BFE = 1, RXDn input to the SCI core is disabled until the Break Field is detected (because XSR0.RXDSF is set to 1). Once the Break Field is detected, RXDn input can be received to the SCI core (XSR0.RXDSF = 0).
2. When a low level is input from the RXDn pin, the Break Field detection count starts. A timer count clock source can be selected by XCR0.TCSS[1:0].
3. When a low level that is equal to or longer than the period set in XCR2.BFLW[15:0] is input from the RXDn pin, it is determined as Break Field. At this time, XSR0.BFDF is set to 1. If XCR0.BFDIE has been set to 1 at this time, a SCIn_BFD interrupt is generated.
   The timer continues counting until the RXDn rising edge or counter overflow.
4. After the Break Field is detected, when the input level from the RXDn pin becomes high, the count value is captured to XSR1.TCNT[15:0] when BMEN = 0. At this time, XSR0.RXDSF is cleared to 0 and the SCI core starts receiving the RXDn input.
5. The SCI core starts receiving Control Field 0. Because the simple LIN continuously counts the edge interval, it determines a low level that is equal to or longer than the period set in XCR2.BFLW[15:0] as detection of the Break Field. When the Break Field is detected in the Control Field 0 phase, the SCI core waits for reception of Control Field 0 again (Figure 26.104).
6. When Control Field 0 has been received, an SCIn_RXI interrupt is generated and the Control Field 0 data is stored in XSR0.CF0RD[7:0]. When the received data matches the set XCR2.CF0D[7:0] value, XSR0.CF0MF is set to 1. If the received data differs from the set XCR2.CF0D[7:0] value, the SCI transitions to the state before the Break Field is detected.
7. The SCI core starts receiving Control Field 1. When BFE = 1, the Break Field detection function is continuously enabled while SDST = 1 as in the case of Control Field 0. When the Break Field is detected in the Control Field 1 phase, the SCI core waits for reception of Control Field 0 again.
8. When Control Field 1 has been received, an SCIn_RXI interrupt is generated and the Control Field 1 data is stored in XSR0.CF1RD[7:0]. When the received data matches the set XCR1.PCF1D[7:0] value or the set XCR1.SCF1D[7:0] value, XSR0.CF1MF is set to 1. If the received Control Field 1 data matches neither the set XCR1.PCF1D[7:0] value nor the set XCR1.SCF1D[7:0] value, the SCI transitions to the state before the Break Field is detected.
9. The SCI core performs Information Frame communication.
10. When communication is completed, write 0 to XCR1.SDST and 0 to CCR0.RE to stop reception.

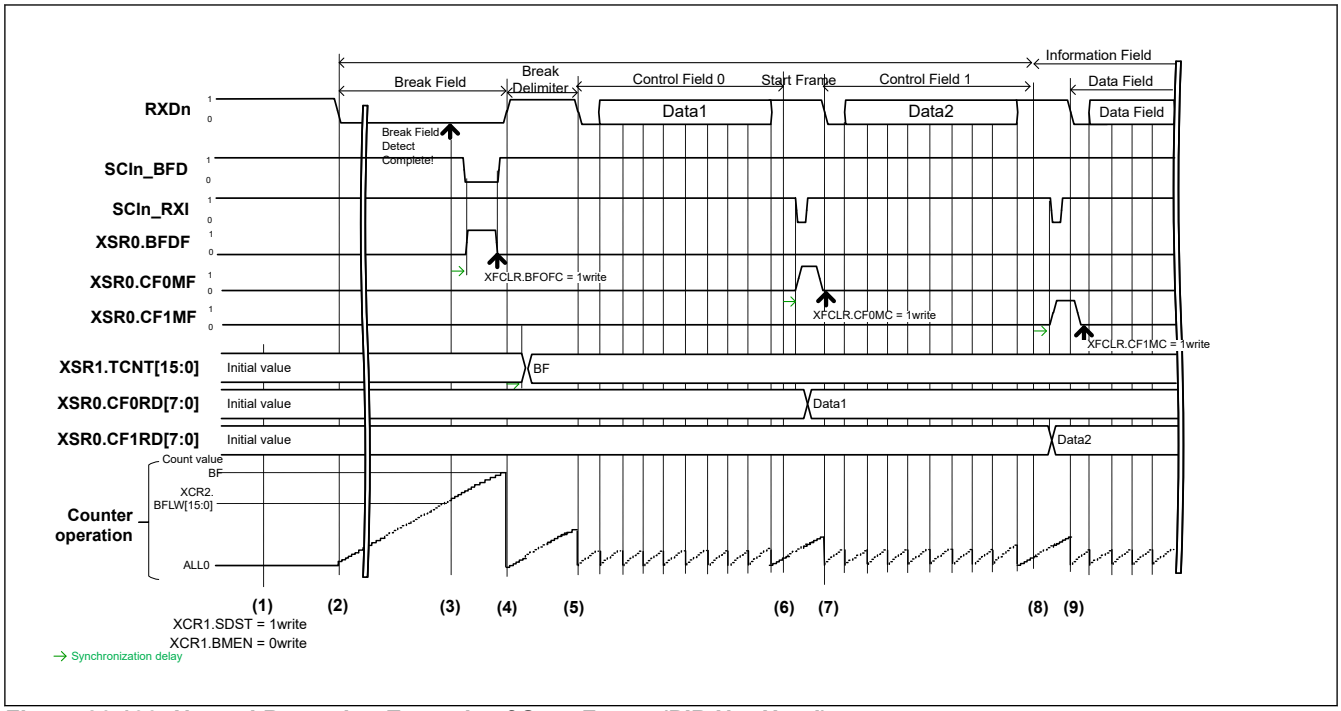


Figure 26.103 Normal Reception Example of Start Frame (PIB Not Used)

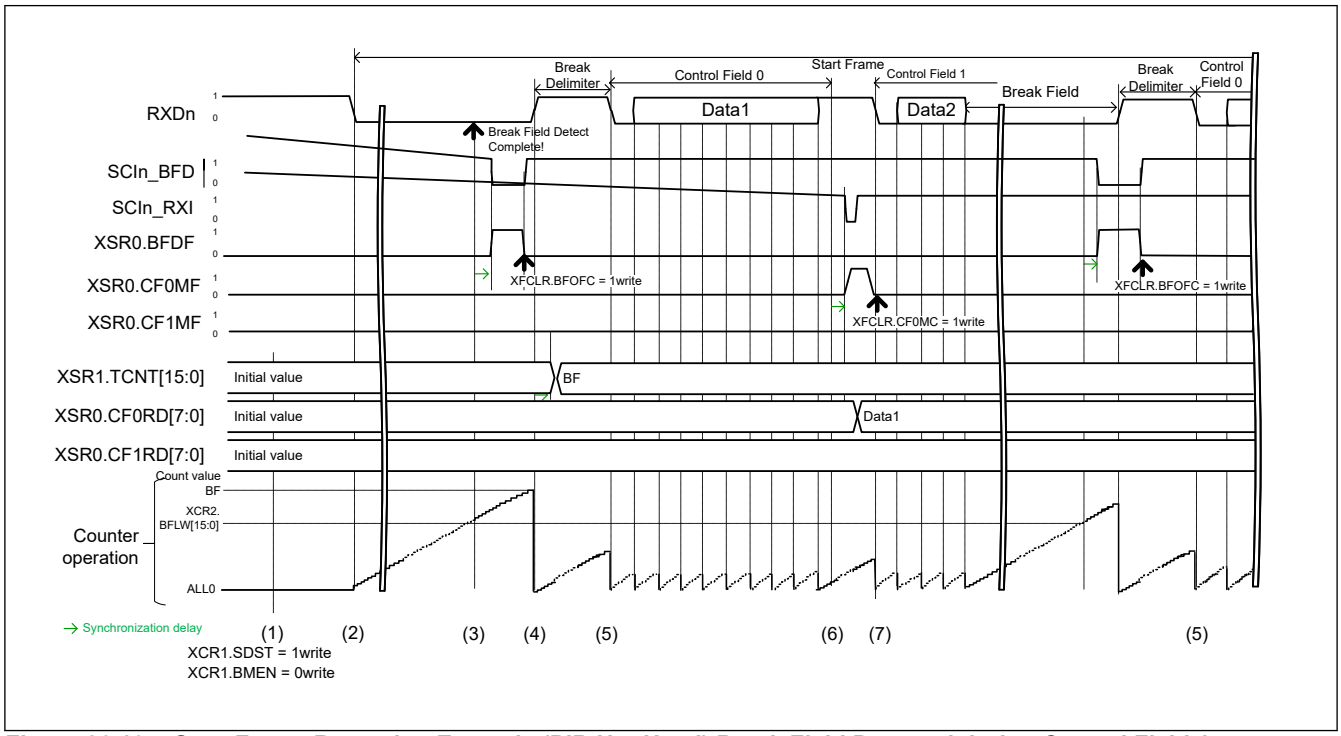


Figure 26.104 Start Frame Reception Example (PIB Not Used) Break Field Detected during Control Field 1

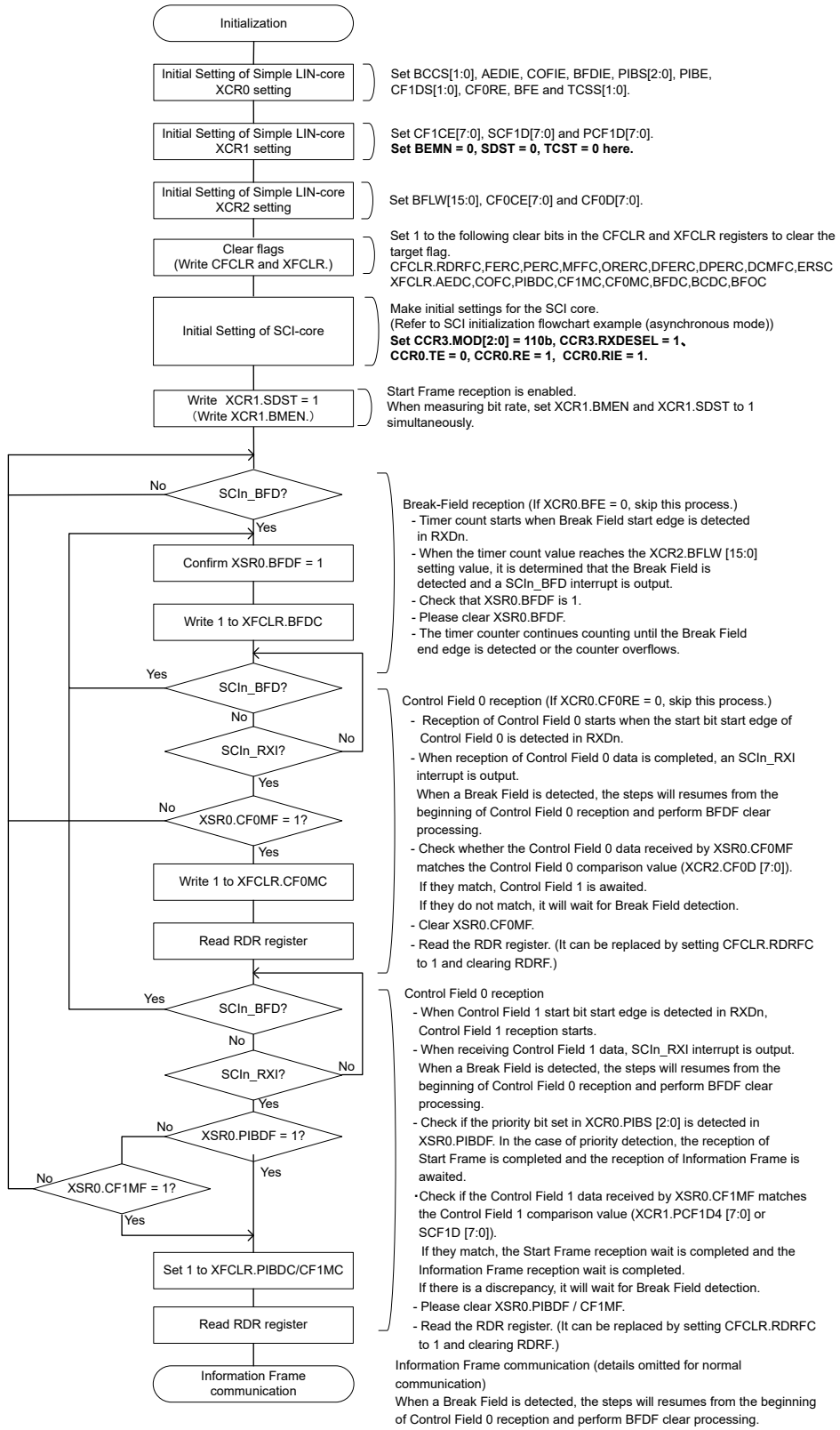


Figure 26.105 Example of Start Frame Reception Flowchart

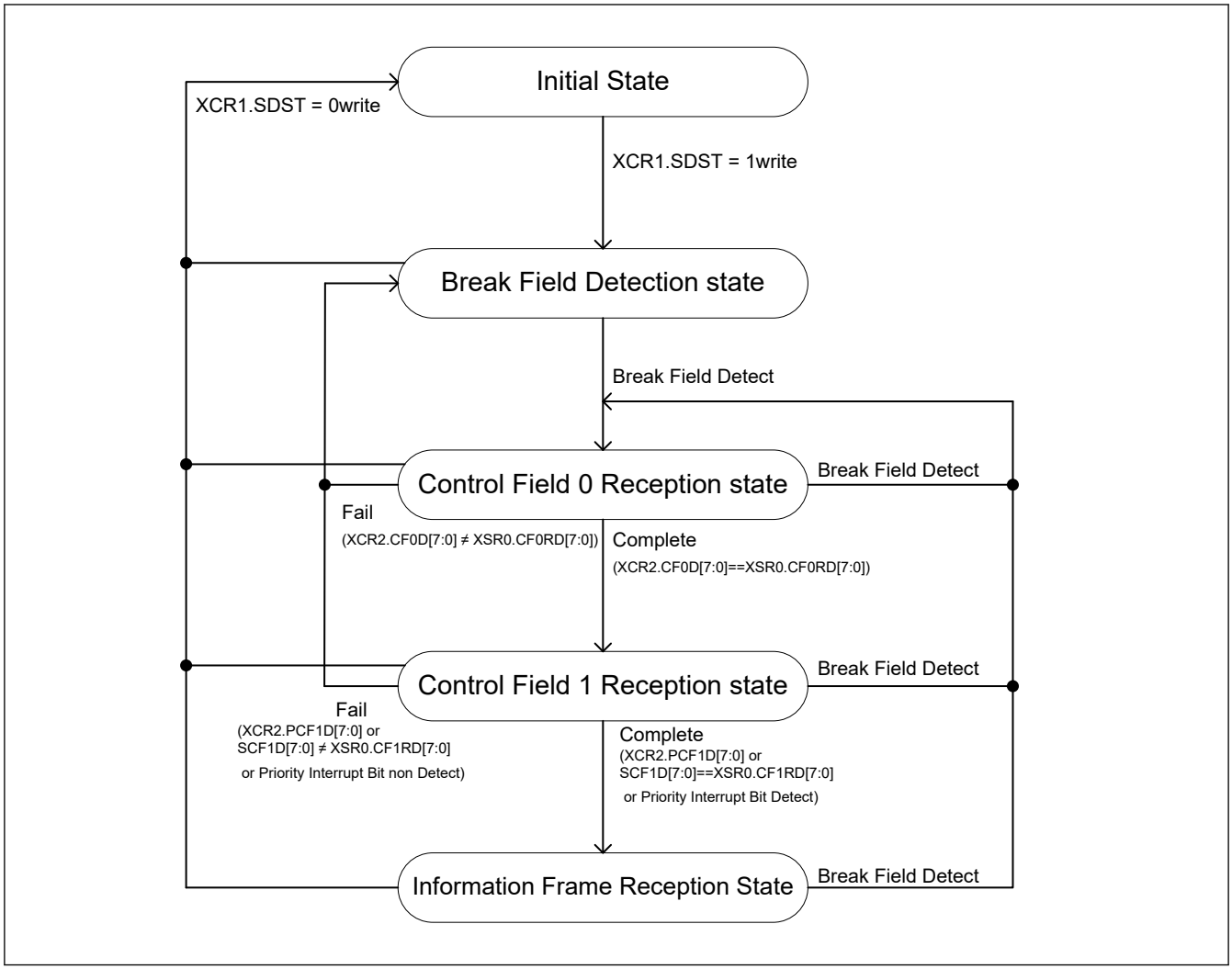


Figure 26.106 State Transition Diagram of Start Frame Reception

(2) Simple LIN Start Frame reception (using the priority interrupt bit)

Figure 26.107 shows an example of Start Frame reception using the priority interrupt bit. The priority interrupt bit is enabled by setting XCR0.PIBE to 1.

The SCI operates as follows during Start Frame reception using the priority interrupt bit.

Steps (1) to (7) are the same as steps (1) to (7) in the Start Frame reception example in Figure 26.103.

(8) When the value specified in the XCR0.PIBS[2:0] bits matches the set XCR1.PCF1D[7:0] value, XSR0.PIBDF is set to 1 and the SCI core performs communication of the Information Frame. If the data received in Control Field 1 matches neither the set XCR1.PCF1D[7:0] value nor the set XCR1.SCF1D[7:0] value and the priority interrupt bit is not detected, the SCI transitions to the state before the Break Field is detected.

(9) Communicate information frame at the SCI core.

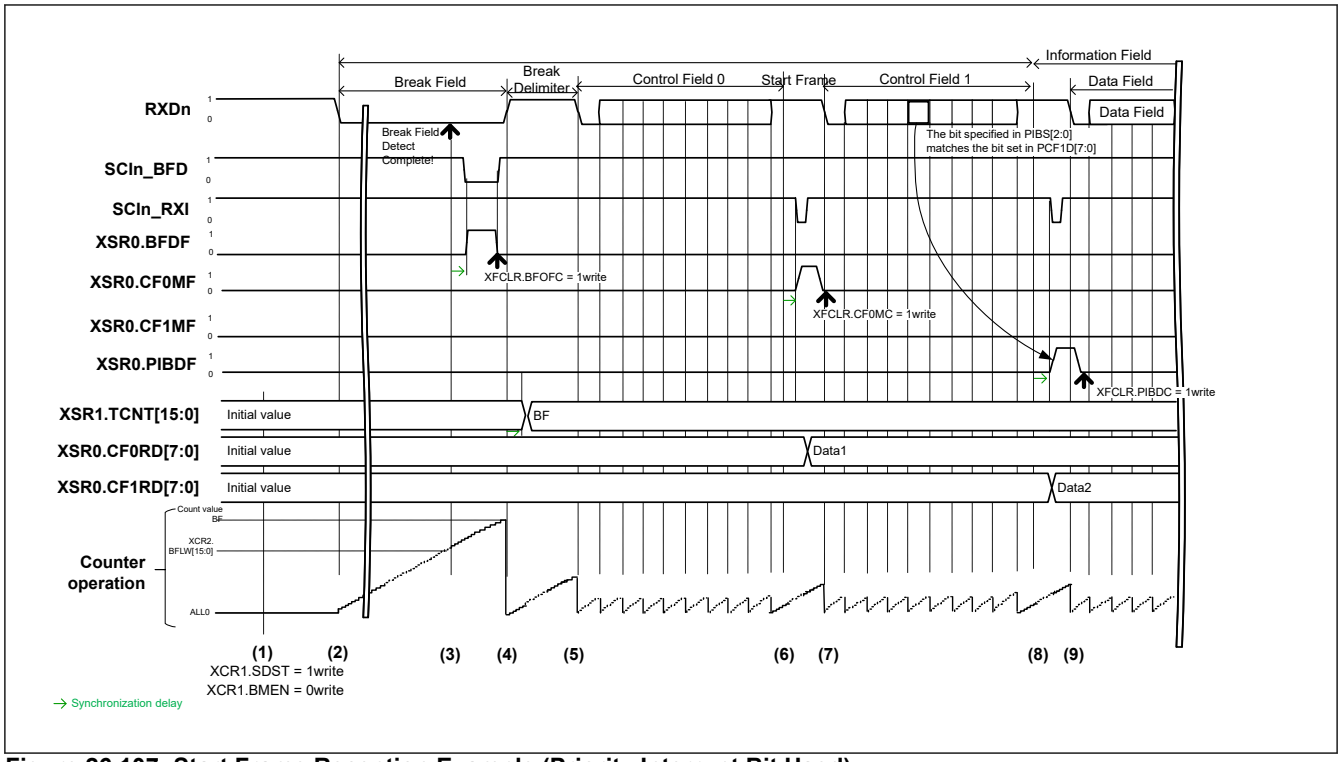


Figure 26.107 Start Frame Reception Example (Priority Interrupt Bit Used)

26.11.3 Simple LIN Bus Conflict Detection Function

In Simple LIN mode (CCR3MOD[2:0] = 110) when TE = 1, the bus conflict detection function works during Break Field output and during data transmission.

Figure 26.108 shows an operation example of the bus conflict detection function. The TXDn pin output and the RXDn pin input are sampled by the bus conflict detection clock set in XCR0.BCCS[1:0]. When a mismatch occurs three times in a row, XSR0.BCDF is set to 1, and if XCR0.BCDIE has been set to 1 at this time, an SCIn_ERI interrupt is generated.

When an SCIn_ERI interrupt is generated, stop transmission according to Figure 26.109. Check the bus state to decide whether to resume transmission.

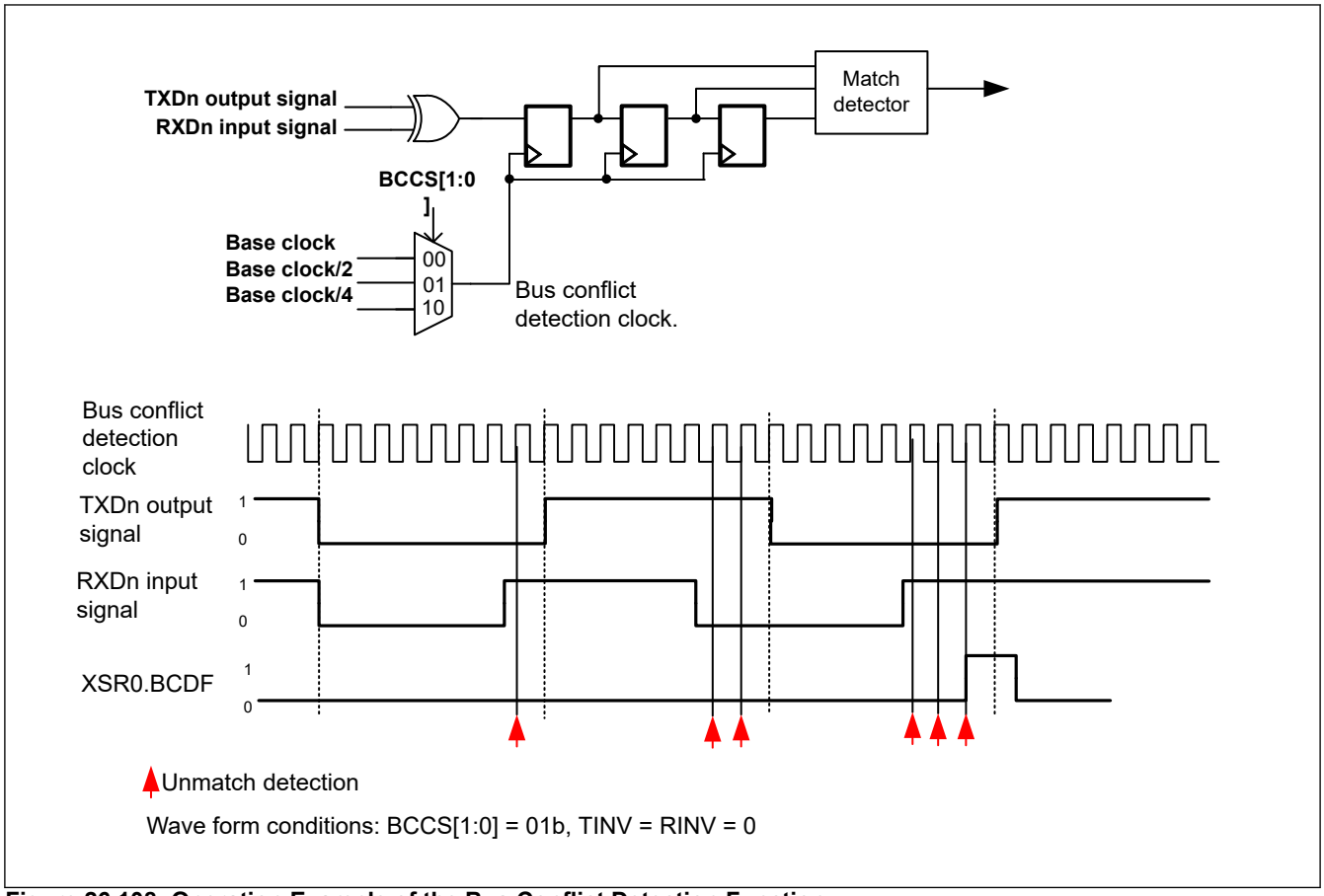


Figure 26.108 Operation Example of the Bus Conflict Detection Function

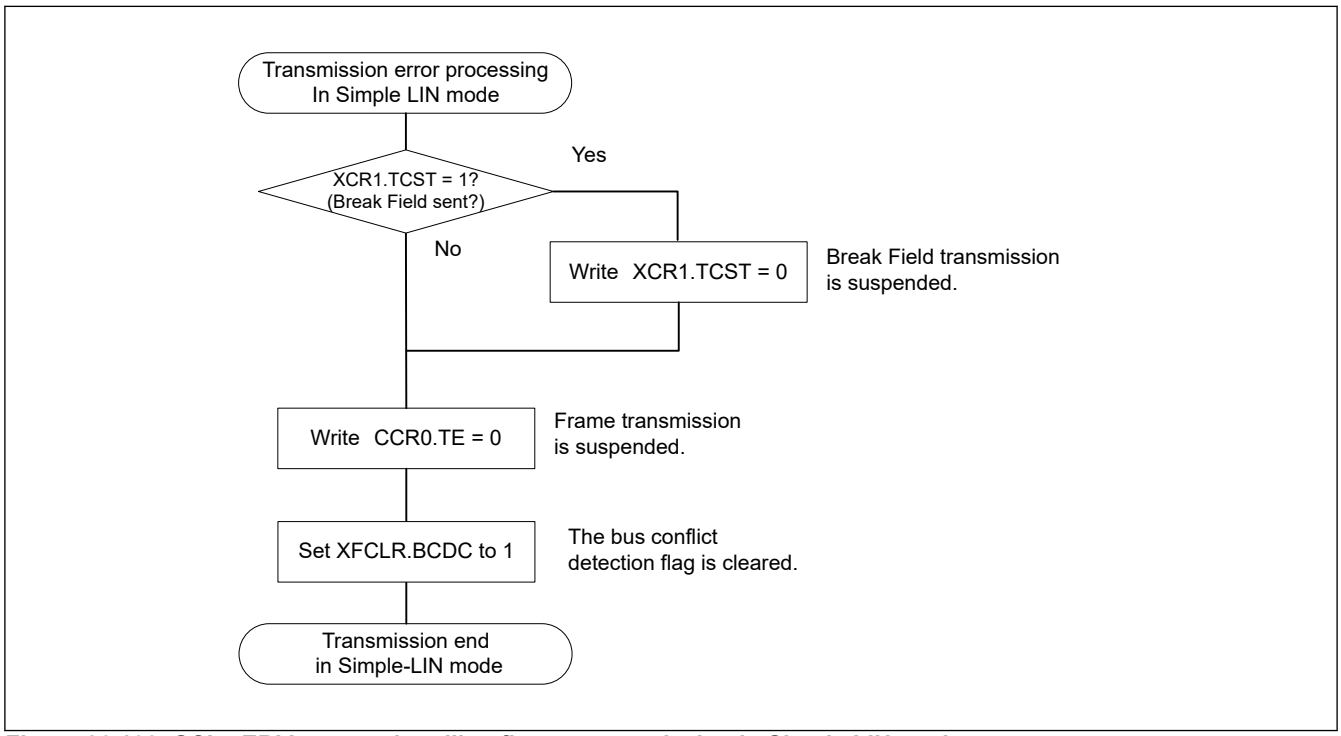


Figure 26.109 SCIn_ERI Interrupt handling flow at transmission in Simple LIN mode

26.11.4 Simple LIN Bit Rate Measurement Function

This function measures a bit rate between the effective edges of the input signal from the RXDn pin. Figure 26.110 shows an operation example of the bit rate measurement function.

1. Writing 1 to XCR1.SDST and XCR1.BMEN enables bit rate measurement. When this bit is set to 1, the valid edge interval of Control Field 0 and Control Field 1 data is measured. However, bit rate is not measured between the Break Field and the Break Delimiter. Set XCR1.BMEN and XCR1.SDST to 1 simultaneously, only when measuring bit rate.
2. Because bit rate is not measured in the Break Field, the effective edge detection flag is not set to 1 at the rising edge at the end of the Break Field, and the counter capture value is not stored in XSR1.TCNT[15:0].
3. The counter starts counting from the falling edge of the start bit in Control Field 0. The Break Delimiter count value is not captured in XSR1.TCNT[15:0].
4. The rising edge of the start bit is detected as an effective edge, and then the XSR0.AEDF flag is set to 1. If XCR0.AEDIE has been set to 1 at this time, an SCIn_AED interrupt is output. The start bit count value is stored in XSR1.TCNT[15:0]. The XSR1.TCNT[15:0] value is retained until the effective capture value is read.
5. Even if an effective edge is input from the RXDn input pin, the count value of this effective edge timing is not captured because the XRS1.TCNT[15:0] value has not been read and retention has not been released. In this case, an SCIn_AED interrupt is not output.
6. The XSR1.TCNT[15:0] value is read. Then the retention of XSR1.TCNT[15:0] is released and the XSR0.AEDF flag is cleared by hardware.
7. Because the retention of XSR1.TCNT[15:0] has been released, the count value is captured at the effective edge and is retained. At the same time, the XSR0.AEDF flag is set to 1, and if XCR0.AEDIE has been set to 1, an SCIn_AED interrupt is output. The bit rate can be adjusted by calculating it from the count value between effective edges by software and by changing the SCI settings.
8. To disable bit rate measurement, write 0 to XCR1.BMEN.
9. The XSR0.AEDF value and the XSR1.TCNT[15:0] value remain unchanged at the effective edge timing because the bit rate measurement function is disabled.

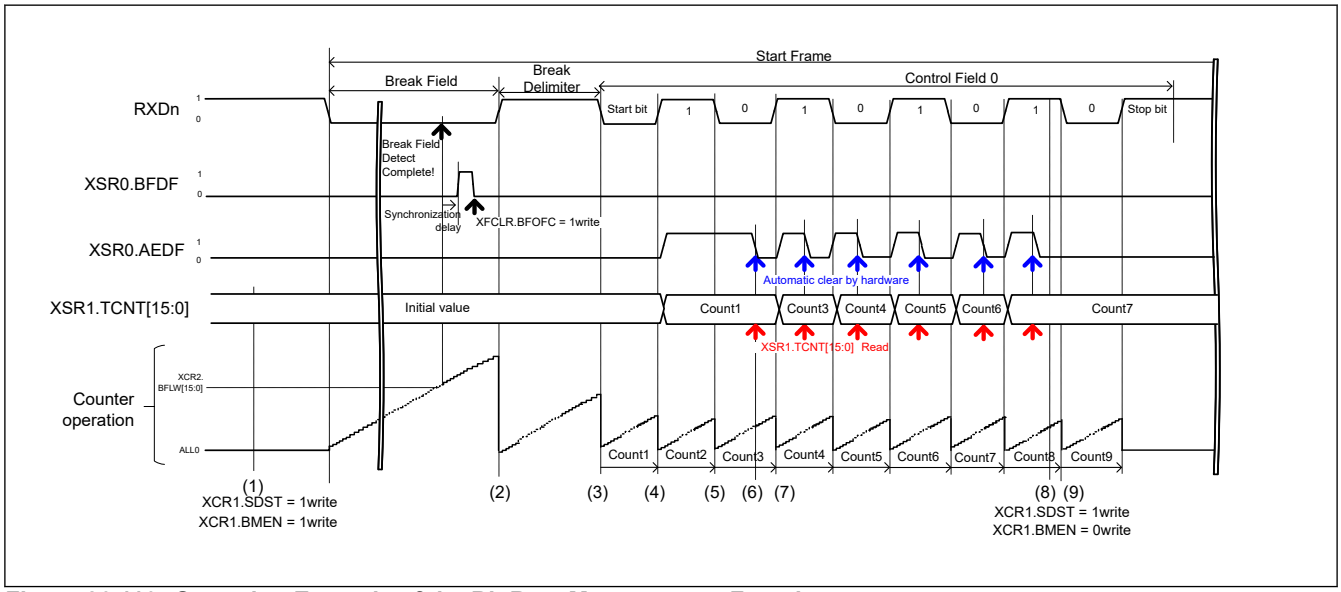


Figure 26.110 Operation Example of the Bit Rate Measurement Function

26.12 Interrupt Sources

26.12.1 Buffer Operation for SCIn_TXI and SCIn_RXI Interrupts

If the conditions for an SCIn_TXI and SCIn_RXI interrupt are satisfied while the interrupt status flag in the ICU is 1, the ICU does not output the interrupt request but saves it internally with a capacity for retention of one request per source.

26.12.2 Interrupts in Asynchronous, Manchester, Clock Synchronous, and Simple SPI Modes

(1) Non-FIFO selected

Table 26.40 lists interrupt sources in asynchronous mode, Manchester mode, clock synchronous mode, and simple SPI mode.

A different interrupt vector can be assigned to each interrupt source. Individual interrupt sources can be enabled or disabled with the enable bits in the CCR0 register.

If the CCR0.TIE bit is 1, an SCIn_TXI interrupt request is generated when transmit data is transferred from the TDR register to the TSR register. An SCIn_TXI interrupt request can also be generated by using a single instruction to set the CCR0.TE and CCR0.TIE bits to 1 at the same time. An SCIn_TXI interrupt request can activate the DTC or DMAC to handle data transfer.

An SCIn_TXI interrupt request is not generated by setting the CCR0.TE bit to 1 when CCR0.TIE is 0 or by setting the CCR0.TIE bit to 1 when the CCR0.TE is 1.*1

When new data is not written by the time of transmission of the last bit of the current transmit data and CCR0.TEIE is 1, the CSR.TEND flag is set to 1 and an SCIn_TEI interrupt request is generated. Additionally, when CCR0.TE is 1, the CSR.TEND flag retains the value 1 until more transmit data is written to the TDR register, and setting CCR0.TEIE to 1 leads to the generation of an SCIn_TEI interrupt request.

Writing data to the TDR register leads to clearing of the CSR.TEND flag and, after a certain time, discarding of the SCIn_TEI interrupt request.

If the CCR0.RIE bit is 1, an SCIn_RXI interrupt request is generated when received data is stored in the RDR register. An SCIn_RXI interrupt request can activate the DTC or DMAC to handle data transfer.

Setting any of the CSR.ORER, FER, PER or MSR.MER*2, SYER*2, PFER*2, and SBER*2 flags to 1 when the CCR0.RIE bit is 1 leads to the generation of an SCIn_ERI interrupt request.

An SCIn_RXI interrupt request is not generated in this case. Clearing all these flags (ORER, FER, PER, MER*2, SYER*2, PFER*2 and SBER*2) leads to discarding of the SCIn_ERI interrupt request.

Note 1. To temporarily prohibit SCIn_TXI interrupts on transmission of the last of the data when a new round of transmission is to be started, after handling the transmission-completed interrupt, control activation of the interrupt by using the interrupt request enable bit in the ICU rather than using the CCR0.TIE bit. This approach can prevent the suppression of SCIn_TXI interrupt requests in the transfer of new data.

Note 2. MER, SYER, PFER, and SBER work as a factor of SCIn_ERI interrupt only in Manchester mode. SYER, PFER, and SBER also only work if its enable bits (SYEREN, PFEREN, SBEREN in MECR) are set to "1".

(2) FIFO selected

Table 26.41 lists interrupt sources in FIFO selected mode.

If the CCR0.TIE bit is 1, an SCIn_TXI interrupt request is generated when the stored amount of data in the transmit-FIFO (TDR) register becomes the threshold value indicated in FCR.TTRG or below. An SCIn_TXI interrupt request can also be generated by using a single instruction to set the CCR0.TIE and CCR0.TE bits to 1 simultaneously or by setting CCR0.TIE to 1 when CCR0.TE is 1.

An SCIn_TXI interrupt request is not generated by setting CCR0.TE to 1 when CCR0.TIE is 0 or by setting the CCR0.TIE bit to 1 while the setting of the CCR0.TE bit is 1.

If CCR0.TEIE is 1 and if the next data is not written to the transmit-FIFO (TDR) register by the time the last bit of the transmit data is sent, the CSR.TEND flag is set to 1 and the SCIn_TEI interrupt request is generated.

If CCR0.RIE is 1, the SCIn_RXI interrupt request is generated when the stored amount of data in the transmit-FIFO (TDR) register is equal to or greater than the threshold value indicated in FCR.RTRG. When RTRG is 0, an SCIn_RXI interrupt does not occur even when the amount of data in the receive FIFO is equal to 0.

If the CCR0.RIE bit is 1, when the CSR.ORER flag is set to 1 or data with a framing error or a parity error is stored in the transmit-FIFO (TDR) register, the SCIn_ERI interrupt request is generated. When the amount of data stored in the transmit-FIFO (TDR) register is at the threshold value or above, the SCIn_RXI interrupt request is also generated. The SCIn_ERI interrupt request can be canceled, in which case CSR.ORER, FER, and PER flags are all cleared.

Table 26.40 SCI interrupt sources with non-FIFO selected

Name	Interrupt source	Interrupt flag	Interrupt enable	DTC or DMAC activation
SCIn_ERI (n = 0 to 4, 9)	Receive error	CSR.ORER, CSR.FER, CSR.PER, CSR.DFER, CSR.DPER, (MSR.MER, MSR.SYER, MSR.PFER, MSR.SBER)*1	CCR0.RIE	Not possible
SCIn_RXI (n = 0 to 4, 9)	Receive data full	CSR.RDRF	CCR0.RIE	Possible
	Address match	CSR.DCMF	CCR0.RIE	Possible
SCIn_TXI (n = 0 to 4, 9)	Transmit data empty	CSR.TDRE	CCR0.TIE	Possible
	TE = 0->1 detection			
SCIn_TEI (n = 0 to 4, 9)	Transmit end	CSR.TEND	CCR0.TEIE	Not possible

Note 1. MER, SYER, PFER, and SBER work as a factor of SCIn_ERI interrupt only in Manchester mode. SYER, PFER, and SBER also only work if its enable bits (SYEREN, PFEREN, SBEREN in MCR) are set to 1.

Table 26.41 SCI interrupt sources with FIFO selected

Name	Interrupt source	Interrupt flag	Interrupt enable	DTC or DMAC activation
SCIn_ERI (n = 0 to 4, 9)	Receive error	CSR.ORER, CSR.FER, CSR.PER, CSR.DFER, CSR.DPER	CCR0.RIE	Not possible
		FRSR.DR (when FCR.DRES = 1)	CCR0.RIE	Not possible
SCIn_RXI (n = 0 to 4, 9)	Receive data full	CSR.RDRF	CCR0.RIE	Possible
	Receive data ready	FRSR.DR (when FCR.DRES = 0)	CCR0.RIE	Possible
	Address match	CSR.DCMF	CCR0.RIE	Possible
SCIn_TXI (n = 0 to 4, 9)	Transmit data empty	CSR.TDFE	CCR0.TIE	Possible
	TE=0->1 detection			
SCIn_TEI (n = 0 to 4, 9)	Transmit end	CSR.TEND	CCR0.TEIE	Not possible

26.12.3 Interrupts in Smart Card Interface Mode

Table 26.42 lists interrupt sources in smart card interface mode. A transmit end interrupt (SCIn_TEI) request and an address match (SCIn_AM) request cannot be used in this mode.

Table 26.42 SCI Interrupt sources in Smart Card Interface Mode

Name	Interrupt source	Interrupt flag	Interrupt enable	DTC or DMAC activation
SCIn_ERI (n = 0 to 4, 9)	Receive error or error signal detection	CSR.ORER, CSR.PER, CSR.ERS	CCR0.RIE	Not possible
SCIn_RXI (n = 0 to 4, 9)	Receive data full	CSR.RDRF	CCR0.RIE	Possible
SCIn_TXI (n = 0 to 4, 9)	Transmit data empty	CSR.TEND	CCR0.TIE	Possible
	When set TE = 0->1			

Data transmission or reception using the DTC or DMAC is also possible in smart card interface mode, similar to normal SCI mode. In transmission, when the CCR0.TEND flag is set to 1, an SCIn_TXI interrupt request is generated. This SCIn_TXI interrupt request activates the DTC or DMAC, allowing transfer of transmit data if the SCIn_TXI request is previously specified as a source of DTC or DMAC activation. The TEND flag is automatically set to 0 when the DTC or DMAC transfers the data.

If an error occurs, the SCI automatically retransmits the same data. During the retransmission, the TEND flag is kept at 0 and the DTC or DMAC is not activated. Therefore, the SCI and DTC or DMAC automatically transmit the specified number of bytes, including retransmission after an error occurrence. However, the CSR.ERS flag is not automatically set to

0 at error occurrence. Therefore, the ERS flag must be cleared by previously setting the CCR0.RIE bit to 1 to enable an SCIn_ERI interrupt request to be generated at error occurrence.

When transmitting or receiving data using the DTC or DMAC, always enable the DTC or DMAC before making the SCI settings. For DTC or DMAC settings, see section 16, Data Transfer Controller (DTC), section 15, DMA Controller (DMAC).

In reception, an SCIn_RXI interrupt request is generated when receive data is set to the RDR register. This SCIn_RXI interrupt request activates the DTC or DMAC, allowing transfer of the receive data if the SCIn_RXI request is previously specified as a source of DTC or DMAC activation. If an error occurs, the error flag is set. Therefore, the DTC or DMAC is not activated and an SCIn_ERI interrupt request is issued to the CPU instead. The error flag must be cleared.

26.12.4 Interrupts in Simple IIC Mode

Table 26.43 lists the interrupt sources in simple IIC mode. The STI interrupt is allocated to the transmit end interrupt (SCIn_TEI) request. The receive error interrupt (SCIn_ERI) and the address match (SCIn_AM) request cannot be used.

The DTC or DMAC can also be used to handle transfer in simple IIC mode.

When the ICR.IICINTM bit is 1:

- An SCIn_RXI request is generated on the falling edge of the SCLn signal for the 8th bit. If SCIn_RXI is previously set up as an activation source for the DTC or DMAC, the SCIn_RXI request activates the DTC or DMAC to handle transfer of the received data.
- An SCIn_TXI request is generated on the falling edge of the SCLn signal for the 9th bit (acknowledge bit). If SCIn_TXI is previously set up as an activation source for the DTC or DMAC, the SCIn_TXI request activates the DTC or DMAC to handle transfer of the transmit data.

When the ICR.IICINTM bit is 0:

- An SCIn_RXI request (ACK detection) is generated if the input on the SDAn pin is low on the rising edge of the SCLn signal for the 9th bit (acknowledge bit)
- An SCIn_TXI request (NACK detection) is generated if the input on the SDAn pin is high on the rising edge of the SCLn signal for the 9th bit (acknowledge bit)
- If SCIn_RXI is previously set up as an activation source for the DTC or DMAC, the SCIn_RXI request activates the DTC or DMAC to handle transfer of the received data.

If the DTC or DMAC is used for data transfer in reception or transmission, always set up and enable the DTC or DMAC before setting up the SCI.

When the IICSTAREQ, IICRSTAREQ, and IICSTPREQ bits in ICR are used to generate a start condition, restart condition, or stop condition, the STI request is issued when generation is complete.

Table 26.43 SCI interrupt sources in Simple IIC Mode

Name	Interrupt source		Interrupt flag	Interrupt enable	DTC or DMAC activation
	ICR.IICINTM = 1	ICR.IICINTM = 0			
SCIn_RXI (n = 0 to 4, 9)	Reception end	—	—	CCR0.RIE	Possible*1
	—	ACK detection	—		Possible
SCIn_TXI (n = 0 to 4, 9)	Transmission end		—	CCR0.TIE	Possible*1
	—	NACK detection	—		Possible
SCIn_TEI(STIn) (n = 0 to 4, 9)	Completion of generation of a start, restart, or stop condition		ICR.IICSTIF	CCR0.TEIE	Not possible

Note 1. If the DMAC or DTC are being used, you cannot confirm whether ACK or NACK.

26.12.5 Interrupts in Simple LIN mode

Table 26.44 lists interrupt sources in Simple LIN mode.

Table 26.44 SCI interrupt sources in Simple LIN mode

Name	Interrupt Sources	Interrupt Flag	Flag the needs to be confirmed	Interrupt Enable	DTC/DMAC Activation
SCIn_ERI (n = 0 to 4, 9)	Receive error	CSR.ORER, CSR.FER, CSR.PER	—	CCR0.RIE	Not Possible
		XSR0.BCDF		XCR0.BCDIE	
		XSR0.COF		CCR0.RIE, XCR0.COFIE	
SCIn_RXI (n = 0 to 4, 9)	Receive data full flag	CSR.RDRF	XSR0.CF0MF XSR0.CF1MF XSR0.PIBDF	CCR0.RIE	XSR0.SFSF = 0: Possible XSR0.SFSF = 1: Not Possible
SCIn_AED (n = 0 to 4, 9)	Active edge detection	XSR0.AEDF	—	XCR0.AEDIE	Possible
SCIn_TXI (n = 0 to 4, 9)	Transmit data empty interrupt	CSR.TDRE	—	CCR0.TIE	Possible
	When set TE = 0->1				
	Break Field output completion	XSR0.BFOF		CCR0.TIE, XCR0.BFOIE	
SCIn_TEI (n = 0 to 4, 9)	Transmit end	CSR.TEND	—	CCR0.TEIE	Not Possible
SCIn_BFD (n = 0 to 4, 9)	Break Field Detection	XSR0.BFDF	—	XCR0.BFDIE	Not Possible (Unnecessary)

In Simple LIN mode, in addition to reception errors (ORER, FER, PER), an SCIn_ERI interrupt request is output when a bus conflict is detected during transmission, or when a counter overflow of the Simple LIN module occurs. At this time, a SCIn_RXI interrupt request is not output. The SCIn_ERI interrupt request can be canceled by clearing all the flags.

When transmitting Start Frame, if CCR0.TIE = 1 and XCR0.BFOIE = 1, a SCIn_TXI interrupt request is output when Break Field transmission is completed. When Control Field 0 data is written to the TDR register, data transmission starts. Therefore, transmission using DTC or DMAC is possible.

Set CCR0.TEIE = 1 after writing the last transmit data to the TDR register and transmission starts.

During Start Frame reception (XSR0.SFSF = 1), reception using DTC or DMAC by SCIn_RXI interrupt is not possible. Check the CSR register and XSR0 register, check the reception status (See Figure 26.81), and then clear the flag. Also read the RDR register (if you do not need to check the received data value, clear the RDRF flag without reading the RDR register). When reception of Control Field 1 is completed (XSR0.CF1MF = 1), Start Frame detection is disabled (XSR0.SFSF = 0) and reception using DTC or DMAC is possible. Be sure to read the RDR register.

When Start Frame / Break Field detection is enabled (XCR1.SDST = 1), if a Break Field longer than the period set in XCR2.BFLW [15:0] is received, the BFDF flag is set and a SCIn_BFD interrupt request is output. Then SCI becomes the Start Frame reception state. Clear the BFDF flag.

When Start Frame / Break Field detection is enabled (XCR1.SDST = 1) and the bit rate measurement function is enabled (XCR1.BMEN = 1), an SCIn_AED interrupt factor is output when an active edge is detected. Read the timer count capture value (XSR1.TCNT [15:0]).

26.13 Event Linking

By using interrupt request signals as event signals, the SCIn can provide linked operation through the ELC for modules selected in advance.

Event signals can be output regardless of the values of the associated interrupt request enable bits.

(1) Error event output (receive error or error signal detected) (SCIn_ERI, n = 0 to 4, 9)

- Indicates abnormal termination because of a parity error during reception in asynchronous mode
- Indicates abnormal termination because of a framing error during reception in asynchronous mode
- Indicates abnormal termination because of an overrun error during reception

- Indicates abnormal termination due to a Manchester error during reception (Only in Manchester mode).
- Indicates that a preface error occurred upon reception and abnormal termination occurred (only in Manchester mode and MCR.PFEREN = 1).
- Indicates that a start bit error occurred during reception and abnormal termination occurred (only in Manchester mode and MCR. SBEREN = 1).
- Indicates that a reception sync error occurred during reception and abnormal termination occurred (only in Manchester mode and only when MCR.SYEREN = 1).
- Indicates detection of the error signal during transmission in smart card interface mode
- The CSR.FER and PER flags are 0, and receive data less than the receive FIFO data trigger number is set in a reception FIFO buffer, and it indicates that 15 etus elapse when FIFO is selected and the FCR.DRES bit is 1
- In Simple LIN mode, indicates that the 16-bit counter in the Simple LIN module has overflowed.
- In Simple LIN mode, a bus collision is detected during transmission (CCR0.TE = 1).

(2) Receive data full event output (SCIn_RXI, n = 0 to 4, 9)

- Indicates that ACK is detected if the ICR.IICINTM bit is 0 in simple IIC mode
- Indicates that the 8th-bit SCLn falling edge is detected if the ICR.IICINTM bit is 1 in simple IIC mode
- When the ICR.IICINTM bit is 1 during master transmission in simple IIC mode, set the ELC so that receive data full events are not used

Non-FIFO selected

- Indicates that received data is set in the Receive Data Register (RDR).

FIFO selected

- Using this event output is prohibited.

(3) Transmit data empty event output (SCIn_TXI, n = 0 to 4, 9)

- Indicates that the CCR0.TE bit is changed from 0 to 1
- Indicates that transmission is complete in smart card interface mode
- Indicates that NACK is detected if the ICR.IICINTM bit is 0 in simple IIC mode
- Indicates that the 9th-bit SCLn falling edge is detected if the ICR.IICINTM bit is 1 in simple IIC mode
- In Simple LIN mode, indicates that Break Field output is complete.

Non-FIFO selected

- Indicates that transmit data is transferred from the Transmit Data Register (TDR) to the Transmit Shift Register (TSR).

FIFO selected

- Using this event output is prohibited.

(4) Transmit end event output (SCIn_TEI, n = 0 to 4, 9)

- Indicates the completion of transmission
- Indicates that the starting condition, resumption condition, or termination condition is generated in simple IIC mode
- In Smart Card mode, the transmit end event is not output.

Note: When FIFO is selected, using this event output is prohibited

(5) Address match event output (SCIn_AM, n = 0 to 4, 9)

- Indicates a match of the comparison data (CCR4.CMPD) with one frame of receive data when CCR0.DCME is set to 1 in asynchronous mode, including multi-processor mode.

(6) Active edge detection event output

- In Simple LIN mode, when CCR1.BMEN is 1, it indicates that a valid edge has been detected in the RXD input signal.

26.14 Address Non-match Event Output (SCI0_DCUF)

SCI0_DCUF indicates the non-match of comparison data (CCR4.CMPD) with receive data that is one frame of the data that is received when CCR0.DCME is set to 1 in asynchronous mode, including multi-processor mode. This event can be used for Snooze end request only. For details, see section 10, Low Power Modes.

26.15 Noise Cancellation Function

Figure 26.111 shows the configuration of the noise filter used for noise cancellation. The noise filter consists of a 2-stage flip-flop circuit and a match detection circuit. When the input signals of the noise filter and the output signals of the 2- stage flip-flop circuits completely match, the matched level is conveyed as an internal signal. Unless otherwise matched, the previous value is retained. When the same level is retained for 3 cycles or longer on the sampling clock of the noise filter, it is considered as a valid receive signal. A change in pulse for 3 cycles or shorter is considered as noise, not as a receive signal.

In asynchronous mode, Manchester and Simple LIN modes, the noise cancellation function can be applied to the receive signal input to the RXDn pin. The sampling period of the noise filter can be selected from the base clock period and the divided clock of the baud rate generator clock source by CCR1.NFCS[2:0].

- When CCR1.NFCS[2:0] = 000b, CCR2.ABCS = 0 and CCR2.ABCSE = 0, the cycle is 1/16 of a 1-bit period.
- When CCR1.NFCS[2:0] = 000b, CCR2.ABCS = 1 and CCR2.ABCSE = 0, the cycle is 1/8 of a 1-bit period.
- When CCR1.NFCS[2:0] = 000b, CCR2.ABCSE = 1, the cycle is 1/6 of a 1-bit period.

In simple IIC mode, this function can be used for each input on SDAn and SCLn. The sampling clock is selected from divided clock of baud rate generator settings by CCR1.NFCS[2:0].

If the base clock is stopped once with the noise filter enabled and then the base clock input is restarted again, the noise filter operation resumes from the state where the clock was stopped. When CCR0.TE and CCR0.RE are set to 0 during base clock input, all of the noise filter flip-flop values are initialized to 1. Accordingly, if the input data is 1 when reception operation resumes, the function determines that a level match is detected and the result is conveyed as an internal signal. When the level being input corresponds to 0, the initial output of the noise filter is retained until the level matches in three consecutive sampling cycles.

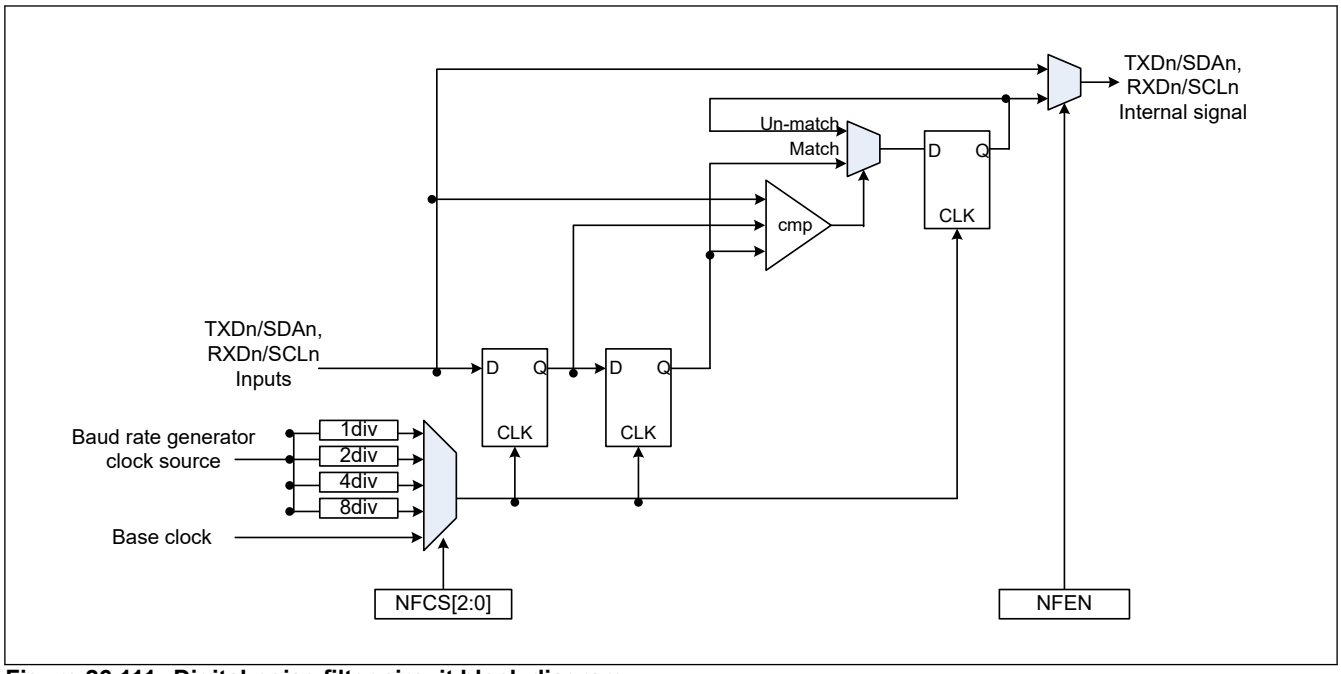


Figure 26.111 Digital noise filter circuit block diagram

26.16 RS-485 Driver Control Function

Setting the DEN bit in the SCI common control register3 (CCR3) to 1 enables the RS-485 driver control function and generates a DEn (Driver Enable) signal that enables the external transceiver transmission mode. The DEn signal outputs a valid level for the period with driver assertion time and driver negate time added before and after data transmission. The DEn signal valid level is set by the DEPOL bit in the driver control register (DCR).

The driver assertion time is the time from when the DEn signal is valid until the start bit starts. Set by DEAST [4:0] of driver control register (DCR).

The driver negate time is the time from the end of the last stop bit of the transmitted message to the invalidation of the DEn signal. Set with DENGT [4:0] of the driver control register (DCR).

DEAST and DENGT are expressed in base clock period (1/8 or 1/16 bit period). For details, see section 26.2.13. DCR : Driver Control Register.

When this function is used (CCR3.DEN = 1), the CSR.TEND set timing and SCIn_TEI interrupt output timing are at the end of the driver negation time.

When transmission is completed and the next transmission data is not written before the DEn signal is negated, the DEn signal is negated once. If the timing for writing the next transmit data is not in time, assert the DEn signal after negating it again, insert the driver assertion time, and transmit the next data. If you want to perform the next transmission with the DEn signal asserted, write the next transmission data to the TDR quickly enough in consideration of the synchronization delay time of the register.

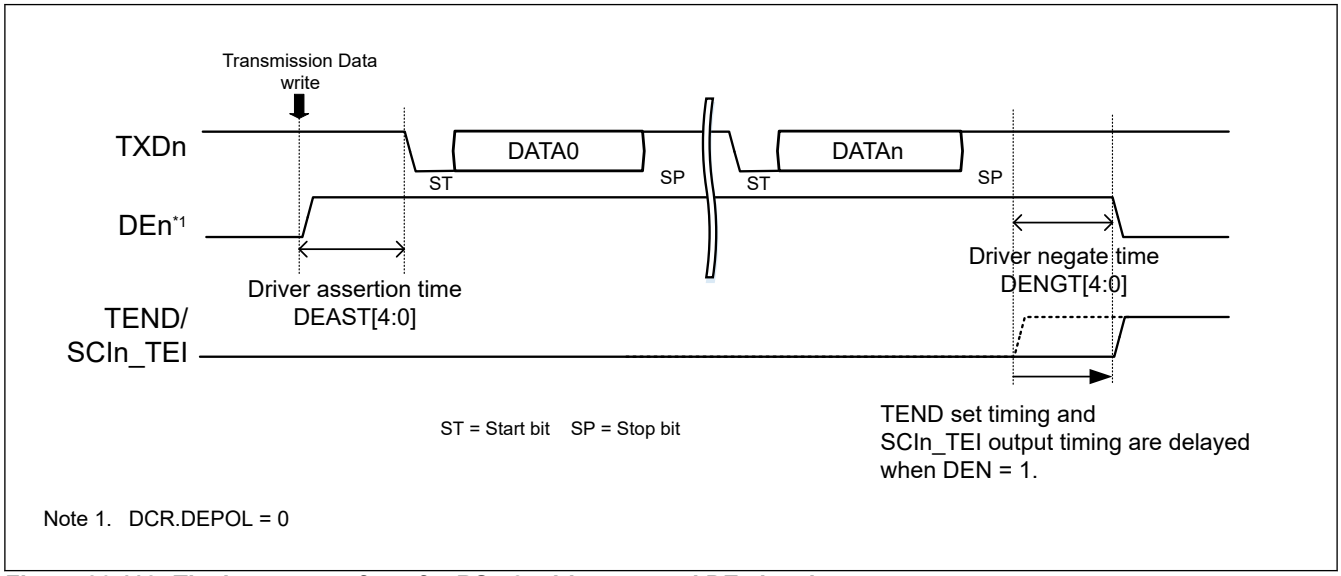


Figure 26.112 The image waveform for RS-485 driver control DE signal output

26.17 Loopback Function

The loopback function can be used in Asynchronous mode with the internal clock, and Manchester mode with the internal clock, and Clock synchronous mode with the internal clock.

When 1 is written to the SPLP bit in the CCR1 register, SCI blocks the external input (RXDn) path and connects the output path of the transmit data register and the input path of the receive data register.

When this function is used with TINV bit = 1, inversion of transmission data becomes reception data. However, this function can be used with TINV = 1 only when operating in clock synchronous mode internal clock.

Table 26.45 shows the relationship between the TINV and SPLP bit settings and the received data.

Table 26.45 TINV and SPLP bit settings and received data

CCR1.TINV	CCR1.SPLP	Receive Data	Communication mode		
			Asynchronous	Manchester	Clock synchronous
			internal clock	internal clock	internal clock
—	0	Receive Data from RXDn pin	Possible	Possible	Possible
0	1	Transmit Data	Possible	Possible	Possible
1	1	Inverted transmit data	Impossible	Impossible	Possible

Note: —: don't care

Figure 26.113 shows the configuration of the shift register input / output path in loopback mode.

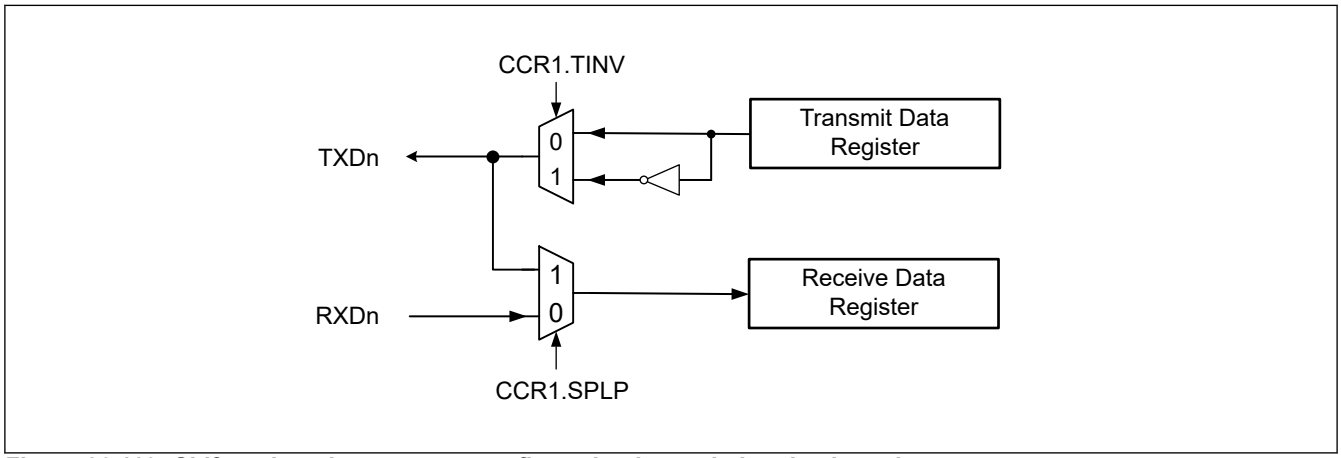


Figure 26.113 Shift register input output configuration image in loopback mode

26.18 Half-Duplex communication Function

Do not use the half-duplex communication function in Simple IIC, Simple SPI and Smart Card Interface modes.

In other communication modes, if the CCR1.SHARPS bit is set to 1, half-duplex communication using the TXDn pin is possible. When half-duplex communication is used, transmission and reception must be performed exclusively. Transmission and reception settings (CCR0.TE = 1 and CCR0.RE = 1) is prohibited.

However, if half-duplex communication is performed as the master reception in clock synchronous mode, perform transmission / reception settings (CCR0.TE = 1 and CCR0.RE = 1) and perform dummy transmission. By dummy transmission (arbitrary transmission data is written to TDR), SCKn is output and reception is enabled. The dummy transmission data is discarded inside the IP and is not actually transmitted.

During half-duplex communication, only the TXDn pin is used as the communication pin. Output when CCR0.TE = 1, input when CCR0.TE = 0.

26.19 Synchronizer Bypass Function

The SCI has a bus clock and the operation clock (TCLK). And these have each operating circuit. Therefore, there is a synchronization circuit for signal transfer between different clocks, and synchronization delay time is required for signal propagation between different clocks.

However, the synchronization circuit can be bypassed by the CCR3.BPEN bit only when the same clock is input to the bus clock and the operation clock. In this case, eliminates synchronization delay time and improves responsiveness. Figure 26.114 shows the image waveform of the bypass function.

This IP also has a synchronization circuit between the communication clock (SCKn) and the operation clock (TCLK), but this synchronization circuit cannot be bypassed.

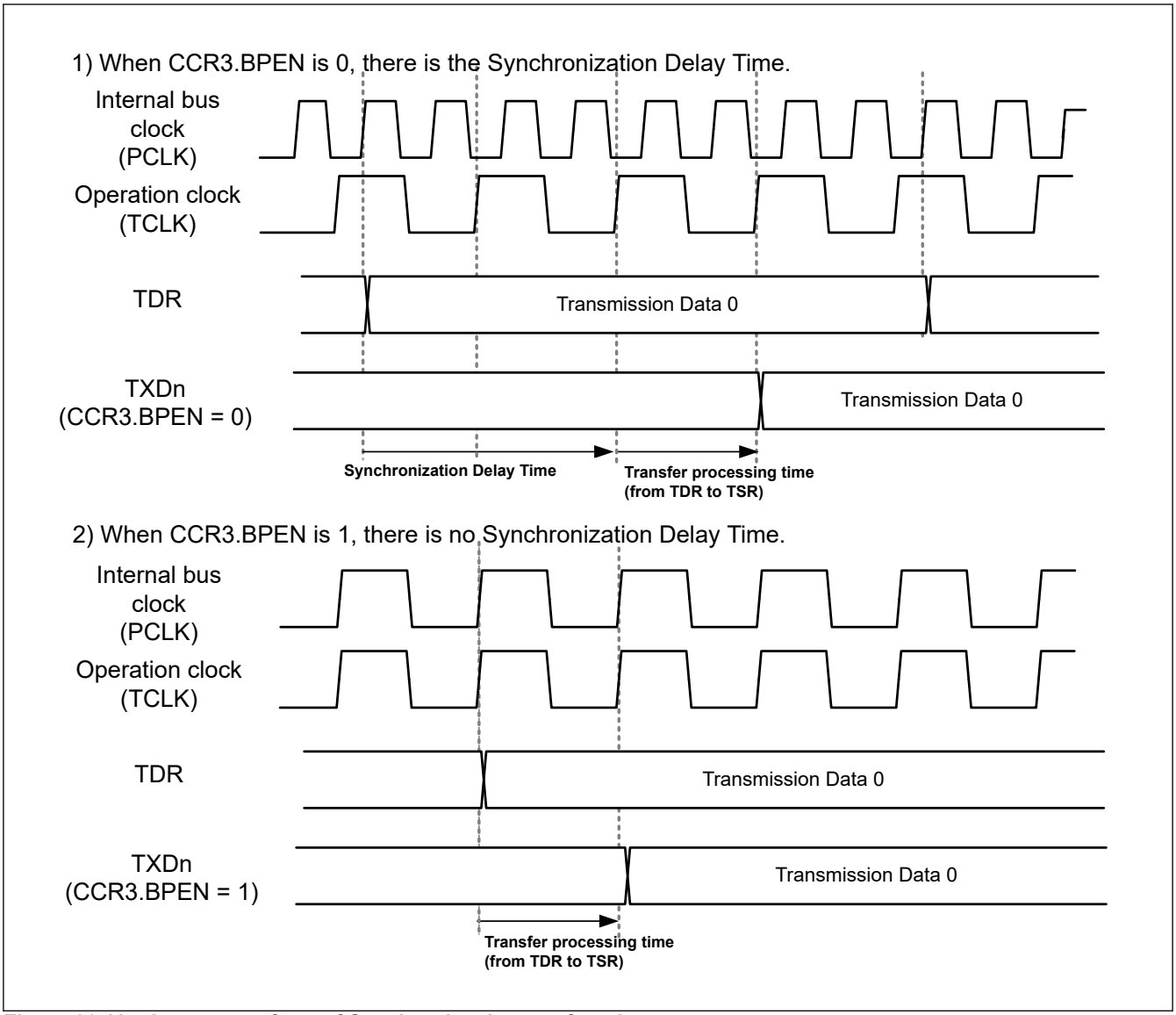


Figure 26.114 Image waveform of Synchronizer bypass function

26.20 Usage Notes

26.20.1 Settings for the Module-Stop Function

The Module Stop Control Register B (MSTPCRB) can enable or disable SCI operation. The SCI is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 10, Low Power Modes.

26.20.2 SCI Operation during Low Power State

(1) Transmission

Before using the power consumption reduction function to reduce SCI's power consumption, do the following to confirm transmission end (CSR.TEND = 1):

- Set the output pin state after transmission operation is stopped by CCR1.SPB2DT, SPB2IO.
- Stop the transmission (CCR0.TIE = 0, TE = 0, TEIE = 0)

When transitions to these states are made during transmission, the data being transmitted become indeterminate.

To transmit data in the same transmission mode after cancellation of the low power consumption state, set the TE bit to 1, read CSR, and write data to TDR sequentially to start data transmission. To transmit data with a different transmission mode, initialize the SCI first.

To start transmission using the DMAC/DTC after cancellation from software standby mode, set the CCR0.TE and CCR0.TIE bit to at the same time. Then SCIn_TXI interrupt flag is generated, which causes the DMAC/DTC to write the transmit data, which starts transmission.

Figure 26.115 shows a sample flowchart for transition to software standby mode during transmission. Figure 26.116 and Figure 26.117 show the port pin states during transition to software standby mode.

(2) Reception

When Address match function is non used as condition of resumption (wake-up)

Before specifying the module stop state or making a transition to software standby mode, stop the receive operations (CCR0.RE = 0). If transition is made during data reception, the data being received will be invalid.

Figure 26.118 shows a sample flowchart for reception to software standby mode during reception.

When Address match function is used as condition of resumption (wake-up)

When using the power consumption reduction function to reduce SCI's power consumption, do the following:

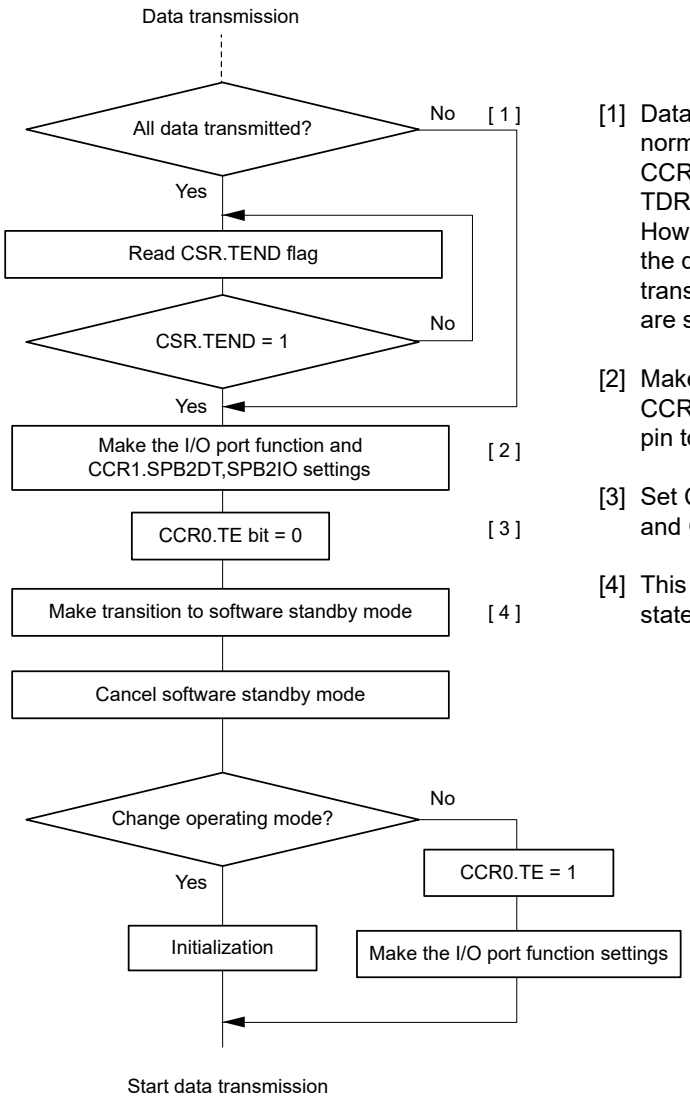
- Set the operation mode to asynchronous mode, after released
- Set the compare data to CCR4.CMPD and set 1 to CCR0.DCME
- Set the receive operation (CCR0.RE = 1)

Set CCR3.RXDESEL = 0. Because there is a possibility that a start bit (fall edge of RXDn pin) cannot be detected at the time of low power consumption mode release.

After detecting the falling edge of the RXDn pin and transitioning from standby mode to snooze mode, the operating clock is supplied to SCI and data is received by SCI. The address match function determines the received data, and if it matches, it transitions from snooze mode to normal mode, and if it does not match, it transitions to standby mode again. This operation behaves as if the standby mode is continued until the communication data matching the CMPD is received, and after the standby mode is released, the normal reception operation is continued.

When using this function, the reception speed must be slow enough because it is necessary to perform the transition from the fall detection of the RXDn pin to the snooze mode, and from the clock supply to the SCI to the reception.

Figure 26.119 shows a sample flowchart for reception to software standby mode during reception.



[1] Data being transmitted is lost halfway. Data can be normally transmitted from the CPU by setting the CCR0.TE to “1”, reading CSR, and writing data to TDR after canceling software stand by mode. However, if the DMAC or DTC has been activated, the data remaining in the DMAC or DTC will be transmitted when both the TE and TIE bits in CCR0 are set to “1”.

[2] Make the I/O port function and CCR1.SPB2DT,SPB2IO settings to switch the TxDn pin to operate as a general I/O port.

[3] Set CCR0.TE bit to “0”. Set also 0 to CCR0.TIE and CCR0.TEIE if you set these bits were 1.

[4] This includes the setting for the module stopped state.

Figure 26.115 Example of Flowchart for Transition to Software Standby Mode during Transmission

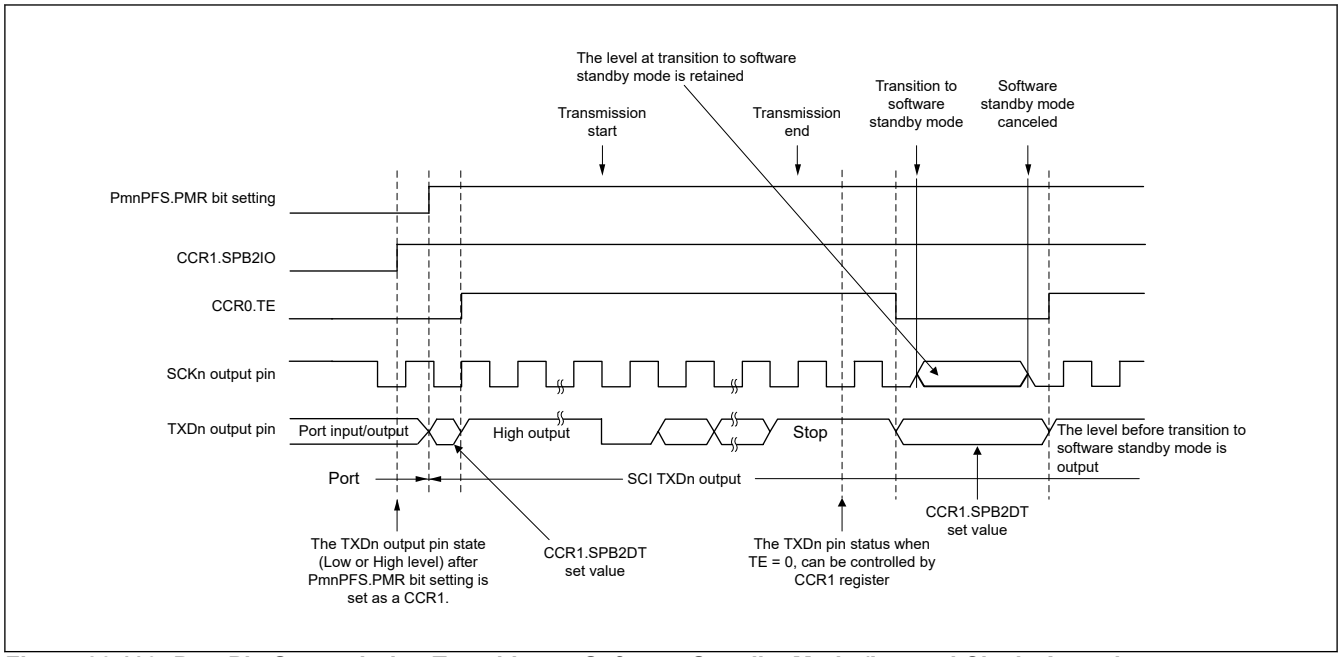


Figure 26.116 Port Pin States during Transition to Software Standby Mode (Internal Clock, Asynchronous Transmission)

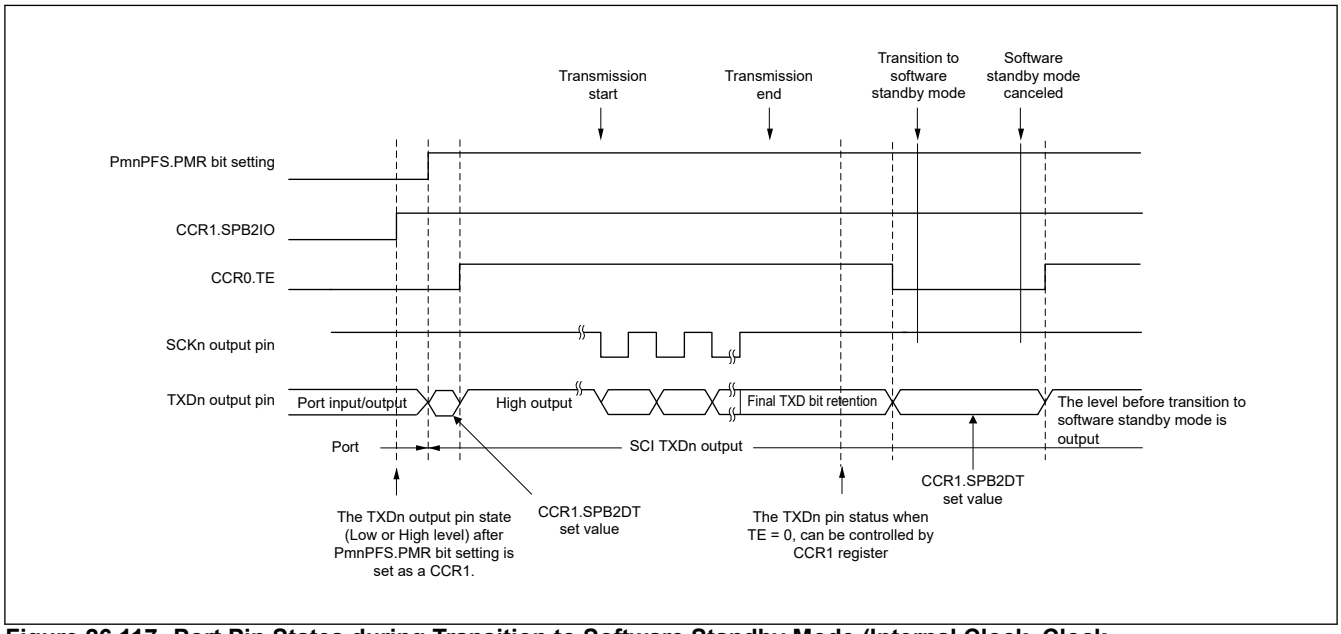


Figure 26.117 Port Pin States during Transition to Software Standby Mode (Internal Clock, Clock Synchronous Transmission)

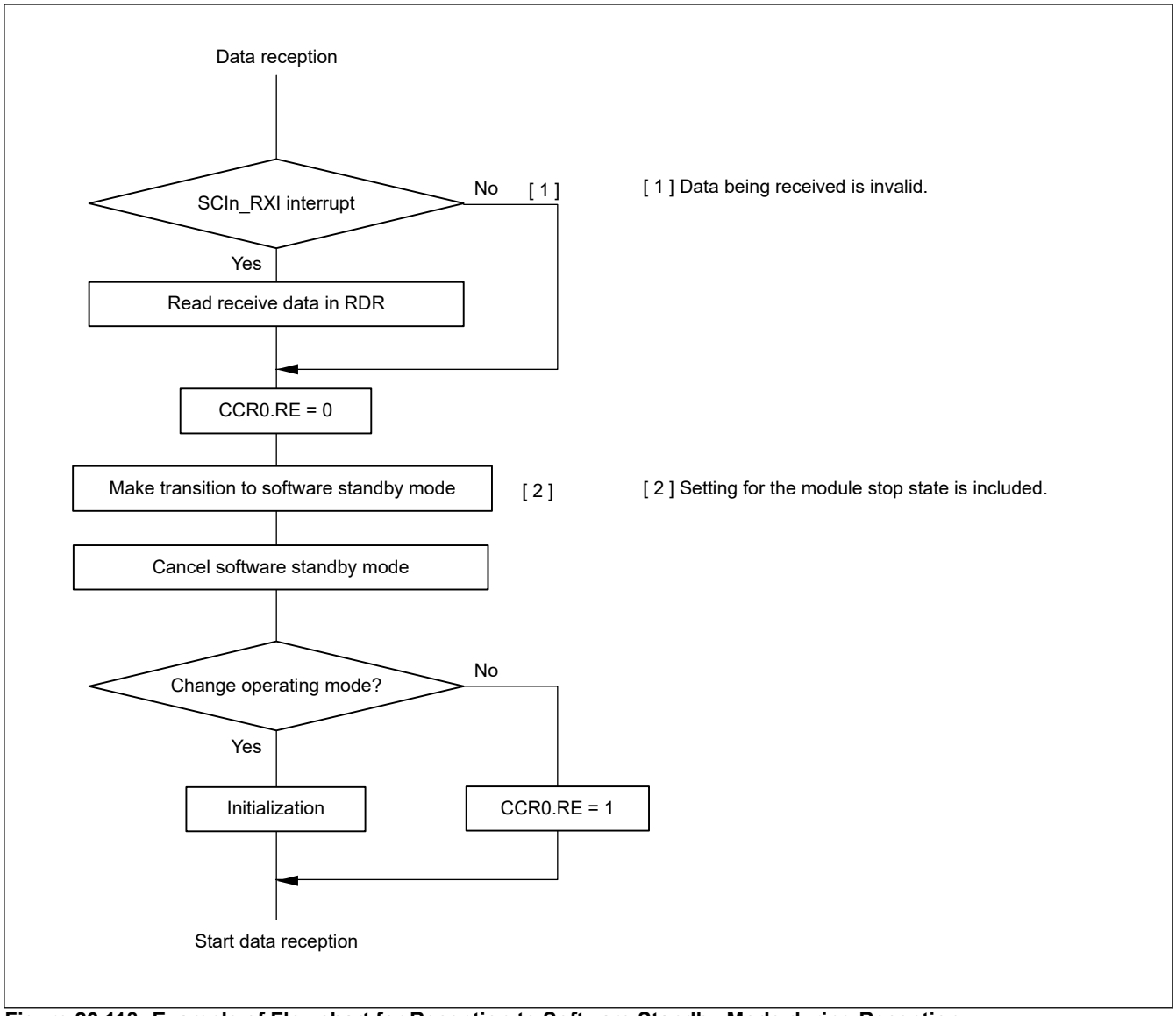


Figure 26.118 Example of Flowchart for Reception to Software Standby Mode during Reception

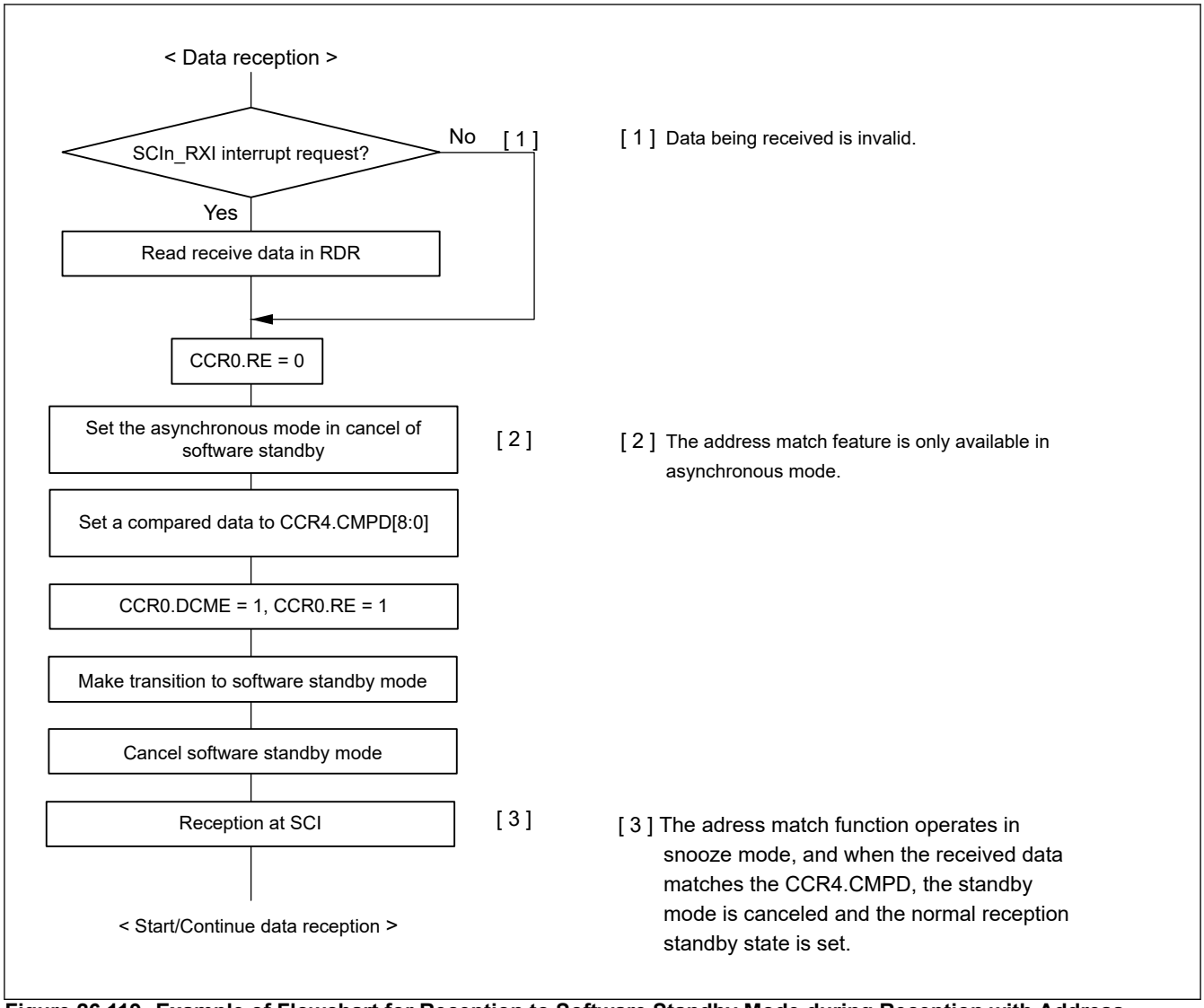


Figure 26.119 Example of Flowchart for Reception to Software Standby Mode during Reception with Address match

26.20.3 Break Detection and Processing

(1) Non-FIFO selected

When a framing error is detected, a break can be detected by reading CSR.RXDMON bit value. In a break, the input from the RXDn pin becomes all 0s, and the CSR.FER flag is set to 1 to indicate a framing error, and the CSR.PER flag might also be set to 1 to indicate a parity error. The SCI continues the receive operation even after a break is received. Therefore, even if the FER flag is 0, indicating that no framing error occurred, it is set to 1 again. When the CCR3.RXDESEL bit is 1, the SCI sets the CSR.FER flag to 1 and stops receiving operations until a start bit of the next data frame is detected. If the CSR.FER flag is set to 0, the CSR.FER flag retains 0 during the break.

When the RXDn pin is set to 1 and the break ends, detecting the beginning of the start bit on the first falling edge of the RXDn pin allows the SCI to start the receiving operation.

(2) FIFO selected

After a framing error is detected and when the SCI detects that continuous receive data is 0 for 1 frame, reception stops. When a framing error is detected, a break can be detected by reading the CSR.RXDMON flag value. After the RXDn signal is in high and the break is finished, data reception to the receive-FIFO (RDR) register resumes.

26.20.4 Mark State and Production of Breaks

When the CCR0.TE bit is 0, disabling serial transmission, the state of the TXDn pin can be set using the CCR1.SPB2IO and CCR1.SPB2DT bits. With this approach, a TXDn pin can be placed in the mark state to transmit a break.

Before setting the CCR0.TE bit to 1, enabling serial transmission, set the SPB2IO and SPB2DT bits to put the communication line in the mark state (the state of 1), and change the TXDn pin using I/O port function. To output a break on data transmission, after setting the TXDn pin to output 0 by setting the SPB2IO and SPB2DT bits, change the TXDn pin using the I/O port function and set the CCR0.TE bit to 0. When the CCR0.TE bit is set to 0, the transmitter is initialized regardless of the current state of transmission.

26.20.5 Receive Error Flags and Transmit Operations (Clock Synchronous Mode and Simple SPI Mode)

Transmission can be start by writing transmit-data to TDR even if CSR.ORER is 1. However, reception cannot be started. Note also that the receive error flags cannot be set to 0 even if the CCR0.RE is set to 0 (serial reception is disabled).

26.20.6 Writing Data to TDR

(1) Non-FIFO selected

Data can be written to TDR anytime when CCR0.TE is 1. However, if new data is written to TDR when transmit data is remaining in TDR, the previous data in TDR is lost because it has not been transferred to TSR yet. If you use DTC or DMAC, be sure to write transmit data to TDR in the SCIn_TXI interrupt request handling routine.

(2) FIFO selected

Data can be written to transmit-FIFO(TDR) when CCR0.TE is 1. Check the number of writable data with the FTSR.T [5:0] bit.

26.20.7 Restrictions on Clock Synchronous Transmission (Clock Synchronous Mode and Simple SPI Mode)

When the external clock source is used as a synchronization clock, the following restrictions apply.

(1) Start of transmission

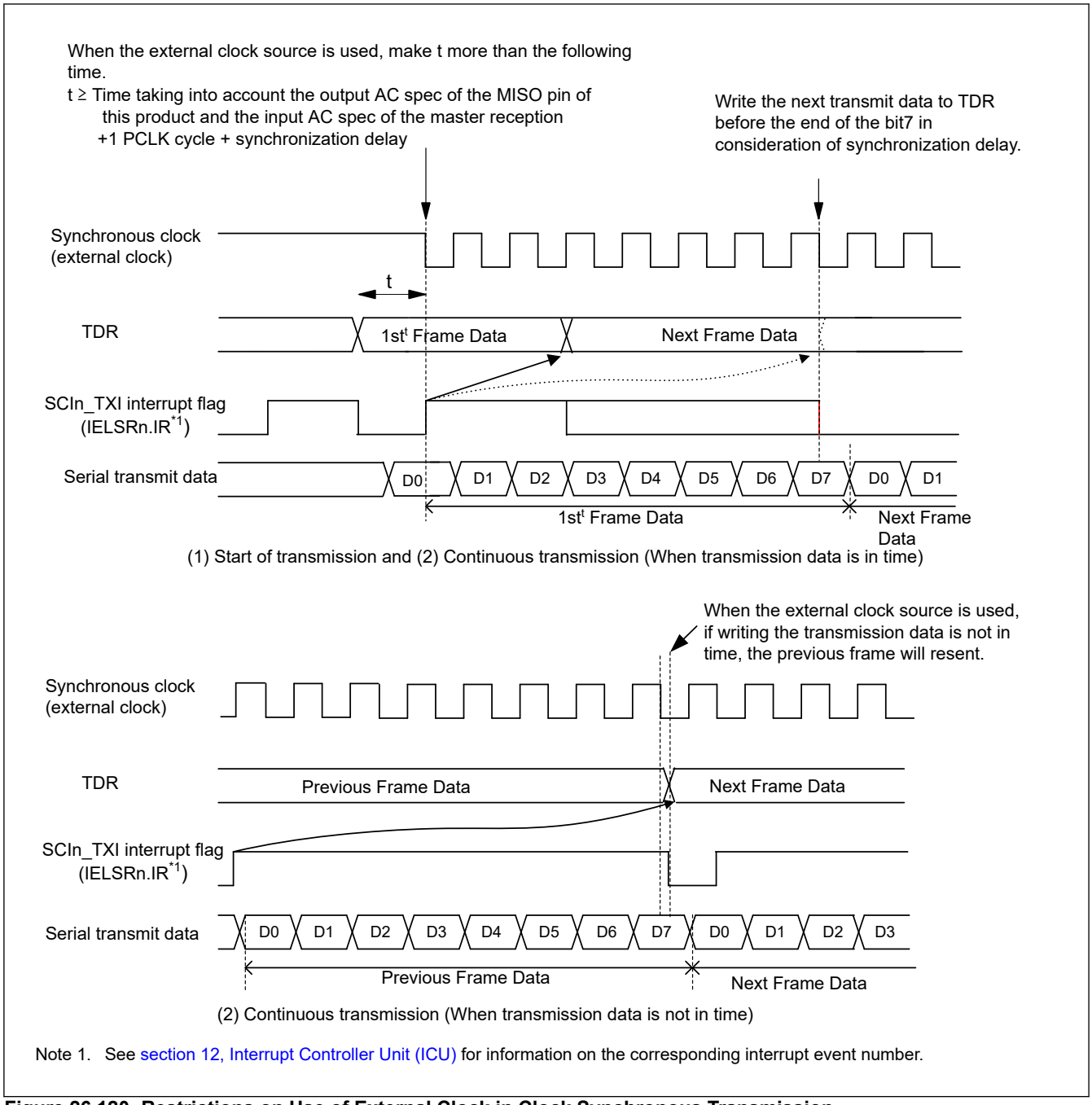
Update TDR by the CPU, DMAC, or DTC and wait at least the following time until the start of the external clock input: (See Figure 26.120)

Take the following time into account: the output AC spec of the MISO pin of this product and the input AC spec of the master reception + 1 PCLK cycle + synchronization delay.

(2) Continuous transmission

Write the next transmit data to TDR before the falling edge[*1] of the transmit clock for bit 7. Write the transmit data to TDR in consideration of synchronization delay. If the transmit data cannot be written in time, the previous frame data is resent. (See Figure 26.120)

Note 1. When CCR3.CPOL = 1 and CCR3.CPHA = 0, or CCR3.CPOL = 0 and CCR3.CPHA = 1. In the case of CCR3.CPOL = 0 and CCR3.CPHA = 0, or CCR3.CPOL = 1 and CCR3.CPHA = 1, it's the rising edge.



Note 1. See section 12, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 26.120 Restrictions on Use of External Clock in Clock Synchronous Transmission

26.20.8 Restrictions on Using DMAC or DTC

When using the DMAC or DTC to read RDR, be sure to set the receive data full interrupt (SCIn_RXI) as the activation source of the relevant SCI.

During the operation in transmission / reception using the DMAC or DTC, it should not set transfer information of DMAC or DTC.

26.20.9 Notes on Starting Transfer

At the point where transfer starts when the interrupt status flag (IELSRn.IR flag) in the ICU is 1, follow the procedure in this section to clear interrupt requests before permitting operations (by setting the CCR0.TE or CCR0.RE bit to 1). For details on the interrupt status flag, see section 12, Interrupt Controller Unit (ICU).

1. Confirm that transfer has stopped (the CCR0.TE or CCR0.RE bit is 0)

2. Set the associated interrupt enable bit (CCR0.TIE or CCR0.RIE bit) to 0
3. Read the associated interrupt enable bit (CCR0.TIE or CCR0.RIE bit) to check that it actually becomes 0
4. Set the interrupt status flag, IELSRn.IR, in the ICU to 0

26.20.10 Limitations on Simple SPI Mode

(1) Master mode

- Use a resistor to pull up or pull down the clock line matching the initial settings for the transfer clock set in the CCR3.CPHA and CPOL bits when the CCR0.SSE bit is 1.

  This prevents the clock line from being placed in the high-impedance state when the CCR0.TE bit is set to 0 or unexpected edges from being generated on the clock line when the CCR0.TE bit changes from 0 to 1. When the CCR0.SSE bit is 0 in single master mode, pulling up or pulling down the clock line is not required because the clock line is not placed in the high-impedance state even when the SCR.TE bit is set to 0.
- For the clock delay setting (CCR3.CPHA bit is 1), the receive data full interrupt (SCIn_RXI) is generated before the final clock edge on the SCKn pin as indicated in Figure 26.121. If the TE and RE bits in the CCR0 register become 0 before the final edge of the clock signal on the SCKn pin, the SCKn pin is placed in the high-impedance state, so the width of the last clock pulse of the transfer clock is shortened. Additionally, an SCIn_RXI interrupt might lead to the input signal on the SSn pin of a connected slave going to the high level before the final edge of the clock signal on the SCKn pin, leading to incorrect operation of the slave.
- In a multi-master configuration, the SCKn pin output goes to high-impedance while the input on the SSn pin is at the low level if a mode fault error occurs while a character is being transferred, stopping supply of the clock signal to the connected slave. Reset the connected slave to avoid misaligned bits when transfer is restarted.

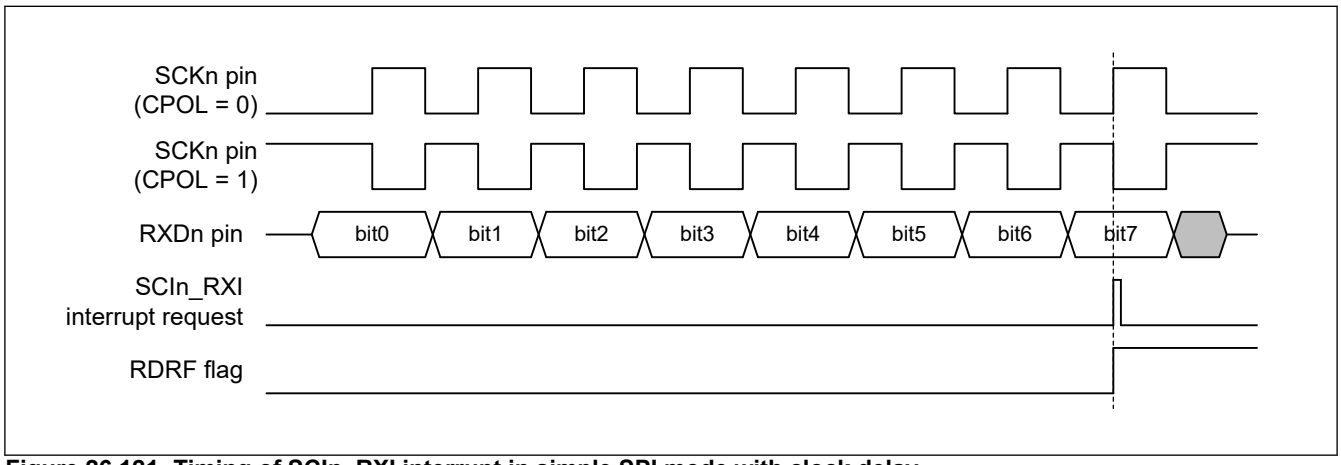


Figure 26.121 Timing of SCIn_RXI interrupt in simple SPI mode with clock delay

(2) Slave mode

- It takes 1PCLK + synchronization delay time + data output delay time (AC spec) from writing the transmit data to the TDR register until the data is output to the RXDn pin. Take these into account when starting external clock input.
- Provide an external clock signal to the master the same as the data length for transfer
- Secure the SSn input setup time (AC spec) from the SSn low-level input to the start of external clock input.
- Control the input on the SSn pin before the start and after the end of data transfer
- When the input level on the SSn pin is to be changed from low to high while a character is being transferred, set the TE and RE bits in the CCR0 register to 0 and, after restoring the settings, restart transfer of the first byte

26.20.11 Notes on Transmit Enable Bit (CCR0.TE)

In initial register value, when CCR0.TE = 0, the state of the TXDn pin is high impedance. The TXDn line should not be high impedance by the following one of ways.

1. The pull-up resistance is connected to the TXDn line.

2. Before setting the CCR0.TE bit to 0, the function of the pin should be changed to a general-purpose output port. After that, set the CCR0.TE bit to 1, and then change the function of the pin to TXDn.
3. In asynchronous mode and Manchester mode, set CCR1 and decided level of TXDn pin during CCR0.TE = 0.

In the Simple SPI mode slave operation, the MISOn pin operates in the same way as the above TXDn pin. The MISOn pin, the same as TXDn pin, should not be high impedance by the above list number 1 or list number 2.

26.20.12 Notes on Simple LIN mode

In Simple LIN mode (CCR3.MOD[2:0] = 110), the following functions cannot be used.

- Multi-processor communication function
- Bit Rate Modulation function
- Loopback function
- FIFO buffer

26.20.13 Notes on RS-485 Driver Control function

RS-485 Driver control function is valid only in Asynchronous mode.

When RS-485 Driver control function is active (CCR3.DEN = 1), the CSR.TEND set timing / SCIn_TEI output timing changes as follows. Wait for the SCIn_TEI interrupt and set the CCR0.TE bit in SCI to 0.

When RS-485 Driver control function is inactive: When STOP bit output is completed.

When RS-485 Driver control function is active: At the end of DEn negation time.

26.20.14 Notes on Loopback function

The Loopback function is valid in Asynchronous mode with internal clock, in Manchester mode with internal clock and Clock synchronous mode with internal clock.

26.20.15 Notes regarding register access when operation clock (TCLK) is slower than bus clock (PCLK)

If the operating clock (TCLK) is slower than the bus clock (PCLK), the time until this information is transmitted internally after writing to the CCR0.TE and CCR0.RE registers is slower than the bus access time. In particular, when trying to change the setting register after writing 0 and interrupting communication, do not change the setting register before the signal inside the IP is in the communication stopped state. To prevent this, after setting CCR0.TE and CCR0.RE to 0, check the CESR.TIST and CESR.RIST bits until they are 0 before setting the next register.

26.20.16 Notes on interrupting operation

If 0 is written to CCR0.RE during data reception and the reception operation is interrupted, there is a possibility of an invalid state, do not use the received data (RDR register stored value) and the flag value of each status register. To interrupt the reception operation, stop the interrupt or event link reception side and then write 0 to the CCR0.RE bit.

26.20.17 Notes on CCR3.BPEN bit setting

Set the BPEN bit only once when setting the CCR3 register in the SCI initialization flow.

This bit cannot be changed after the initialization.

When this bit setting is changed, start from the SCI initialization flow again.

27. I²C Bus Interface (IIC)

This is the IIC_B version of the IIC peripheral module.

IIC_B is referred to as IIC in this chapter.

27.1 Overview

27.1.1 Functional Overview

The I²C bus interface (IIC) has 2 channels. The IIC module conform with and provide a subset of the NXP I²C (Inter-Integrated Circuit) bus interface functions.

Table 27.1 lists the I²C specifications.

Table 27.1 I²C specifications

Item	Description
Operation mode	Master mode and slave mode selectable
Data handler	Single buffer transfer
Communication protocol	• I²C bus format – Standard-mode (Sm) : 0 to 100 kbps – Fast-mode (Fm) : 0 to 400 kbps – Fast-mode Plus (Fm+) : 0 to 1 Mbps*1 – High-speed mode (Hs-mode) : 0 to 3.2 Mbps*1 • SMBus format : 10 to 100 kbps
Address format	• 7-bit address • 10-bit address
Address detection	• Slave address (static address) (max 3 addresses) • General call address • Hs-mode master code*1 • Device ID • Host address • 10-bit slave addressing
Clock stretching	Clock stretching capability
Noise-filter	• Analog noise-filter*2 • Digital noise-filter
Interrupt source	• Rx data buffer full • Tx data buffer empty • START condition detection • STOP condition detection • Transmit end • NACK detection • Arbitration lost • Timeout detection • Wake-up condition detection*2
Error detection	• NACK received • Arbitration lost error • Timeout error
Event link output	• Communication event • Rx data buffer full event • Tx data buffer empty event • Transmit end event
Wake-up source*2	Address detection of slave address

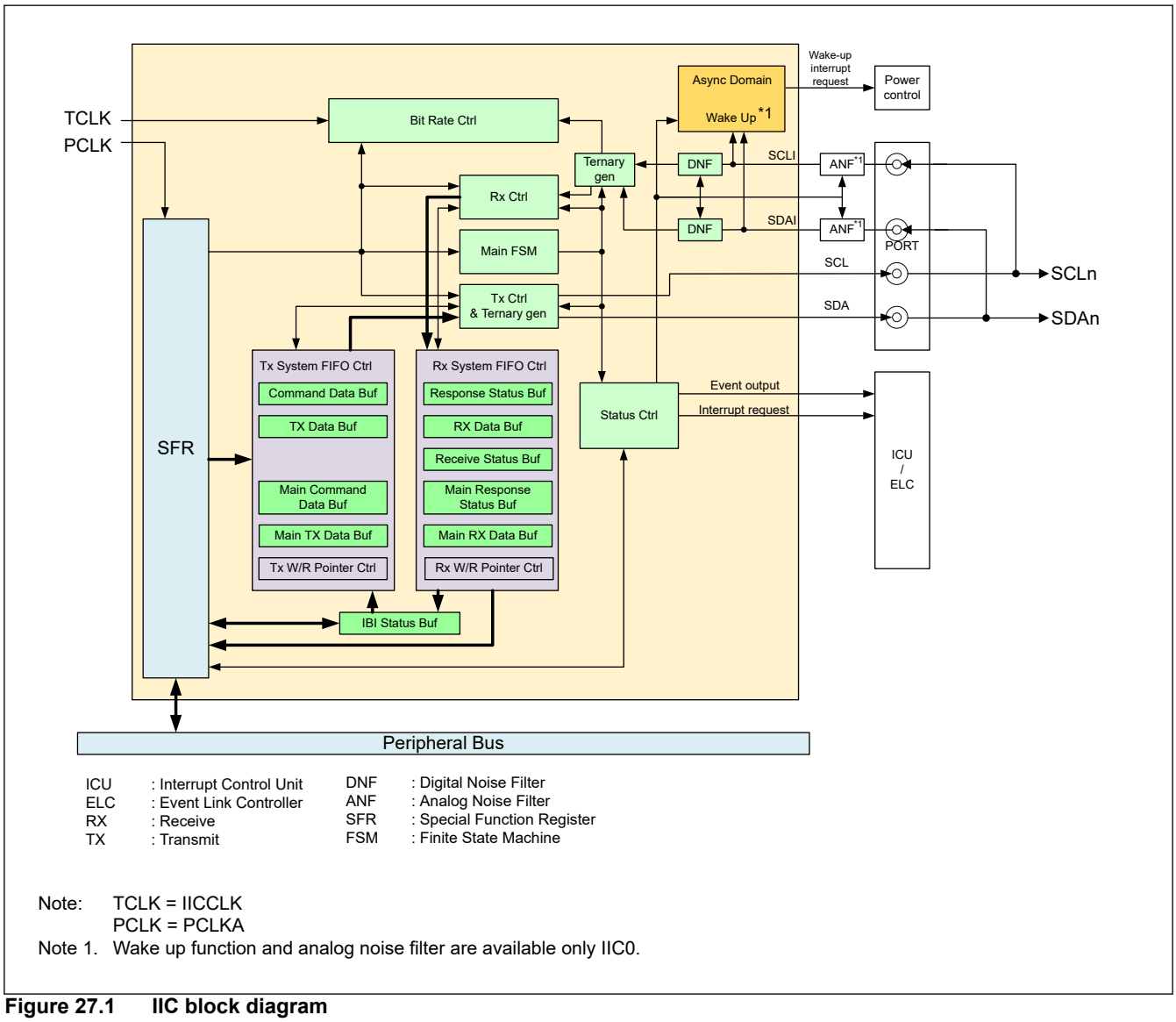
Note 1. Fast-mode Plus and High-speed mode are supported by IIC0 (SCL0_A, SDA0_A)
Note 2. Wake-up function and analog noise filter are available only IIC0.

Table 27.2 IIC I/O pins (n = 0, 1)

Function	Pin name	I/O	Description
IICn	SCLn	I/O	Input/output pins for clock
	SDAn	I/O	Input/output pins for data

27.1.2 Block Diagram

Figure 27.1 shows the main components of this IIC.



Note: TCLK = IICCLK
PCLK = PCLKA
Note 1. Wake up function and analog noise filter are available only IIC0.

Figure 27.1 IIC block diagram

27.2 Registers

27.2.1 List of Registers

IIC registers are listed in the following table.

Table 27.3 List of IIC registers (1 of 2)

Register	Symbol	Offset address
Bus Control Register	BCTL	0x014
Reset Control Register	RSTCTL	0x020

Table 27.3 List of IIC registers (2 of 2)

Register	Symbol	Offset address
Present State Register	PRSST	0x024
Bus Function Control Register	BFCTL	0x060
Slave Control Register	SVCTL	0x064
Reference Clock Control Register	REFCKCTL	0x070
Standard Bit Rate Register	STDBR	0x074
Extended Bit Rate Register	EXTBR	0x078
Bus Free Condition Detection Time Register	BFRECDT	0x07C
Output Control Register	OUTCTL	0x088
Input Control Register	INCTL	0x08C
Timeout Control Register	TMOCTL	0x090
Wake Up Unit Control Register*1	WUCTL	0x098
Acknowledge Control Register	ACKCTL	0x0A0
SCL Stretch Control Register	SCSTRCTL	0x0A4
Condition Control Register	CNDCTL	0x140
Normal Transfer Data Buffer Port Register 0	NTDTBP0/ NTDTBP0_BY	0x158
Bus Status Register	BST	0x1D0
Bus Status Enable Register	BSTE	0x1D4
Bus Interrupt Enable Register	BIE	0x1D8
Bus Status Force Register	BSTFC	0x1DC
Normal Transfer Status Register	NTST	0x1E0
Normal Transfer Status Enable Register	NTSTE	0x1E4
Normal Transfer Interrupt Enable Register	NTIE	0x1E8
Normal Transfer Status Force Register	NTSTFC	0x1EC
Bus Condition Status Register	BCST	0x210
Slave Status Register	SVST	0x214
Wake Up Unit Operating Status Register*1	WUST	0x218
Slave Device Address Table Basic Register 0	SDATBAS0	0x2B0
Slave Device Address Table Basic Register 1	SDATBAS1	0x2B4
Slave Device Address Table Basic Register 2	SDATBAS2	0x2B8
Slave Device Address Register 0	SVDVAD0	0x330
Slave Device Address Register 1	SVDVAD1	0x334
Slave Device Address Register 2	SVDVAD2	0x338
Bit Count Register	BITCNT	0x380
Present State Debug Register	PRSTDBG	0x3CC

Note 1. Reserved register in IIC1

27.2.2 BCTL : Bus Control Register

Base address: IIC_Bn = 0x4011_F000 + 0x0400 × n (n = 0, 1)

Offset address: 0x014

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

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

Bit	Symbol	Function	R/W
30:0	—	These bits are read as 0. The write value should be 0.	R/W
31	BUSE	Bus Enable 0: IIC bus operation is disabled. 1: IIC bus operation is enabled.	R/W

BUSE bit (Bus Enable)

Enables or disables the operation on the I^2C Bus by IIC.

Set the BUSE bit to 1 when using IIC. The SCL and SDA pins are placed in the active state when the BUSE bit is set to 1. Set the BUSE bit to 0 when IIC is not to be used. The SCL and SDA pins are placed in the inactive state when the BUSE bit is set to 0.

If the software sets this bit, then it also confirms that initialization is done, and that IIC can use the programmed register values.

Software may disable IIC bus operation while it is active, However:

- When the software reads the value 0 from this field, this indicates that IIC bus operation disable operation has completed.

27.2.3 RSTCTL : Reset Control Register

Base address: IIC_Bn = 0x4011_F000 + 0x0400 × n (n = 0, 1)

Offset address: 0x020

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RI2CRST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RI2CRST	IIC Software Reset 0: Reset of all registers and internal state. 1: Releases of all registers and internal state.	R/W
15:1	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
16	INTLRST	Internal Software Reset 0: Releases of some registers and internal state. 1: Resets of some registers and internal state.	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

For details on reset for each register, see section 27.6. Reset Descriptions.

RI2CRST bit (IIC Software Reset)

On Driver setting this bit to 1, IIC shall be reset and disabled.

All registers shall return to their reset values, and the software shall re-initialize IIC.

This field is cleared automatically upon IIC reset completion. This field also resets all Queues in IIC.

Note: Programming this field while it contains a value of 1 may result in undefined behavior.

INTLRST bit (Internal Software Reset)

When set to 1, some of registers is reset, and internal state is reset. For details on the registers to be reset, see section 27.6. Reset Descriptions.

27.2.4 PRSST : Present State Register

Base address: IIC_Bn = 0x4011_F000 + 0x0400 × n (n = 0, 1)

Offset address: 0x024

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

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

Bit	Symbol	Function	R/W
1:0	—	These bits are read as 0. The write value should be 0.	R/W
2	CRMS	Current Master 0: The Master is not the Current Master, and must request and acquire bus ownership before initiating any transfer. 1: The Master is the Current Master, and as a result can initiate transfers.	R/W*1
3	—	This bit is read as 0. The write value should be 0.	R/W
4	TRMD	Transmit/Receive Mode 0: Receive mode 1: Transmit mode	R
6:5	—	These bits are read as 0. The write value should be 0.	R/W
7	PRSSTWP	Present State Write Protect 0: CRMS bit is protected. 1: CRMS bit can be written when writing simultaneously with the value of the target bit.	W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. When the PRSSTWP bit is set to 1, the CRMS bit can be written to.

CRMS bit (Current Master)

Indicates the set condition and reset condition of each operation mode.

[Clearing conditions]

- When 0 written to the PRSST.CRMS by the software.
- When STOP is issued.
- When Master Arbitration-Lost.

[Setting conditions]

- When 1 written to the PRSST.CRMS by the software.
- When START is issued.

The PRSST register returns IIC current state.

State has two parts: this register which is mandatory, and an additional optional PRSST_DEBUG register intended for debug purposes (see the Debug Capability registers in the Extended Capabilities list).

TRMD bit (Transmit/Receive Mode)

This bit indicates transmit or receive mode.

IIC is in receive mode when the TRMD bit is set to 0 and is in transmit mode when the bit is set to 1. Combination of this bit and the CRMS bit indicates the operating mode of IIC.

The value of TRMD bit is automatically changed to 1 for transmit mode or 0 for receive mode by issuing or detection of a START condition and setting of the R/W# bit.

[Setting conditions]

- When a START condition is issued normally according to the START condition issuance request (when a START condition is detected with the CNDCTL.STCND bit set to 1).
- When a Repeated START condition is issued normally according to the Repeated START condition issuance request (when a Repeated START condition is detected with the CNDCTL.SRCND bit set to 1).
- When the R/W# bit added to the slave address is set to 0 in master mode.
- When the address received in slave mode matches the address enabled in SVCTL, with the R/W# bit set to 1.

[Clearing conditions]

- When a STOP condition is detected.
- The ALF (arbitration-lost) flag in BST being set to 1.
- In master mode, reception of a slave address to which an R/W# bit with the value 1 is appended.
- In slave mode, a match between the received address and the address enabled in SVCTL when the value of the received R/W# bit is 0 (including cases where the received address is the general call address).
- In slave mode, a Repeated START condition is detected (a Repeated START condition is detected with BCST.BFREF = 0 and CRMS = 0).

PRSSTWP bit (Present State Write Protect)

PRSSTWP is always 0 when reading.

When writing to PRSST, writing 1 to this bit at the same time enables writing to CRMS bit.

27.2.5 BFCTL : Bus Function Control Register

Base address: IIC_Bn = 0x4011_F000 + 0x0400 × n (n = 0, 1)

Offset address: 0x060

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	HSME	FMPE	—	SMBS	—	—	—	SCSYNE	—	—	—	—	—	SALE	NALE	MALE
Value after reset:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	MALE	Master Arbitration-Lost Detection Enable 0: Master arbitration-lost detection disables. Disables the arbitration-lost detection function and does not clear the CRMS and TRMD bits in PRSST automatically when arbitration is lost. 1: Master arbitration-lost detection enables. Enables the arbitration-lost detection function and clears the CRMS and TRMD bits in PRSST automatically when arbitration is lost.	R/W
1	NALE	NACK Transmission Arbitration-Lost Detection Enable 0: NACK transmission arbitration-lost detection disables. 1: NACK transmission arbitration-lost detection enables.	R/W
2	SALE	Slave Arbitration-Lost Detection Enable 0: Slave arbitration-lost detection disables. 1: Slave arbitration-lost detection enables.	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
8	SCSYNE	SCL Synchronous Circuit Enable 0: No SCL synchronous circuit uses. 1: An SCL synchronous circuit uses.	R/W
11:9	—	These bits are read as 0. The write value should be 0.	R/W
12	SMBS	SMBus/I^{2}C Bus Selection 0: The I^{2}C bus select. 1: The SMBus select.	R/W
13	—	This bit is read as 0. The write value should be 0.	R/W
14	FMPE*1	Fast-mode Plus Enable 0: No Fm+ slope control circuit uses for the SCLn pin and SDAn pin. (n = 0, 1) 1: An Fm+ slope control circuit uses for the SCLn pin and SDAn pin. (n = 0, 1)	R/W
15	HSME*2	High Speed Mode Enable 0: Disable High Speed Mode. 1: Enable High Speed Mode.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. The Fast-mode Plus Enable bit (FMPE) is supported by IIC0 (SCL0_A, SDA0_A). Bit[14] is the reserved bit in the not supported channel.

Note 2. The High Speed Mode Enable bit (HSME) is supported by IIC0 (SCL0_A, SDA0_A). Bit[15] is the reserved bit in the not supported channel.

MALE bit (Master Arbitration-Lost Detection Enable)

This bit is used to specify whether to use the arbitration-lost detection function in master mode. Normally, set this bit to 1.

NALE bit (NACK Transmission Arbitration-Lost Detection Enable)

This bit is used to specify whether to cause arbitration to be lost when ACK is detected during transmission of NACK in receive mode (such as when slaves with the same address exist on the bus or when two or more masters select the same slave device simultaneously with different number of receive bytes).

SALE bit (Slave Arbitration-Lost Detection Enable)

This bit is used to specify whether to cause arbitration to be lost when a value different from the value being transmitted is detected on the bus in slave transmit mode (such as when slaves with the same address exist on the bus or when a mismatch with the transmit data occurs due to noise).

SCSYNE bit (SCL Synchronous Circuit Enable)

This bit is used to specify whether to synchronize the SCL clock with the SCL input clock. Normally, set this bit to 1.

When the SCSYNE bit set to 0 (no SCL synchronous circuit used), IIC does not synchronize the SCL clock with the SCL input clock. In this setting, IIC outputs the SCL clock with the transfer rate set in STDBR and EXTBR regardless of the SCLn line state. For this reason, if the bus load of the I^2C bus line is much larger than the specification value or if the SCL clock output overlaps in multiple masters, the short-cycle SCL clock that does not meet the specification may be output. When no SCL synchronous circuit uses, it also affects the issuance of a START condition, Repeated START condition, and STOP condition, and the continuous output of extra SCL clock cycles.

This bit must not be set to 0 except for checking the output of the set transfer rate.

FMPE bit (Fast-mode Plus Enable)

This bit is used to specify whether to use a slope control circuit for Fast-mode Plus [Fm+].

When this bit is set to 1, a slope control circuit conforming to the Fast-mode Plus [Fm+] slope control specification (tof) of the IIC-bus is selected. When this bit is set to 0, a slope control circuit conforming to the Standard-mode [Sm] and Fast-mode [fm] slope control specification (tof) of the IIC-bus is selected.

Set this bit to 1 when using the transmission rate within a range up to 1 Mbps (Fast-mode Plus [Fm+]) of the IIC-bus specification. Set this bit to 0 when using the transmission rate at other rates (up to 100 kbps [Sm], up to 400 kbps [Fm]) or for SMBus (10 to 100 kbps).

Note: When communicating in Hs-mode, set as follows.
- Set FMPE to 0 when sending Hs-mode master code (0000 1XXXb) with Fast-mode.
- Set FMPE to 1 when sending Hs-mode master code (0000 1XXXb) with Fast-mode Plus.

HSME bit (High Speed Mode Enable)

This bit is used for communicating in Hs-mode.

When this bit is set to 1, the Hs-mode master code is recognized and Hs-mode communication is possible.

After the START condition is detected, if Hs-mode master code (0000 1XXXb) transmission is recognized, Hs-mode communication starts from Repeated START after receiving the NACK response.

It communicates at the bit rate set in STDBR until the NACK response, and automatically switches from Repeated START condition issuance after receiving the NACK response to the bit rate set in EXTBR.

Hs-mode continues until a STOP condition is detected.

When the STOP condition is detected, the bit rate is automatically switched to the bit rate set in STDBR.

Note: When this bit is set to 1, the BST.NACKDF bit will not be set even if a NACK response is received after sending the Hs-mode master code.

27.2.6 SVCTL : Slave Control Register

Base address: IIC_Bn = 0x4011_F000 + 0x0400 × n (n = 0, 1)

Offset address: 0x064

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	SVAE2	SVAE1	SVAE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
0	GCAE	General Call Address Enable 0: General call address detection disables. 1: General call address detection enables.	R/W
4:1	—	These bits are read as 0. The write value should be 0.	R/W
5	HSMCE*1	Hs-mode Master Code Enable 0: Hs-mode Master Code Detection disables. 1: Hs-mode Master Code Detection enables.	R/W
6	DVIDE	Device-ID Address Enable 0: Device-ID address detection disables. 1: Device-ID address detection enables.	R/W
14:7	—	These bits are read as 0. The write value should be 0.	R/W
15	HOAE	Host Address Enable 0: Host address detection disables. 1: Host address detection enables.	R/W
16	SVAE0	Slave Address Enable 0 0: Slave 0 disables 1: Slave 0 enables	R/W
17	SVAE1	Slave Address Enable 1 0: Slave 1 disables 1: Slave 1 enables	R/W
18	SVAE2	Slave Address Enable 2 0: Slave 2 disables 1: Slave 2 enables	R/W
31:19	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. The Hs-mode Master Code Enable bit (HSMCE) is supported by IIC0 (SCL0_A, SDA0_A). Bit[5] is the reserved bit in the not suported channel.

GCAE bit (General Call Address Enable)

This bit is used to specify whether to ignore the general call address (0000 000 + 0 (write): All 0) when it is received. When this bit is set to 1, if the received slave address matches the general call address, IIC recognizes the received slave address as the general call address independently of the slave addresses set in the SVDVADy.SVAD[9:0] bits (y = 0 to 2) and performs data receive operation.

When this bit is set to 0, the received slave address is ignored even if it matches the general call address.

HSMCE bit (Hs-mode Master Code Enable)

This bit is used to specify whether to recognize and execute the Hs-mode master code (0000 1XXXb) is received in the first byte after a START condition is detected.

When this bit is set to 1, if the received first byte matches the Hs-mode master code, IIC recognizes that the Hs-mode master code has been received.

The first byte after Repeated START after NACK response to Hs-mode master code is recognized as a slave address and compared with the slave address set by SVDVADy.SVAD[9:0].

If the addresses match, the transmission / reception operation continues according to the R/W# bit value.

Hs-mode continues until a STOP condition is detected.

When this bit is set to 0, IIC will ignore the pattern until a STOP condition is detected, even if it matches the Hs-mode master code.

Note: When this bit is set to 1, SCSTRCTL.ACKTWE bit must be set to 0 and SCSTRCTL.RWE bit must be set to 1.

DVIDE bit (Device-ID Address Enable)

This bit is used to specify whether to recognize and execute the Device-ID address when a device ID (1111 100) is received in the first byte after a START condition or Repeated START condition is detected.

When this bit is set to 1, if the received first byte matches the Device-ID, IIC recognizes that the Device-ID address has been received. When the following R/W# bit is 0 (write), IIC recognizes the second and the following bytes as slave addresses and continues the receive operation.

When this bit is set to 0, IIC ignores the received first byte even if it matches the Device ID address and recognizes the first byte as a normal slave address.

For details on the Device-ID address detection, see (3)Device-ID Address Detection .

HOAE bit (Host Address Enable)

This bit is used to specify whether to ignore received host address (0001 000) when the BFCTL.SMBS bit = 1.

When this bit is set to 1 while the SMBS bit = 1, if the received slave address matches the host address, IIC recognizes the received slave address as the host address independently of the slave addresses set in the SVDVADy.SVAD[9:0] bits (y = 0 to 2) and performs the receive operation.

When the SMBS bit or the HOAE bit is set to 0, the received slave address is ignored even if it matches the host address.

SVAEy bits (Slave Address Enable y (y = 0 to 2))

This bit is used to enable or disable the slave address set in the SVDVADy.SVAD[9:0] bits.

When this bit is set to 1, the slave address set in the SVAD[9:0] bits is enabled and is compared with the received slave address.

When this bit is set to 0, the slave address set in the SVAD[9:0] bits is disabled and is ignored even if it matches the received slave address.

27.2.7 REFCKCTL : Reference Clock Control Register

Base address: IIC_Bn = 0x4011_F000 + 0x0400 × n (n = 0, 1)

Offset address: 0x070

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

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

Bit	Symbol	Function	R/W
2:0	IREFCKS[2:0]	Internal Reference Clock Selection Selects the internal reference clock source (IICφ) for IIC. 0 0 0: IICCLK/1 clock 0 0 1: IICCLK/2 clock 0 1 0: IICCLK/4 clock 0 1 1: IICCLK/8 clock 1 0 0: IICCLK/16 clock 1 0 1: IICCLK/32 clock 1 1 0: IICCLK/64 clock 1 1 1: IICCLK/128 clock	R/W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

27.2.8 STDBR : Standard Bit Rate Register

Base address: IIC_Bn = 0x4011_F000 + 0x0400 × n (n = 0, 1)

Offset address: 0x074

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

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

Bit	Symbol	Function	R/W
7:0	SBRLO[7:0]	Count value of the Low-level period of SCL clock	R/W
15:8	SBRHO[7:0]	Count value of the High-level period of SCL clock	R/W
30:16	—	These bits are read as 0. The write value should be 0.	R/W
31	DSBRPO	Double the Standard Bit Rate Period for Open-Drain 0: The time period set for SBRHO[7:0] and SBRLO[7:0] is not doubled. 1: The time period set for SBRHO[7:0] and SBRLO[7:0] is doubled.	R/W

The STDBR register sets the bit rate according to the operating speed (Standard-mode / Fast-mode / Fast-mode plus).

The I^2C transfer rate and the SCL clock duty are calculated using the following expression.

Transfer rate = 1 / {[(High-Level Period + α[*1]) + (Low-Level Period + α)] / IICφ[*2] + SCLn line rising time [tr][*3] + SCLn line falling time [tf][*3]}

Duty cycle = {SCLn line rising time [tr] + (High-Level Period + α) / IICφ} / {SCLn line falling time [tf] + (Low- Level Period + α) / IICφ}

Note 1. α depend on the number of stages in the noise filter.
Note 2. IICφ = TCLK × Division ratio
Note 3. The SCLn line rising time [tr] and SCLn line falling time [tf] depend on the total bus line capacitance [Cb] and the pull-up resistor [Rp]. For details, see the I^2C-bus specification from NXP Semiconductors.

SBRLO[7:0] bits (Count value of the Low-level period of SCL clock)

The SBRLO[7:0] bits are used to set the low-level period of SCL clock in Open-Drain mode.

IIC counts the low-level period with the internal reference clock source (IICφ) specified by the REFCKCTL.IREFCKS[2:0] bits. It also works to generate the data setup time for automatic SCL low-hold operation (see section 27.3.1.3.5. Clock Stretching); when IIC is used in I^2C slave mode, these bits need to be set to a value longer than the data setup time[*1].

If the digital noise filter is enabled (INCTL.DNFE = 1), set the SBRLO[7:0] bits to a value at least one greater than the number of stages in the noise filter. Regarding the number of stages in the noise filter, see the description of the INCTL.DNFS[3:0] bits.

Note 1. Data setup time (tSU: DAT)

250 ns (up to 100 kbps: Standard-mode [Sm])

100 ns (up to 400 kbps: Fast-mode [Fm])

50 ns (up to 1 Mbps: Fast-mode plus [Fm+])

10 ns (up to 3.4 Mbps: Hs-mode [HS])

SBRHO[7:0] bits (Count value of the High-level period of SCL clock)

The SBRHO[7:0] bits use to set the high-level period of SCL clock in Open-Drain mode. SBRHO[7:0] bits are valid in master mode. If IIC is used only in I^2C slave mode, these bits need not to set the high-level period.

IIC counts the high-level period with the internal reference clock source (IICφ) specified by the REFCKCTL.IREFCKS[2:0] bits.

If the digital noise filter is enabled (the INCTL.DNFE bit = 1), set the SBRHO[7:0] bits to a value at least one greater than the number of stages in the noise filter. Regarding the number of stages in the noise filter, see the description of the INCTL.DNFS[3:0] bits.

DSBRPO bit (Double the Standard Bit Rate Period for Open-Drain)

When DSBRPO = 1, double the high-level period that is set in SBRHO[7:0] and double the low-level period that is set in SBRLO[7:0].

Table 27.4 Requirement and usage of setting in each mode

Bit name	Device mode	
	I^2C master	I^2C slave
SBRHO[7:0]	Setting required*1	Do not use
SBRLO[7:0]	Setting required*1	Setting required*2

Note 1. The setting value is used for the data rate of ST, FM, and FM+ mode.
Note 2. The setting value is used for the data setup time of automatic SCL low-hold operation.

27.2.9 EXTBR : Extended Bit Rate Register

Base address: IIC_Bn = 0x4011_F000 + 0x0400 × n (n = 0, 1)

Offset address: 0x078

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

Bit	Symbol	Function	R/W
7:0	EBRLO[7:0]	Extended Bit Rate Low-Level Period Open-Drain Count value of the low-level period of SCL clock	R/W
15:8	EBRHO[7:0]	Extended Bit Rate High-Level Period Open-Drain Count value of the high-level period of SCL clock	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

The EXTBR register sets the bit rate for communication in high-speed mode.

EBRLO[7:0] bits (Extended Bit Rate Low-Level Period Open-Drain)

See SBRLO[7:0] bits of section 27.2.8. STDBR : Standard Bit Rate Register for details. Watch SBRHO, SBRLO as EBRHO[7:0], EBRLO[7:0].

EBRHO[7:0] bits (Extended Bit Rate High-Level Period Open-Drain)

See SBRHO[7:0] bits of section 27.2.8. STDBR : Standard Bit Rate Register for details. Watch SBRHO, SBRLO as EBRHO[7:0], EBRLO[7:0].

Table 27.5 Requirement and usage of setting in each mode

Bit name	Device mode	
	I²C master	I²C slave
EBRHO[7:0]	Setting required*1	Do not use
EBRLO[7:0]	Setting required*1	Setting required*2

Note 1. The setting value is used for the data rate of HS-mode.
Note 2. The setting value is used for the data setup time of automatic SCL low-hold operation in HS-mode.

27.2.10 BFRECDT : Bus Free Condition Detection Time Register

Base address: IIC_Bn = 0x4011_F000 + 0x0400 × n (n = 0, 1)

Offset address: 0x07C

Bit position:	31																					9	8								0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FRECYC[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	FRECYC[8:0]	Bus Free Condition Detection Cycle The count value is a period for detecting the Bus free condition.	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

FRECYC[8:0] bits (Bus Free Condition Detection Cycle)

IIC counts the period for detecting the Bus free condition with the IICφ.

These bits set the Bus Free period. This Bus Free period is counted by the internal reference clock (IICφ) selected by the REFCKCTL.IREFCKS[2:0] bits. See the BCST.BFREF flag for Bus Free detection behavior.

27.2.11 OUTCTL : Output Control Register

Base address: IIC_Bn = 0x4011_F000 + 0x0400 × n (n = 0, 1)

Offset address: 0x088

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SDODCS	—	—	—	—	SDOD[2:0]			—	—	—	EXCYC	—	SOCWP	SCOC	SDOC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bit	Symbol	Function	R/W
0	SDOC	SDA Output Control 0: IIC drives the SDAn pin low. 1: IIC releases the SDAn pin.	R/W
1	SCOC	SCL Output Control High level output is achieved through an external pull-up resistor. 0: IIC drives the SCLn pin low. 1: IIC releases the SCLn pin.	R/W

Bit	Symbol	Function	R/W
2	SOCWP	SCL/SDA Output Control Write Protect 0: Bits SCOC and SDOC are protected. 1: Bits SCOC and SDOC can be written (When writing simultaneously with the value of the target bit). This bit is read as 0.	W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	EXCYC	Extra SCL Clock Cycle Output The EXCYC bit is cleared automatically after one clock cycle is output. 0: Does not output an extra SCL clock cycle (default). 1: Outputs an extra SCL clock cycle.	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
10:8	SDOD[2:0]	SDA Output Delay 0 0 0: No output delay 0 0 1: 1 IICφ cycle (When OUTCTL.SDODCS = 0 (IICφ)) 1 or 2 IICφ cycles (When OUTCTL.SDODCS = 1 (IICφ/2)) 0 1 0: 2 IICφ cycles (When OUTCTL.SDODCS = 0 (IICφ)) 3 or 4 IICφ cycles (When OUTCTL.SDODCS = 1 (IICφ/2)) 0 1 1: 3 IICφ cycles (When OUTCTL.SDODCS = 0 (IICφ)) 5 or 6 IICφ cycles (When OUTCTL.SDODCS = 1 (IICφ/2)) 1 0 0: 4 IICφ cycles (When OUTCTL.SDODCS = 0 (IICφ)) 7 or 8 IICφ cycles (When OUTCTL.SDODCS = 1 (IICφ/2)) 1 0 1: 5 IICφ cycles (When OUTCTL.SDODCS = 0 (IICφ)) 9 or 10 IICφ cycles (When OUTCTL.SDODCS = 1 (IICφ/2)) 1 1 0: 6 IICφ cycles (When OUTCTL.SDODCS = 0 (IICφ)) 11 or 12 IICφ cycles (When OUTCTL.SDODCS = 1 (IICφ/2)) 1 1 1: 7 IICφ cycles (When OUTCTL.SDODCS = 0 (IICφ)) 13 or 14 IICφ cycles (When OUTCTL.SDODCS = 1 (IICφ/2))	R/W
14:11	—	These bits are read as 0. The write value should be 0.	R/W
15	SDODCS	SDA Output Delay Clock Source Selection 0: The internal reference clock (IICφ) is selected as the clock source of the SDA output delay counter. 1: The internal reference clock divided by 2 (IICφ/2) is selected as the clock source of the SDA output delay counter.*1	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. The setting SDODCS = 1 (IICφ/2) only becomes valid when SCL is at the low level. When SCL is at the high level, the setting SDODCS = 1 becomes invalid and the clock source becomes the internal reference clock (IICφ).

SDOC bit (SDA Output Control) and SCOC bit (SCL Output Control)

These bits are used to directly control the SDAn and SCLn signals output from this module.

When writing to these bits, also write 1 to the SOCWP bit at the same time.

The result of setting these bits is input to IIC via the input buffer. When slave mode is selected, a START condition may be detected and the bus may be released depending on the bit settings.

Do not rewrite these bits during a START condition, STOP condition, Repeated START condition, or during transmission or reception. Operation after rewriting under the above conditions is not guaranteed.

EXCYC bit (Extra SCL Clock Cycle Output)

This bit is used to output an extra SCL clock cycle for debugging or error processing.

Normally, set the bit to 0. Setting the bit to 1 in a normal communication state causes a communication error.

For details on this function, see section 27.3.1.3.6. Port Control , (1)Extra SCL Clock Cycle Output Function.

27.2.12 INCTL : Input Control Register

Base address: IIC_Bn = 0x4011_F000 + 0x0400 × n (n = 0, 1)

Offset address: 0x08C

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

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

Bit	Symbol	Function	R/W
3:0	DNFS[3:0]	Digital Noise Filter Stage Selection 0x0: Noise of up to one IICφ cycle is filtered out (singlestage filter). 0x1: Noise of up to two IICφ cycles is filtered out (2-stage filter). 0x2: Noise of up to three IICφ cycles is filtered out (3-stage filter). 0x3: Noise of up to four IICφ cycles is filtered out (4-stage filter). 0x4: Noise of up to five IICφ cycles is filtered out (5-stage filter). ⋮ 0xF: Noise of up to sixteen IICφ cycles is filtered out (16-stage filter).	R/W
4	DNFE	Digital Noise Filter Circuit Enable 0: No digital noise filter circuit is used. 1: A digital noise filter circuit is used.	R/W
5	—	This bit is read as 0. The write value should be 0.	R/W
7:6	—	These bits are read as 1. The write value should be 1.	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

DNFS[3:0] bits (Digital Noise Filter Stage Selection)

These bits are used to select the number of stages in the digital noise filter.

For details on the digital noise filter function, see section 27.3.1.6.3. Digital Noise-Filter Circuits .

In I^2C High Speed mode, the module changes the number of noise filter stage to a quarter of the number of noise filter stage automatically.

Note:
- Set the noise range to be filtered out by the noise filter within a range less than the SCLn line high-level period or low-level period. If the noise range is set to a value of (SCL clock width: high-level period or lowlevel period, whichever is shorter) - [1.5 internal reference clock (IICφ) cycles] or more, the SCL clock is regarded as noise by the noise filter function of IIC, which may prevent IIC from operating normally.
- In I^2C High Speed mode, the lower 2 bits of the DNFS [3:0] bits are ignored, and the number of filter stages for 1 to 4 stages is selected by the upper 2 bits.

27.2.13 TMOCTL : Timeout Control Register

Base address: IIC_Bn = 0x4011_F000 + 0x0400 × n (n = 0, 1)

Offset address: 0x090

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

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

Bit	Symbol	Function	R/W
1:0	TODTS[1:0]	Timeout Detection Time Selection 0 0: 16bit-timeout 0 1: 14bit-timeout 1 0: 8bit-timeout 1 1: 6bit-timeout	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	TOLCTL	Timeout L Count Control 0: Count is disabled while the SCLn line is at a low level. 1: Count is enabled while the SCLn line is at a low level.	R/W
5	TOHCTL	Timeout H Count Control 0: Count is disabled while the SCLn line is at a high level. 1: Count is enabled while the SCLn line is at a high level.	R/W
7:6	TOMDS[1:0]	Timeout Operation Mode Selection 0 0: Timeout is detected during the following conditions: • The bus is busy (BCST.BFREF = 0) in master mode. • IIC's own slave address is detected and the bus is busy in slave mode. • The bus is free (BCST.BFREF = 1) while generation of a START condition is requested (CNDCTL.STCND = 1). 0 1: Timeout is detected while the bus is busy. 1 0: Timeout is detected while the bus is free. 1 1: Setting prohibited	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

TODTS[1:0] bits (Timeout Detection Time Selection)

These bits are used to select for the timeout detection time when the timeout function is enabled (BSTE.TODE bit = 1).

When these bits are set to 00b, the timeout detection internal counter functions as a 16-bit counter.

When these bits are set to 01b, the counter functions as a 14-bit counter.

When these bits are set to 10b, the counter functions as a 8-bit counter.

When these bits are set to 11b, the counter functions as a 6-bit counter.

While the SCLn line is in the state that enables this counter as specified by bits TOHCTL and TOLCTL, the counter counts up in synchronization with the internal reference clock (IICφ) as a count source.

For details on the timeout function, see section 27.3.1.4.1. Timeout Error Detection .

TOLCTL bit (Timeout L Count Control)

This bit is used to enable or disable the internal counter of the timeout function to count up while the SCLn line is held low when the timeout function is enabled (BSTE.TODE = 1).

TOHCTL bit (Timeout H Count Control)

This bit is used to enable or disable the internal counter of the timeout function to count up while the SCLn line is held high when the timeout function is enabled (BSTE.TODE = 1).

TOMDS[1:0] bits (Timeout Operation Mode Selection)

These bits are used to select the detection condition for timeout when the timeout function is enabled.

27.2.14 WUCTL : Wake Up Unit Control Register

Base address: IIC0WU_B = 0x4011_F098

Offset address: 0x000

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

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

Bit	Symbol	Function	R/W
0	WUACKS	Wake-Up Acknowledge Selection Choice of four response mode with a combination of RSTCTL.INTLRST bit and WUACKS bit. Shown in Table 27.6.	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
4	WUANFS	Wake-Up Analog Noise Filter Selection 0: Do not add the Wake Up analog filter. 1: Add the Wake Up analog filter.	R/W
5	—	This bit is read as 0. The write value should be 0.	R/W
6	WUFSYNE	Wake-Up function PCLKA Synchronous Enable 0: IIC asynchronous circuit enable 1: IIC synchronous circuit enable	R/W
7	WUFE	Wake-Up function Enable Do not set WUFE = 0 during Wake-Up operation. 0: Wake-up function disables 1: Wake-up function enables	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

Table 27.6 Wake-Up Mode

INTLRST	WUACKS	Operation mode	Description
0	0	Normal Wake-Up mode 1	ACK response at 9th SCL and SCL low hold after at 9th SCL.
0	1	Normal Wake-Up mode 2	No ACK response immediately and SCL low hold between 8th and 9th SCL. Release SCL low hold and ACK response at 9th SCL.
1	0	Command recovery mode	ACK response at 9th SCL and not SCL low hold.
1	1	EEP response mode	NACK response at 9th SCL and not SCL low hold.

Note: In WakeUp mode 2, HS mode cannot be used.

WUFSYNE bit (Wake-Up function PCLKA Synchronous Enable)

This bit is used to switch between the PCLKA/IICCLK synchronous operation and the PCLKA/IICCLK asynchronous operation.

The bit is used in combination with the WUASYNF flag at Wake-Up effective function (WUCTL.WUFE bit = 1).

[When switching from the PCLKA/IICCLK synchronous operation to the PCLKA/IICCLK asynchronous operation]

IIC operation changes into the PCLKA/IICCLK asynchronous operation during BCST.BFREF flag = 1, when the WUASYNF flag set to 1 during WUFSYNE = 0.

The reception can operate without depending on the state of operation of PCLKA/IICCLK (With PCLKA/IICCLK stopped) after it switches to the PCLKA/IICCLK asynchronous operation (Wake-Up event detection operation).

[When switching from the PCLKA/IICCLK asynchronous operation to the PCLKA/IICCLK synchronous operation]

IIC operation changes into the PCLKA/IICCLK synchronous operation at the following conditions. (At the same timing when WUFSYNE flag becomes 0)

In the case Wake-Up event detects : right after WUFSYNE bit is set to 1.

In the case Wake-Up event does not detect : when STOP condition is detected after WUFSYNE bit is set to 1.

[Setting condition]

- When 1 is written to the WUFSYNE bit.
- WUCTL.WUFE = 0

[Clearing conditions]

- When 0 is written to the WUFSYNE bit.

27.2.15 ACKCTL : Acknowledge Control Register

Base address: IIC_Bn = 0x4011_F000 + 0x0400 × n (n = 0, 1)

Offset address: 0x0A0

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

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

Bit	Symbol	Function	R/W
0	ACKR	Acknowledge Reception 0: A 0 is received as the acknowledge bit (ACK reception). 1: A 1 is received as the acknowledge bit (NACK reception).	R
1	ACKT	Acknowledge Transmission 0: A 0 is sent as the acknowledge bit (ACK transmission). 1: A 1 is sent as the acknowledge bit (NACK transmission).	R/W
2	ACKTWP	ACKT Write Protect 0: The ACKT bit are protected. 1: The ACKT bit can be written (when writing simultaneously with the value of the target bit). This bit is read as 0.	W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

ACKR bit (Acknowledge Reception)

This bit is used to store the acknowledge bit information received from the receive device in transmit mode.

[Setting condition]

- When 1 is received as the acknowledge bit with the PRSST.TRMD bit set to 1.

[Clearing condition]

- When 0 is received as the acknowledge bit with the PRSST.TRMD bit set to 1.

ACKT bit (Acknowledge Transmission)

[Setting condition]

- When 1 is written to the ACKT bit and 1 is written to the ACKTWP bit at the same time.

[Clearing conditions]

- When 0 is written to the ACKT bit and 1 is written to the ACKTWP bit at the same time.
- When a STOP condition is detected. (when a STOP condition is detected with the CNDCTL.SPCND bit set to 1.)

Note: Set the ACKT bit to 0 in I2C Slave mode.

ACKTWP bit (ACKT Write Protect)

This bit is used to control the modification of the ACKT bit.

When changing the ACKT bit, setting this bit to 1 at the same time can change the ACKT bit.

When this bit is read, 0 is always read.

27.2.16 SCSTRCTL : SCL Stretch Control Register

Base address: IIC_Bn = 0x4011_F000 + 0x0400 × n (n = 0, 1)

Offset address: 0x0A4

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

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

Bit	Symbol	Function	R/W
0	ACKTWE	Acknowledge Transmission Wait Enable 0: NTST.RDBFF0 is set at the rising edge of the ninth SCL clock cycle. (The SCLn line is not held low at the falling edge of the eighth clock cycle.) 1: NTST.RDBFF0 is set at the rising edge of the eighth SCL clock cycle. (The SCLn line is held low at the falling edge of the eighth clock cycle.) Low-hold is released by writing a value to the ACKCTL.ACKT bit.	R/W
1	RWE	Receive Wait Enable 0: No WAIT (The period between ninth clock cycle and first clock cycle is not held low.) 1: WAIT (The period between ninth clock cycle and first clock cycle is held low.) Low-hold is released by reading NTDTBP0.	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

ACKTWE bit (Acknowledge Transmission Wait Enable)

This bit is used to select the NTST.RDBFF0 flag set timing in receive mode and also to select whether to hold the SCLn line low at the falling edge of the eighth SCL clock cycle.

When ACKTWE = 0, the SCLn line is not held low at the falling edge of the eighth SCL clock cycle, and the NTST.RDBFF0 flag is set to 1 at the rising edge of the ninth SCL clock cycle.

When ACKTWE = 1, the NTST.RDBFF0 flag is set to 1 at the rising edge of the eighth SCL clock cycle and the SCLn line is held low at the falling edge of the eighth SCL clock cycle. The low-hold of the SCLn line is released by writing a value to the ACKCTL.ACKT bit.

After data is received with this setting, the SCLn line is automatically held low before the acknowledge bit is sent. This enables processing to send ACK (ACKCTL.ACKT = 0) or NACK (ACKCTL.ACKT = 1) according to receive data.

RWE bit (Receive Wait Enable)

This bit is used to control whether to hold the period between the ninth SCL clock cycle and the first SCL clock cycle low until the receive data buffer (NTDTBP0) is completely read each time single-byte data is received in receive mode.

When RWE = 0, the receive operation is continued without holding the period between the ninth and the first SCL clock cycle low. When both the ACKTWE and RWE bits = 0, continuous receive operation is enabled with the double buffer.

When RWE = 1, the SCLn line is held low from the falling edge of the ninth clock cycle until the NTDTBP0 value is read each time single-byte data is received.

This enables receive operation in byte units.

Note: When the value of the RWE bit is to be read, be sure to read the NTDTBP0 beforehand.

27.2.17 CNDCTL : Condition Control Register

Base address: IIC_Bn = 0x4011_F000 + 0x0400 × n (n = 0, 1)

Offset address: 0x140

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

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

Bit	Symbol	Function	R/W
0	STCND	START (S) Condition Issuance 0: Does not request to issue a START condition. 1: Requests to issue a START condition.	R/W
1	SRCND	Repeated START (Sr) Condition Issuance 0: Does not request to issue a Repeated START condition. 1: Requests to issue a Repeated START condition.	R/W
2	SPCND	STOP (P) Condition Issuance 0: Does not request to issue a STOP condition. 1: Requests to issue a STOP condition.	R/W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

STCND bit (START (S) Condition Issuance)

This bit is used to request transition to master mode and issuance of a START condition.

For details on the START condition issuance, see section 27.3.1.3.2. START Condition / Repeated START Condition / STOP Condition Issuing Function .

[Setting condition]

- When 1 is written to the STCND bit

[Clearing conditions]

- When 0 is written to the STCND bit
- When a START condition has been issued (A START condition is detected)
- When the BST.ALF (arbitration-lost) flag is set to 1

Note: Set the STCND bit to 1 (START condition issuance request) when the BCST.BFREF flag is set to 1 (bus free state).

Note that arbitration may be lost due to a START condition issuance error if the STCND bit is set to 1 (START condition issuance request) when the BFREF flag is set to 0 (bus busy state).

SRCND bit (Repeated START (Sr) Condition Issuance)

This bit is used to request that a Repeated START condition be issued in master mode.

When this bit is set to 1 to request to issue a Repeated START condition, a Repeated START condition is issued when the BFREF flag is set to 0 (bus busy state) and the PRSST.CRMS bit is set to 1 (master mode).

For details on the Repeated START condition issuance, see section 27.3.1.3.2. START Condition / Repeated START Condition / STOP Condition Issuing Function .

[Setting condition]

- When 1 is written to the SRCND bit with the BCST.BFREF flag set to 0

[Clearing conditions]

- When 0 is written to the SRCND bit
- When a Repeated START condition has been issued (A Repeated START condition is detected)
- When the BST.ALF (arbitration-lost) flag is set to 1

Note: Do not set the SRCND bit to 1 while issuing a STOP condition.

Note: If 1 (requests to issue a Repeated START condition) is written to the SRCND bit in slave mode, the Repeated START condition is not issued but the SRCND bit remains set to 1.

If the operating mode changes to master mode with the bit not being cleared, note that the Repeated START condition may be issued.

SPCND bit (STOP (P) Condition Issuance)

This bit is used to request that a STOP condition be issued in master mode.

When this bit is set to 1 to request to issue a STOP condition, a STOP condition is issued when the BCST.BFREF flag is set to 0 (bus busy state) and the PRSST.CRMS bit is set to 1 (master mode).

For details on the STOP condition issuance, see section 27.3.1.3.2. START Condition / Repeated START Condition / STOP Condition Issuing Function .

[Setting condition]

- When 1 is written to the SPCND bit with the BCST.BFREF flag set to 0 and the PRSST.CRMS bit set to 1

[Clearing conditions]

- When 0 is written to the SPCND bit
- When a STOP condition has been issued (A STOP condition is detected)
- When the BST.ALF (arbitration-lost) flag is set to 1
- When a START condition and a Repeated START condition are detected

Note: Writing to the SPCND bit is not possible while the setting of the BCST.BFREF flag = 1 (bus free state).

Note: Do not set the SPCND bit to 1 while a Repeated START condition is being issued.

27.2.18 NTDTBP0/NTDTBP0_BY : Normal Transfer Data Buffer Port Register 0

Base address: IIC_Bn = 0x4011_F000 + 0x0400 × n (n = 0, 1)

Offset address: 0x158

Bit position:	31																												0
Bit field:																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	n/a	Normal Transfer Data Buffer Port NTDTBP0 is a 32-bit read/write register. NTDTBP0_BY (NTDTBP0[7:0]) is a 8-bit read/write register.	R/W

32-bit mailbox register NTDTBP0 is a 32-bit bi-directional data transfer register which is used both to read from the Normal Receive Data Buffer, and to write to the Normal Transmit Data Buffer.

In other words, the Normal Receive Data Buffer and the Normal Transmit Data Buffer have the same offset, forming a single bidirectional port for transmitting or receiving IIC data.

Read Operations:

When 1 byte of data has been received, the received data is transferred from the internal shift register to NTDTBP0 to enable the next data to be received. The double-buffer structure of the internal shift register and NTDTBP0 allows continuous receive operation if the received data has been read from NTDTBP0 while the internal shift register is receiving data. Read data from NTDTBP0 once when a receive data full interrupt (IICn_RX) request is generated. If NTDTBP0 receives the next receive data before the current data is read from NTDTBP0 (while the RDBFF0 flag in NTST is 1), this module automatically holds the SCL clock low one cycle before the RDBFF0 flag is set to 1 next. The lower 8 bits of the read 32-bit data are valid as received data.

Write Operations:

When NTDTBP0 detects a space in the internal shift register, it transfers the transmit data that has been written to NTDTBP0 to the internal shift register and starts transmitting data in transmit mode. The double-buffer structure of NTDTBP0 and the internal shift register allows continuous transmit operation if the next transmit data has been written to NTDTBP0 while the the internal shift register data is being transmitted. Write transmit data to NTDTBP0 once when a transmit data empty interrupt (IICn_TX) request is generated. The lower 8 bits of the written 32-bit data are valid as transmission data.

27.2.19 BST : Bus Status Register

Base address: IIC_Bn = 0x4011_F000 + 0x0400 × n (n = 0, 1)

Offset address: 0x1D0

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

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

Bit	Symbol	Function	R/W
0	STCNDDF	START Condition Detection Flag 0: START condition is not detected. 1: START condition is detected.	R/W*1
1	SPCNDDF	STOP Condition Detection Flag 0: STOP condition is not detected. 1: STOP condition is detected.	R/W*1
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	NACKDF	NACK Detection Flag 0: NACK is not detected. 1: NACK is detected.	R/W*1
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
8	TENDF	Transmit End Flag 0: Data is being transmitted. 1: Data has been transmitted.	R/W*1
15:9	—	These bits are read as 0. The write value should be 0.	R/W
16	ALF	Arbitration Lost Flag 0: Arbitration is not lost 1: Arbitration is lost.	R/W*1
19:17	—	These bits are read as 0. The write value should be 0.	R/W
20	TODF	Timeout Detection Flag 0: Timeout is not detected. 1: Timeout is detected.	R/W*1
23:21	—	These bits are read as 0. The write value should be 0.	R/W
24	WUCNDDF*2	Wake-Up Condition Detection Flag 0: Wake-Up is not detected. 1: Wake-Up is detected.	R/W*1
31:25	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Clearing (to 0) condition : Writing 0 after 1 is read.
Note 2. This bit is available only IIC0. This bit are read as 0 and the write value should be 0 in IIC1.

STCNDDF bit (START Condition Detection Flag)

[Setting conditions]

- All of the followings are satisfied:
  1. The BSTE.STCNDDE bit = 1.
  2. When a START condition (or a Repeated START condition) is detected.

[Clearing conditions]

- When 0 is written to the STCNDDF flag after reading STCNDDF flag = 1.
- When a STOP condition is detected.

SPCNDDF bit (STOP Condition Detection Flag)

[Setting conditions]

- All of the followings are satisfied:
  1. The BSTE.SPCNDDE bit = 1.
  2. When a STOP condition is detected.

[Clearing condition]

- When 0 is written to the SPCNDDF flag after reading SPCNDDF flag = 1.

NACKDF bit (NACK Detection Flag)

[Setting conditions]

- All of the followings are satisfied:
  1. The BSTE.NACKDE bit = 1 (Enables NACK detection interrupt status logging).
  2. When acknowledge is not received (NACK is received) from the receive device in transmit mode.

[Clearing condition]

- When 0 is written to the NACKDF flag after reading NACKDF flag = 1.

TENDF bit (Transmit End Flag)

[Setting conditions]

- All of the followings are satisfied:

1. The BSTE.TENDE bit = 1 (Enables Transmit End Interrupt Status logging).
2. At the rising edge of the ninth SCL clock cycle while the NTST.TDBEF0 flag = 1.
   Excluding when sending an address.

[Clearing conditions]

- When 0 is written to the TENDF flag after reading TENDF flag = 1.
- When data is written to the NTDTBP0 register.
- When a STOP condition is detected.

ALF bit (Arbitration Lost Flag)

[Setting conditions]

When master arbitration-lost detection is enabled: BSTE.ALE bit = 1, BFCTL.MALE = 1.

- When the internal SDA output state does not match the SDA line level at the rising edge of SCL clock except for the ACK period during data (including slave address) transmission in master transmit mode (when the SDA line is driven low while the internal SDA output is at a high level (the SDA pin is in the highimpedance state)).
- All of the followings are satisfied.
  1. When the START condition is detected while the CNDCTL.STCND bit = 1.
  2. When the internal SDA output state does not match the SDA line level.
- When the CNDCTL.STCND bit is set to 1 (START condition issuance request) while the BCST.BFREF flag = 0.

When NACK arbitration-lost detection is enabled: BSTE.ALE bit = 1, BFCTL.NALE = 1.

- When the internal SDA output state does not match the SDA line level at the rising edge of SCL clock in the ACK period during NACK transmission in receive mode.

When slave arbitration-lost detection is enabled: BSTE.ALE bit = 1, BFCTL.SALE = 1.

- When the internal SDA output state does not match the SDA line level at the rising edge of SCL clock except for the ACK period during data transmission in slave transmit mode.

[Clearing condition]

- When 0 is written to the ALF flag after reading ALF flag = 1.

TODF bit (Timeout Detection Flag)

[Setting conditions]

- All of the followings are satisfied.
  1. The BSTE.TODE bit = 1 (Enables Timeout Detection Interrupt Status logging).
  2. When the master mode or the received slave address matches the slave address n in Slave mode.
  3. When the SCL line state remains unchanged for the period specified by TMOCTL register.

[Clearing condition]

- When 0 is written to the TODF flag after reading TODF flag = 1.

WUCNDDF bit (Wake-Up Condition Detection Flag)

[Setting condition]

- When PCLKA and IICCLK are supplied after all of the followings are satisfied.
  1. The WUCTL.WUFE bit = 1 (Wake-up function is enabled).
  2. The BSTE.WUCNDDE bit = 1 (Enables Wake-up Condition Detection Status logging).
  3. The WUST.WUASYNF flag = 1.
  4. When the address received in slave mode matches the address of slave enabled in the SVCTL.SVAEy bit (except for the Device-ID address).

[Clearing condition]

- When 0 is written to the WUCNDDF flag after reading WUCNDDF flag = 1 while the WUST.WUASYNF flag = 0.

27.2.20 BSTE : Bus Status Enable Register

Base address: IIC_Bn = 0x4011_F000 + 0x0400 × n (n = 0, 1)

Offset address: 0x1D4

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

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

Bit	Symbol	Function	R/W
0	STCNDDE	START Condition Detection Enable 0: Disables START condition Detection Interrupt Status logging. 1: Enables START condition Detection Interrupt Status logging.	R/W
1	SPCNDDE	STOP Condition Detection Enable 0: Disables STOP condition Detection Interrupt Status logging. 1: Enables STOP condition Detection Interrupt Status logging.	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	NACKDE	NACK Detection Enable 0: Disables NACK Detection Interrupt Status logging. 1: Enables NACK Detection Interrupt Status logging.	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
8	TENDE	Transmit End Enable 0: Disables Transmit End Interrupt Status logging. 1: Enables Transmit End Interrupt Status logging.	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W
16	ALE	Arbitration Lost Enable 0: Disables Arbitration Lost Interrupt Status logging. 1: Enables Arbitration Lost Interrupt Status logging.	R/W
19:17	—	These bits are read as 0. The write value should be 0.	R/W
20	TODE	Timeout Detection Enable 0: Disables Timeout Detection Interrupt Status logging. 1: Enables Timeout Detection Interrupt Status logging.	R/W
23:21	—	These bits are read as 0. The write value should be 0.	R/W
24	WUCNDDE*1	Wake-up Condition Detection Enable 0: Disables Wake-up Condition Detection Status logging. 1: Enables Wake-up Condition Detection Status logging.	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. This bit is available only IIC0. This bit are read as 0 and the write value should be 0 in IIC1.

STCNDDE bit (START Condition Detection Enable)

When this bit is 1, operation of BST.STCNDDF is enabled. For the setting conditions and clearing conditions of the BST.STCNDDF flag, see the details of BST.STCNDDF.

SPCNDDE bit (STOP Condition Detection Enable)

When this bit is 1, operation of BST.SPCNDDF is enabled. For the setting conditions and clearing conditions of the BST.SPCNDDF flag, see the details of BST.SPCNDDF.

NACKDE bit (NACK Detection Enable)

When this bit is 1, the operation of BST.NACKDF is enabled. This bit is used to specify whether to continue or discontinue the transfer operation when NACK is received from the slave device in transmit mode. Normally, set this bit to 1. For the setting conditions and clearing conditions of the BST.NACKDF flag, see the details of BST.NACKDF.

TENDE bit (Transmit End Enable)

When this bit is 1, the operation of BST.TENDF is enabled. For the setting conditions and clearing conditions of the BST.TENDF flag, see the details of BST.TENDF.

ALE bit (Arbitration Lost Enable)

When this bit is 1, the operation of BST.ALF is enabled. For the setting conditions and clearing conditions of the BST.ALF flag, see the details of BST.ALF.

TODE bit (Timeout Detection Enable)

When this bit is 1, the operation of BST.TODF is enabled. For the setting conditions and clearing conditions of the BST.TODF flag, see the details of BST.TODF.

WUCNDDE bit (Wake-up Condition Detection Enable)

When this bit is 1, the operation of BST.WUCNDDF is enabled. For the setting conditions and clearing conditions of the BST.WUCNDDF flag, see the details of BST.WUCNDDF.

27.2.21 BIE : Bus Interrupt Enable Register

Base address: IIC_Bn = 0x4011_F000 + 0x0400 × n (n = 0, 1)

Offset address: 0x1D8

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

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

Bit	Symbol	Function	R/W
0	STCNDDIE	START Condition Detection Interrupt Enable 0: Disables START condition Detection Interrupt Signal. 1: Enables START condition Detection Interrupt Signal.	R/W
1	SPCNDDIE	STOP Condition Detection Interrupt Enable 0: Disables STOP condition Detection Interrupt Signal. 1: Enables STOP condition Detection Interrupt Signal.	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	NACKDIE	NACK Detection Interrupt Enable 0: Disables NACK Detection Interrupt Signal. 1: Enables NACK Detection Interrupt Signal.	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
8	TENDIE	Transmit End Interrupt Enable 0: Disables Transmit End Interrupt Signal. 1: Enables Transmit End Interrupt Signal.	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W
16	ALIE	Arbitration Lost Interrupt Enable 0: Disables Arbitration Lost Interrupt Signal. 1: Enables Arbitration Lost Interrupt Signal.	R/W

Bit	Symbol	Function	R/W
19:17	—	These bits are read as 0. The write value should be 0.	R/W
20	TODIE	Timeout Detection Interrupt Enable 0: Disables Timeout Detection Interrupt Signal. 1: Enables Timeout Detection Interrupt Signal.	R/W
23:21	—	These bits are read as 0. The write value should be 0.	R/W
24	WUCNDDIE*1	Wake-Up Condition Detection Interrupt Enable 0: Disables Wake-Up Condition Detection Interrupt Signal. 1: Enables Wake-Up Condition Detection Interrupt Signal.	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. This bit is available only IIC0. This bit are read as 0 and the write value should be 0 in IIC1.

The BIE register enables signaling of outstanding bus interrupts received by IIC.

STCNDDIE bit (START Condition Detection Interrupt Enable)

This bit enables or disables the START Condition Detection interrupt requests when the BST.STCNDDF flag is set to 1.

SPCNDDIE bit (STOP Condition Detection Interrupt Enable)

This bit enables or disables the STOP Condition Detection interrupt requests when the BST.SPCNDDF flag is set to 1.

NACKDIE bit (NACK Detection Interrupt Enable)

This bit enables or disables the NACK Detection interrupt requests when the BST.NACKDF flag is set to 1.

TENDIE bit (Transmit End Interrupt Enable)

This bit enables or disables the Transmit End interrupt (IICn_TEND) requests when the BST.TENDF flag is set to 1.

ALIE bit (Arbitration Lost Interrupt Enable)

This bit enables or disables the Arbitration Llost interrupt requests when the BST.ALF flag is set to 1.

TODIE bit (Timeout Detection Interrupt Enable)

This bit enables or disables the Timeout Detection interrupt requests when the BST.TODF flag is set to 1.

WUCNDDIE bit (Wake-Up Condition Detection Interrupt Enable)

This bit enables or disables the Wake-up Condition Detection interrupt (IIC0_WU) requests when the BST.WUCNDDF flag is set to 1.

27.2.22 BSTFC : Bus Status Force Register

Base address: IIC_Bn = 0x4011_F000 + 0x0400 × n (n = 0, 1)

Offset address: 0x1DC

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

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

Bit	Symbol	Function	R/W
0	STCNDDFC	START condition Detection Force 0: Not Force START condition Detection Interrupt for software testing. 1: Force START condition Detection Interrupt for software testing.	W

Bit	Symbol	Function	R/W
1	SPCNDDFC	STOP condition Detection Force 0: Not Force STOP condition Detection Interrupt for software testing. 1: Force STOP condition Detection Interrupt for software testing.	W
3:2	—	These bits are read as 0.	R
4	NACKDFC	NACK Detection Force 0: Not Force NACK Detection Interrupt for software testing. 1: Force NACK Detection Interrupt for software testing.	W
7:5	—	These bits are read as 0.	R
8	TENDFC*1	Transmit End Force 0: Not Force Transmit End Interrupt for software testing. 1: Force Transmit End Interrupt for software testing.	W
15:9	—	These bits are read as 0.	R
16	ALFC	Arbitration Lost Force 0: Not Force Arbitration Lost Interrupt for software testing. 1: Force Arbitration Lost Interrupt for software testing.	W
19:17	—	These bits are read as 0.	R
20	TODFC	Timeout Detection Force 0: Not Force Timeout Detection Interrupt for software testing. 1: Force Timeout Detection Interrupt for software testing.	W
23:21	—	These bits are read as 0.	R
24	WUCNDDFC*2	Wake-Up Condition Detection Force 0: Not Force Wake-Up Condition Detection Interrupt for software testing. 1: Force Wake-Up Condition Detection Interrupt for software testing.	W
31:25	—	These bits are read as 0.	R

Note 1. TENDFC does not work unless TDBEF0 = 1.
Note 2. This bit is available only IIC0. This bit are read as 0 and the write value should be 0 in IIC1.

27.2.23 NTST : Normal Transfer Status Register

Base address: IIC_Bn = 0x4011_F000 + 0x0400 × n (n = 0, 1)

Offset address: 0x1E0

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RDBFF0	TDBEF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TDBEF0	Normal Transmit Data Buffer Empty Flag 0 0: Normal Transmit Data Buffer 0 contains transmit data. 1: Normal Transmit Data Buffer 0 contains no transmit data.	R/W*1
1	RDBFF0	Normal Receive Data Buffer Full Flag 0 0: Normal Receive Data Buffer0 contains no receive data. 1: Normal Receive Data Buffer0 contains receive data.	R/W*1
31:2	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Clearing (to 0) condition : Writing 0 after the 1 state is read.

TDBEF0 bit (Normal Transmit Data Buffer Empty Flag 0)

[Setting conditions]

The following condition 1 is satisfied and any of the following conditions 2 to 4 are satisfied:

1. The NTSTE.TDBEE0 bit = 1 (enables Tx0 Data Buffer Empty Interrupt Status logging).
2. When data has been transferred from the Normal Transmit Data Buffer 0 to the Shift Register and the Normal Transmit Data Buffer 0 becomes empty*1.
3. When the PRSST.TRMD bit is set to 1.
4. When the received slave address matches while the TRMD bit = 1.

[Clearing conditions]

- When data is written to NTDTBP0.
- When the TRMD bit in PRSST is set to 0.

Note 1. When the BST.NACKDF flag is set to 1 while the BSTE.NACKDE bit = 1, IIC aborts data transmission/reception. If the TDBEF0 flag = 0 (next transmit data has been written), data is transferred to the Shift Register and the Normal Transmit Data Buffer 0 register becomes empty at the rising edge of the 9th clock cycle, but the TDBEF0 flag is not set to 1.

RDBFF0 bit (Normal Receive Data Buffer Full Flag 0)

[Setting conditions]

The following condition 1 is satisfied and any of the following condition 2 or 3 is satisfied:

1. The NTSTE.RDBFE0 bit = 1 (enables Rx0 Data Buffer Full Interrupt Status logging).
2. When receive data is transferred from Shift Register to Normal Receive Data Buffer 0.
   The RDBFF0 flag is set to 1 on the rising edge of the 8th or 9th SCL clock cycle (selected in the ACKTWE bit in SCSTRCTL).
3. When the received slave address matches after a START (or Repeated START) condition is detected with the TRMD bit in PRSST set to 0.

[Clearing conditions]

- When data is read from NTDTBP0.

27.2.24 NTSTE : Normal Transfer Status Enable Register

Base address: IIC_Bn = 0x4011_F000 + 0x0400 × n (n = 0, 1)

Offset address: 0x1E4

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RDBFE0	TDBEE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TDBEE0	Normal Transmit Data Buffer Empty Enable 0 0: Disables Tx0 Data Buffer Empty Interrupt Status logging. 1: Enables Tx0 Data Buffer Empty Interrupt Status logging.	R/W
1	RDBFE0	Normal Receive Data Buffer Full Enable 0 0: Disables Rx0 Data Buffer Full Interrupt Status logging. 1: Enables Rx0 Data Buffer Full Interrupt Status logging.	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

TDBEE0 bit (Normal Transmit Data Buffer Empty Enable 0)

When this bit is 1, the operation of NTST.TDBEF0 is enabled.

For the setting conditions and clearing conditions of the NTST.TDBEF0 flag, see the details of NTST.TDBEF0.

RDBFE0 bit (Normal Receive Data Buffer Full Enable 0)

When this bit is 1, the operation of NTST.RDBFF0 is enabled.

For the setting conditions and clearing conditions of the NTST.RDBFF0 flag, see the details of NTST.RDBFF0.

27.2.25 NTIE : Normal Transfer Interrupt Enable Register

Base address: IIC_Bn = 0x4011_F000 + 0x0400 × n (n = 0, 1)

Offset address: 0x1E8

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RDBFIE0	TDBEIE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TDBEIE0	Normal Transmit Data Buffer Empty Interrupt Enable 0 0: Disables Tx0 Data Buffer Empty Interrupt Signal. 1: Enables Tx0 Data Buffer Empty Interrupt Signal.	R/W
1	RDBFIE0	Normal Receive Data Buffer Full Interrupt Enable 0 0: Disables Rx0 Data Buffer Full Interrupt Signal. 1: Enables Rx0 Data Buffer Full Interrupt Signal.	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

The PIO Interrupt Signal Enable register enables signaling of outstanding interrupts received by IIC.

TDBEIE0 bit (Normal Transmit Data Buffer Empty Interrupt Enable 0)

This bit is used to enable or disable the Normal Tx Data buffer 0 empty interrupt (IICn_TX) requests when the NTST.TDBEF0 flag is set to 1.

RDBFIE0 bit (Normal Receive Data Buffer Full Interrupt Enable 0)

This bit is used to enable or disable the Normal Rx Data buffer 0 full interrupt (IICn_RX) requests when the NTST.RDBFF0 flag is set to 1.

27.2.26 NTSTFC : Normal Transfer Status Force Register

Base address: IIC_Bn = 0x4011_F000 + 0x0400 × n (n = 0, 1)

Offset address: 0x1EC

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RDBFFC0	TDBEFC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TDBEFC0	Normal Transmit Data Buffer Empty Force 0 0: Not Force Tx0 Data Buffer Empty Interrupt for software testing. 1: Force Tx0 Data Buffer Empty Interrupt for software testing.	W
1	RDBFFC0	Normal Receive Data Buffer Full Force 0 0: Not Force Rx0 Data Buffer Full Interrupt for software testing. 1: Force Rx0 Data Buffer Full Interrupt for software testing.	W
31:2	—	The write value should be 0.	W

The PIO Interrupt Force register is used to force specific interrupt. It can be used for debug purposes.

TDBEFC0 bit (Normal Transmit Data Buffer Empty Force 0)

For software testing, when set to 1, forces the corresponding interrupt, subject to TDBEE0 and TDBEIE0 configuration.

RDBFFC0 bit (Normal Receive Data Buffer Full Force 0)

For software testing, when set to 1, forces the corresponding interrupt, subject to RDBFE0 and RDBFIE0 configuration.

27.2.27 BCST : Bus Condition Status Register

Base address: IIC_Bn = 0x4011_F000 + 0x0400 × n (n = 0, 1)

Offset address: 0x210

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

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

Bit	Symbol	Function	R/W
0	BFREF	Bus Free Detection Flag 0: Have not Detected Bus Free 1: Have Detected Bus Free	R
31:1	—	These bits are read as 0.	R

BFREF bit (Bus Free Detection Flag)

[Setting conditions]

- After a STOP condition is detected, when the number of cycles (IICφ) that are set by BFRECDT.FRECYC[8:0] has passed in the state of SCL = SDA = 1.
- After setting BCTL.BUSE to 1, when the number of cycles (IICφ) that are set by BFRECDT.FRECYC[8:0] has passed in the state of SCL = SDA = 1.

[Clearing conditions]

- When SCL and SDA are other than high.
- When the BCTL.BUSE bit is set to 0.

27.2.28 SVST : Slave Status Register

Base address: IIC_Bn = 0x4011_F000 + 0x0400 × n (n = 0, 1)

Offset address: 0x214

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	SVAF2	SVAF1	SVAF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
0	GCAF	General Call Address Detection Flag 0: General call address does not detect. 1: General call address detects.	R/W*1
4:1	—	These bits are read as 0. The write value should be 0.	R/W
5	HSMCF*2	Hs-mode Master Code Detection Flag 0: Hs-mode Master Code does not detect. 1: Hs-mode Master Code detects.	R/W*1
6	DVIDF	Device-ID Address Detection Flag 0: Device-ID command does not detect. 1: Device-ID command detects. ● This bit set to 1 when the first frame received immediately after a START condition is detected matches a value of (device ID (1111 100) + 0[W]).	R/W*1
14:7	—	These bits are read as 0. The write value should be 0.	R/W
15	HOAF	Host Address Detection Flag 0: Host address does not detect. 1: Host address detects. ● This bit set to 1 when the received slave address matches the host address (0001 000).	R/W*1
16	SVAF0	Slave Address Detection Flag 0 0: Slave 0 does not detect 1: Slave 0 detect	R/W*1
17	SVAF1	Slave Address Detection Flag 1 0: Slave 1 does not detect 1: Slave 1 detect	R/W*1
18	SVAF2	Slave Address Detection Flag 2 0: Slave 2 does not detect 1: Slave 2 detect	R/W*1
31:19	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Clearing (to 0) condition : Writing 0 after the 1 state is read.
Note 2. The HSMCF bit is supported by IIC0(SCL0_A, SDA0_A). The HSMCF is a reserved bit in the not supported channel.

GCAF flag (General Call Address Detection Flag)

I2C Normal Wake-Up Mode1 / 2 sets GCAF to 1 when switching from asynchronous operation to synchronous unit.

[Setting condition]

- This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the first byte when all of the followings are satisfied.
  1. The SVCTL.GCAE bit = 1 (General call address detection is enabled).
  2. When the received slave address matches the general call address (0000 000 + 0 (write)).

[Clearing condition]

- When 0 is written to the GCAF flag after reading GCAF flag to be 1.
- When a STOP condition is detected.
- When a Repeated START condition is detected.

HSMCF flag (Hs-mode Master Code Detection Flag)

The I^{2}C Normal Wake-Up Mode 1/2 sets 1 to HSMCF when switching from asynchronous operation to synchronous unit.

[Setting condition]

- This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the first byte when all of the followings are satisfied.
  1. The SVCTL.HSMCE bit = 1 (Hs-mode master code detection is enabled).
  2. When the first byte received immediately after a START condition is detected matches a value of Hs-mode master code (0000 1XXX) + 1 (NACK).

[Clearing condition]

- When 0 is written to the HSMCF flag after reading HSMCF flag to be 1.
- When a STOP condition is detected.

DVIDF flag (Device-ID Address Detection Flag)

[Setting condition]

- This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the first byte when all of the followings are satisfied.
  1. The SVCTL.DVIDE bit = 1 (Device-ID address detection is enabled).
  2. When the first byte received immediately after a START condition or Repeated START condition is detected matches a value of Device ID (1111 100) + 0 (write).

[Clearing condition]

- When 0 is written to the DVIDF flag after reading DVIDF flag to be 1.
- When a STOP condition is detected.
- This flag is set to 0 at the rising edge of the ninth SCL clock cycle in the first byte when the following 1 and 2 or 1 and 3 are satisfied.
  1. The SVCTL.DVIDE bit = 1 (Device-ID address detection is enabled).
  2. When the first byte received immediately after a START condition or Repeated START condition is detected does not match a value of Device ID (1111 100).
  3. When the first byte received immediately after a START condition or Repeated START condition is detected matches a value of (device ID (1111 100) + 0 [W]) and the second byte does not match any of slave addresses 0 to 2.

HOAF flag (Host Address Detection Flag)

I^{2}C Normal Wake-Up Mode1 / 2 sets HOAF to 1 at the time of switching from asynchronous operation to synchronous unit.

[Setting condition]

- This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the first byte when all of the followings are satisfied.
  1. The SVCTL.HOAE bit = 1 (Host address detection is enabled).
  2. When the received slave address matches the host address (0001 000).

[Clearing condition]

- When 0 is written to the HOAF flag after reading HOAF flag to be 1.
- When a STOP condition is detected.
- When a Repeated START condition is detected.

SVAFy flags (Slave Address Detection Flag y (y = 0 to 2))

I^2C Normal Wake-Up Mode1 / 2 sets 1 to SVAF2 / 1 / 0 when switching from asynchronous operation to synchronous unit.

[Setting condition]

For 7-bit address format: SVDVADy.SADLG bit = 0.

- This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the first byte when all of the followings are satisfied.
  1. The SVCTL.SVAEy bit = 1 (Slave n enabled).
  2. When the received slave address matches the SVDVADy.SVAD[6:0] bits value.

For 10-bit address format: SVDVADy.SADLG bit = 1.

- This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the second byte when all of the followings are satisfied.
  1. The SVCTL.SVAEy bit = 1 (Slave n enabled).
  2. When the received slave address matches a value of 11110 + SVDVADy.SVAD[9:8] bits and the following address matches the SVDVADy.SVAD[7:0] value.

[Clearing condition]

- When 0 is written to the SVAFy flag after reading SVAFy flag to be 1.
- When a STOP condition is detected.

For 7-bit address format: SVDVADy.SADLG bit = 0.

- This flag is set to 0 at the rising edge of the ninth SCL clock cycle in the first byte when all of the followings are satisfied.
  1. The SVCTL.SVAEy bit = 1 (Slave y enabled).
  2. When the received slave address does not match SVDVADy.SVAD[6:0] bits value.

For 10-bit address format: SVDVADy.SADLG bit = 1.

- This flag is set to 0 at the rising edge of the ninth SCL clock cycle in the first byte when all of the followings are satisfied.
  1. The SVCTL.SVAEy bit = 1 (Slave y enabled).
  2. When the received slave address does not match a value of 11110 + SVDVADy.SVAD[9:8] bits.
- This flag is set to 0 at the rising edge of the ninth SCL clock cycle in the second byte when all of the followings are satisfied.
  1. The SVCTL.SVAEy bit = 1 (Slave y enabled).
  2. When the received slave address matches a value of 11110 + SVDVADy.SVAD[9:8] bits and the following address does not match the SVDVADy.SVAD[7:0] value.

27.2.29 WUST : Wake Up Unit Operating Status Register

Base address: IIC0WU_B = 0x4011_F098

Offset address: 0x180

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

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

Bit	Symbol	Function	R/W
0	WUASYNF	Wake-up function asynchronous operation status flag 0: IIC synchronous circuit enable condition. 1: IIC asynchronous circuit enable condition.	R
31:1	—	These bits are read as 0.	R

WUASYNF flag (Wake-up function asynchronous operation status flag)

This bit shows whether IIC is in the PCLKA/IICCLK asynchronous operation (WUCTL.WUFE bit = 1).

[Setting condition]

- All of the followings are satisfied.
  1. The WUCTL.WUFE bit = 1 (Wake-up function is enabled)
  2. When the BCST.BFREF flag = 1 after 0 is written to the WUCTL.WUFSYNE bit

[Clearing condition]

- The WUCTL.WUFE bit = 0 (Wake-up function is disabled)
- All of the followings are satisfied.
  1. The WUCTL.WUFE bit = 1 (Wake-up function is enabled)
  2. Wake-up event is detected
  3. When 1 is written to the WUCTL.WUFSYNE bit during WUASYNF flag = 1

27.2.30 SDATBASy : Slave Device Address Table Basic Register y (y = 0 to 2)

Base address: IIC_Bn = 0x4011_F000 + 0x0400 × n (n = 0, 1)

Offset address: 0x2B0 + 0x04 × y

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

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

Bit	Symbol	Function	R/W
9:0	SDSTAD[9:0]	Slave Device Static Address IIC Static Address	R/W

Bit	Symbol	Function	R/W
10	SDADLS	Slave Device Address Length Selection 0: Slave Device address length 7 bits selected. 1: Slave Device address length 10 bits selected. (I^2C device only)	R/W
31:11	—	These bits are read as 0. The write value should be 0.	R/W

Note: SW write to the SDATBAS register of the main master is prohibited.

SDSTAD[9:0] bit (Slave Device Static Address)

When the 7-bit address format is selected (SDADLS bit is 0), the lower 7 bits of SDSTAD[9:0] function as the 7-bit address. When the 10-bit address format is selected (SDADLS bit is 1), the SDSTAD[9:0] function as the 10-bit address. While the SVCTL.SVAEy bit is 0, the setting of this bit is ignored.

27.2.31 SVDVADy : Slave Device Address Register y (y = 0 to 2)

Base address: IIC_Bn = 0x4011_F000 + 0x0400 × n (n = 0, 1)

Offset address: 0x330 + 0x04 × y

Bit position:	31	30	29	28	27	26	25									16
Bit field:	—	SSTADV	—	—	SADLG	—	SVAD[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
15:0	—	These bits are read as 0.	R
25:16	SVAD[9:0]	Slave Address A slave address is set. When rewriting SVAD, change to SVAE = 0 and rewrite.	R
26	—	This bit is read as 0.	R
27	SADLG	Slave Address Length 0: The 7-bit address format is selected. 1: The 10-bit address format is selected.	R
29:28	—	These bits are read as 0.	R
30	SSTADV	Slave Static Address Valid 0: Slave address is disabled. 1: Slave address is enabled.	R
31	—	These bits are read as 0.	R

SVAD[9:0] bits (Slave Address)

The SVAD[9:0] bits indicate a valid slave address.

[The SVDVAD0.SDYADV bit = 1]

Note: This condition is only for SVDVAD0.SVAD[9:0].

- The SVAD[9:7] bits = 0
- The SVAD[6:0] bits = the SDATBAS0.SDDYAD[6:0] bits

[The SVDVADy.SSTADV bit = 1 and the SVDVADy.SADLG bit = 0]

- The SVAD[9:7] bits = 0
- The SVAD[6:0] bits = the SDATBASy.SDSTAD[6:0] bits

[The SVDVADy.SSTADV bit = 1 and the SVDVADy.SADLG bit = 1]

- The SVAD[9:0] bits = the SDATBASy.SDSTAD[9:0] bits

SADLG bit (Slave Address Length)

[Setting conditions]

- All of the followings are satisfied:

1. The SVCTL.SVAEy bit = 1 (Slave y is enabled)
2. The SDATBASy.SDADLS bit = 1 (The address length is 10 bits)

[Clearing condition]

- [Setting condition] is not satisfied.

SSTADV bit (Slave Static Address Valid)

[Setting conditions]

- All of the followings are satisfied:

1. The SVCTL.SVAEy bit = 1 (Slave y is enabled)
2. The SVDVAD0.SDYADV bit = 0 (Dynamic Address is disabled)

   Note: This condition is only for SVDVAD0.SSTADV.
3. If the SVDVADy.SADLG bit = 0, the SDATBASy.SDSTAD[6:0] bits are not all 0
   If the SVDVADy.SADLG bit = 1, the SDATBASy.SDSTAD[9:0] bits are not all 0

[Clearing condition]

- [Setting condition] is not satisfied.

27.2.32 BITCNT : Bit Count Register

Base address: IIC_Bn = 0x4011_F000 + 0x0400 × n (n = 0, 1)

Offset address: 0x380

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

Bit	Symbol	Function	R/W
4:0	BCNT[4:0]	Bit Counter Indicates the number of bits remaining to be transferred. For details on the values, see Table 27.7.	R
31:5	—	These bits are read as 0.	R

BCNT[4:0] bits (Bit Counter)

These bits function as a counter that indicates the number of bits remaining to be transferred at the detection of a sampling edge on the SCLn line.

Table 27.7 I²C transfer (1 of 2)

BCNT[4:0]	Master		Slave	
	Address phase	Data phase	Address phase	Data phase
0x00	2 to 1 bits	2 to 1 bits	3 to 1 bits	2 to 1 bits
0x01	3 bits	3 bits	4 bits	3 bits

Table 27.7 I^2C transfer (2 of 2)

BCNT[4:0]	Master		Slave	
	Address phase	Data phase	Address phase	Data phase
0x02	4 bits	4 bits	5 bits	4 bits
0x03	5 bits	5 bits	6 bits	5 bits
0x04	6 bits	6 bits	7 bits	6 bits
0x05	7 bits	7 bits	8 bits	7 bits
0x06	8 bits	8 bits	9 bits	8 bits
0x07	9 bits	9 bits	—	9 bits

27.2.33 PRSTDBG : Present State Debug Register

Base address: IIC_Bn = 0x4011_F000 + 0x0400 × n (n = 0, 1)

Offset address: 0x3CC

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

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

Bit	Symbol	Function	R/W
0	SCILV	SCL Line Signal Level This bit is used to check the SCL Line level, in order to recover from errors and for debugging.	R
1	SDILV	SDA Line Signal Level This bit is used to check the SDA Line level, in order to recover from errors and for debugging.	R
2	SCOLV	SCL Output Level 0: IIC has driven the SCL pin low. 1: IIC has released the SCL pin.	R
3	SDOLV	SDA Output Level 0: IIC has driven the SDA pin low. 1: IIC has released the SDA pin.	R
31:4	—	These bits are read as 0.	R

SDOLV bit (SDA Output Level) and SCOLV bit (SCL Output Level)

When reading these bits, the state of signals output from IIC can be read.

27.3 Operation

27.3.1 Details of Function

27.3.1.1 Operation Mode

IIC has 2 operation modes which are master mode operation and slave mode operation.

27.3.1.1.1 Master Mode Operation

(1) I2C Master Operation

(a) Data Write Transfer (Single Buffer transfer)

In master transmit operation, IIC outputs the SCL clock and transmitted data signals as the master device, and the slave device returns acknowledgments. Figure 27.47 shows an example of usage of master transmission and Figure 27.2 to Figure 27.4 show the timing of operations in master transmission.

The following describes the procedure and operations for master transmission.

1. Initial settings. For details, see section 27.3.2.1. Initial Setting Flow.
2. Read the BCST.BFREF flag to check that the bus is open, and then set the CNDCTL.STCND bit to 1 (START condition issuance request). Upon receiving the request, IIC issues a START condition. At the same time, the BFREF flag bit is automatically set to 0, the BST.STCNDDF flag is automatically set to 1 and the STCND bit is automatically set to 0. At this time, if the START condition is detected and the internal levels for the SDA output state and the levels on the SDAn line have matched while the STCND bit = 1, IIC recognizes that issuing of the START condition as requested by the STCND bit has been successfully completed, and bits CRMS and TRMD in the PRSST register are automatically set to 1, placing IIC in master transmit mode. The NTST.TDBEF0 flag is also automatically set to 1 in response to setting of the TRMD bit to 1.
3. Check that the NTST.TDBEF0 flag = 1, and then write the value for transmission (the slave address and the R/W# bit) to the NTDTBP0 register. Once the data for transmission are written to the NTDTBP0 register, the TDBEF0 flag is automatically set to 0, the data are transferred from the Normal Transmit Data Buffer 0 to the Shift Register, and the TDBEF0 flag is again set to 1. After the byte containing the slave address and R/W# bit has been transmitted, the value of the TRMD bit is automatically updated to select master transmit or master receive mode in accord with the value of the transmitted R/W# bit. If the value of the R/W# bit was 0, IIC continues in master transmit mode. Because the BST.NACKDF flag being 1 at this time indicates that no slave device recognized the address or there was an error in communications, write 1 to the CNDCTL.SPCND bit to issue a STOP condition. For data transmission with an address in the 10-bit format, start by writing 1111 0, the 2 higher-order bits of the slave address, and W to the NTDTBP0 register as the first address transmission. Then, as the second address transmission, write the 8 lower-order bits of the slave address to the NTDTBP0 register.
4. After confirming that the NTST.TDBEF0 flag = 1, write the data for transmission to the NTDTBP0 register. IIC automatically holds the SCLn line low until the data for transmission are ready or a STOP condition is issued.
5. After all bytes of data for transmission have been written to the NTDTBP0 register, wait until the value of the BST.TENDF flag returns to 1, and then set the CNDCTL.SPCND bit to 1 (STOP condition issuance request). Upon receiving a STOP condition issuance request, IIC issues the STOP condition.
6. Upon detecting the STOP condition, IIC automatically sets bits CRMS and TRMD in the PRSST register to 00 and enters slave receive mode. Furthermore, it automatically sets the TDBEF0 and TENDF flags to 0, and sets the BST.SPCNDDF flag to 1.
7. After checking that the BST.SPCNDDF flag = 1, set the BST.NACKDF and SPCNDDF flags to 0 for the next transfer operation.

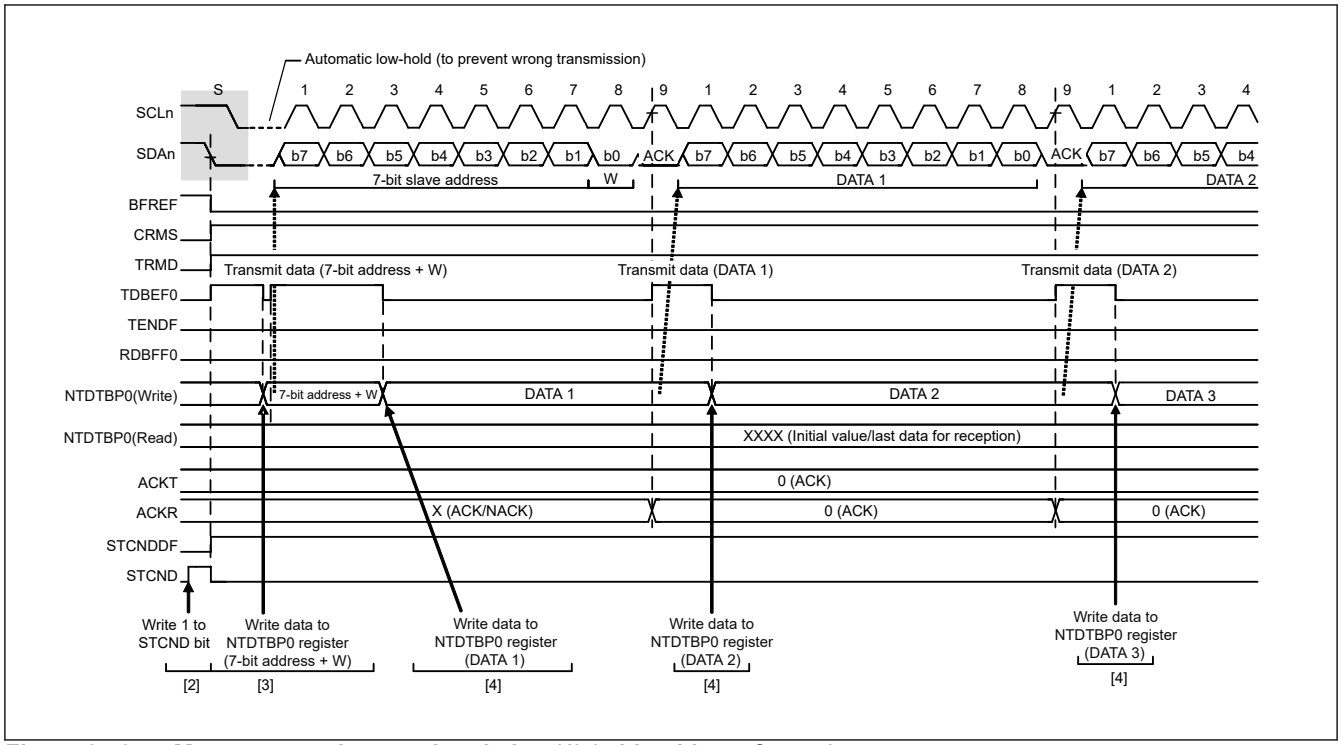


Figure 27.2 Master transmit operation timing (1) (7-bit address format)

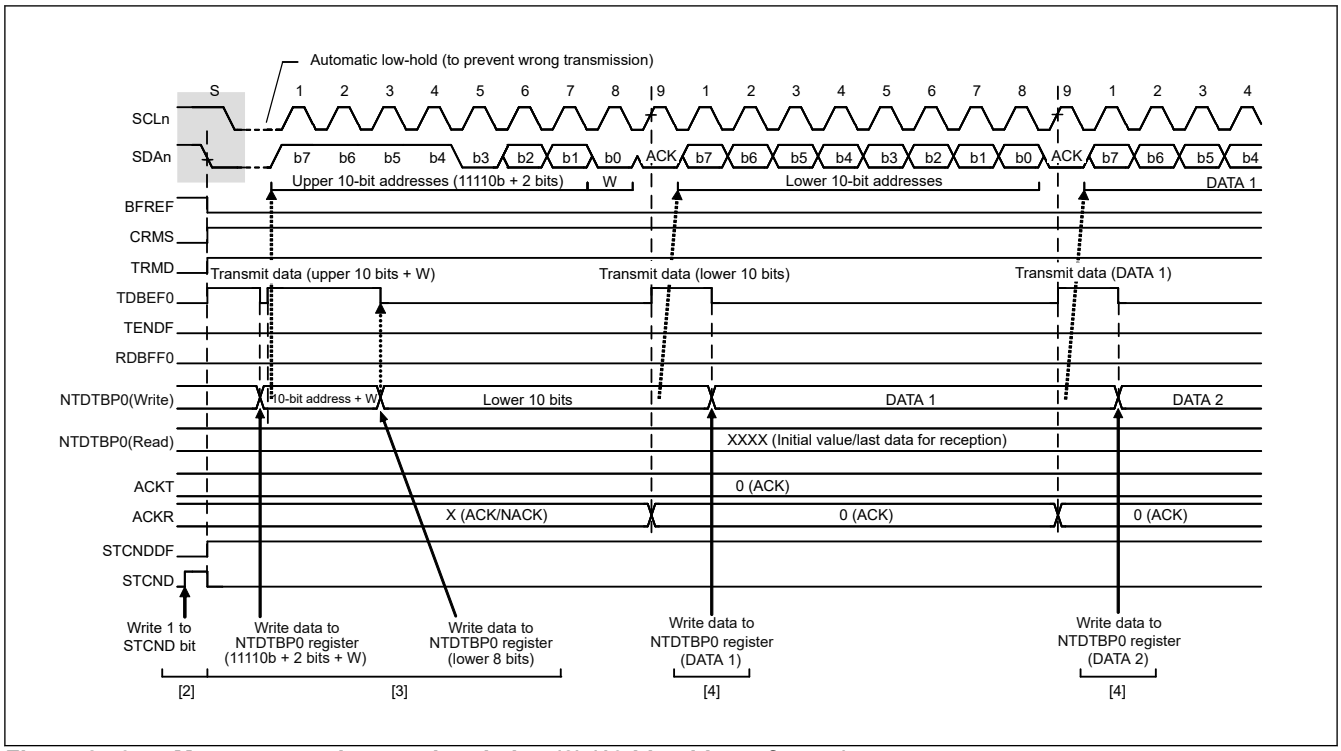


Figure 27.3 Master transmit operation timing (2) (10-bit address format)

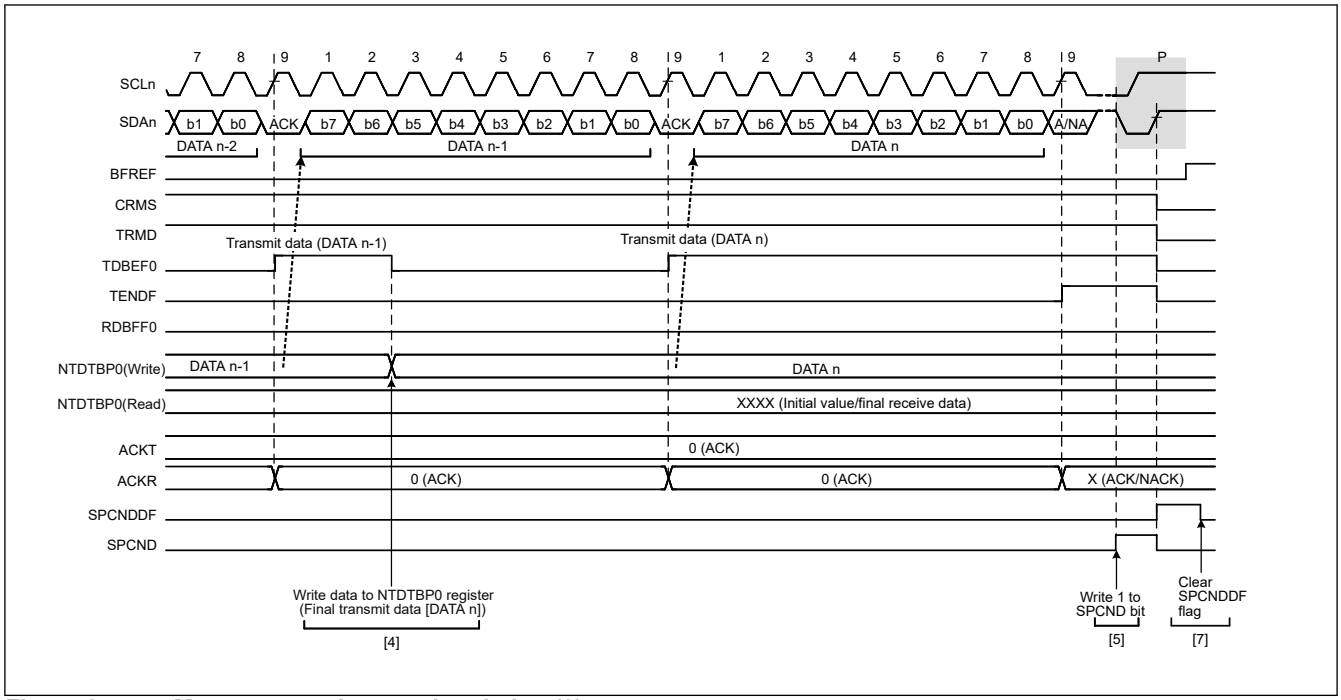


Figure 27.4 Master transmit operation timing (3)

(b) Data Read Transfer (Single Buffer transfer)

In master receive operation, IIC as a master device outputs the SCL clock, receives data from the slave device, and returns acknowledgments. Because IIC must start by sending a slave address to the corresponding slave device, this part of the procedure is performed in master transmit mode, but the subsequent steps are in master receive mode.

Figure 27.48 and Figure 27.49 show examples of usage of master reception (7-bit address format) and Figure 27.5 to Figure 27.7 show the timing of operations in master reception.

The following describes the procedure and operations for master reception.

1. Initial settings. For details, see section 27.3.2.1. Initial Setting Flow.
2. Read the BCST.BFREF flag to check that the bus is open, and then set the CNDCTL.STCND bit to 1 (START condition issuance request). Upon receiving the request, IIC issues a START condition. When IIC detects the START condition, the BFREF flag is automatically set to 0 and the BST.STCNDDF flag is automatically set to 1 and the STCND bit is automatically set to 0. At this time, if the START condition is detected and the levels for the SDA output and the levels on the SDAn line have matched while the STCND bit = 1, IIC recognizes that issuing of the START condition as requested by the STCND bit has been successfully completed, and bits CRMS and TRMD in the PRSST register are automatically set to 1, placing IIC in master transmit mode. The NTST.TDBEF0 flag is also automatically set to 1 in response to setting of the TRMD bit to 1.
3. Check that the NTST.TDBEF0 flag = 1, and then write the value for transmission (the first byte indicates the slave address and value of the R/W# bit) to the NTDTBP0 register. Once the data for transmission are written to the NTDTBP0 register, the TDBEF0 flag is automatically set to 0, the data are transferred from the Normal Transmit Data Buffer 0 to the Shift Register, and the TDBEF0 flag is again set to 1. Once the byte containing the slave address and R/W# bit has been transmitted, the value of the PRSST.TRMD bit is automatically updated to select transmit or receive mode in accord with the value of the transmitted R/W# bit. If the value of the R/W# bit was 1, the TRMD bit is set to 0 on the rising edge of the ninth cycle of SCL clock, placing IIC in master receive mode. At this time, the TDBEF0 flag is set to 0. The NTST.RDBFF0 flag is automatically set to 1 when ACK response is received from the slave device. If the slave device is not recognized or a communication failure occurs, the BST.NACKDF flag will be set to 1. At this time, set 1 to the CNDCTL.SPCND bit to issue a STOP condition. For master reception from a device with a 10-bit address, start by using master transmission to issue the 10-bit address, and then issue a Repeated START condition. After that, transmitting 1111 0, the two higher-order bits of the slave address, and the R bit places IIC in master receive mode.
4. Dummy read the NTDTBP0 register after confirming that the NTST.RDBFF0 flag = 1; this makes IIC start output of the SCL clock and start data reception.
5. After 1 byte of data has been received, the NTST.RDBFF0 flag is set to 1 on the rising edge of the eighth or ninth cycle of SCL clock (the clock signal) as selected by the SCSTRCTL.ACKTWE bit. Reading the NTDTBP0 register at this

time will produce the received data, and the RDBFF0 flag is automatically set to 0 at the same time. Furthermore, the value of the acknowledgment field received during the ninth cycle of SCL clock is returned as the value set in the ACKCTL.ACKT bit. Furthermore, if the next byte to be received is the next to last byte, set the SCSTRCTL.RWE bit to 1 (for wait insertion) before reading the NTDTBP0 register (containing the second byte from last). As well as enabling NACK output even in the case of delays in processing to set the ACKCTL.ACKT bit to 1 (NACK) in step 6, due to other interrupts, etc., this fixes the SCLn line to the low level on the falling edge of the ninth clock cycle in reception of the last byte, so the state is such that issuing a STOP condition is possible.

6. When the SCSTRCTL.ACKTWE bit = 0 and the slave device must be notified that it is to end transfer for data reception after transfer of the next (final) byte, set the ACKCTL.ACKT bit to 1 (NACK).
7. After reading the byte before last from the NTDTBP0 register, if the value of the NTST.RDBFF0 flag is confirmed to be 1, write 1 to the CNDCTL.SPCND bit (STOP condition issuance request) and then read the last byte from the NTDTBP0 register. When 1 is written to the CNDCTL.SPCND bit, IIC is released from the wait state and issues the STOP condition after low-level output in the ninth clock cycle is completed or the SCLn line is released from the low-hold state.
8. Upon detecting the STOP condition, IIC automatically sets bits CRMS and TRMD in the PRSST register to 00 and enters slave receive mode. Furthermore, detection of the STOP condition leads to setting of the BST.SPCNDDF flag to 1.
9. After checking that the BST.SPCNDDF flag = 1, set the BST.NACKDF and SPCNDDF flags to 0 for the next transfer operation.

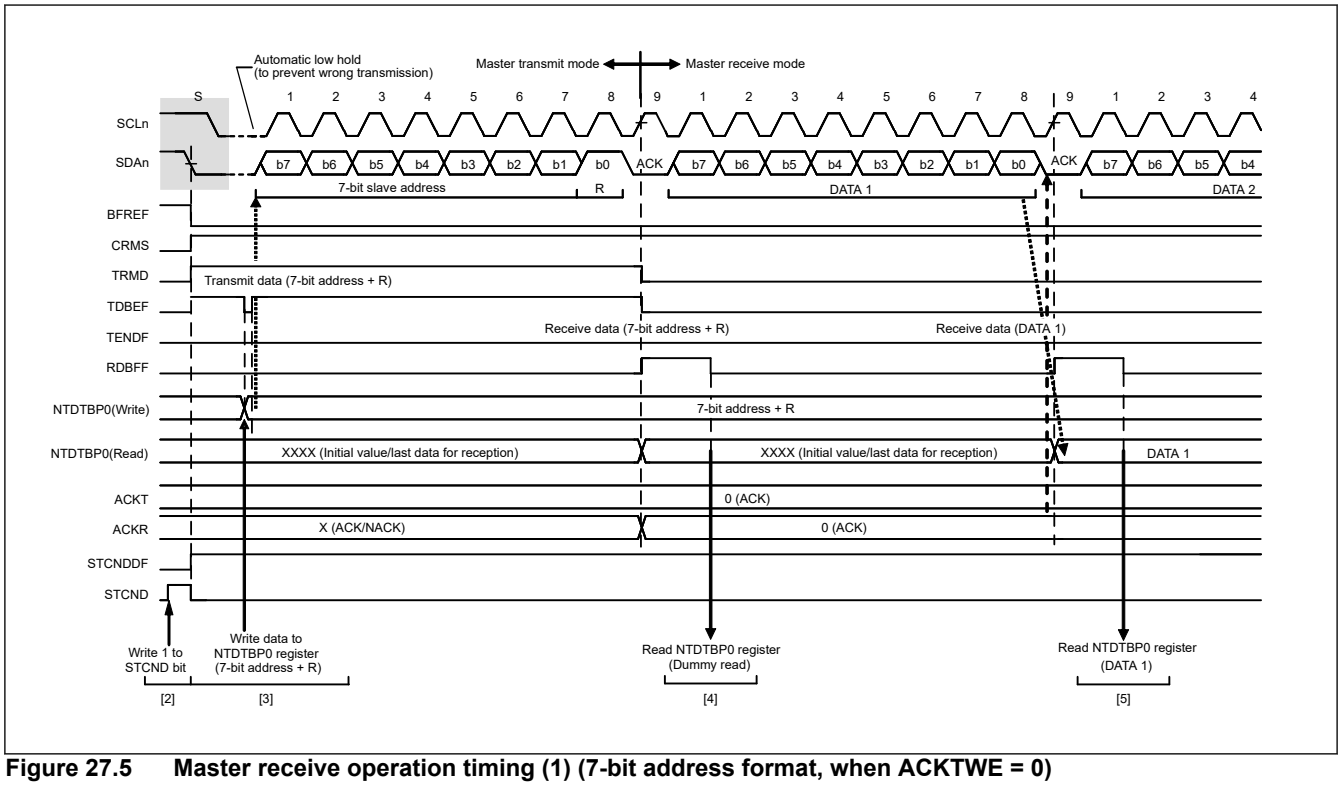


Figure 27.5 Master receive operation timing (1) (7-bit address format, when ACKTWE = 0)

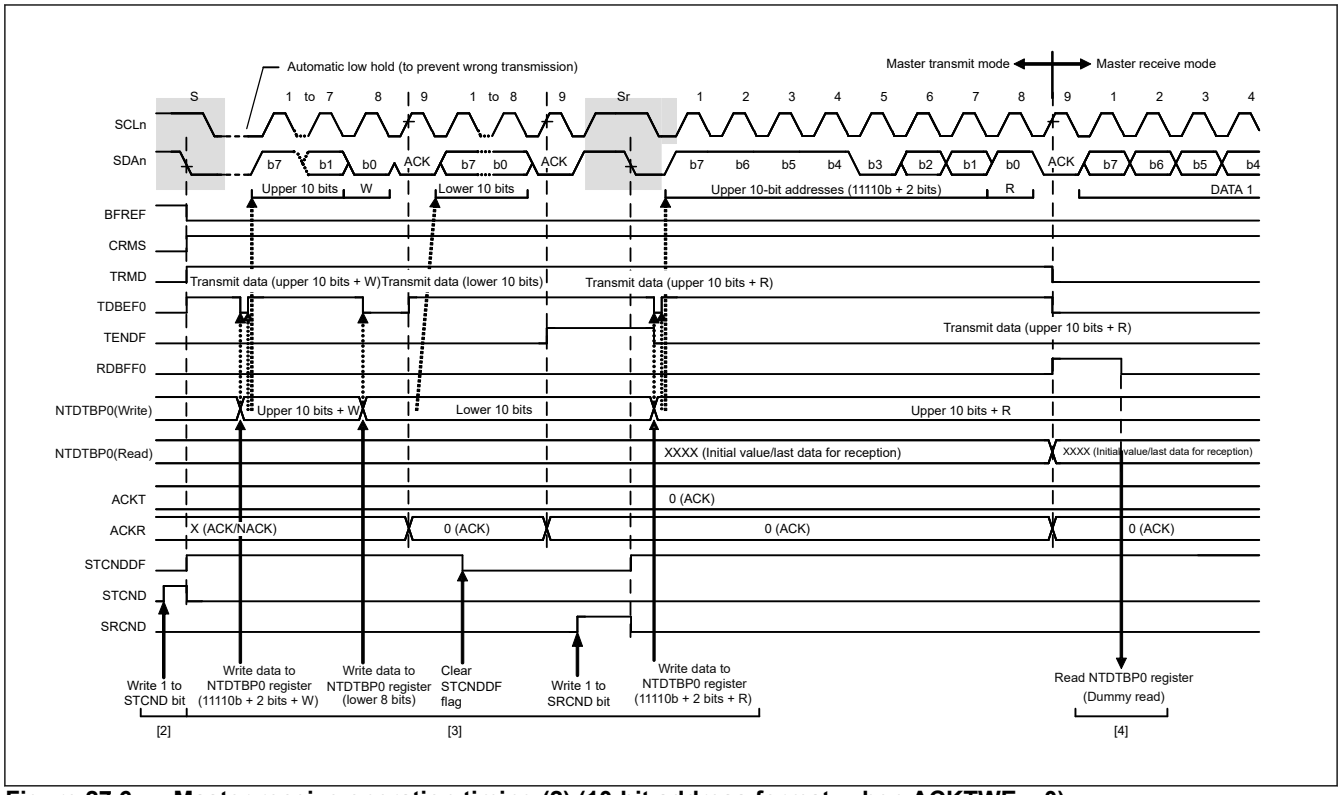


Figure 27.6 Master receive operation timing (2) (10-bit address format, when ACKTWE = 0)

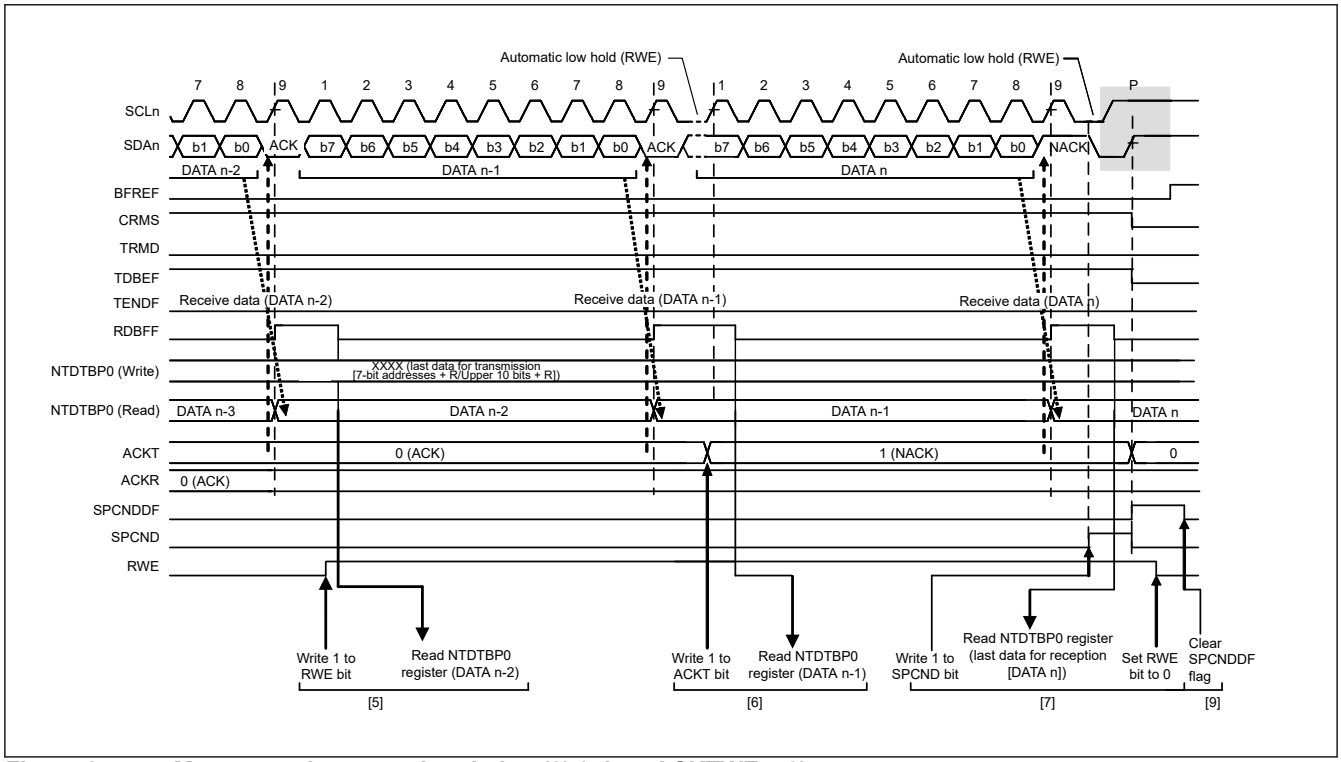


Figure 27.7 Master receive operation timing (3) (when ACKTWE = 0)

27.3.1.1.2 Slave Mode Operation

(1) I²C Slave Operation

(a) Data Write Transfer (Single Buffer transfer)

In slave receive operation, the master device outputs the SCL clock and transmit data, and IIC returns acknowledgments as a slave device.

Figure 27.51 shows an example of usage of slave reception and Figure 27.8 and Figure 27.9 show the timing of operations in slave reception.

The following describes the procedure and operations for slave reception.

1. Initial settings. For details, see section 27.3.2.1. Initial Setting Flow. After initial settings, IIC will stay in the standby state until it receives a slave address that it matches.
2. After receiving a matching slave address, IIC sets one of the corresponding bits SVST.HOAF, GCAF, and SVAFy (y = 0 to 2) to 1 on the rising edge of the ninth cycle of SCL clock (the clock signal) and outputs the acknowledge bit (ACK) on the ninth cycle of SCL clock. If the value of the R/W# bit that was also received at this time is 0, IIC continues to place itself in slave receive mode and sets the NTST.RDBFF0 flag to 1.
3. After the BST.SPCNDDF flag is confirmed to be 0 and the NTST.RDBFF0 flag to be 1, dummy read the NTDTBP0 register (the dummy value consists of the slave address and R/W# bit when the 7-bit address format is selected, or the lower 8 bits when the 10-bit address format is selected).
4. When the NTDTBP0 register is read, IIC automatically sets the NTST.RDBFF0 flag to 0. If reading of the NTDTBP0 register is delayed and a next byte is received while the RDBFF0 flag is still set to 1, IIC holds the SCLn line low from one SCL cycle before the timing with which RDBFF0 should be set. In this case, reading the NTDTBP0 register releases the SCLn line from being held at the low level. When the BST.SPCNDDF flag = 1 and the NTST.RDBFF0 flag is also 1, read the NTDTBP0 register until all the data is completely received.
5. Upon detecting the STOP condition, IIC automatically clears bits SVST.HOAF, GCAF, and SVAFy (y = 0 to 2) to 0.
6. After checking that the BST.SPCNDDF flag = 1, set the BST.SPCNDDF flag to 0 for the next transfer operation.

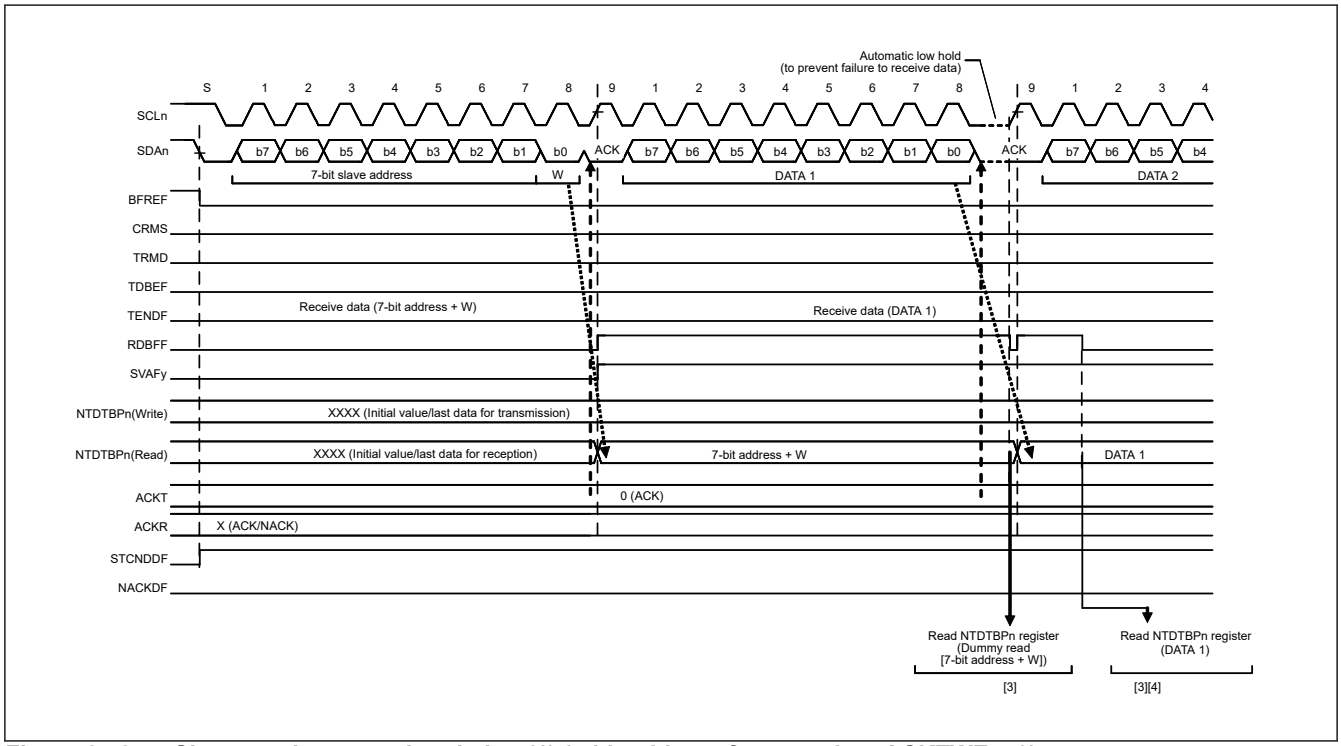


Figure 27.8 Slave receive operation timing (1) (7-bit address format, when ACKTWE = 0)

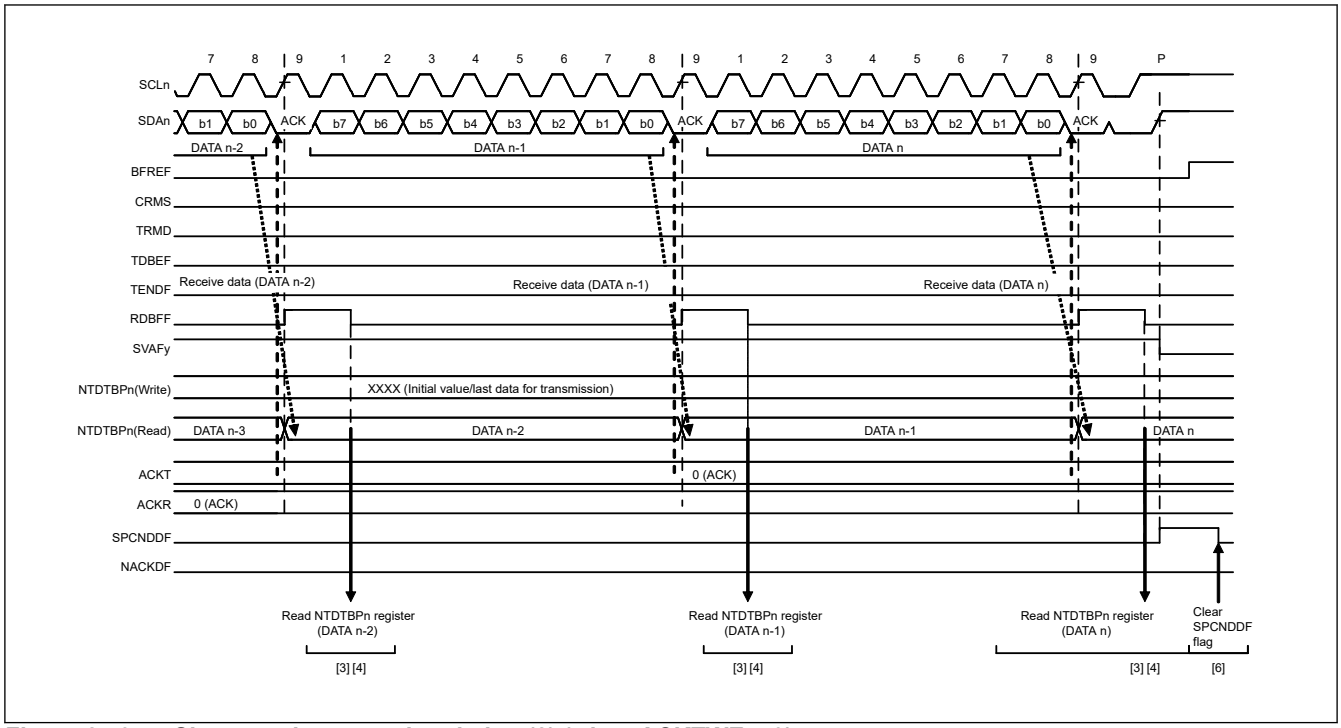


Figure 27.9 Slave receive operation timing (2) (when ACKTWE = 0)

(b) Data Read Transfer (Single Buffer transfer)

In slave transmit operation, the master device outputs the SCL clock, IIC transmits data as a slave device, and the master device returns acknowledgments.

Figure 27.50 shows an example of usage of slave transmission and Figure 27.10 and Figure 27.11 show the timing of operations in slave transmission.

The following describes the procedure and operations for slave transmission.

1. Initial settings. For details, see section 27.3.2.1. Initial Setting Flow.
   After initial settings, IIC will stay in the standby state until it receives a slave address that it matches.
2. After receiving a matching slave address, IIC sets one of the corresponding bits SVST.HOAF, GCAF, and SVAFy (y = 0 to 2) to 1 on the rising edge of the ninth cycle of SCL clock (the clock signal) and outputs the acknowledge bit (ACK) on the ninth cycle of SCL clock. If the value of the R/W# bit that was also received at this time is 1, IIC automatically places itself in slave transmit mode by setting both the PRSST.TRMD bit and the NTST.TDBEF0 flag to 1.
3. After the NTST.TDBEF0 flag is confirmed to be 1, write the data for transmission to the NTDTBP0 register. At this time, if IIC does not receive acknowledge from the master device (receives an NACK signal) while the BSTE.NACKDE bit = 1, IIC aborts transfer of the next data.
4. Wait until the following (a) or (b) condition.
   (a) The BST.NACKDF flag is set to 1.
   (b) The BST.TENDF flag is set to 1 while the NTST.TDBEF0 flag = 1, after the last byte for transmission is written to the NTDTBP0 register.

   When the BST.NACKDF flag or the TENDF flag = 1, IIC drives the SCLn line low on the ninth falling edge of SCL clock.
5. When the BST.NACKDF flag or the BST.TENDF flag = 1, dummy read the NTDTBP0 register to complete the processing. This releases the SCLn line.
6. Upon detecting the STOP condition, IIC automatically sets bits SVST.HOAF, GCAF, and SVAFy (y = 0 to 2), flags NTST.TDBEF0 and BST.TENDF, and the PRSST.TRMD bit to 0, and enters slave receive mode.
7. After checking that the BST.SPCNDDF flag = 1, set the BST.NACKDF and SPCNDDF flags to 0 for the next transfer operation.

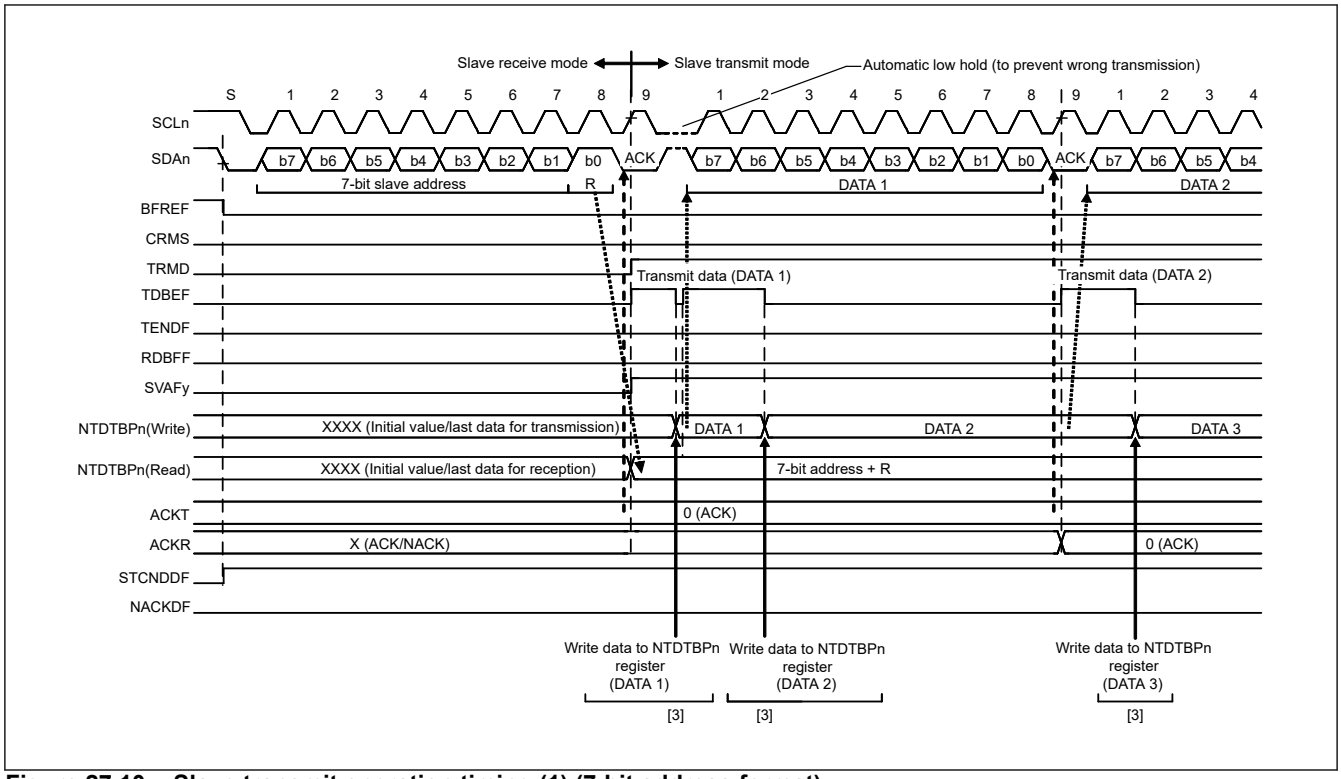


Figure 27.10 Slave transmit operation timing (1) (7-bit address format)

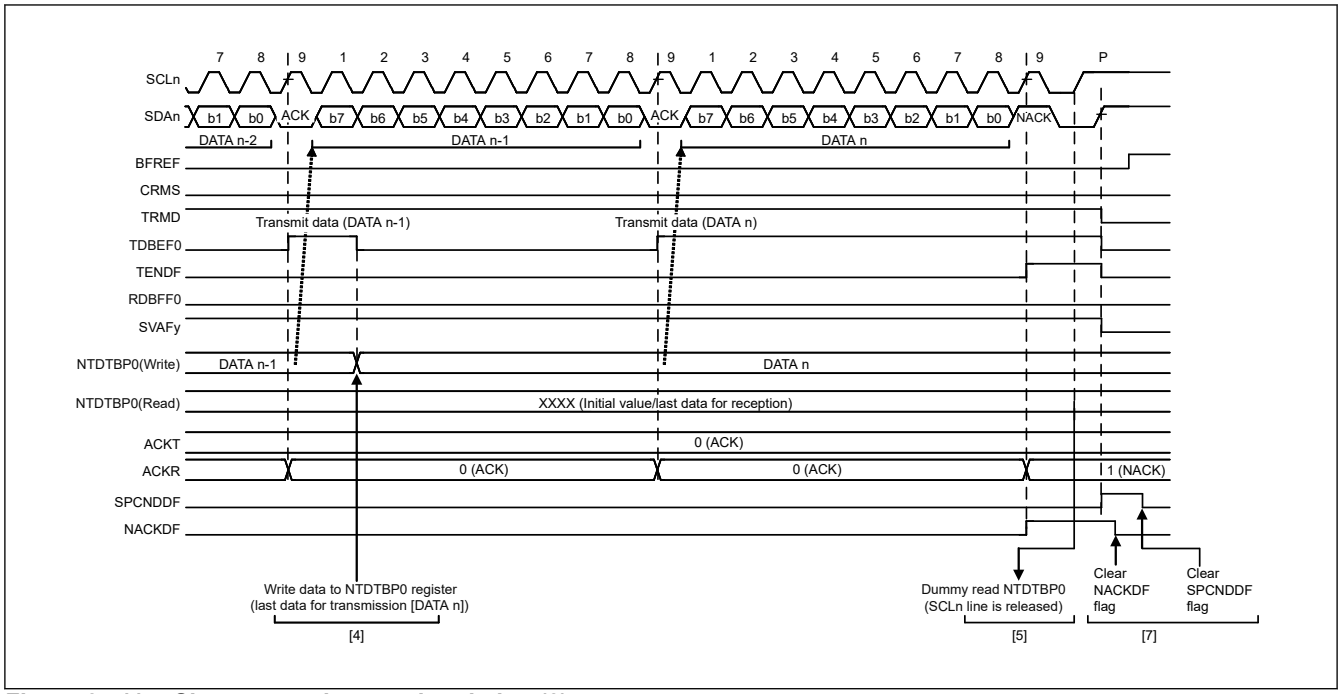


Figure 27.11 Slave transmit operation timing (2)

27.3.1.2 Data Handler

The relationship between the transfer method and the queue is shown in Table 27.8.

Table 27.8 Transfer method

Transfer method	Buffer	size	Master	Slave
Single buffer transfer	Normal Transmit Data	1 byte	✓	✓
	Normal Receive Data	1 byte	✓	✓

27.3.1.2.1 Transfer Method

(1) Single Buffer transfer

Each process (condition issue, data transfer, ACK / NACK response) is controlled by software.

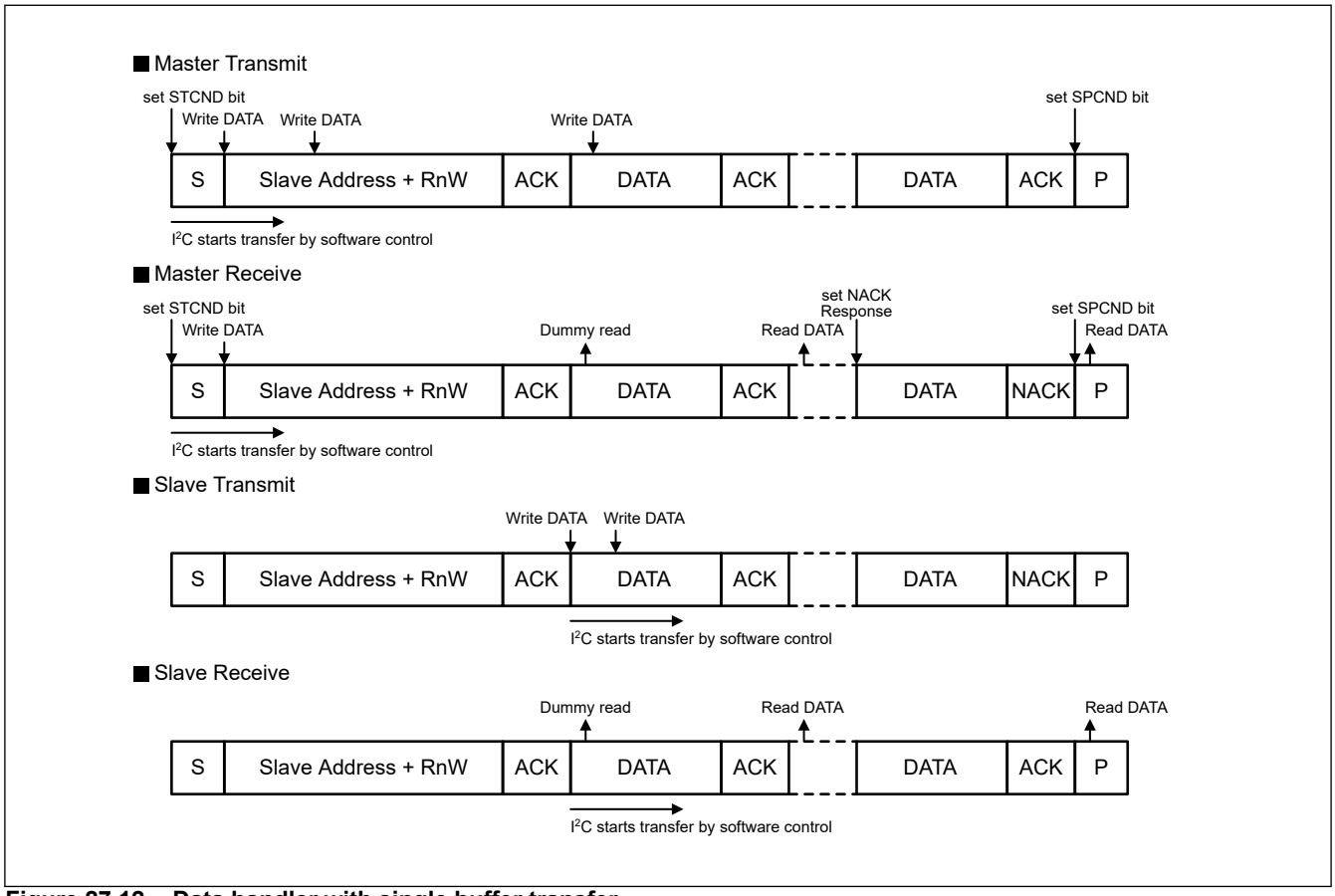


Figure 27.12 Data handler with single buffer transfer

27.3.1.3 I2C Protocol

27.3.1.3.1 Communication Protocol

(1) I2C Communication Data Format

The I2C bus format consists of 8-bit data and 1-bit acknowledge. The frame following a START condition or Repeated START condition is an address frame used to specify a slave device with which the master device communicates. The specified slave is valid until a new slave is specified or a STOP condition is issued.

Figure 27.13 shows the I2C bus format, and Figure 27.14 shows the I2C bus timing.

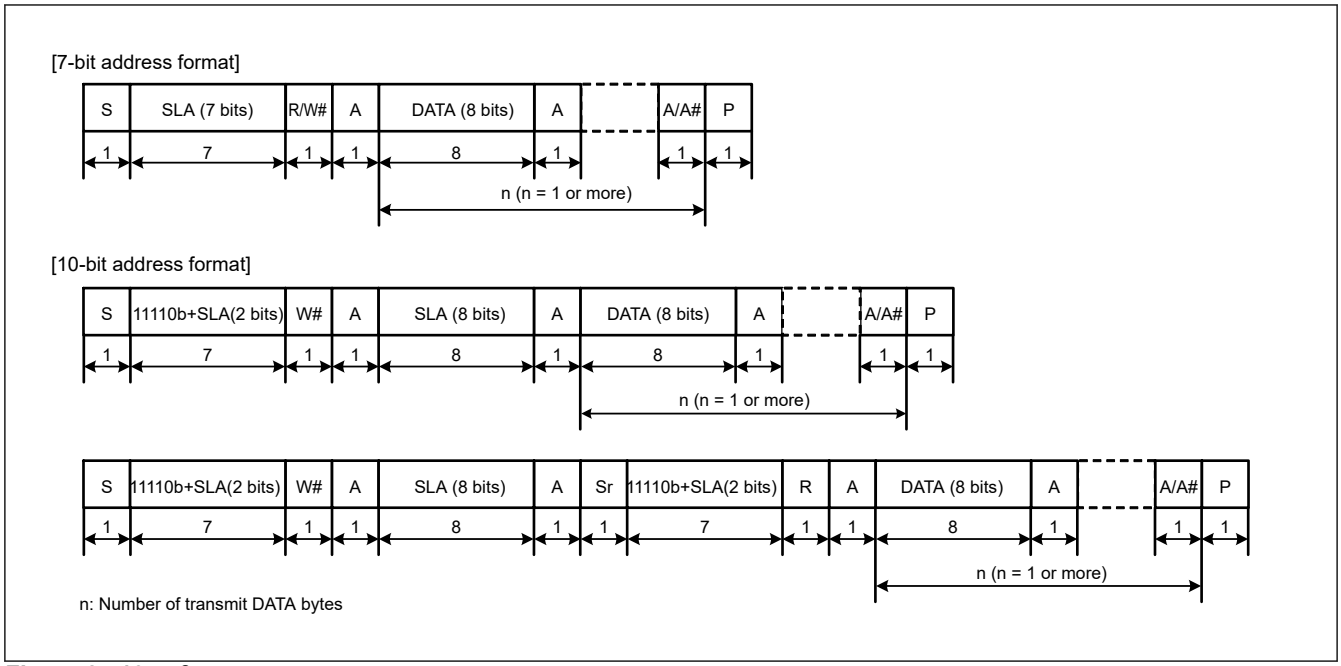


Figure 27.13 I²C bus format

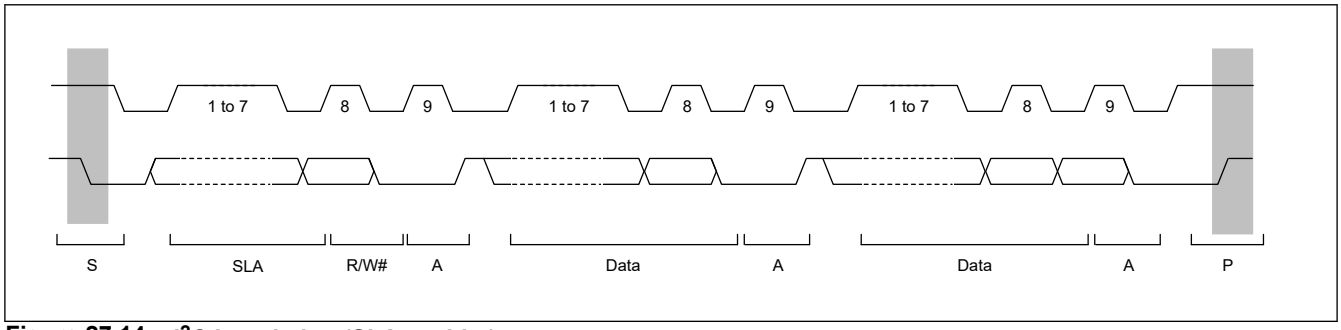


Figure 27.14 I²C bus timing (SLA = 7 bits)

S:	START condition. The master device drives the SDAn line low from high level while the SCLn line is at a high level.
SLA:	Slave address, by which the master device selects a slave device.
R/W#:	Indicates the direction of data transfer: from the slave device to the master device when R/W = 1, or from the master device to the slave device when R/W = 0.
A:	Acknowledge. The receive device drives the SDAn line low. (In master transmit mode, the slave device returns acknowledge. In master receive mode, the master device returns acknowledge.)
A#:	Not Acknowledge. The receive device drives the SDAn line high.
Sr:	Repeated START condition. The master device drives the SDAn line low from the high level after the setup time has elapsed with the SCLn line at the high level.
DATA:	Transmitted or received data
P:	STOP condition. The master device drives the SDAn line high from low level while the SCLn line is at a high level.

27.3.1.3.2 START Condition / Repeated START Condition / STOP Condition Issuing Function

(1) Issuing a START Condition

IIC issues a START condition when the CNDCTL.STCND bit is set to 1.

Set the STCND bit to 1 (START condition issuance request) when the BCST.BFREF flag is set to 1 (bus free state).

IIC issues a START condition.

When a START condition is issued normally, IIC automatically shifts to the master transmit mode. A START condition is issued in the following sequence.

[START condition issuance]

- Drive the SDAn line low (high level to low level).
- Ensure the time set in STDBR.SBRHO[7:0] and the START condition hold time.
- Drive the SCLn line low (high level to low level).
- Detect low level of the SCLn line and ensure the low-level period of SCLn line set in STDBR.SBRLO[7:0].

(2) Issuing a Repeated START Condition

IIC issues a Repeated START condition when the CNDCTL.SRCND bit is set to 1.

When the SRCND bit is set to 1, a Repeated START condition issuance request is made and IIC issues a Repeated START condition when the BCST.BFREF flag = 0 (bus busy state) and the PRSST.CRMS bit = 1 (master mode).

A Repeated START condition is issued in the following sequence.

[Repeated START condition issuance]

- Release the SDAn line.
- Ensure the low-level period of SCLn line set in STDBR.SBRLO[7:0].
- Release the SCLn line (low level to high level).
- Detect a high level of the SCLn line and ensure the time set in STDBR.SBRLO[7:0] and the Repeated START condition setup time.
- Drive the SDAn line low (high level to low level).
- Ensure the time set in STDBR.SBRHO[7:0] and the Repeated START condition hold time.
- Drive the SCLn line low (high level to low level).
- Detect a low level of the SCLn line and ensure the low-level period of SCLn line set in STDBR.SBRLO[7:0].

Note: When issuing Repeated START conditions request, write the slave address to NTDTBP0 after confirming CNDCTL.SRCND = 0. Data written in the period of CNDCTL.SRCND = 1 is not forwarded because retransmission condition before the occurrence.

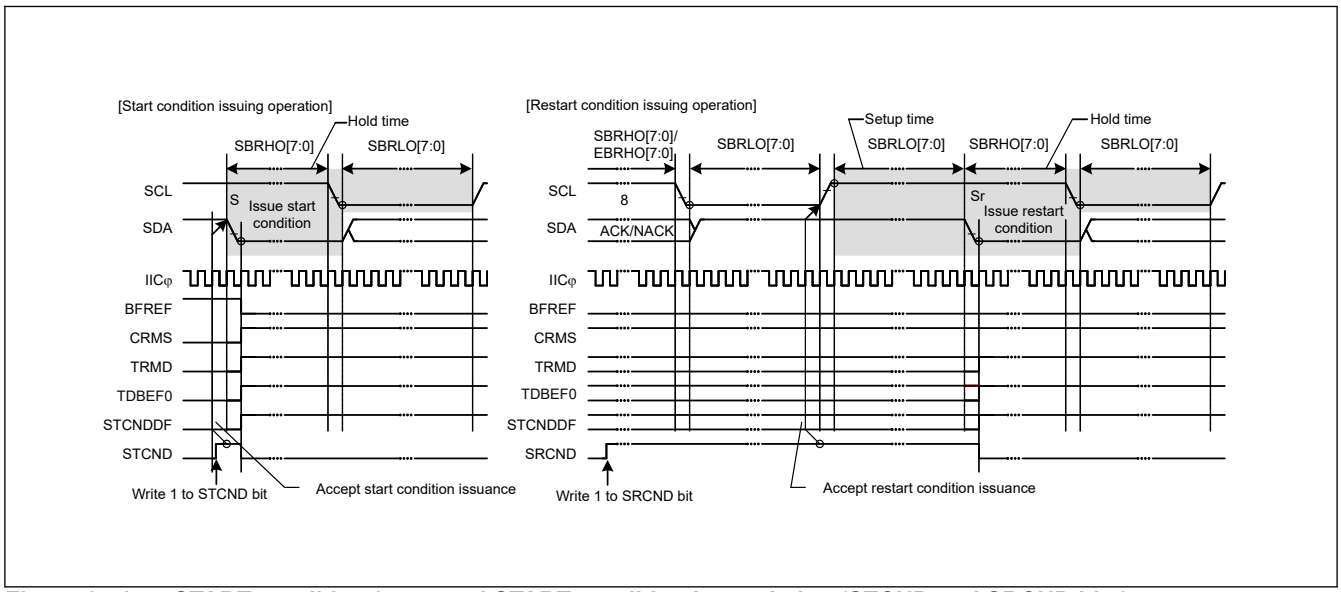


Figure 27.15 START condition / repeated START condition issue timing (STCND and SRCND bits)

Figure 27.16 shows the operation to issue a Repeated START condition after the master transmission.

[Repeated START condition issuance after the master transmission]

- Initial setting. For details, see section 27.3.2.1. Initial Setting Flow.

- Read the BFREF flag in BCST to check that the bus is open, and then set the STCND bit in CNDCTL to 1 (START condition issuance request). Upon receiving the request, IIC issues a START condition. At the same time, the BFREF flag is automatically set to 0 and the STCNDDF flag in BST is automatically set to 1 and the STCND bit is automatically set to 0. At this time, if the START condition is detected and the internal levels for the SDA output state and the levels on the SDAn line have matched while the STCND bit = 1, IIC recognizes that issuing of the START condition as requested by the STCND bit has been successfully completed, and CRMS and TRMD bits in PRSST is automatically set to 1, placing IIC in master transmit mode. The NTST.TDBEF0 flag is also automatically set to 1 in response to setting of the TRMD bit to 1.
- Check that the NTST.TDBEF0 flag = 1, and then write the value for transmission (the slave address and the R/W# bit) to NTDTBP0. Once the data for transmission are written to NTDTBP0, the TDBEF0 flag is automatically set to 0, the data are transferred from NTDTBP0, and the TDBEF0 flag is again set to 1. After the byte containing the slave address and R/W# bit has been transmitted, the value of the TRMD bit is automatically updated to select master transmit or master receive mode in accord with the value of the transmitted R/W# bit. If the value of the R/W# bit was 0, IIC continues in master transmit mode. Since the BST.NACKDF flag being 1 at this time indicates that no slave device recognized the address or there was an error in communications, write 1 to CNDCTL.SPCND bit to issue a STOP condition. For data transmission with an address in the 10-bit format, start by writing 1111 0, the 2 higher-order bits of the slave address, and W to NTDTBP0 as the first address transmission. Then, as the second address transmission, write the 8 lower-order bits of the slave address to NTDTBP0.
- After confirming that the NTST.TDBEF0 flag = 1, write the data for transmission to the NTDTBP0 register. IIC automatically holds the SCLn line low until the data for transmission are ready, a Repeated START condition is issued or a STOP condition is issued.
- After all bytes of data for transmission have been written to the NTDTBP0 register, wait until the value of the BST.TENDF flag returns to 1, and then, after check that the BST.STCNDDF flag = 1, set the BST.STCNDDF flag to 0.
- Set the SRCND bit in CNDCTL to 1 (Repeated START condition issuance request). Upon receiving the request, IIC issues a Repeated START condition.
- After check that the BST.STCNDDF flag = 1, write the value for transmission (the slave address and the R/W# bit) to NTDTBP0.

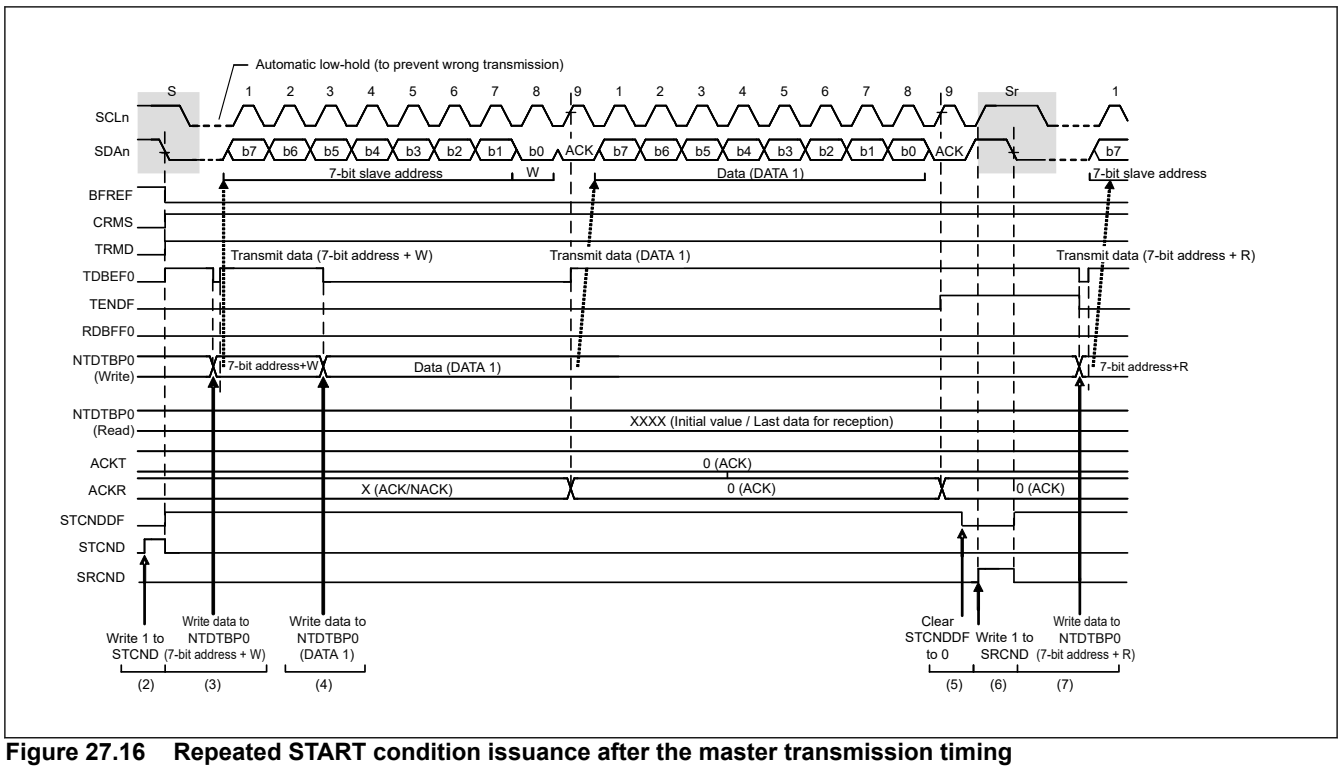


Figure 27.16 Repeated START condition issuance after the master transmission timing

(3) Issuing a STOP Condition

IIC issues a STOP condition when the SPCND bit in CNDCTL is set to 1.

When the SPCND bit is set to 1, a STOP condition issuance request is made and IIC issues a STOP condition when the BCST.BFREF flag = 0 (bus busy state) and the PRSST.MST bit = 1 (master mode).

A STOP condition is issued in the following sequence.

[STOP condition issuance]

- Drive the SDAn line low (high level to low level).
- Ensure the low-level period of SCLn line set in STDBR.SBRLO[7:0].
- Release the SCLn line (low level to high level).
- Detect a high level of the SCLn line and ensure the time set in STDBR.SBRHO[7:0] and the STOP condition setup time.
- Release the SDAn line (low level to high level).
- Ensure the time set in STDBR.SBRLO[7:0] and the bus free time.
- Set the BFREF flag to 1 (to release the bus mastership).

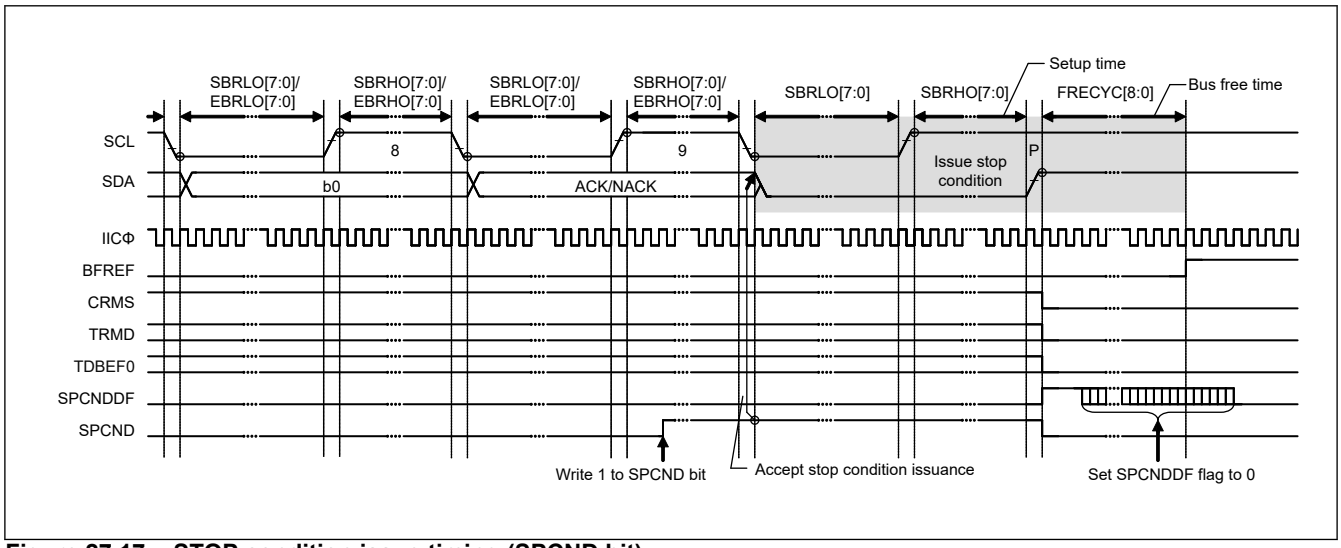


Figure 27.17 STOP condition issue timing (SPCND bit)

27.3.1.3.3 Address Match Detection

IIC can set three unique slave addresses in addition to the general call address and host address, and also can set 7- bit or 10-bit slave addresses.

(1) Slave-Address Match Detection

IIC can set three unique slave addresses, and has a slave address detection function for each unique slave address.

When the SVCTL.SVAEy bit (y = 0 to 2) is set to 1, the slave addresses set in the SVDVADy register (y = 0 to 2) can be detected.

When IIC detects a match of the set slave address, the corresponding SVST.SVAFy flag (y = 0 to 2) is set to 1 at the rising edge of the ninth SCL clock cycle, and the NTST.RDBFF0 flag or the NTST.TDBEF0 flag is set to 1 by the following R/W# bit. This causes a receive data full interrupt (IICn_RX) or transmit data empty interrupt (IICn_TX) to be generated. The SVAFy flag is used to identify which slave address has been specified.

Figure 27.18 to Figure 27.20 show the SVAFy flag set timing in three cases.

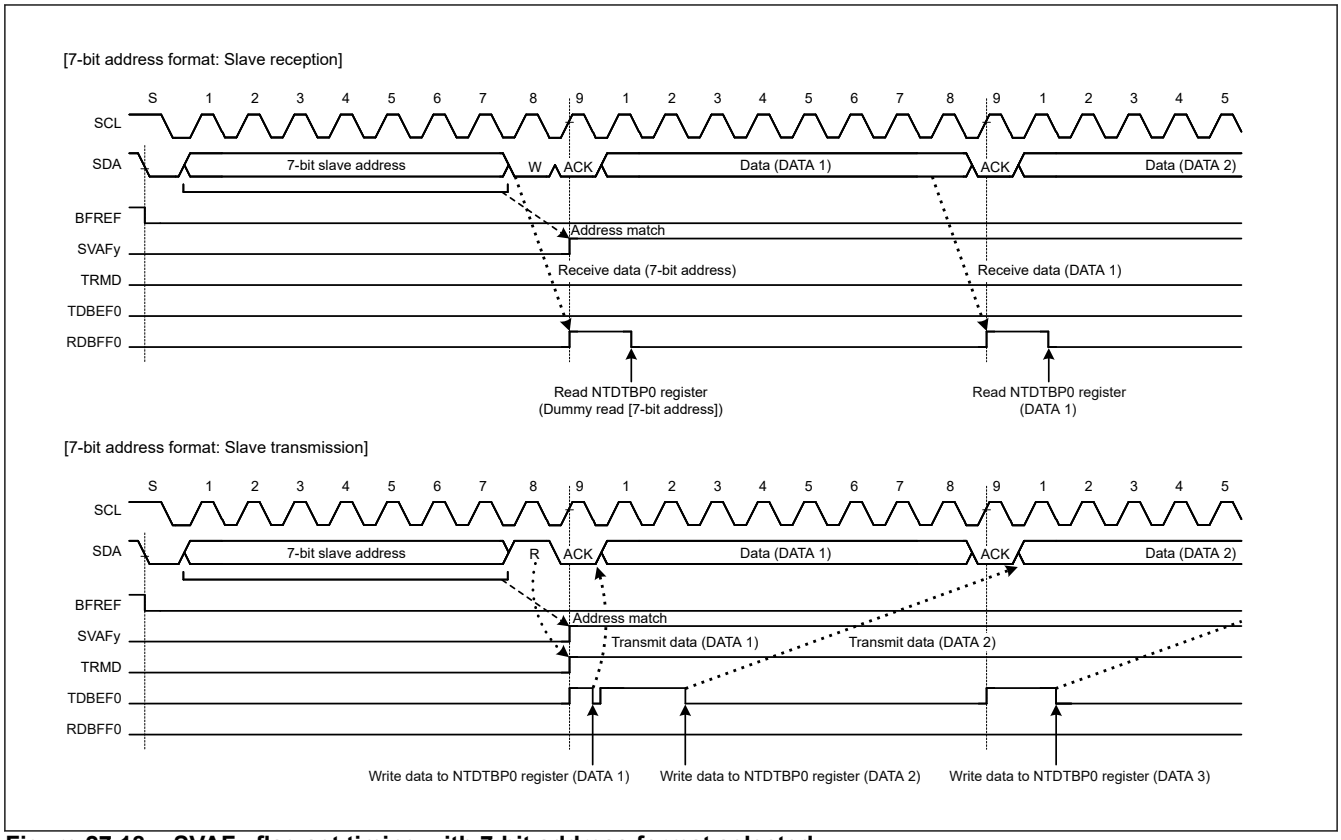


Figure 27.18 SVAFy flag set timing with 7-bit address format selected

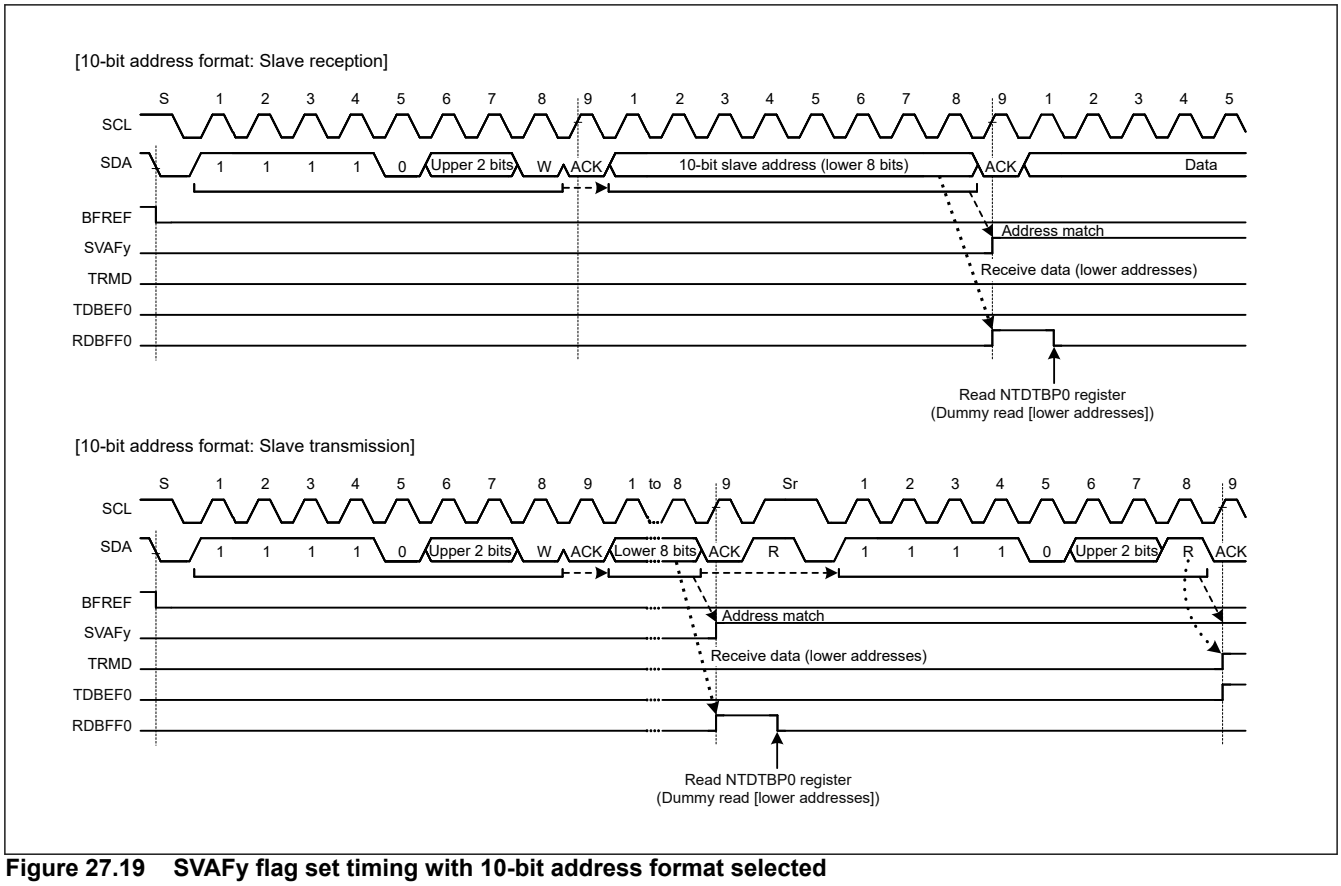


Figure 27.19 SVAFy flag set timing with 10-bit address format selected

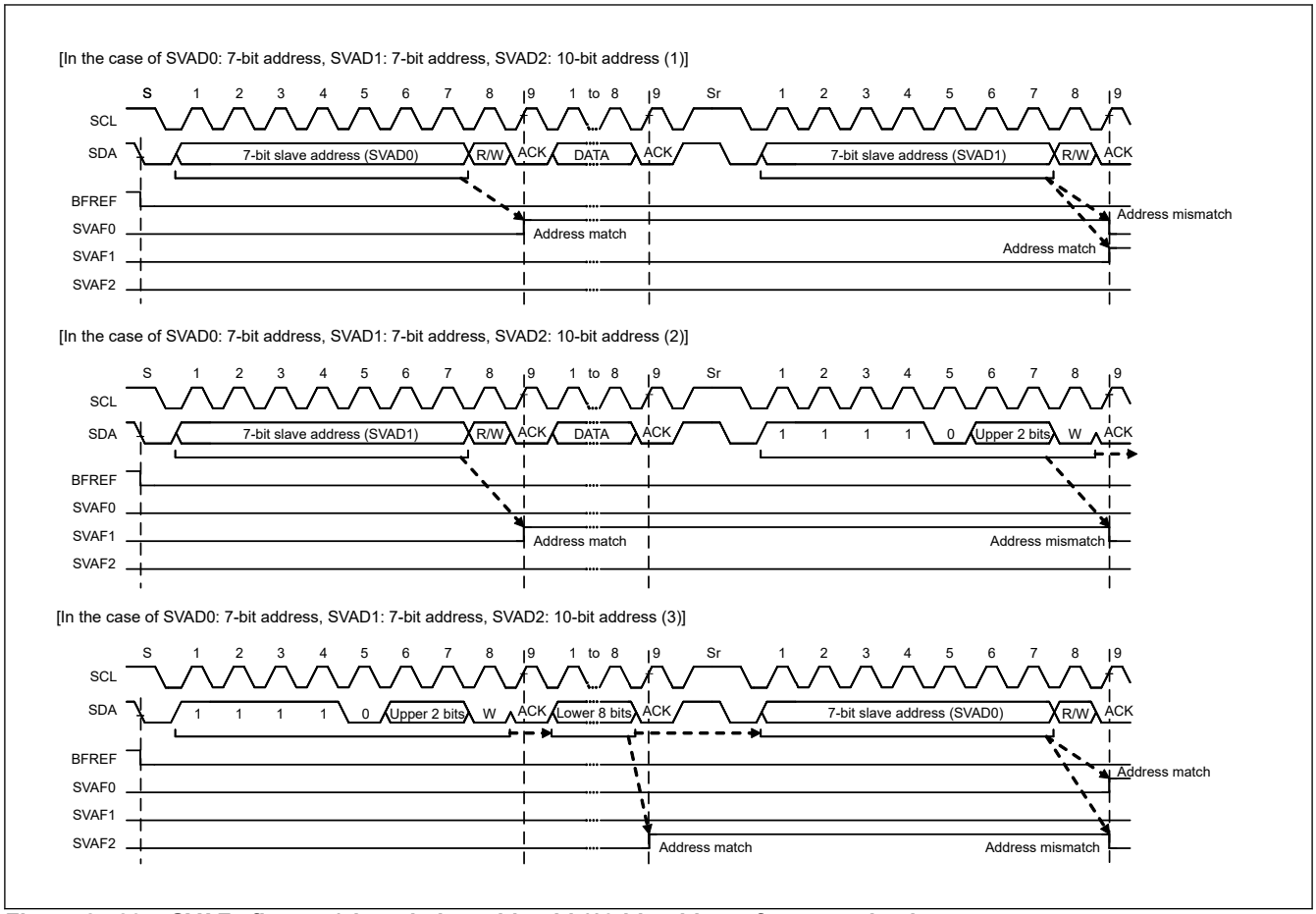


Figure 27.20 SVAFy flag set/clear timing with 7-bit/10-bit address formats mixed

(2) Detection of the General Call Address

IIC has a facility for detecting the general call address (0000 000 + 0 (write)). This is enabled by setting the SVCTL.GCAE bit to 1.

If the address received after a START or Repeated START condition is issued is 0000 000 + 1 (read) (start byte), IIC recognizes this as the address of a slave device with an all-zero address but not as the general call address.

When IIC detects the general call address, both the SVST.GCAF flag and the NTST.RDBFF0 flag are set to 1 on the rising edge of the ninth cycle of SCL clock. This leads to the generation of a receive data full interrupt (IICn_RX). The value of the GCAF flag can be confirmed to recognize that the general call address has been transmitted.

Operation after detection of the general call address is the same as normal slave receive operation.

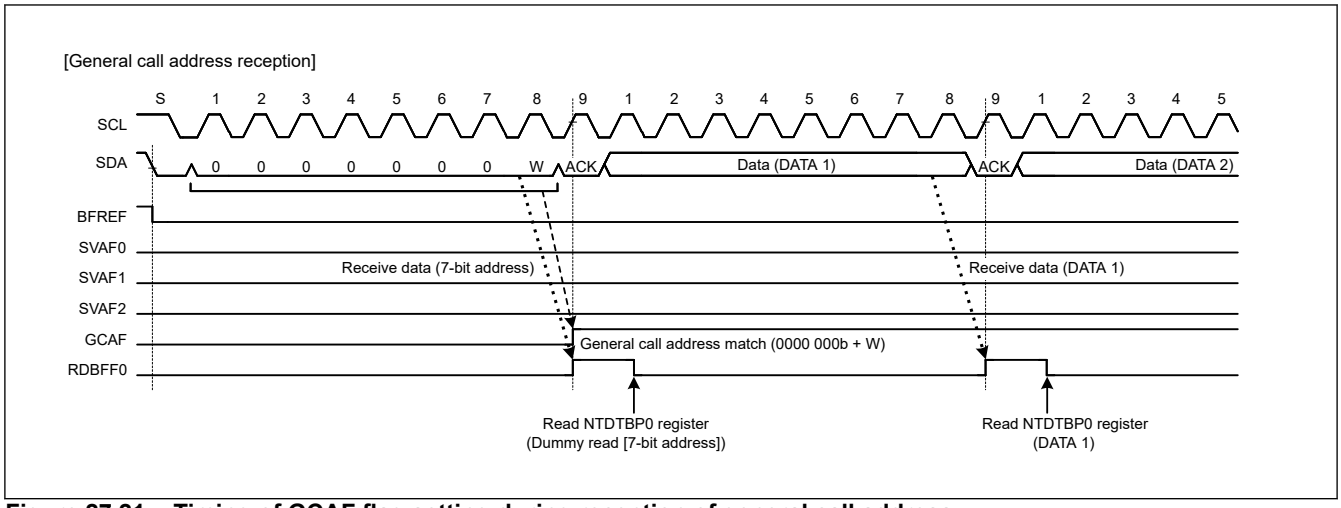


Figure 27.21 Timing of GCAF flag setting during reception of general call address

(3) Device-ID Address Detection

IIC module has a facility for detecting device-ID addresses conformant with the I^2C-bus specification (Rev.03). When IIC receives 1111 100 as the first byte after a START condition or Repeated START condition was issued with the SVCTL.DVIDE bit set to 1, IIC recognizes the address as a device ID, sets the SVST.DVIDF flag to 1 on the rising edge of the ninth SCL clock cycle when the following R/W# bit = 0, and then compares the second and subsequent bytes with its own slave address. If the address matches the value in the slave address register, IIC sets the corresponding SVST.SVAFy flag (y = 0 to 2) to 1.

After that, when the first byte received after a START or Repeated START condition is issued matches the device ID address (1111 100) again and the following R/W# bit = 1, IIC does not compare the second and subsequent bytes and sets the NTST.TDBEF0 flag to 1.

In the device-ID address detection function, IIC sets the DVIDF flag to 0 if a match with IIC's own slave address is not obtained or a match with the device ID address is not obtained after a match with IIC's own slave address and the detection of a Repeated START condition. If the first byte after detection of a START or Repeated START condition matches the device ID address (1111 100) and the R/W# bit = 0, IIC sets the DVIDF flag to 1 and compares the second and subsequent bytes with IIC's slave address. If the R/W# bit = 1, the DVIDF flag holds the previous value and IIC does not compare the second and subsequent bytes. Therefore, the reception of a device-ID address can be checked by reading the DVIDF flag after confirming that TDBEF0 flag = 1.

Furthermore, prepare the device-ID fields (3 bytes: 12 bits indicating the manufacturer + 9 bits identifying the part + 3 bits indicating the revision) that must be sent to the host after reception of a continuous device-ID field as normal data for transmission. For details of the information that must be included in device-ID fields, contact NXP Semiconductors.

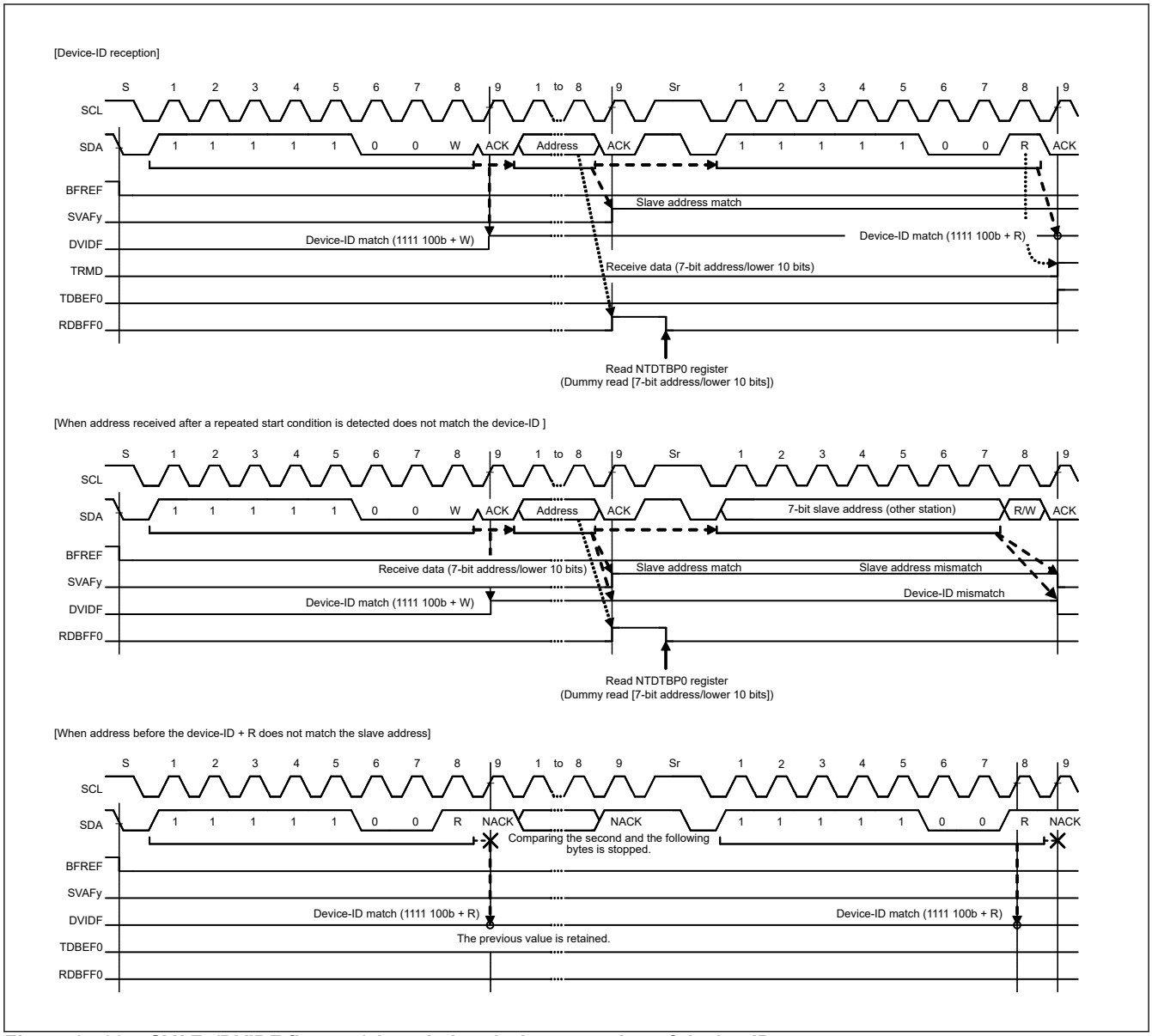


Figure 27.22 SVAFy/DVIDF flag set/clear timing during reception of device-ID

(4) Host Address Detection

IIC has a function to detect the host address while the SMBus is operating. When the SVCTL.HOAE bit is set to 1 while the BFCTL.SMBS bit = 1, IIC can detect the host address (0001 000) in slave receive mode (bits CRMS and TRMD in the PRSST register = 00).

If the bit following the host address (0001 000) is an Rd bit (R/W# bit = 1), IIC can also detect the host address. After the host address is detected, IIC operates in the same manner as normal slave operation.

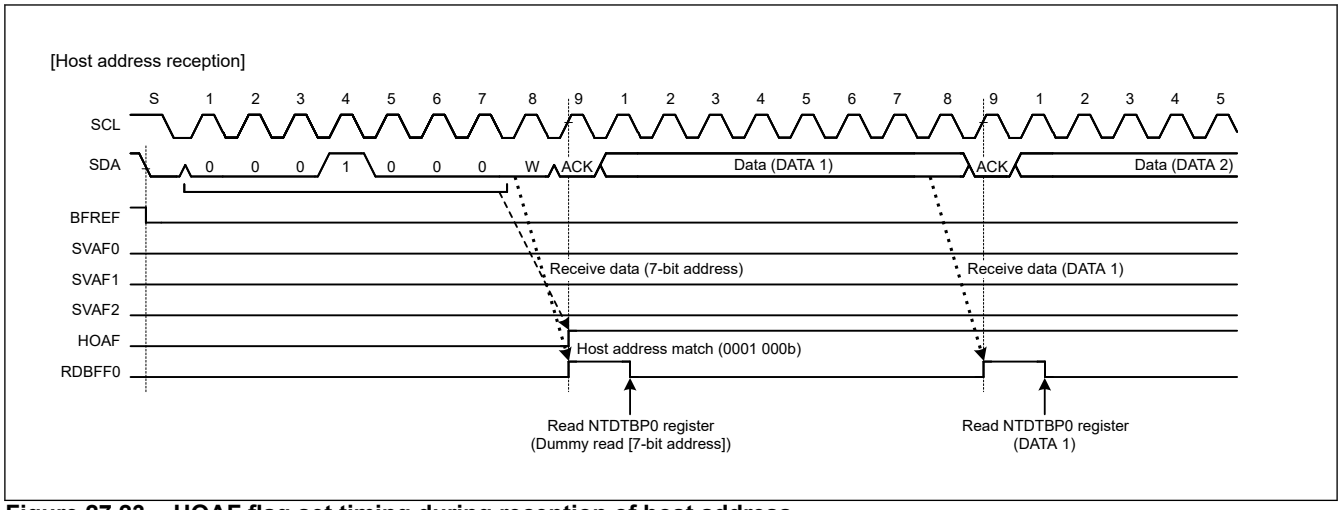


Figure 27.23 HOAF flag set timing during reception of host address

(5) Hs-mode master code Detection

IIC has a facility for detecting the Hs-mode master code (0000 1XXXb). When IIC receives the Hs-mode master code (0000 1XXXb) as the first byte after a START condition was issued with the SVCTL.HSMCE bit set to 1, this module recognizes the address as the Hs-mode master code, sets the SVST.HSMCF flag to 1 on the rising edge of the ninth SCL clock cycle. The first byte after Repeated START after NACK response to Hs-mode master code is recognized as a slave address and compared with the slave address set by SVDVADy.SVAD[9:0] (y = 0 to 2). When IIC detects a match of the set slave address, the corresponding SVST.SVAFy flag (y = 0 to 2) is set to 1 at the rising edge of the ninth SCL clock cycle, and the NTST.RDBFF0 flag or the NTST.TDBEF0 flag is set to 1 by the following R/W# bit. This causes a receive data full interrupt (IICn_RX) or transmit data empty interrupt (IICn_TX) to be generated. The SVAFy flag is used to identify which slave address has been specified. The SVST.HSMCF flag is cleared to 0 when the STOP condition is detected.

Note: If the Hs-mode master code (0000 1XXXb) is received with the SVCTL.HSMCE bit set to 0, other patterns are ignored until the STOP condition is detected.

Note: Hs-mode is supported by IIC0(SCL0_A, SDA0_A).

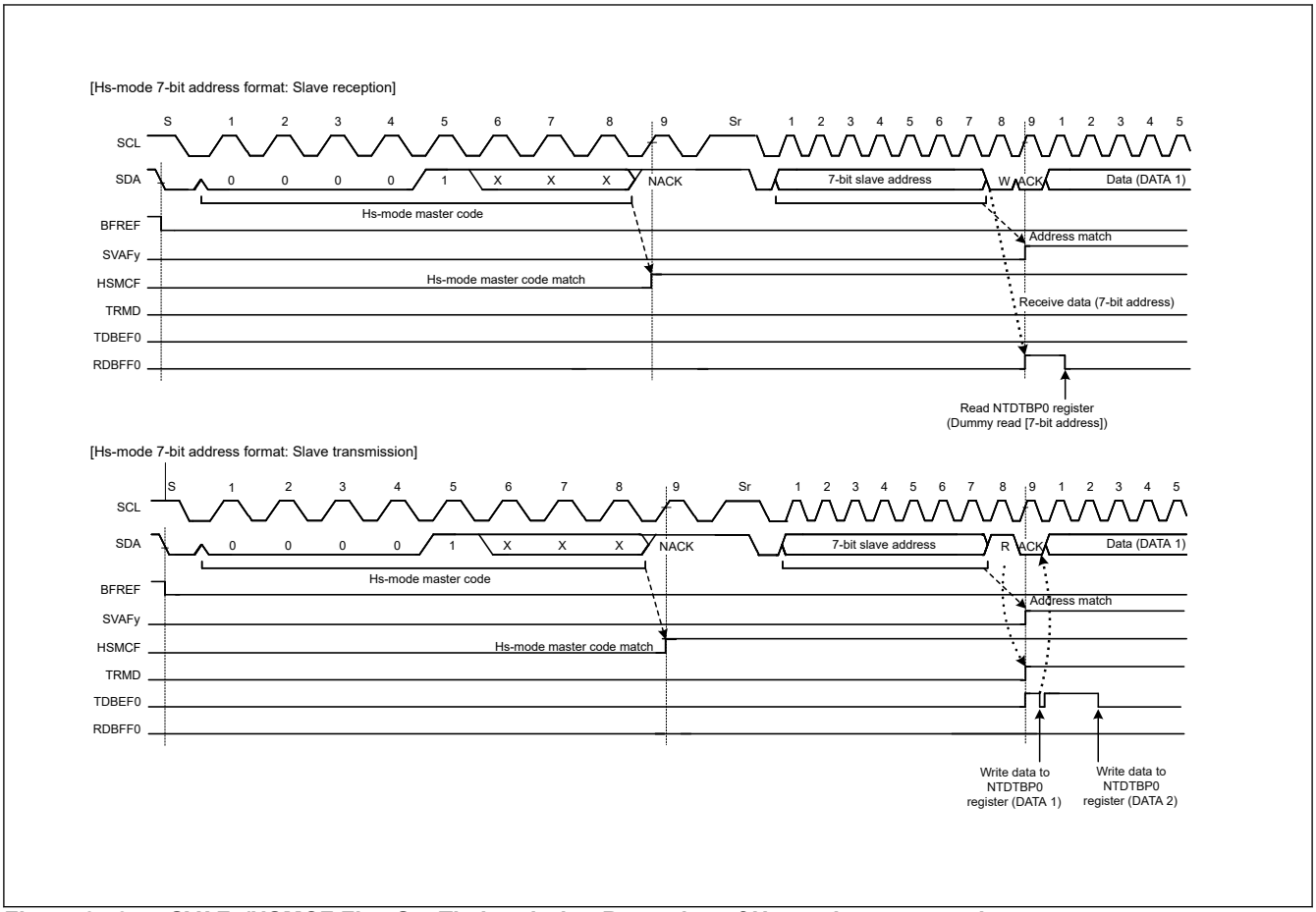


Figure 27.24 SVAFy/HSMCF Flag Set Timing during Reception of Hs-mode master code

27.3.1.3.4 Arbitration-Lost Detection

In addition to the normal arbitration-lost detection function defined by the I^2C-bus specification, the IIC has functions to prevent double-issue of a start condition, to detect arbitration-lost during transmission of NACK, and to detect arbitration-lost in slave transmit mode.

(1) Master Arbitration-Lost Detection (MALE Bit)

The IIC drives the SDAn line low to issue a start condition. However, if the SDAn line has already been driven low by another master device issuing a start condition, this module causes arbitration to be lost, so priority is given to transfer by the other master device. Similarly, if the CNDCTL.STCND bit is set to 1 while the BCST.BFREF flag is 0 (bus busy state), arbitration is lost, so priority is given to transfer by the other master device. No start condition is issued in this case.

When a start condition is issued successfully, if the data for transmission including the address bits (the internal SDA output level) and the level on the SDAn line do not match (the high output as the internal SDA output, that is, the SDA0 pin is in the high-impedance state) and the low level is detected on the SDAn line, the IIC loses in arbitration.

IIC detects master arbitration-lost when the following conditions are met while the BSTE.ALE bit = 1 and the BFCTL.MALE bit = 1 (master arbitration-lost detection enabled).

If arbitration of mastership is lost, IIC immediately enters slave receive mode.

If a slave address (including the general call address) matches its own address at this time, IIC continues in slave operation.

[Conditions for master arbitration-lost]

- Non-matching of the internal level for output on SDA and the level on the SDAn line after a START condition was issued by setting the CNDCTL.STCND bit to 1 while the BCST.BFREF flag was set to 1 (erroneous issuing of a START condition)
- Setting of the CNDCTL.STCND bit to 1 (START condition double-issue error) while the BFREF flag is set to 0

  Note: IIC does not issue a START condition.

- When the transmit data excluding acknowledge (internal SDA output level) does not match the level on the SDAn line in master transmit mode (bits CRMS and TRMD in the PRSST register = 11)

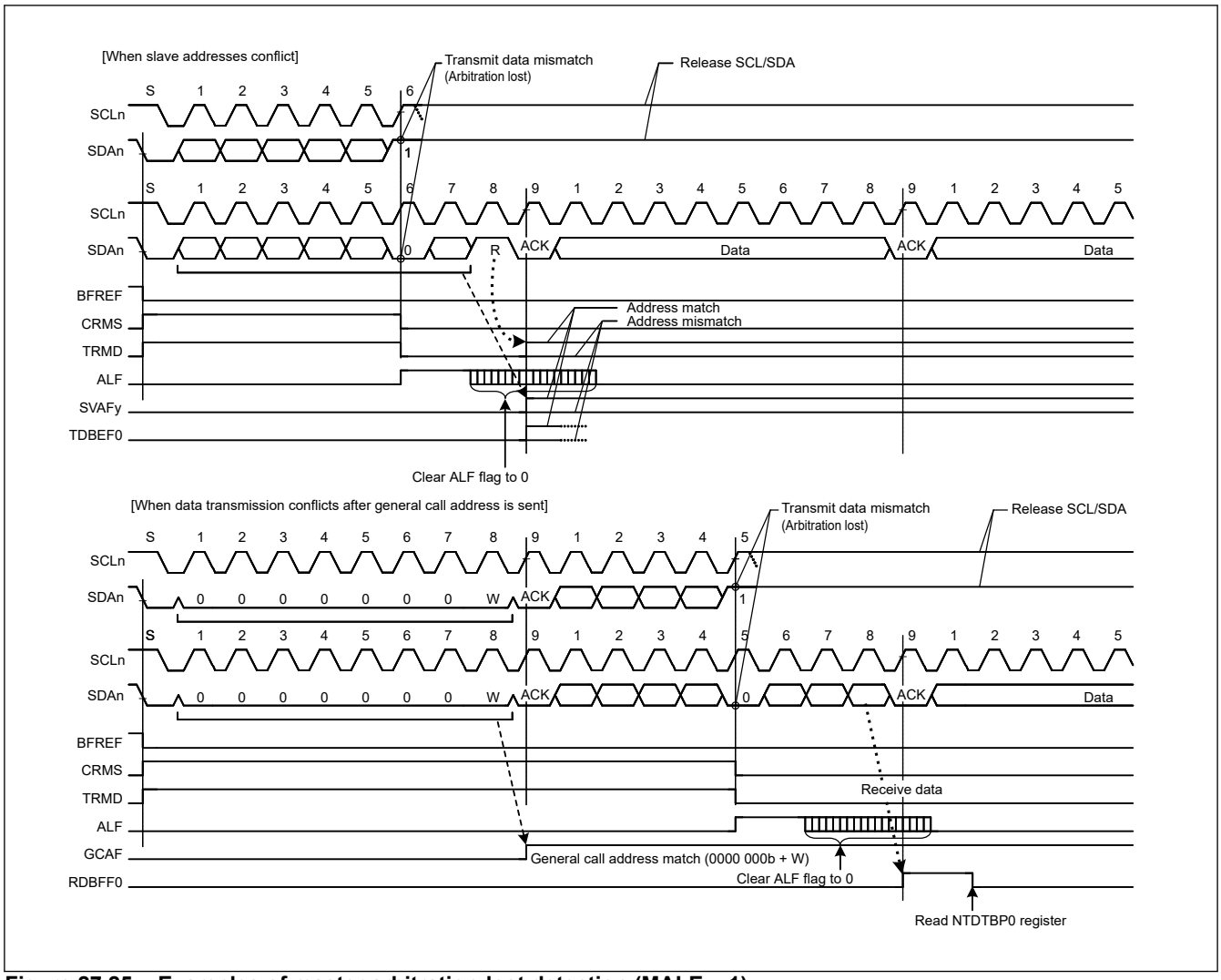


Figure 27.25 Examples of master arbitration-lost detection (MALE = 1)

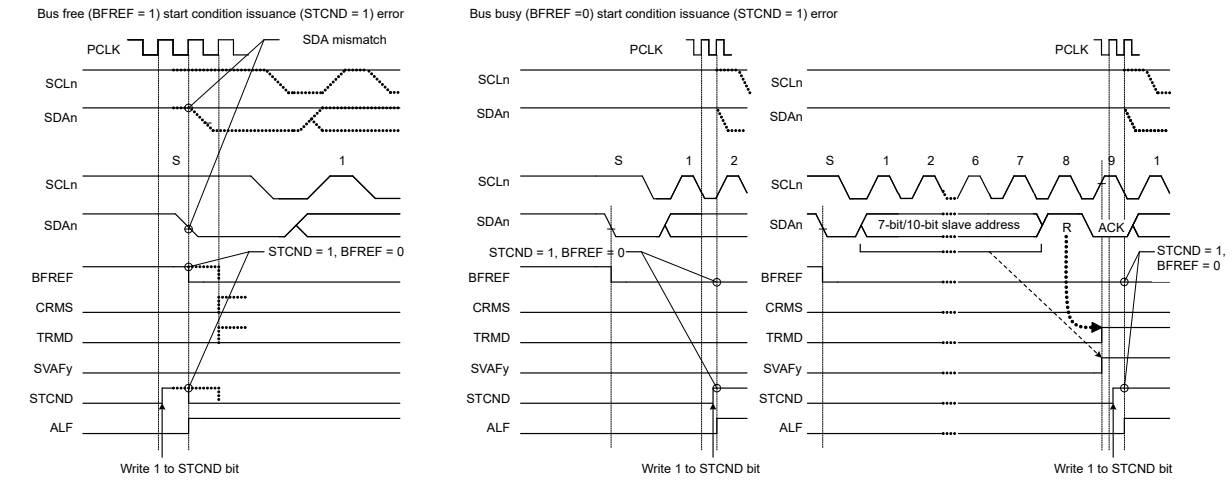


Figure 27.26 Arbitration-lost detection when a START condition is issued (MALE = 1)

(2) Arbitration-Lost Detection during NACK Transmission (NALE Bit)

The IIC has a function to cause arbitration to be lost if the internal SDA output level does not match the level on the SDAn line (the high output as the internal SDA output; i.e. the SDAn pin is in the high-impedance state) and the low level is detected on the SDAn line during transmission of NACK in receive mode. Arbitration is lost due to a conflict of NACK transmission and ACK transmission when two or more master devices receive data from the same slave device simultaneously in a multi-master system. Such conflict occurs when multiple master devices send/receive the same information through a single slave device.

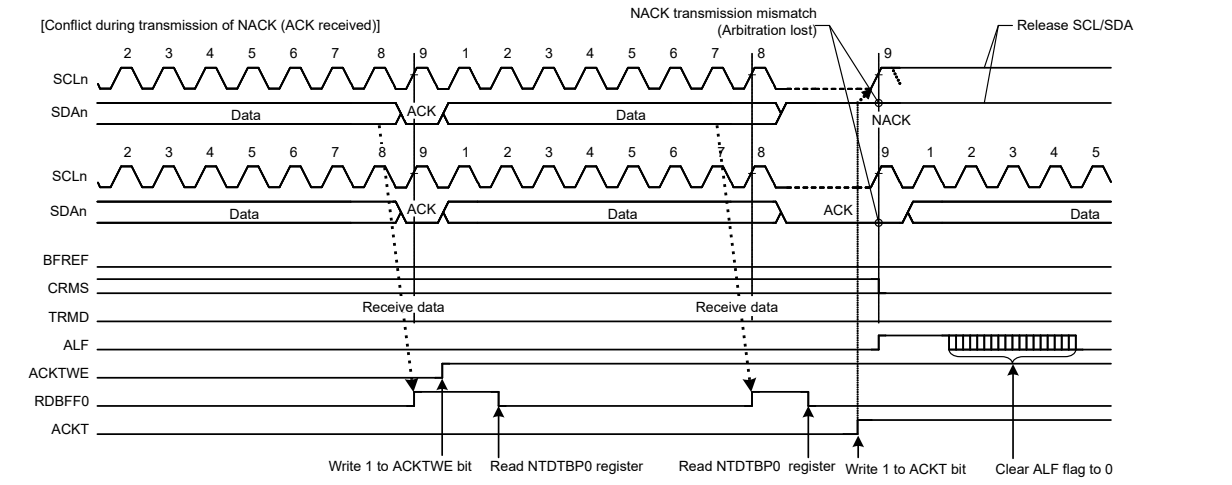


Figure 27.27 Example of arbitration-lost detection during transmission of NACK (NALE = 1)

The following section explains arbitration-lost detection using an example where two master devices (master A and master B) and a single slave device are connected through the bus. In this example, master A receives 2 bytes of data from the slave device, and master B receives 4 bytes of data from the slave device.

If master A and master B access the slave device simultaneously, because the slave address is identical, arbitration is not lost in both master A and master B during access to the slave device. Therefore, both master A and master B recognize that they have obtained the bus mastership and operate as such. In this example,, master A sends NACK when it has received 2 final bytes of data from the slave device. Meanwhile, master B sends ACK because it has not received the necessary 4 bytes of data. At this time, the NACK transmission from master A and the ACK transmission from master B conflict. In general, if a

conflict like this occurs, master A cannot detect ACK transmitted by master B and issues a stop condition. Therefore, the issuance of the stop condition conflicts with the SCL clock output of master B, which disturbs communication.

When this module receives ACK during transmission of NACK, it detects a defeat in conflict with other master devices and causes arbitration to be lost.

If arbitration is lost during transmission of NACK, this module is immediately released from the slave-matched state and enters slave receive mode. This prevents a stop condition from being issued, preventing a communication failure on the bus.

Similarly, in the ARP command processing of SMBus, the function to detect loss of arbitration during transmission of NACK is also available for eliminating the extra clock cycle processing (such as FFh transmission processing) necessary if the UDID (Unique Device Identifier) of assign address does not match in the Get UDID (general) processing after the Assign Address command.

The IIC detects arbitration-lost during transmission of NACK when the following condition is met while the BSTE.ALE bit = 1 and the BFCTL.NALE bit = 1 (arbitration-lost detection during NACK transmission enabled).

[Condition for arbitration-lost during NACK transmission]

- When the internal SDA output level does not match the SDAn line (ACK is received) during transmission of NACK (ACKCTL.ACKT bit = 1)

(3) Slave Arbitration-Lost Detection (SALE Bit)

The IIC has a function to cause arbitration to be lost if the data for transmission (the internal SDA output level) and the level on the SDAn line do not match (the high output as the internal SDA output, that is, the SDAn pin is in the high impedance state) and the low level is detected on the SDAn line in slave transmit mode. This arbitration-lost detection function is mainly used when transmitting a UDID (Unique Device Identifier) over an SMBus.

If arbitration is lost during transmission of DATA, this module is immediately released from the slave-matched state and enters slave receive mode. This function can detect conflicts of data during transmission of UDIDs over an SMBus and eliminate subsequent redundant processing (processing for the transmission of 0xFF).

The IIC detects slave arbitration-lost when the following condition is met while the BSTE.ALE bit = 1 and the BFCTL.SALE bit = 1 (slave arbitration-lost detection enabled).

[Condition for slave arbitration-lost]

- When the transmit data excluding acknowledge (internal SDA output level) does not match the level on the SDAn line in slave transmit mode (bits CRMS and TRMD in the PRSST register = 01).

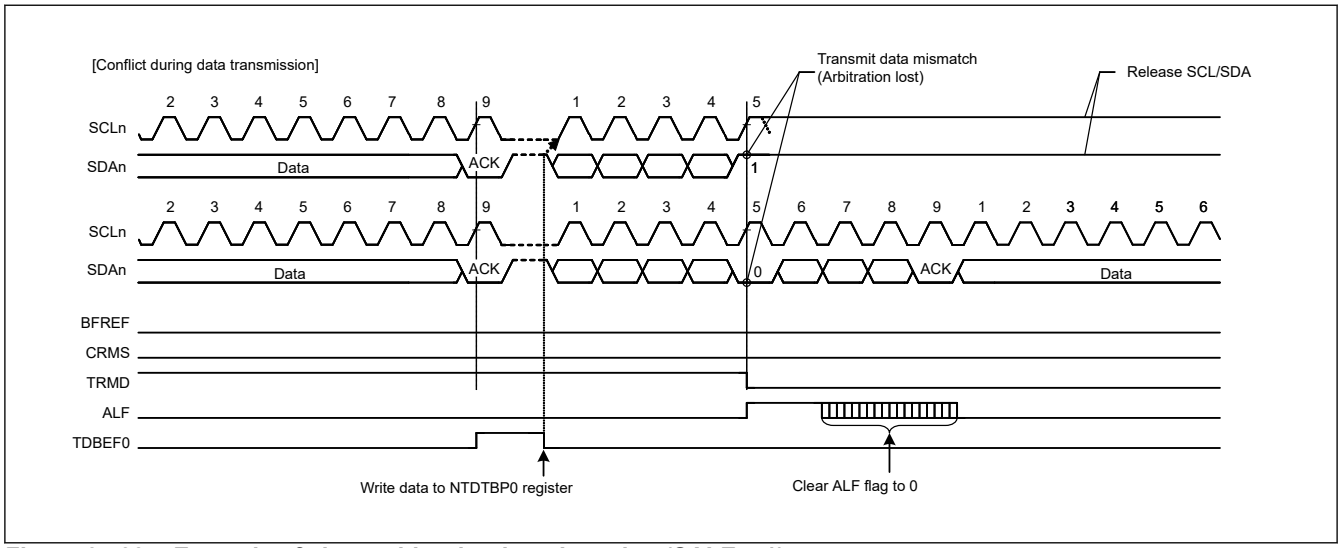


Figure 27.28 Example of slave arbitration-lost detection (SALE = 1)

27.3.1.3.5 Clock Stretching

(1) Function to Prevent Wrong Transmission of Transmit Data

When data have not been written to the I^2C bus transmit data register (NTDTBP0) with IIC in transmission mode (PRSST.TRMD = 1), the SCLn line is automatically held at the low level over the intervals shown below. This low-hold period is extended until data for transmission have been written, which prevents the unintended transmission of erroneous data.

Master transmit mode

- Low-level interval after a START condition or Repeated START condition is issued
- Low-level interval between the ninth clock cycle of one transfer and the first clock cycle of the next

Slave transmit mode

- Low-level interval between the ninth clock cycle of one transfer and the first clock cycle of the next

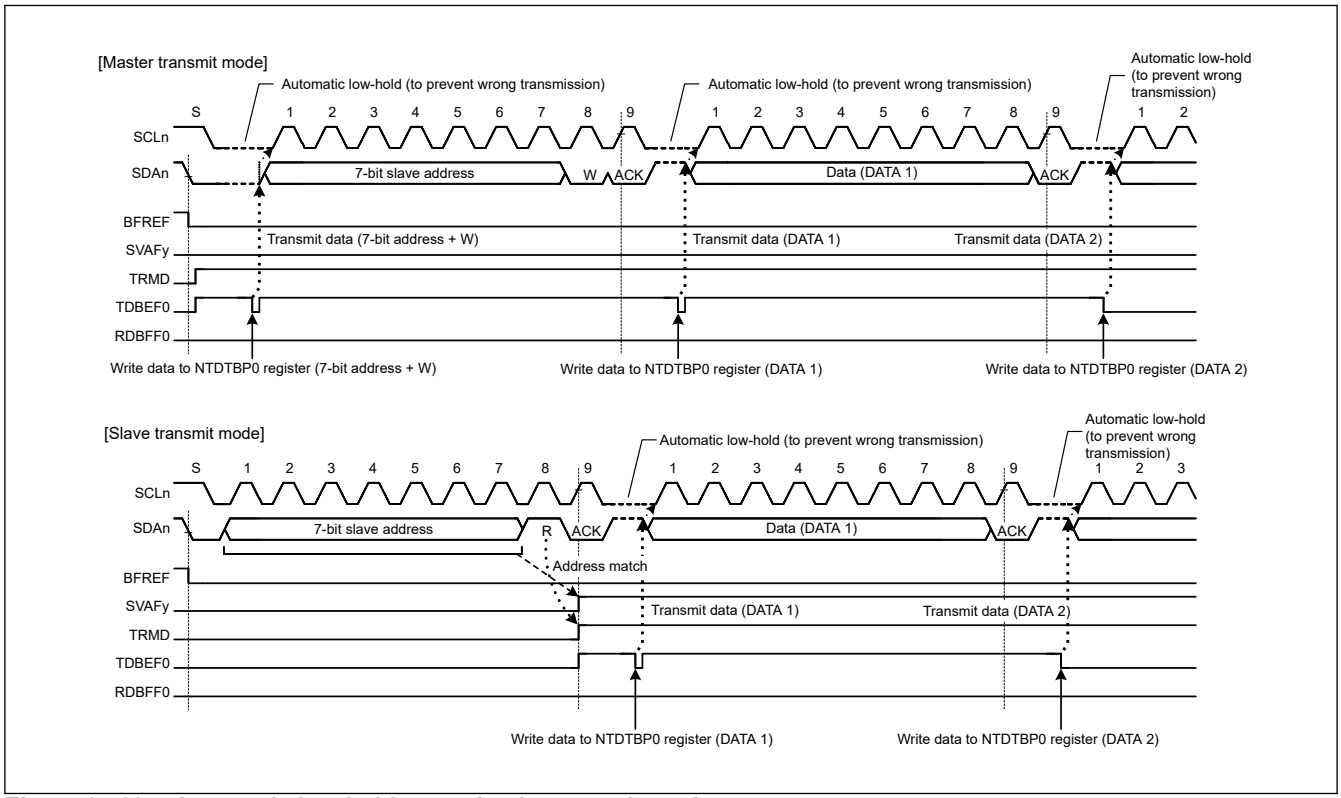


Figure 27.29 Automatic low-hold operation in transmit mode

(2) NACK Reception Transfer Abort Function

IIC has a function to abort transfer operation when NACK is received in transmit mode (PRSST.TRMD = 1). This function is enabled when the BSTE.NACKDE bit is set to 1 (transfer abort enabled). If the next transmit data has already been written (NTST.TDBEF0 = 0) when NACK is received, next data transmission at the falling edge of the ninth SCL clock cycle is automatically aborted. This prevents the SDAn line output level from being held low when the MSB of the next transmit data is 0.

If the transfer operation is aborted by this function (BST.NACKDF = 1), transmit operation and receive operation are discontinued. To restore transmit/receive operation, be sure to set the NACKDF flag to 0. In master transmit mode, restore operation using either of the methods below:

- After issuing a Repeated START condition, set the NACKDF flag to 0
- After issuing a STOP condition, set the NACKDF flag to 0 and then issue a START condition

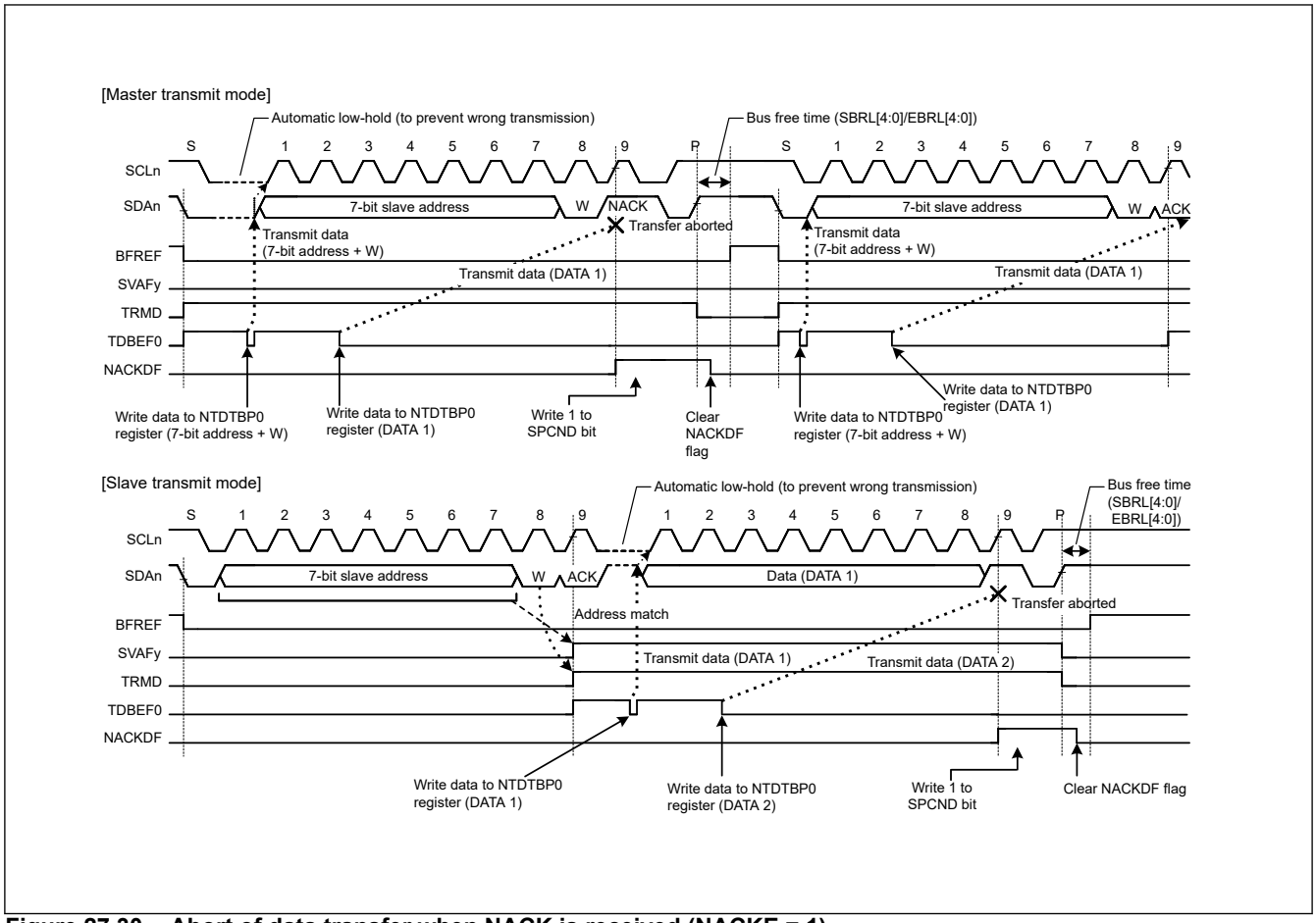


Figure 27.30 Abort of data transfer when NACK is received (NACKE = 1)

(3) Function to Prevent Failure to Receive Data

If response processing is delayed when receive data (NTDTBP0) read is delayed for a period of one transfer frame or more with receive data full (NTST.RDBFF0 = 1) in receive mode (PRSST.TRMD = 0), IIC holds the SCLn line low automatically immediately before the next data is received to prevent failure to receive data.

This function to prevent failure to receive data using the automatic low-hold function is also enabled even if the read processing of the final receive data is delayed and, in the meantime, IIC's own slave address or another slave address is received after a STOP condition is issued.

Sections in which the SCLn line is held low can be selected with a combination of the RWE and ACKTWE bits in SCSTRCTL.

(a) 1-Byte Receive Operation and Automatic Low-Hold Function Using the RWE Bit

When the SCSTRCTL.RWE bit is set to 1, IIC performs 1-byte receive operation using the RWE bit function.

Furthermore, when the SCSTRCTL.ACKTWE bit = 0, IIC automatically sends the ACKCTL.ACKT bit value for the acknowledge bit in the period from the falling edge of the eighth SCL clock cycle to the falling edge of the ninth SCL clock cycle, and automatically holds the SCLn line low at the falling edge of the ninth SCL clock cycle using the RWE bit function. This low-hold is released by reading data from NTDTBP0, which enables bytewise receive operation.

The RWE bit function is enabled for receive frames after a match with IIC's own slave address (including the general call address and host address) is obtained in master receive mode or slave receive mode.

(b) 1-Byte Receive Operation (ACK/NACK Transmission Control) and Automatic Low-Hold Function Using the ACKTWE Bit

When the SCSTRCTL.ACKTWE bit is set to 1, IIC performs 1-byte receive operation using the ACKTWE bit function.

When the ACKTWE bit is set to 1, the NTST.RDBFF0 flag (receive data full) is set to 1 at the rising edge of the eighth SCL clock cycle, and the SCLn line is automatically held low at the falling edge of the eighth SCL clock cycle. This lowhold is

released by writing a value to the ACKCTL.ACKT bit, but cannot be released by reading data from NTDTBP0, which enables receive operation by the ACK/NACK transmission control according to the data received in byte units.

The ACKTWE bit function is enabled for receive frames after a match with IIC's own slave address (including the general call address and host address) is obtained in master receive mode or slave receive mode.

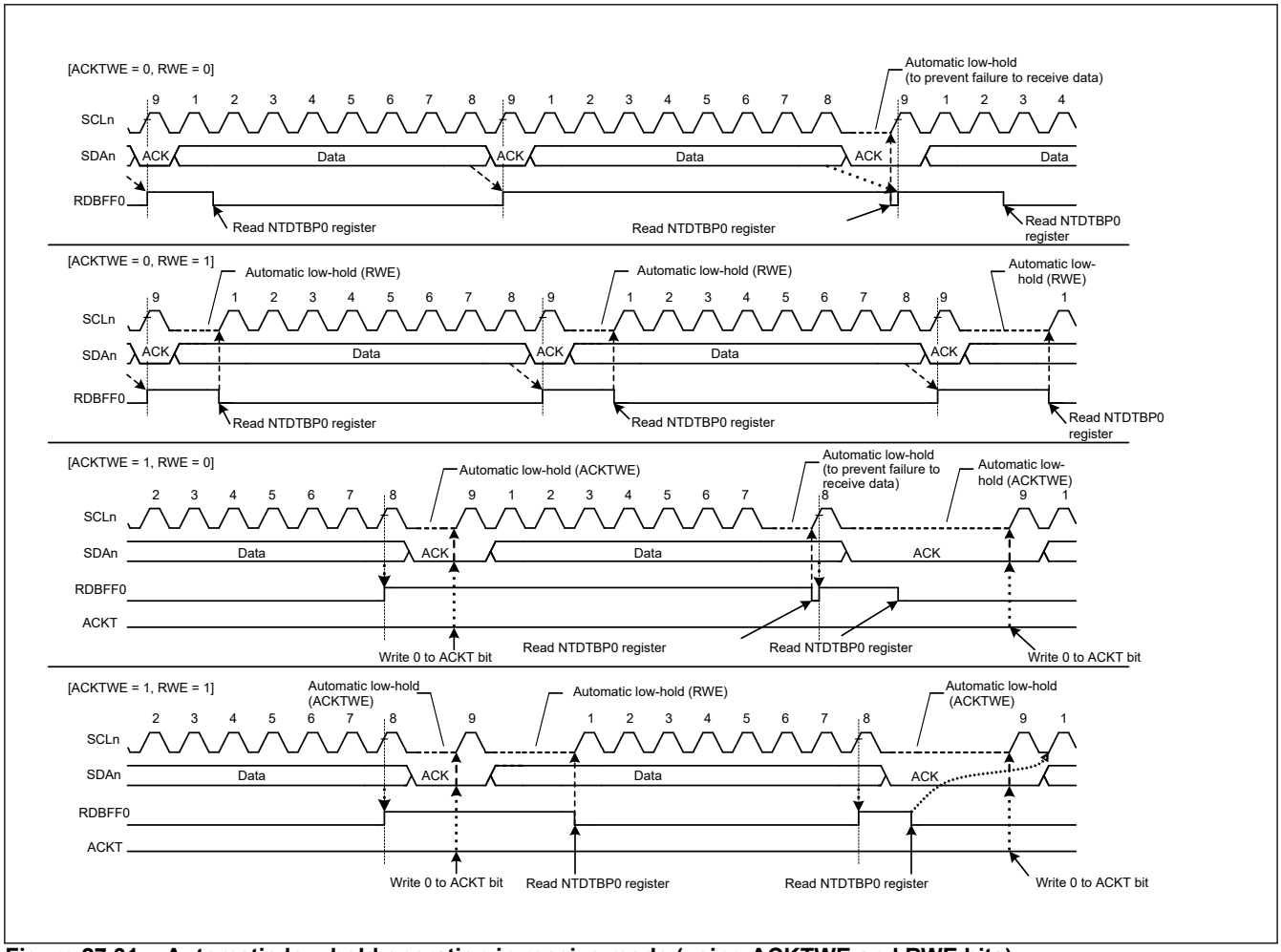


Figure 27.31 Automatic low-hold operation in receive mode (using ACKTWE and RWE bits)

27.3.1.3.6 Port Control

(1) Extra SCL Clock Cycle Output Function

In master mode, IIC module has a facility for the output of extra SCL clock cycles to release the SDAn line of the slave device from being held at the low level due to the master being out of synchronization with the slave device.

This function is mainly used in master mode to release the SDAn line of the slave device from the state of being fixed to the low level by including extra cycles of SCL output from IIC with single cycles of the SCL clock as the unit in the case of a bus error where IIC cannot issue a Repeated START condition or a STOP condition because the slave device is holding the SDAn line at the low level. Do not use this facility in normal situations. Using it when communications are proceeding correctly will lead to malfunctions.

When the OUTCTL.EXCYC bit is set to 1, an additional clock pulse at the frequency set by the REFCKCTL.IREFCKS[2:0] bits and the STDBR.SBRHO[7:0] and STDBR.SBRLO[7:0] registers is output from the SCLn pin. After output of this clock pulse, the EXCYC bit automatically becomes 0. After confirming that the EXCYC bit is 0, wait for the setup time of the Repeated START condition or STOP condition, and then confirm the detection of the Repeated START condition or STOP condition. If the Repeated START condition or STOP condition is not detected, consecutive additional clock pulses can be output by writing 1 to the EXCYC bit again.

When IIC module is in master mode and the slave device is holding the SDAn line at the low level because synchronization with the slave device has been lost due to the effects of noise, etc., the output of a Repeated START condition or a STOP

condition is not possible. The facility for output of an extra cycle of the SCL clock can be used to output extra cycles of SCL one by one to make the slave device release the SDAn line from being held at the low level, thus recovering the bus from an unusable state. Release of the SDAn line by the slave device can be monitored by reading the SDILV bit in PRSTDBG. After the SDAn line has been released by the slave device, the preset of a Repeated START condition or a STOP condition is issued.

Use this function with the BFCTL.MALE bit set to 0 (master arbitration-lost detection is disabled).

[Output conditions for using the EXCYC bit in OUTCTL]

- When the bus is free (BFREF flag in BCST = 1) or in master mode (CRMS bit = 1 in PRSST and BFREF flag = 0 in BCST)
- When the communication device does not hold the SCLn line low

Figure 27.32 shows the operation timing of the extra SCL clock cycle output function (EXCYC bit).

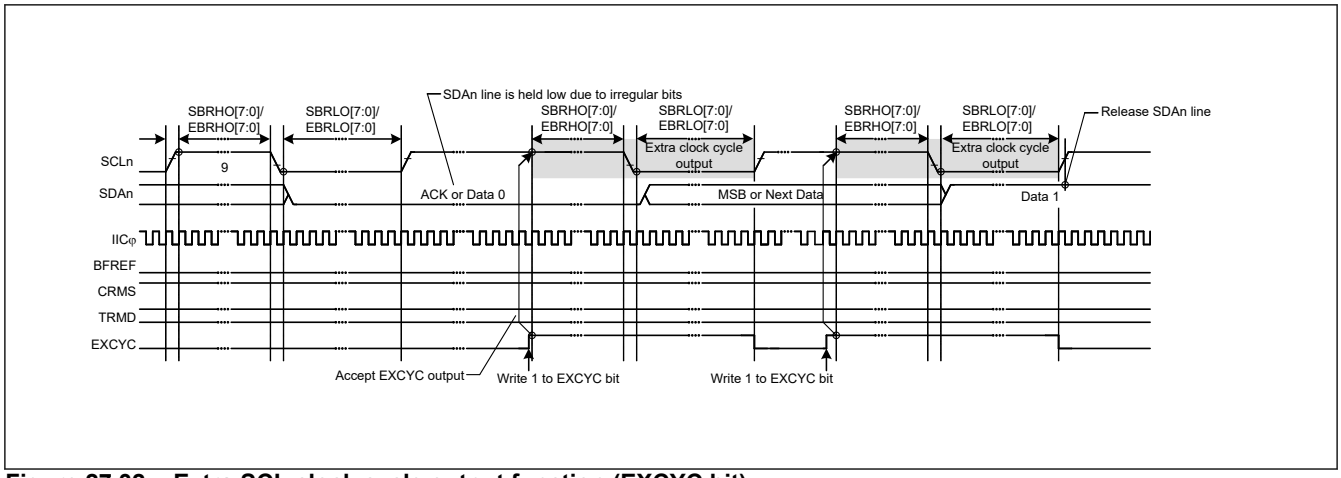


Figure 27.32 Extra SCL clock cycle output function (EXCYC bit)

27.3.1.3.7 SMBus Operation

IIC is available for data communication conforming to the SMBus (Version 2.0). To perform SMBus communication, set the BFCTL.SMBS bit to 1. To use the transfer rate within a range of 10 kbps to 100 kbps of the SMBus specification, set the REFCKCTL.IREFCKS[2:0] bits, the STDBR.SBRHO[7:0] bits, and the STDBR.SBRLO[7:0] bits. In addition, determine the values of the OUTCTL.SDODCS bit and the OUTCTL.SDOD[2:0] bits to meet the data hold time specification of 300 ns or more. If IIC is used only as an I^2C slave device, the transfer rate setting is not necessary, whereas the STDBR.SBRLO[7:0] bits needs to be set to a value longer than the data setup time (250 ns).

For the SMBus device default address (1100 001), use one of the slave device address table basic registers 0 to 2 (SDATBASy.SDSTAD[6:0] bits (y = 0 to 2), and set the corresponding SDATBASy.SDADLS bit (7-bit/10-bit address format select) (y = 0 to 2) to 0 (7-bit address format).

When transmitting the UDID (Unique Device Identifier), set the BFCTL.SALE bit to 1 to enable the slave arbitrationlost detection function.

(1) SMBus Timeout Measurement

(a) Measuring timeout of slave device

The following period (timeout interval: $T_{LOW:SEXT}$) must be measured for slave devices in SMBus communication.

- From START condition to STOP condition

To measure timeout for slave devices, measure the period from START condition detection to STOP condition detection with the GPT timer using a START condition detection interrupt (IICn_EEI) and STOP condition detection interrupt (IICn_EEI) of IIC. The measured timeout period must be within the total clock low-level period [slave device] $T_{LOW:SEXT}$: 25 ms (max.) of the SMBus specification.

If the time measured with the GPT exceeds the clock low-level detection timeout $T_{TIMEOUT}$: 25 ms (min.) of the SMBus specification, the slave device must release the bus by writing 1 to the RSTCTL.INTLRST bit to issue an internal reset of

IIC. When an internal reset is issued, IIC stops driving the bus for the SCLn pin and SDAn pin and make the SCLn/SDAn pin outputs high-impedance, which releases the bus.

(b) Measuring timeout of master device

The following periods (timeout interval: $T_{LOW:MEXT}$) must be measured for master devices in SMBus communication.

- From START condition to acknowledge bit
- Between acknowledge bits
- From acknowledge bit to STOP condition

To measure timeout for master devices, measure these periods with the GPT timer using a START condition detection interrupt (IICn_EEI), STOP condition detection interrupt (IICn_EEI), and transmit end interrupt (IICn_TEND) or receive data buffer full interrupt (IICn_RX) of IIC. The measured timeout period must be within the total clock lowlevel extended period (master device) $T_{LOW:MEXT}$: 10 ms (max.) of the SMBus specification, and the total of all $T_{LOW:MEXT}$ from START condition to STOP condition must be within $T_{LOW:SEXT}$: 25 ms (max.).

For the ACK receive timing (rising edge of the ninth SCL clock cycle), monitor the BST.TENDF flag in master transmit mode (master transmitter) and the NTST.RDBFF0 flag in master receive mode (master receiver). For this reason, perform bytewise transmit operation in master transmit mode, and hold the SCSTRCTL.ACKTWE bit 0 until the byte just before reception of the final byte in master receive mode. While the ACKTWE bit = 0, the RDBFF0 flag is set to 1 at the rising edge of the ninth SCL clock cycle.

If the period measured with the GPT exceeds the total clock low-level extended period (master device) $T_{LOW:MEXT}$: 10 ms (max.) of the SMBus specification or the total of measured periods exceeds the clock low-level detection timeout $T_{TIMEOUT}$: 25 ms (min.) of the SMBus specification, the master device must stop the transaction by issuing a STOP condition. In master transmit mode, immediately stop the transmit operation (writing data to NTDTBP0).

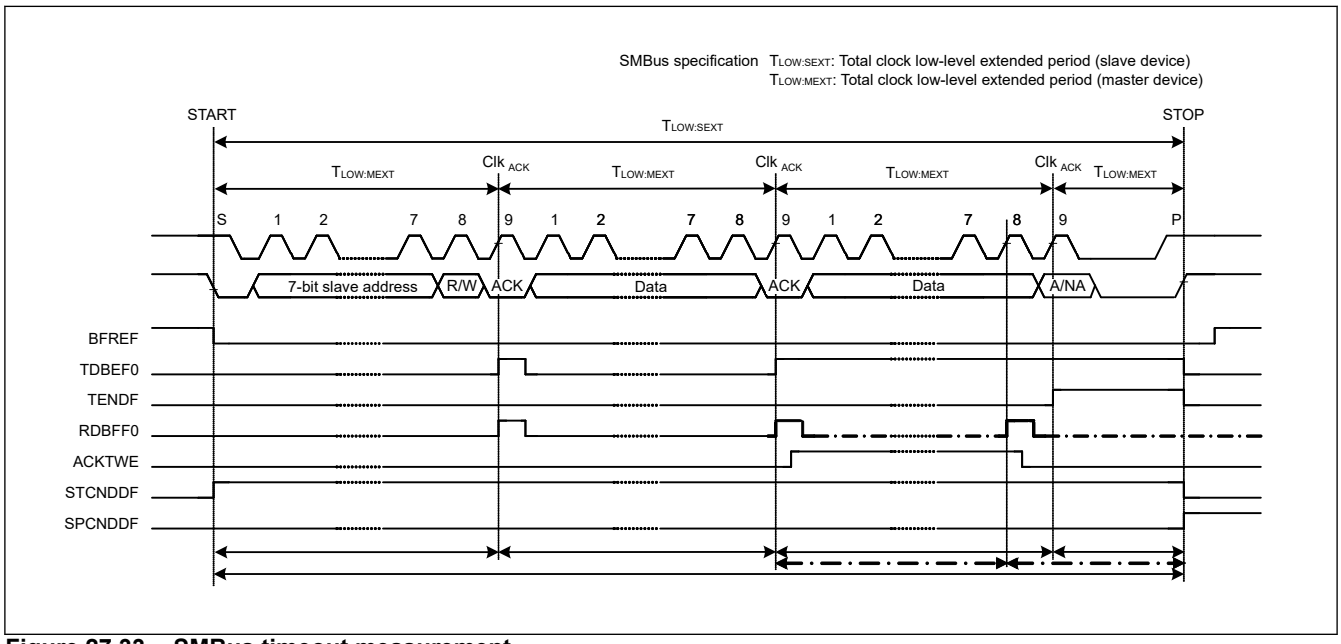


Figure 27.33 SMBus timeout measurement

(2) Packet Error Code (PEC)

This MCU incorporates a CRC calculator. The CRC calculator enables transmission of a packet error code (PEC) or checking the received data of the SMBus in data communication of IIC. For the CRC generating polynomials of the CRC calculator, see section 31, Cyclic Redundancy Check (CRC).

The PEC data in master transmit mode can be generated by writing all transmit data to the CRC data input register (CRCDIR) in the CRC calculator.

The PEC data in master receive mode can be checked by writing all receive data to CRCDIR in the CRC calculator and comparing the obtained value in the CRC data output register (CRCDOR) with the received PEC data.

To send ACK or NACK according to the match or mismatch result when the final byte is received as a result of the PEC code check, set the SCSTRCTL.ACKTWE bit to 1 before the rising edge of the eighth SCL clock cycle during reception of the final byte, and hold the SCLn line low at the falling edge of the eighth clock cycle.

(3) SMBus Host Notification Protocol (Notify ARP Master Command)

In communications over an SMBus, a slave device can temporarily act as a master device to notify the SMBus host (or ARP master) of its own slave address or to request its own slave address from the SMBus host.

For a product of this MCU to operate as an SMBus host (or ARP master), the host address (0001 000) sent from the slave device must be detected as a slave address, so IIC has a function for detecting the host address. To detect the host address as a slave address, set the BFCTL.SMBS bit and the SVCTL.HOAE bit to 1. Operation after the host address has been detected is the same as normal slave operation.

27.3.1.4 Error Detection

27.3.1.4.1 Timeout Error Detection

IIC includes a timeout function for detecting when the SCLn line has been stuck longer than the predetermined time. IIC can detect an abnormal bus state by monitoring that the SCLn line is stuck low or high for a predetermined time.

The timeout function monitors the SCLn line state and counts the low-level period or high-level period using the internal counter. The timeout function resets the internal counter each time the SCLn line changes (rising or falling), but continues to count unless the SCLn line changes. If the internal counter overflows due to no SCLn line change, IIC can detect the timeout and report the bus hung state.

This timeout function is enabled when BSTE.TODE = 1. It detects a hung state that the SCLn line is stuck low or high during the following conditions: (When TMOCTL.TOMDS[1:0] = 00b)

- The bus is busy (BCST.BFREF = 0) in master mode (PRSST.CRMS = 1).
- IIC’s own slave address is detected (SVST register is not 0x0000) and the bus is busy (BCST.BFREF = 0) in slave mode (PRSST.CRMS = 0).
- The bus is free (BCST.BFREF = 1) while generation of a START condition is requested (CNDCTL.STCND = 1).

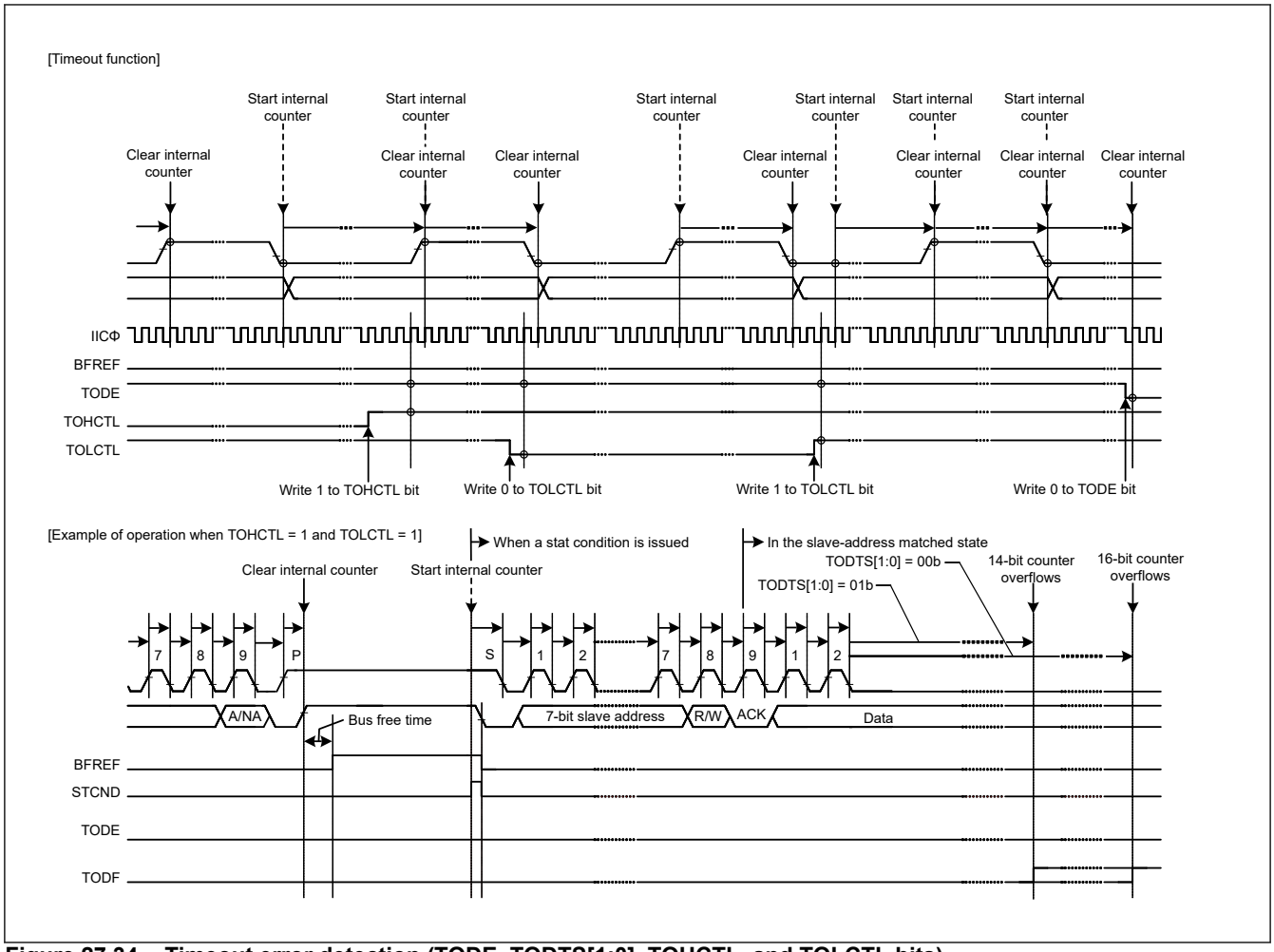


Figure 27.34 Timeout error detection (TODE, TODTS[1:0], TOHCTL, and TOLCTL bits)

27.3.1.5 Low Power Function

27.3.1.5.1 Wake Up function

IIC is equipped with the Wake-up function that causes the microcomputer to transition from low power consumption mode with system clock is stopped (software standby mode, snooze, etc.) to the normal operation. The Wake-up function is used to generate a Wake-up interrupt signal when the received data matches the address set to Wake-up interrupt factor also receives data in a state where the operating clock (PCLKA/IICCLK) is stopped (PCLKA/IICCLK asynchronous operation). This wake-up interrupt signal causes the microcomputer to transition to the normal operation. After Wake-up interrupt occurs, switch IIC to PCLKA/IICCLK synchronous operation, it will be able to continue the communication operation.

The Wake-up function has four wake-up operation modes (normal WU mode 1, normal WU mode 2, command recovery mode, and EEP response mode). The table below describes the behavior in these four wake-up operation modes.

Table 27.9 Wake-up operation mode (1 of 2)

	ACK response timing	ACK Type responded before recovery to PCLKA/IICCLK synchronous operation	SCL state before recovery to PCLKA/IICCLK synchronous operation
Normal WU mode 1	Before recovery to PCLKA/ IICCLK synchronous operation*1	ACK	Fixed to L
Normal WU mode 2	After recovery to PCLKA/IICCLK synchronous operation*2	Before recovery: no response (NACK level retained) After recovery: ACK response	Fixed to L
Command recovery mode	Before recovery to PCLKA/ IICCLK synchronous operation*1	ACK	Open

Table 27.9 Wake-up operation mode (2 of 2)

	ACK response timing	ACK Type responded before recovery to PCLKA/IICCLK synchronous operation	SCL state before recovery to PCLKA/IICCLK synchronous operation
EEP response mode	Before recovery to PCLKA/ IICCLK synchronous operation*1	NACK	Open

Note 1. Switching timing from PCLKA/IICCLK asynchronous operation to PCLKA/IICCLK synchronous operation is the fall of the 9th clock of SCL.
Note 2. Switching timing from PCLKA/IICCLK asynchronous operation to PCLKA/IICCLK synchronous operation is the fall of the 8th clock of SCL.

The following is able to select as Wake-Up interrupt factor.

- Host address detection (valid when SVCTL.HOAE = 1)
- General call address detection (valid when SVCTL.GCAE = 1)
- Slave address 0*1 detection (valid when SVCTL.SVAE0 = 1)
- Slave address 1*1 detection (valid when SVCTL.SVAE1 = 1)
- Slave address 2*1 detection (valid when SVCTL.SVAE2 = 1)

Note 1. 7-bit address only can be set. Set SDADLS bit to 0 in SDATBASy.

(1) Normal Wake-Up mode 1

This section describes the behavior, the timing, and a use case of normal WU mode 1.

A wake-up interrupt triggered by the match of the slave address makes the transition to the normal operation in the manner described below. Also, the detailed timing is provided in Figure 27.37.

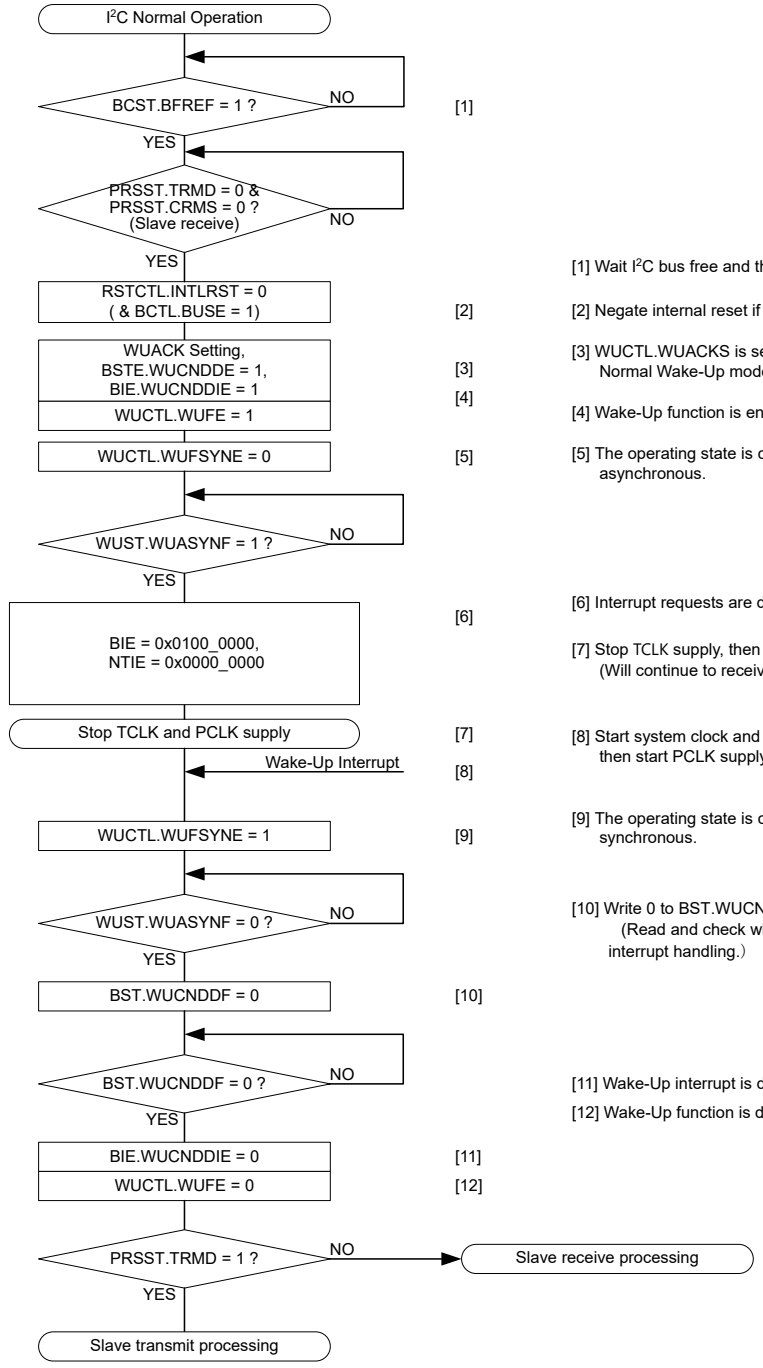
Before wake-up recovery:	ACK is sent in response to the data received with its own slave address.
During wake-up recovery:	ACK response is made at the 9th clock cycle of SCL, and the SCL is held low afterwards.*1
After wake-up recovery:	Normal operation continues.

Note 1. Between ninth clock cycle and first clock cycle during Wake-Up recovery, SCSTRCTL.RWE = 1 does not work.

If the slave address does not match, the SCL line is not held low after the fall of the 9th clock cycle of SCL, and the slave operation continues.

See Figure 27.35 below for a use case.

A wake-up interrupt is not generated at the transition to the normal operation if the transition is triggered by a cause (other recovery causes (IRQ)) other than a wake-up interrupt signal generated by a slave address match. BST.WUCNDDF is not set in this case. Carry out the following processing according to Figure 27.36.



[1] Wait I²C bus free and this module*1 stay in the standby state.

[2] Negate internal reset if assert the internal reset.

[3] WUCTL.WUACKS is set up according to each mode. (Normal Wake-Up mode 1/ Normal Wake-Up mode 2) Wake-Up Interrupt is enabled.

[4] Wake-Up function is enabled.

[5] The operating state is changed from PCLK/TCLK synchronous to PCLK/TCLK asynchronous.

[6] Interrupt requests are disabled except WUI.

[7] Stop TCLK supply, then stop PCLK supply. (Will continue to receive)

[8] Start system clock and TCLK supply by Wake-Up interrupt (address match), then start PCLK supply.

[9] The operating state is changed from PCLK/TCLK asynchronous to PCLK/TCLK synchronous.

[10] Write 0 to BST.WUCNDDF. (Read and check whether clearing has been completed befor return from interrupt handling.)

[11] Wake-Up interrupt is disabled.

[12] Wake-Up function is disabled.

Note: TCLK = IICCLK
PCLK = PCLKA

Figure 27.35 Use case of normal WU mode 1 (wake-up recovery by a wake-up interrupt triggered by the match of the slave address)

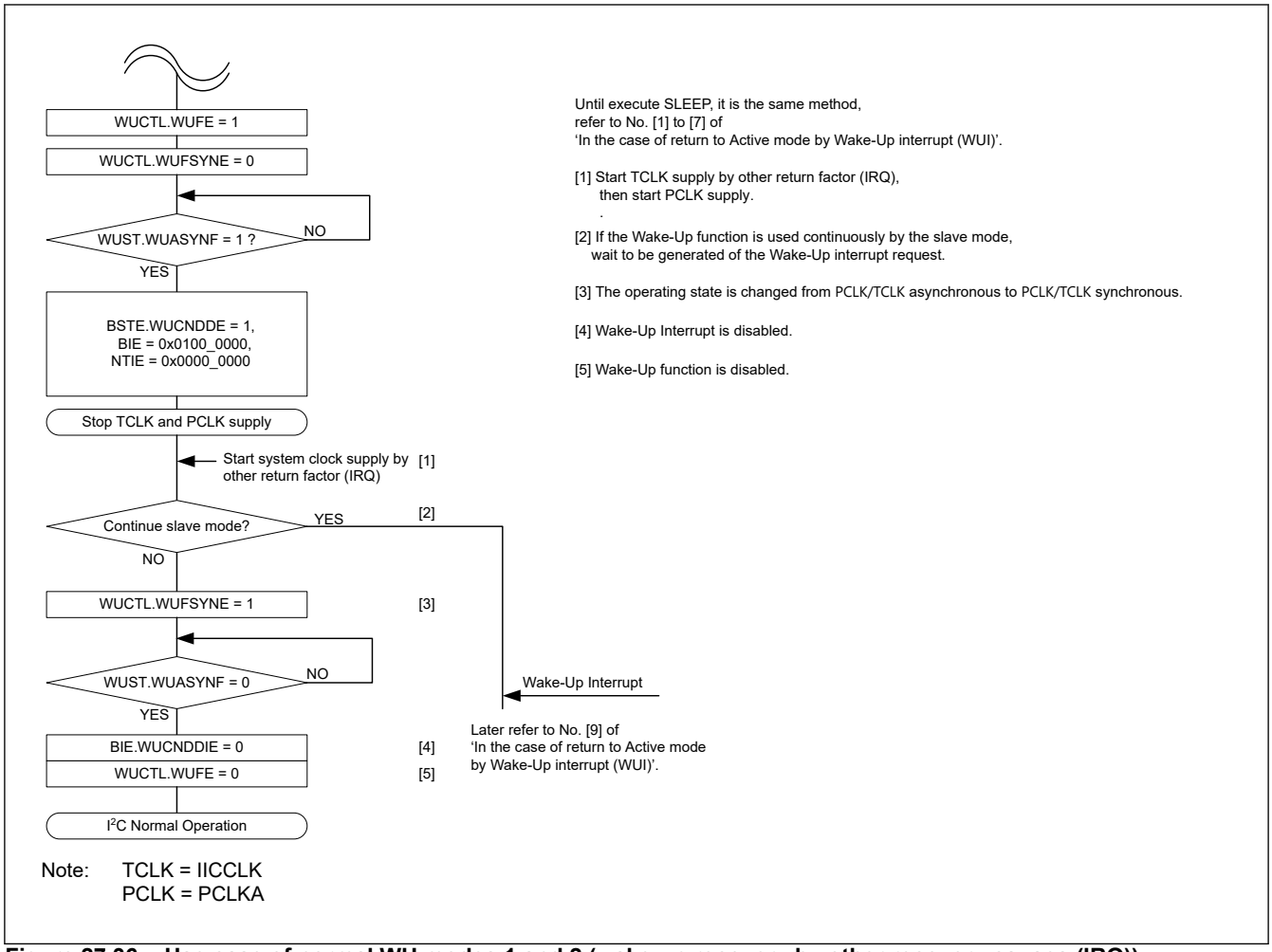
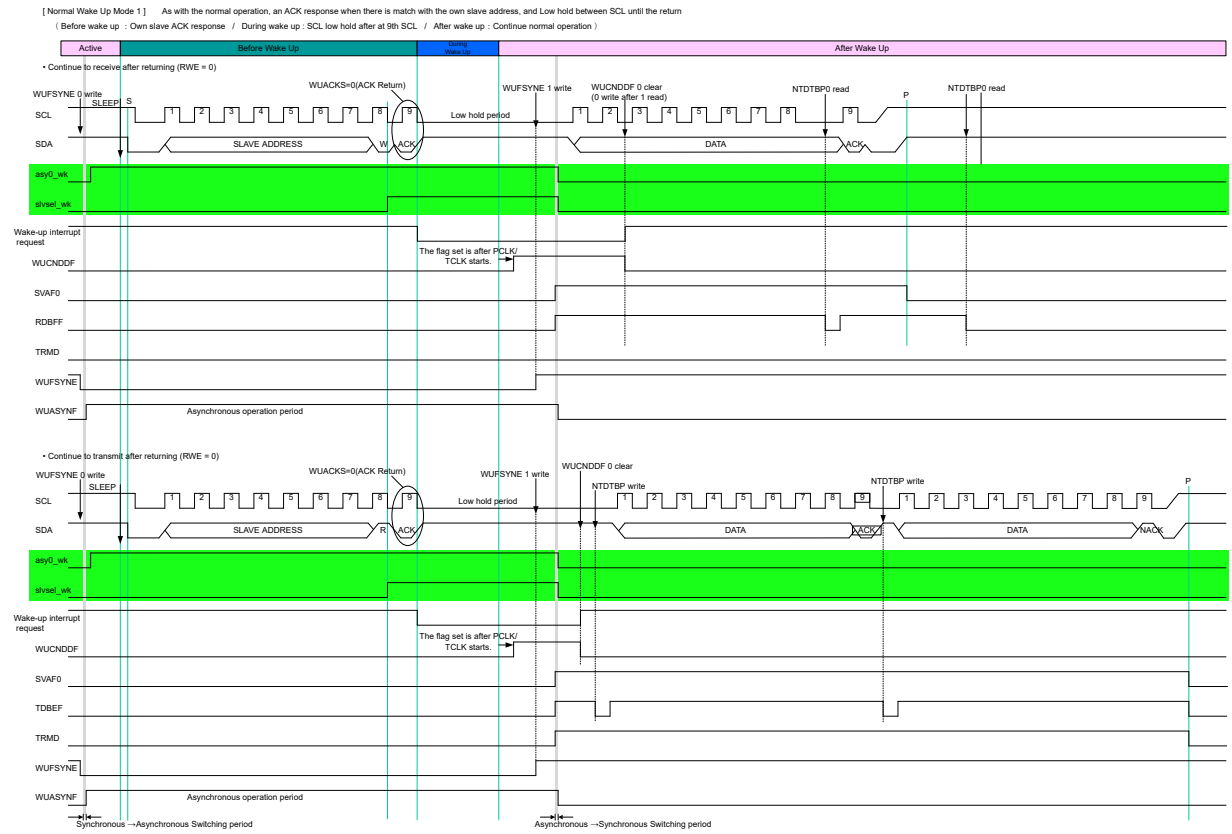


Figure 27.36 Use case of normal WU modes 1 and 2 (wake-up recovery by other recovery causes (IRQ))



Note: TCLK = IICCLK
PCLK = PCLKA

Figure 27.37 Timing of normal wake up mode 1

(2) Normal Wake Up Mode 2

This section describes the behavior, the timing, and a use case of normal WU mode 2.

A wake-up interrupt triggered by the match of the slave address makes the transition to the normal operation in the manner described below.

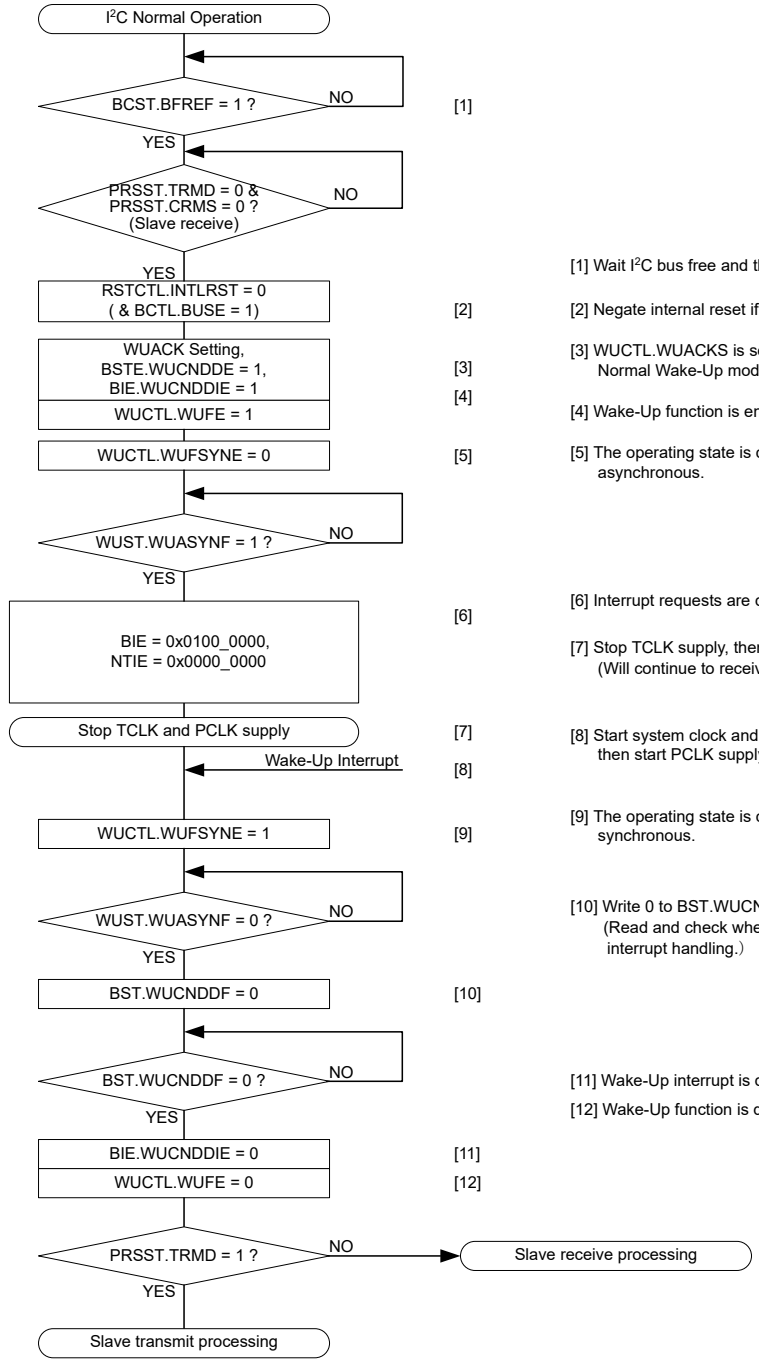
Also, the detailed timing is provided in Figure 27.39.

Before wake-up recovery:	No response to the data received with its own slave address (until 8th SCL cycle end)
During wake-up recovery:	Holding the SCL line low during the 8th and 9th clock cycles
After wake-up recovery:	Returning ACK at the 9th clock cycle of SCL, and continuing the normal operation

If the slave address does not match, the SCL line is not held low after the fall of the 8th SCL v clock cycle. The slave operation continues.

See Figure 27.38 below for a use case.

A wake-up interrupt is not generated at the transition to the normal operation if the transition is triggered by a cause (other recovery causes (IRQ)) other than a wake-up interrupt signal generated by a slave address match. BST.WUCNDDF is not set in this case. Carry out the following processing according to Figure 27.36.



[1] Wait I2C bus free and this module stay in the standby state.

[2] Negate internal reset if assert the internal reset.

[3] WUCTL.WUACKS is set up according to each mode. (Normal Wake-Up mode 1/ Normal Wake-Up mode 2) Wake-Up Interrupt is enabled.

[4] Wake-Up function is enabled.

[5] The operating state is changed from PCLK/TCLK synchronous to PCLK/TCLK asynchronous.

[6] Interrupt requests are disabled except WUI.

[7] Stop TCLK supply, then stop PCLK supply. (Will continue to receive)

[8] Start system clock and TCLK supply by Wake-Up interrupt (address match), then start PCLK supply.

[9] The operating state is changed from PCLK/TCLK asynchronous to PCLK/TCLK synchronous.

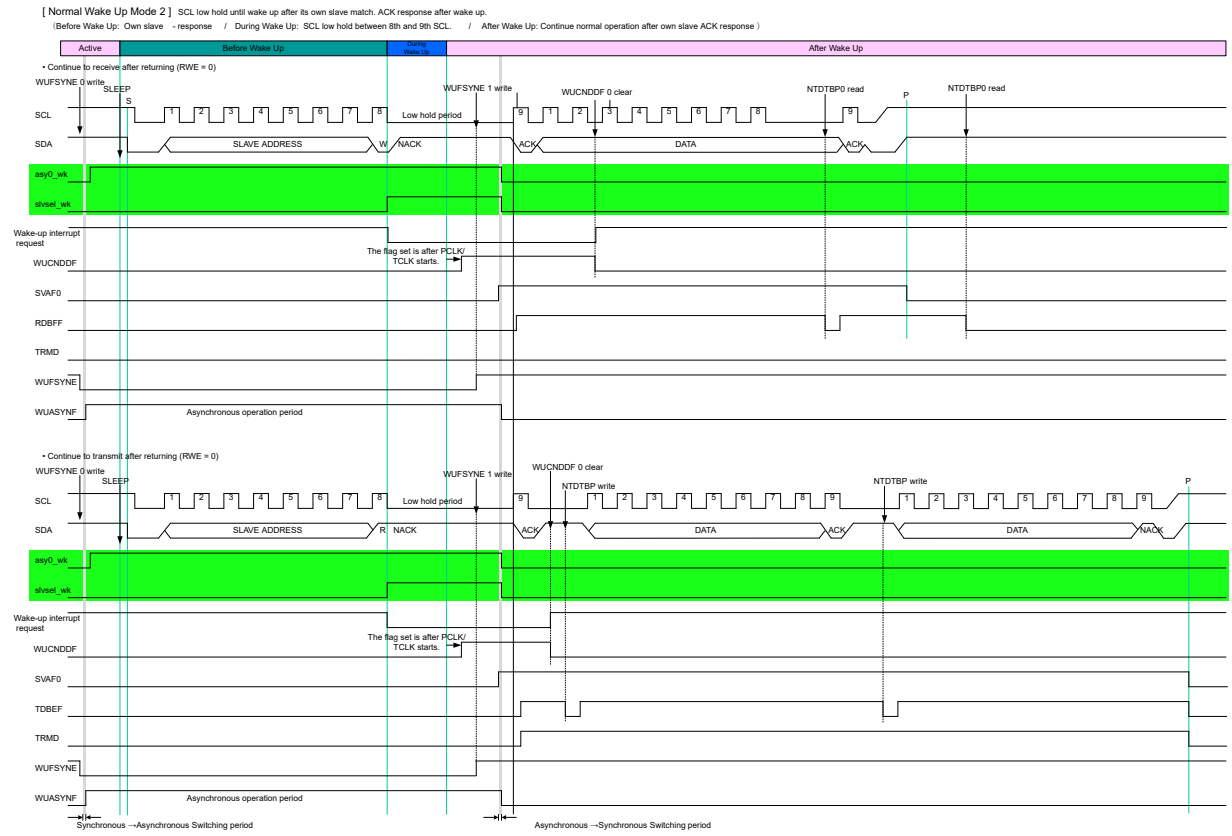
[10] Write 0 to BST.WUCNDDF. (Read and check whether clearing has been completed befor return from interrupt handling.)

[11] Wake-Up interrupt is disabled.

[12] Wake-Up function is disabled.

Note: TCLK = IICCLK
PCLK = PCLKA

Figure 27.38 Use case of normal WU mode 2 (wake-up recovery by a wake-up interrupt triggered by the match of the slave address)



Note: TCLK = IICCLK
PCLK = PCLKA

Figure 27.39 Timing of normal wake up mode 2

(3) Command recovery mode/ EEP response mode (Special Wake Up mode)

In the command recovery mode and EEP response mode, the SCL line is not held low during the wake-up recovery period (after the rise of the 9th clock cycle of SCL), so other I^2C devices can use the I^2C bus during this period. This section describes the behavior, the timing, and use cases of the command recovery mode and the EEP response mode.

A wake-up interrupt triggered by the match of the slave address makes the transition to the normal operation in the manner described below. Also, the detailed timing is provided in Figure 27.42.

Before wake-up recovery:	In response to the data received with its own slave address, ACK (command recovery mode) or NACK (EEP response mode) is returned.
During wake-up recovery:	The SCL line is not held low.
After wake-up recovery:	Normal operation continues after IIC initial setting.

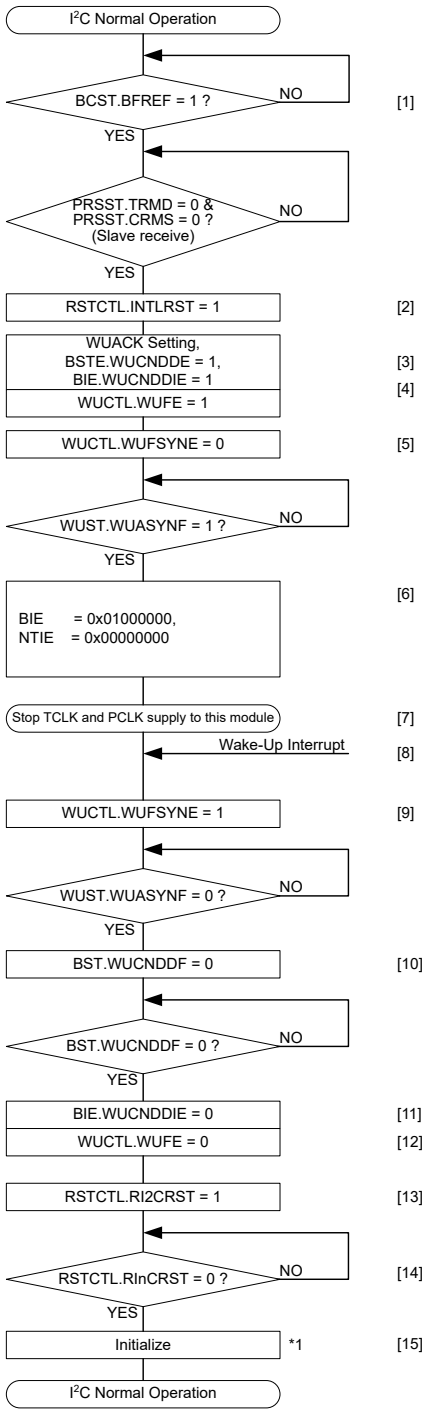
Note: Because the SCL line is not held low during wake-up recovery, the transmission/reception of the data that follows the slave address is not possible.

Note: The command recovery mode and the EEP response mode are internal reset (RSTCTL.INTLRST = 1) states. Therefore, the match of the slave address does not set the SVST flags (HOAF, GCAF, and SVAF2, SVAF1, SVAF0).

If the slave address does not match, the slave operation continues.

See Figure 27.41 below for a use case.

A wake-up interrupt is not generated at the transition to the normal operation if the transition is triggered by a cause (other recovery causes (IRQ)) other than a wake-up interrupt signal generated by a slave address match. BST.WUCNDDF is not set in this case. Carry out the following processing according to Figure 27.41.



[1] Wait I²C bus free and this module stay in the standby state.

[2] Assert the internal reset.

[3] WUCTL.WUACKS is set up according to each mode. (Command return mode/ EEP response mode) Wake-Up Interrupt is enabled.

[4] Wake-Up function is enabled.

[5] This module of operating state is changed from PCLK/TCLK synchronous to PCLK/TCLK asynchronous.

[6] Interrupt requests are disabled except WUI.

[7] Stop TCLK supply to this module, then stop PCLK supply to this module. (This module will continue to receive)

[8] Start system clock and TCLK supply to this module by Wake-Up interrupt (address match), then start PCLK supply to this module.

[9] This module of operating state is changed from PCLK/TCLK asynchronous to PCLK/TCLK synchronous.

[10] Write 0 to BST.WUCNDDF. (Read and check whether clearing has been completed before return from interrupt handling.)

[11] Wake-Up interrupt is disabled.

[12] Wake-Up function is disabled.

[13] Software reset.

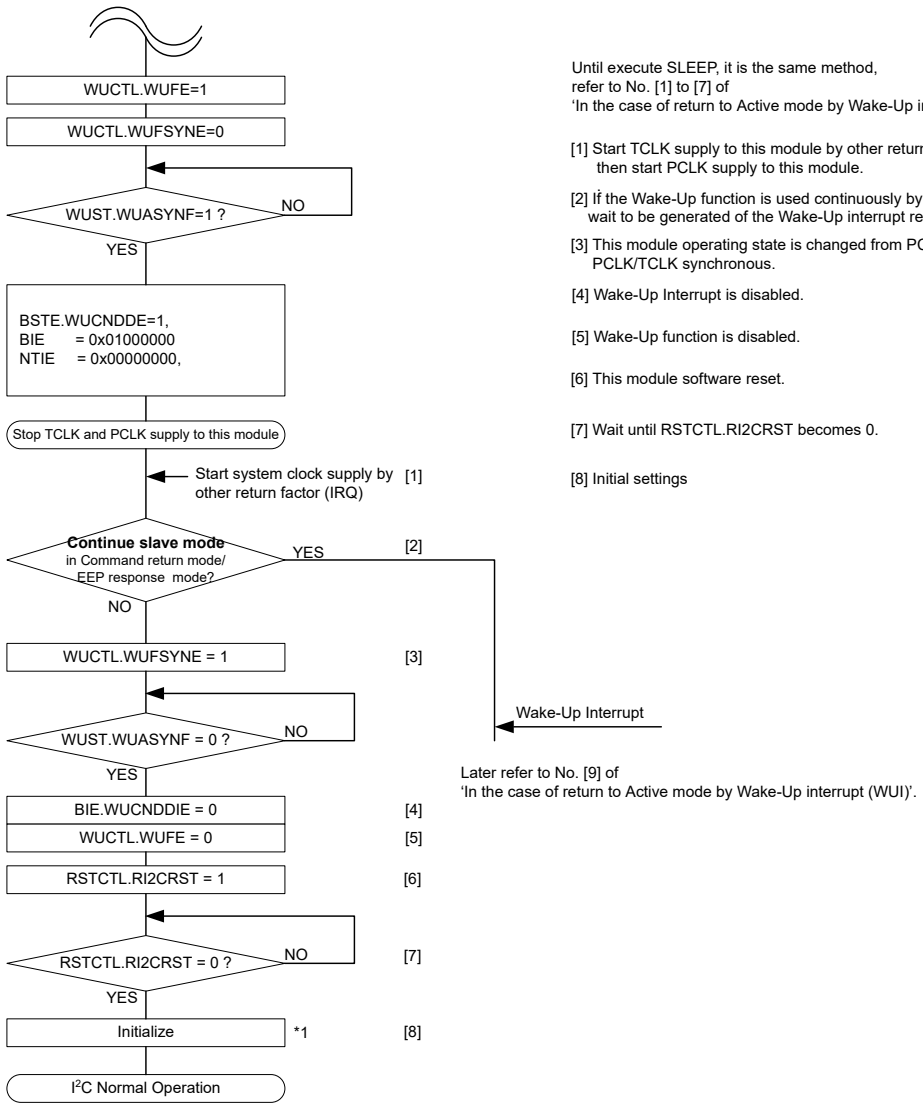
[14] Wait until RSTCTL.RI2CRST becomes 0.

[15] Initial settings

Note: TCLK = IICCLK
PCLK = PCLKA

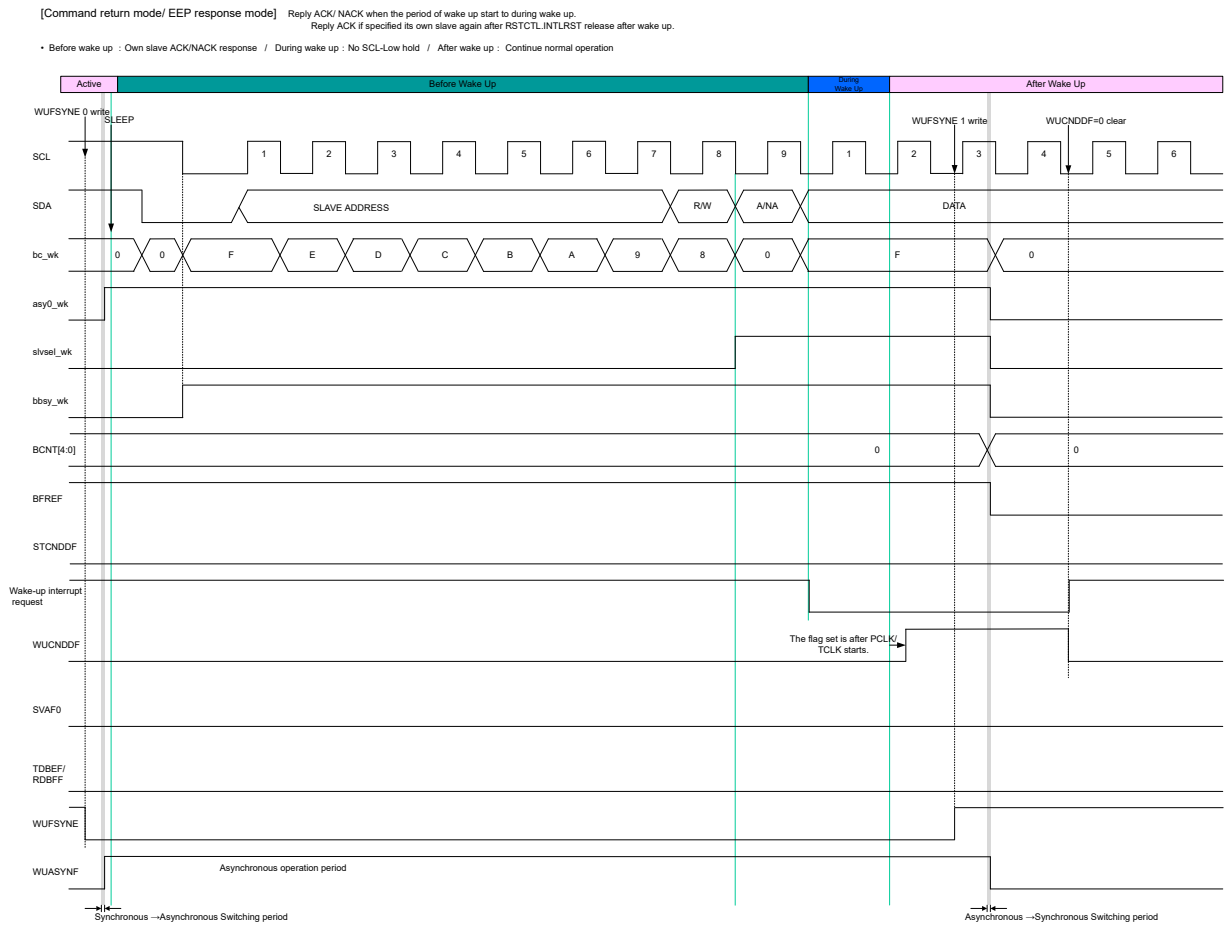
Note 1. See section 27.3.2.1. Initial Setting Flow for details of initialization.

Figure 27.40 Use case of command recover mode and EEP response mode (wake-up recovery by a wake-up interrupt triggered by the match of the slave address)



Note: TCLK = IICCLK
PCLK = PCLKA
Note 1. See section 27.3.2.1. Initial Setting Flow for details of initialization.

Figure 27.41 Use case of command recover mode and EEP response mode (wake-up recovery by other recovery causes (IRQ))



Note: TCLK = IICCLK
PCLK = PCLKA

Figure 27.42 Timing of command recovery mode/EEP response mode

(4) Precautions on the use of the Wake-Up function

Precautions on the use of the Wake-up function is shown below.

- Do not change the registers in IIC except the WUCTL.WUFSYNE bit while the WUST.WUASYNF flag = 1 (while PCLKA/IICCLK asynchronous operation).
- Set WUCTL.WUFE = BSTE.WUCNDDE = BIE.WUCNDDIE = 1 and PRSST.CRMS = PRSST.TRMD = 0 (slave reception mode) before switching PCLKA/IICCLK asynchronous mode.
- Cannot select the device ID and the 10-bit slave address for wake-up interrupt factor. Set the DVIDE bit in SVCTL and SDADLS bit in SDATBASy to 0.
- Sets all bits in BIE (TENDIE, NACKDIE, SPCNDDIE, STCNDDIE, ALIE, TODIE) and TDBEIE0 and RDBFIE0 bits in NTIE to 0 (Interrupt disabled) before switching the asynchronous operation.
- Do not use the timeout function while the Wake-up function is enabled (WUCTL.WUFE = 1).
- Wake-up interrupt is generated while PCLKA/IICCLK asynchronous operation (when WUST.WUASYNF = 1). In case of detecting slave address matching, The case of detect slave address match in PCLKA/IICCLK synchronous mode (WUST.WUASYNF = 0), does not occur Wake-up interrupt, and BST.WUCNDDF flag will be not set also.
- If WUCTL.WUFSYNE bit to 0 write timing and START condition of detecting a conflict, IIC might start the next reception in PCLKA/IICCLK synchronous operation mode. In this case, WUST.WUASYNF flag becomes 1 (switch to PCLKA/IICCLK asynchronous mode) when data communication is finished and detected STOP condition and starts the Wake-up event detection.

- If you want to switch from PCLKA/IICCLK asynchronous operation to PCLKA/IICCLK synchronous operation without address match detection, it will switch in the STOP condition detection. When the WUCTL.WUFSYNE bit was set to 1 in a bus free state, it is continued PCLKA/IICCLK asynchronous operation (Reception operation: waiting communication frame). WUST.WUASYNF flag becomes to 0 when IIC detect the STOP condition of the next communication frame, and IIC switches to PCLKA/IICCLK synchronous operation.
- After writing 0 to WUFSYNE bit in WUCTL, do not change IIC operation mode setting register (BFCTL, SCSTRCTL, ACKCTL, INCTL, SVCTL, SDATBASy) until switched to the PCLKA/IICCLK asynchronous operation from PCLKA/IICCLK synchronous operation (while WUST.WUASYNF flag = 1). If register value changes by the interrupt processing etc. in this period, IIC might malfunction without succeeding to the setting to the asynchronous operation.
- During PCLKA/IICCLK asynchronous operation (WUST.WUASYNF = 1), do not refer to each flag of SVST, BST, NTST register and BCST.BFREF flag.
- Do not set ACKCTL.ACKT = 1 in order to make an ACK response in the synchronization unit when Wake-up is performed by slave address match in Normal wake-up mode 2.

27.3.1.6 Other

27.3.1.6.1 SCL Synchronization Circuit

In generation of the SCL clock, IIC starts counting out the value for width at high level specified in STDBR.SBRHO[7:0] when it detects a rising edge on the SCLn line and drives the SCLn line low once counting of the width at high level is complete.

When IIC detects the falling edge of the SCLn line, it starts counting out the width at low level period specified in STDBR.SBRLO[7:0], and then stops driving the SCLn line (releases the line) once counting of the width at low level is complete. The SCL clock is thus generated.

If multiple master devices are connected to the I^2C bus, a collision of SCL signals may arise due to contention with another master device. In such cases, the master devices have to synchronize their SCL signals. Since this synchronization of SCL signals must be bit by bit, IIC is equipped with a facility (the SCL synchronization circuit) to obtain bit-by-bit synchronization of the SCL clock signals by monitoring the SCLn line while in master mode.

When IIC has detected a rising edge on the SCLn line and thus started counting out the width at high level specified in STDBR.SBRHO[7:0], and the level on the SCLn line falls because an SCL signal is being generated by another master device, IIC stops counting when it detects the falling edge, drives the level on the SCLn line low, and starts counting out the width at low level specified in STDBR.SBRLO[7:0]. When IIC finishes counting out the width at low level, it stops driving the SCLn line to the low level (releases the line). At this time, if the width at low level of the SCL clock signal from the other master device is longer than the width at low level set in this module, the width at low level of the SCL signal will be extended. Once the width at low level for the other master device has ended, the SCL signal rises because the SCLn line has been released. When IIC finishes outputting the low-level period of the SCL clock, the SCLn line is released and the SCL clock rises. That is, in cases of contention of SCL signals from more than one master, the width at high level of the SCL signal is synchronized with that of the clock having the narrower width, and the width at low level of the SCL signal is synchronized with that of the clock having the broader width. However, such synchronization of the SCL signal is only enabled when the SCSYNE bit in BFCTL is set to 1.

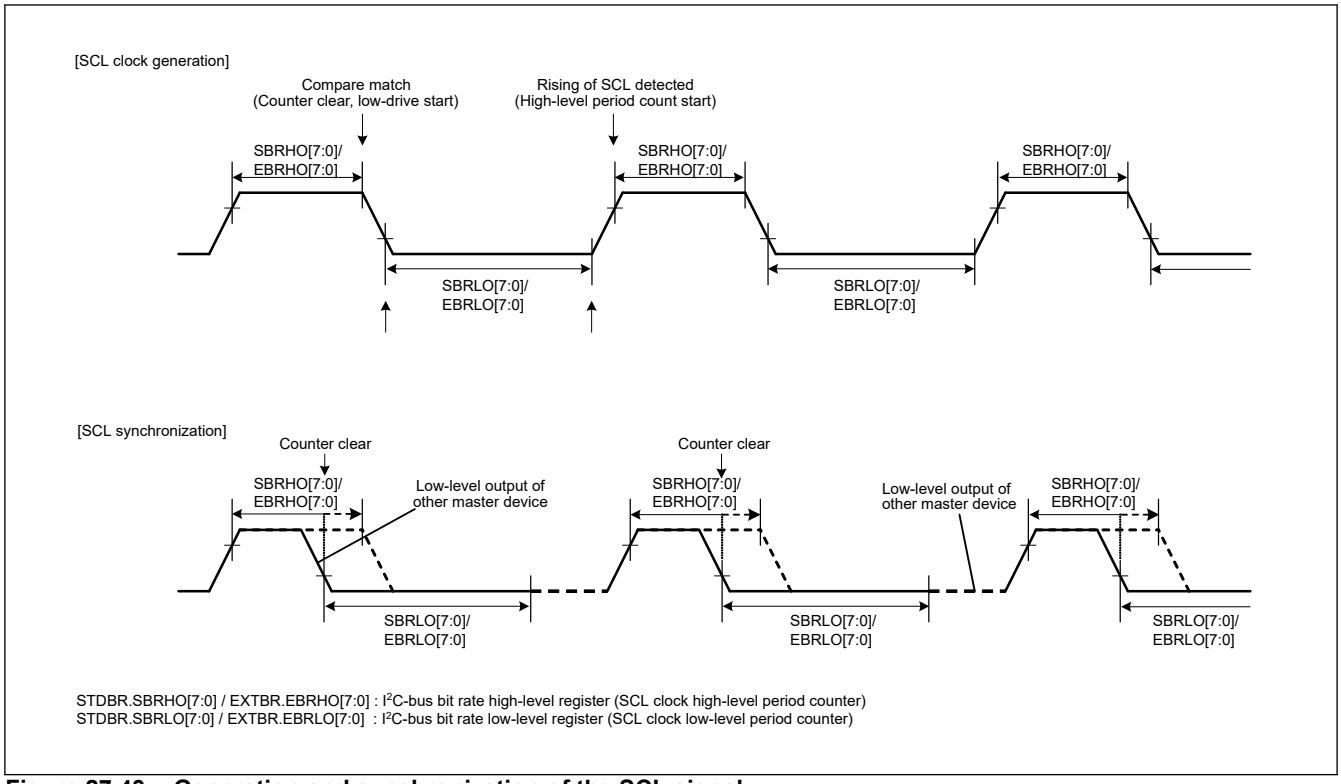


Figure 27.43 Generation and synchronization of the SCL signal

27.3.1.6.2 Facility for Delaying SDA Output

IIC module incorporates a facility for delaying output on the SDA line. The delay can be applied to all output (issuing of the START, Repeated START, and STOP conditions, data, and the ACK and NACK signals) on the SDA line.

With the SDA output delay facility, SDA output is delayed from detection of a falling edge of the SCL signal to ensure that the SDA signal is output within the interval over which the SCL clock is at the low level. Doing this leads to usage with the aim of preventing erroneous operation of communications devices, with the aim of satisfying the 300 ns (minimum) data-hold time requirement of the SMBus specification.

The output delay facility is enabled by setting the SDOD[2:0] bits in OUTCTL to any value other than 000b, and disabled by setting the same bits to 000b.

While the SDA output delay facility is enabled (while the SDOD[2:0] bits in OUTCTL are set to any value other than 000b), the SDODCS bit in OUTCTL selects the clock source for counting by the SDA output delay counter as the internal base clock (IICφ) for IIC module or as a clock signal derived by dividing the frequency of the internal base clock by two (IICφ/2). The counter counts the number of cycles set in the SDOD[2:0] bits in OUTCTL. After counting of the set number of cycles of delay is completed, IIC module places the required output (START, Repeated START, or STOP condition, data, or an ACK or NACK signal) on the SDA line.

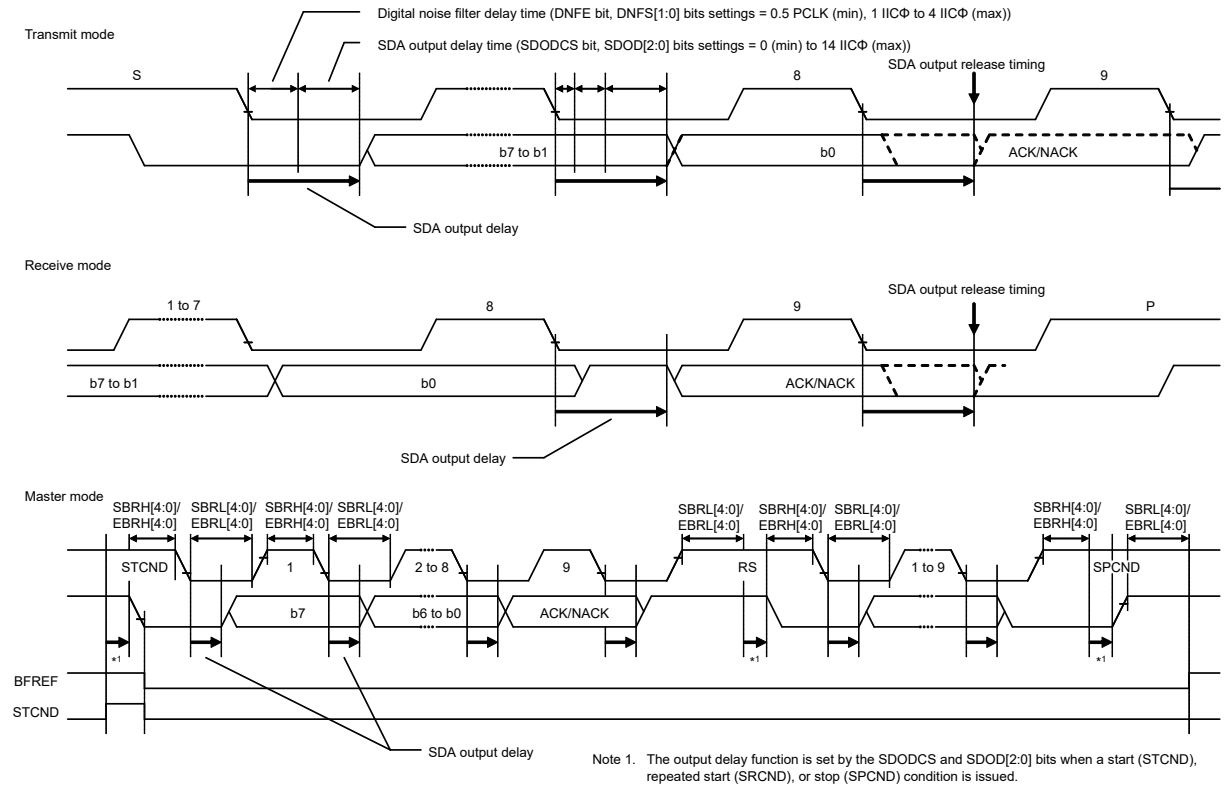


Figure 27.44 SDA output delay facility

27.3.1.6.3 Digital Noise-Filter Circuits

The states of the SCLn and SDAn pins are conveyed to the internal circuitry through digital noise-filter circuits. Figure 27.45 is a block diagram of the digital noise-filter circuit.

The on-chip digital noise-filter circuit of IIC consists of 16 flip-flop circuit stages connected in series and a match detection circuit. When HS mode is selected, only the first four flip-flop circuits stages are enabled.

The number of effective stages in the digital noise filter is selected by the INCTL.DNFS[3:0] bits. The selected number of effective stages determines the noise-filtering capability as a period from one to sixteen IICφ cycles.

The input signal to the SCLn pin (or SDAn pin) is sampled on rising edges of the IICφ signal. When the input signal level matches the output level of the number of effective flip-flop circuit stages as selected by the INCTL.DNFS[3:0] bits, the signal level is conveyed to the subsequent stage. If the signal levels do not match, the previous value is retained.

If the ratio between the frequency of the internal operating clock (IICCLK) and the transfer rate is small (For example, data transfer at 400 kbps with IICCLK = 4 MHz), the characteristics of the digital noise filter may lead to the elimination of needed signals as noise.

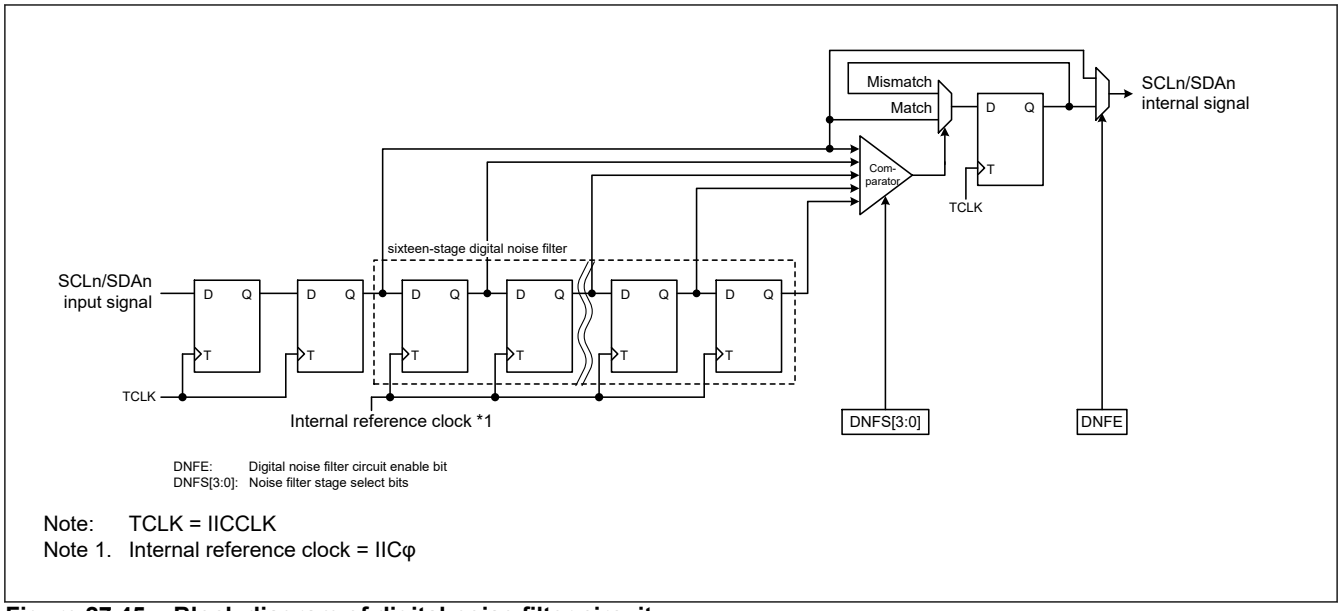


Figure 27.45 Block diagram of digital noise filter circuit

27.3.2 Operation

27.3.2.1 Initial Setting Flow

27.3.2.1.1 I2C Initial Setting Flow (Single Buffer Transfer)

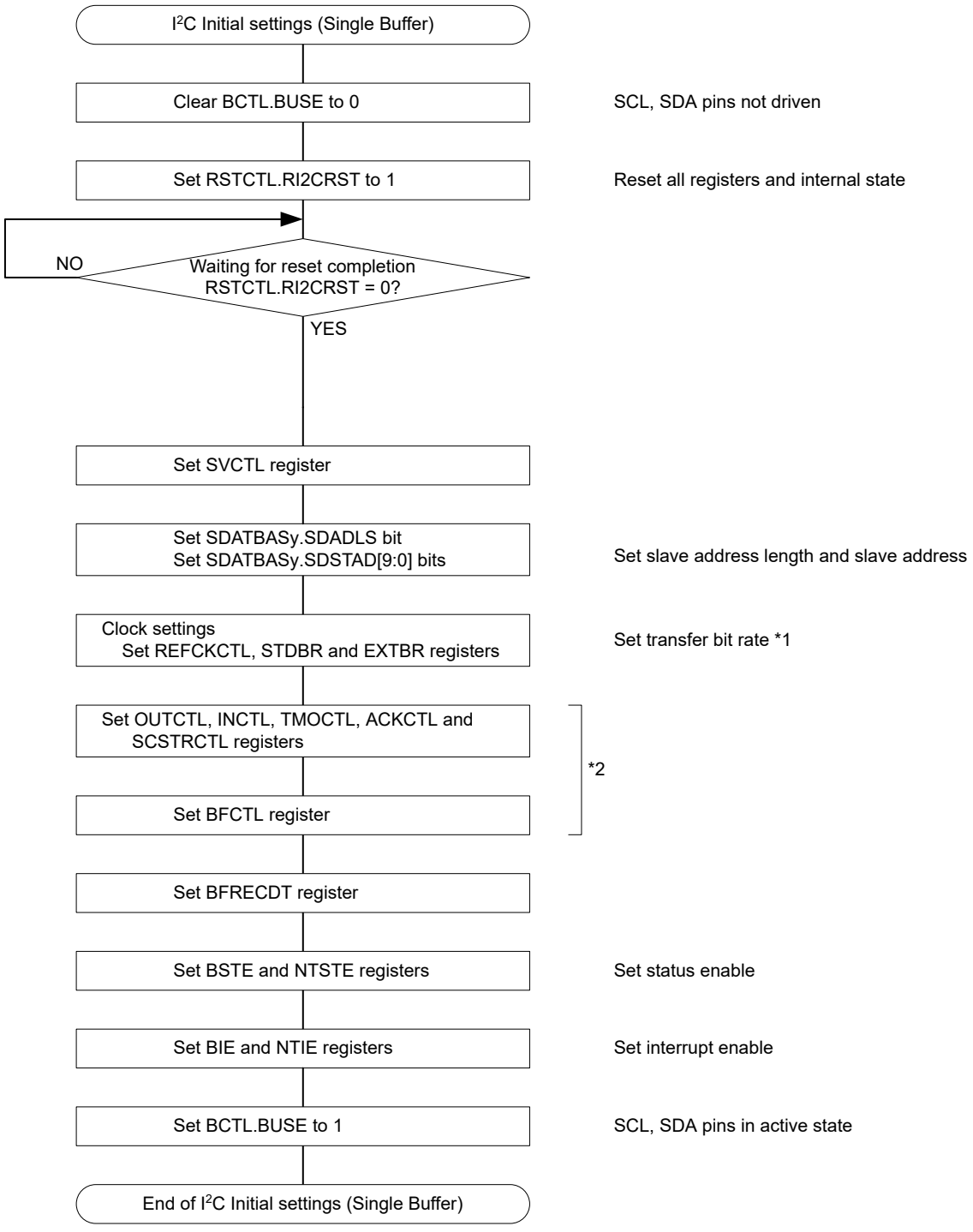
Before starting data transmission and reception, initialize IIC according to the procedure in Figure 27.46.

First, set the BCTL.BUSE bit to 0 (SCLn, SDAn pins not driven).

Next, set the RSTCTL.RI2CRST bit to 1 (IIC reset). This initializes the all registers and internal state. Then, waits for RI2CRST to become 0.

This initializes the various flags and some registers. See Reset Descriptions.

After that, set registers SDATBAS.SDADLS, SDATBAS.SDATAD[9:0], STDBR, INCTL, OUTCTL, TMOCTL, TMOCNT, SCSTRCTL, ACKCTL, and BFCTL, then set the other registers as necessary (for initial settings of IIC, see Figure 27.46).



Note: y = 0 to 2
Note 1. When the IIC is used only in slave mode, set the STDBR register to a value longer than the data setup time.
Note 2. Set these registers as necessary.

Figure 27.46 Block diagram of digital noise filter circuit

27.3.2.2 Master Mode Communication Flow

27.3.2.2.1 I²C Master Transmission Flow (Single Buffer Transfer)

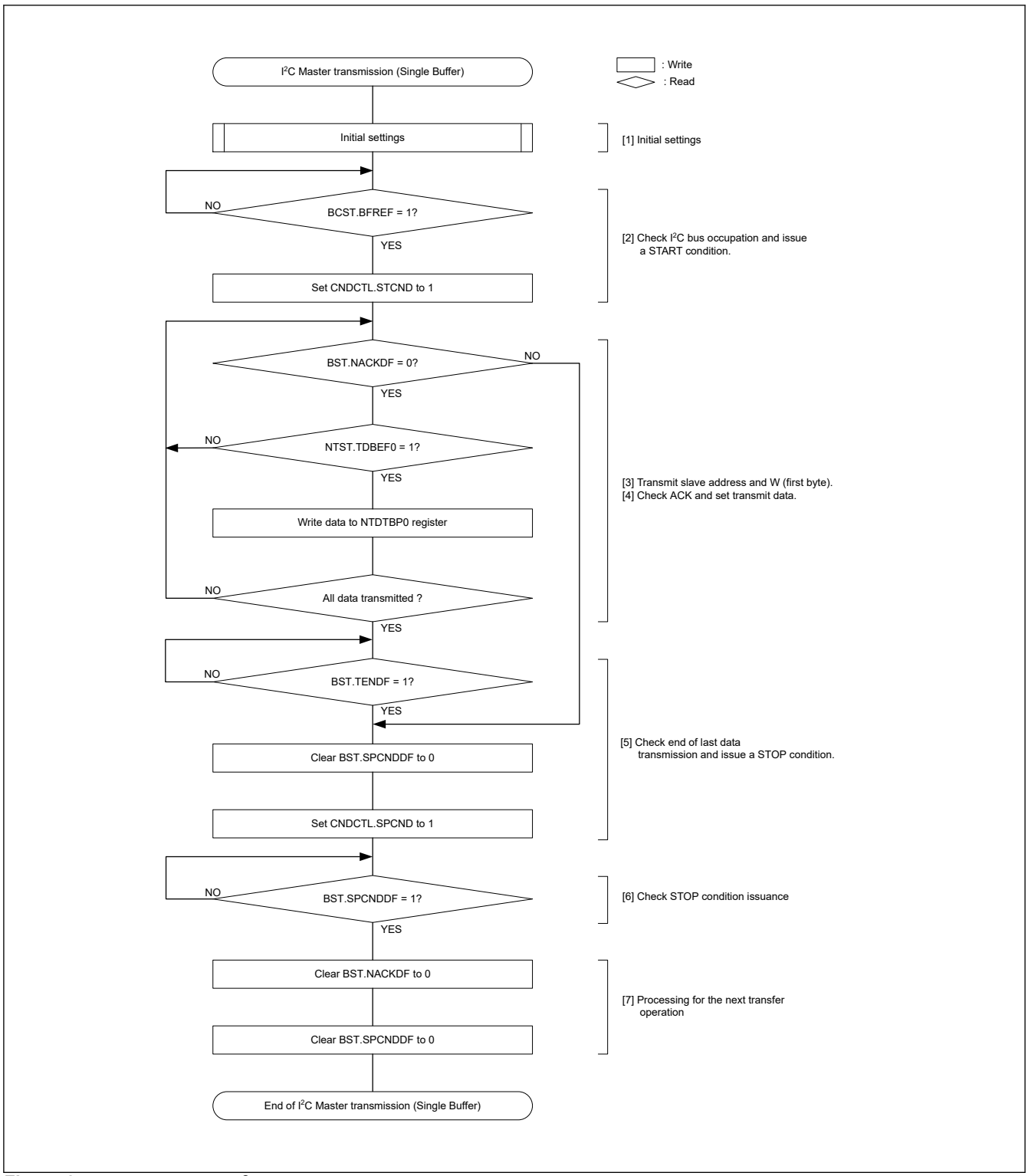


Figure 27.47 Example of I²C master transmission flowchart (single buffer transfer)

27.3.2.2.2 I²C Master Reception Flow (Single Buffer Transfer)

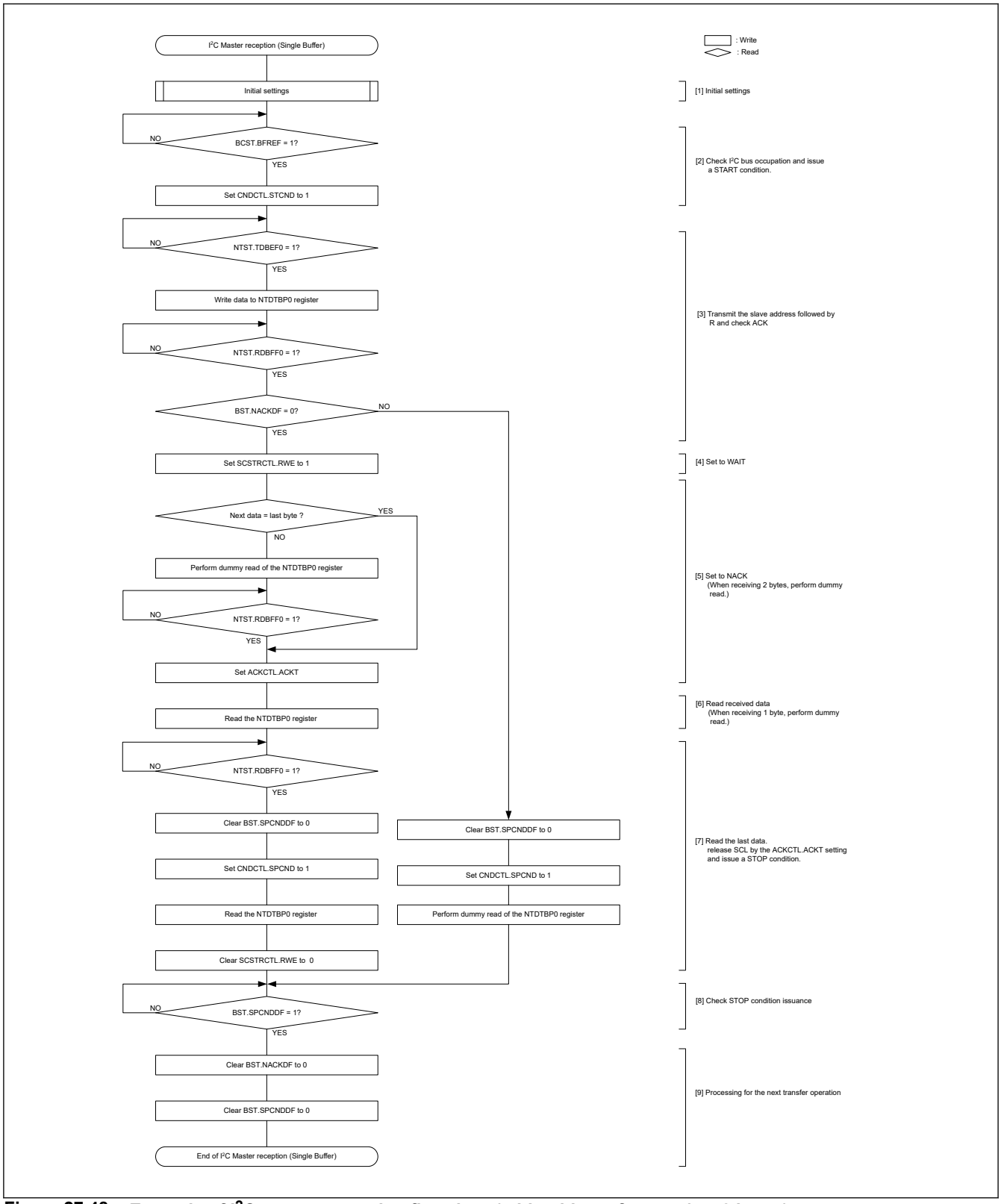


Figure 27.48 Example of I²C master reception flowchart (7-bit address format, 1 or 2 bytes)

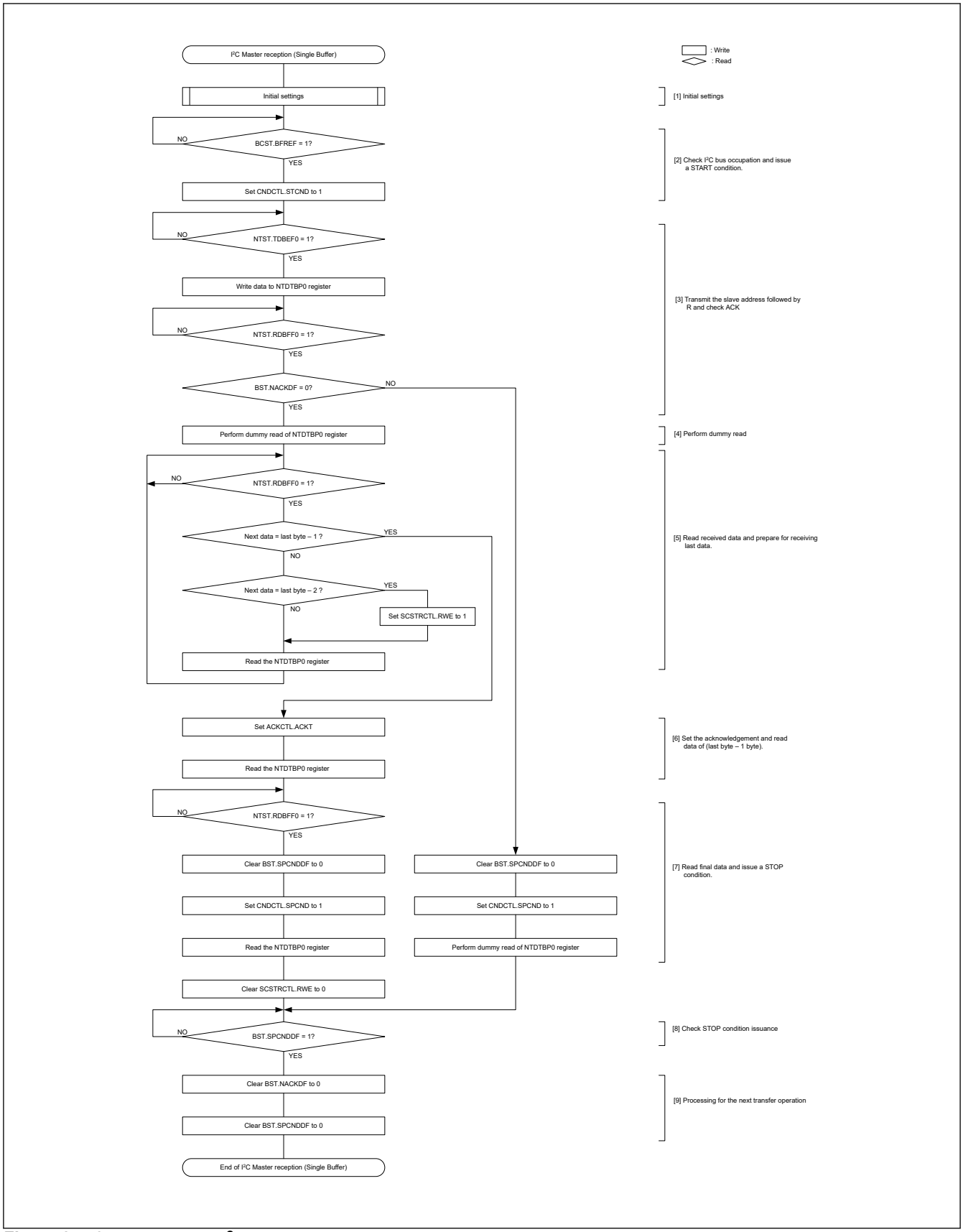


Figure 27.49 Example of I²C master reception flowchart (7-bit address format, 3 bytes or more)

27.3.2.3 Slave Mode Communication Flow

27.3.2.3.1 I²C Slave Transmission Flow (Single Buffer Transfer)

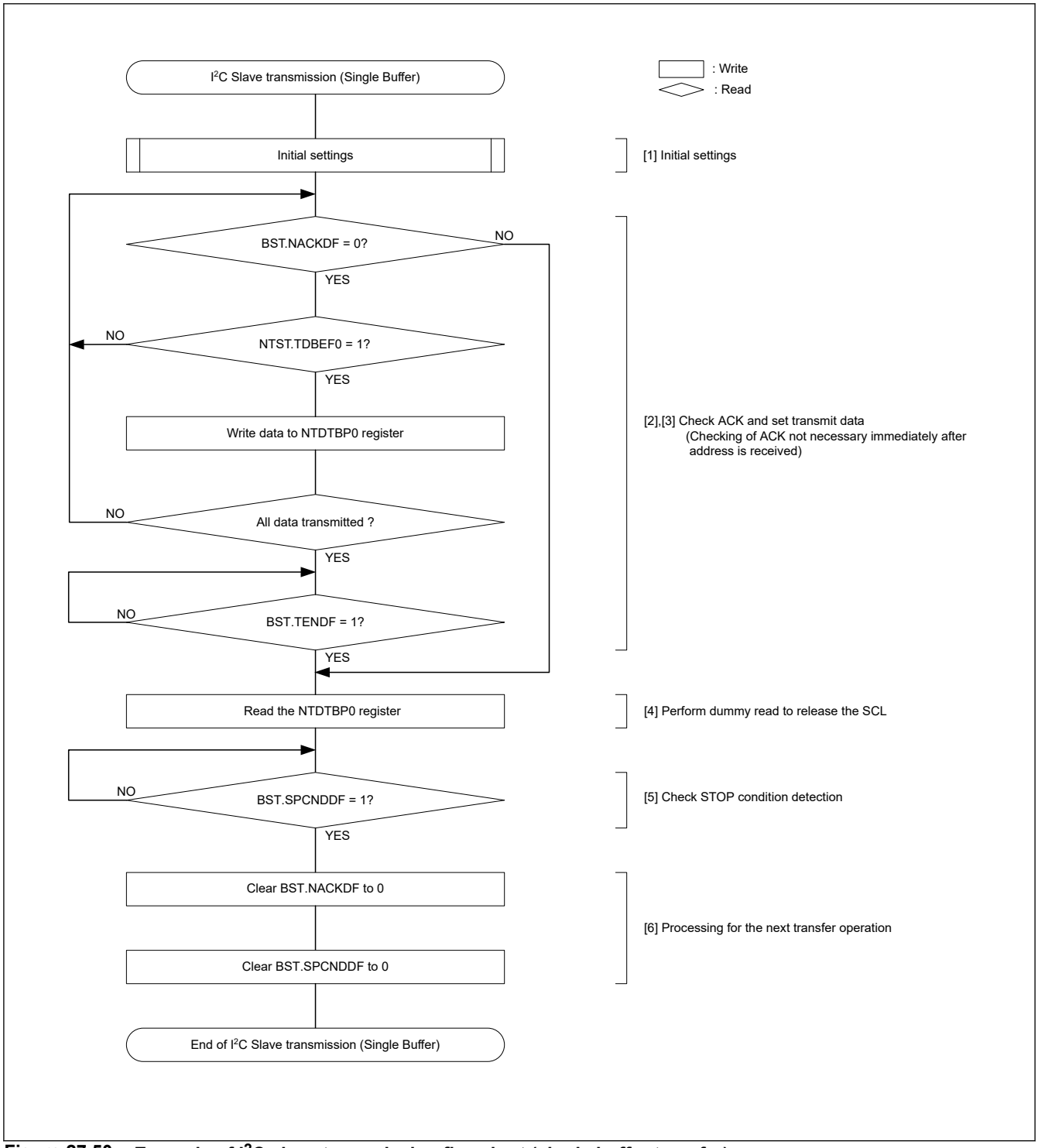


Figure 27.50 Example of I²C slave transmission flowchart (single buffer transfer)

27.3.2.3.2 I²C Slave Reception Flow (Single Buffer Transfer)

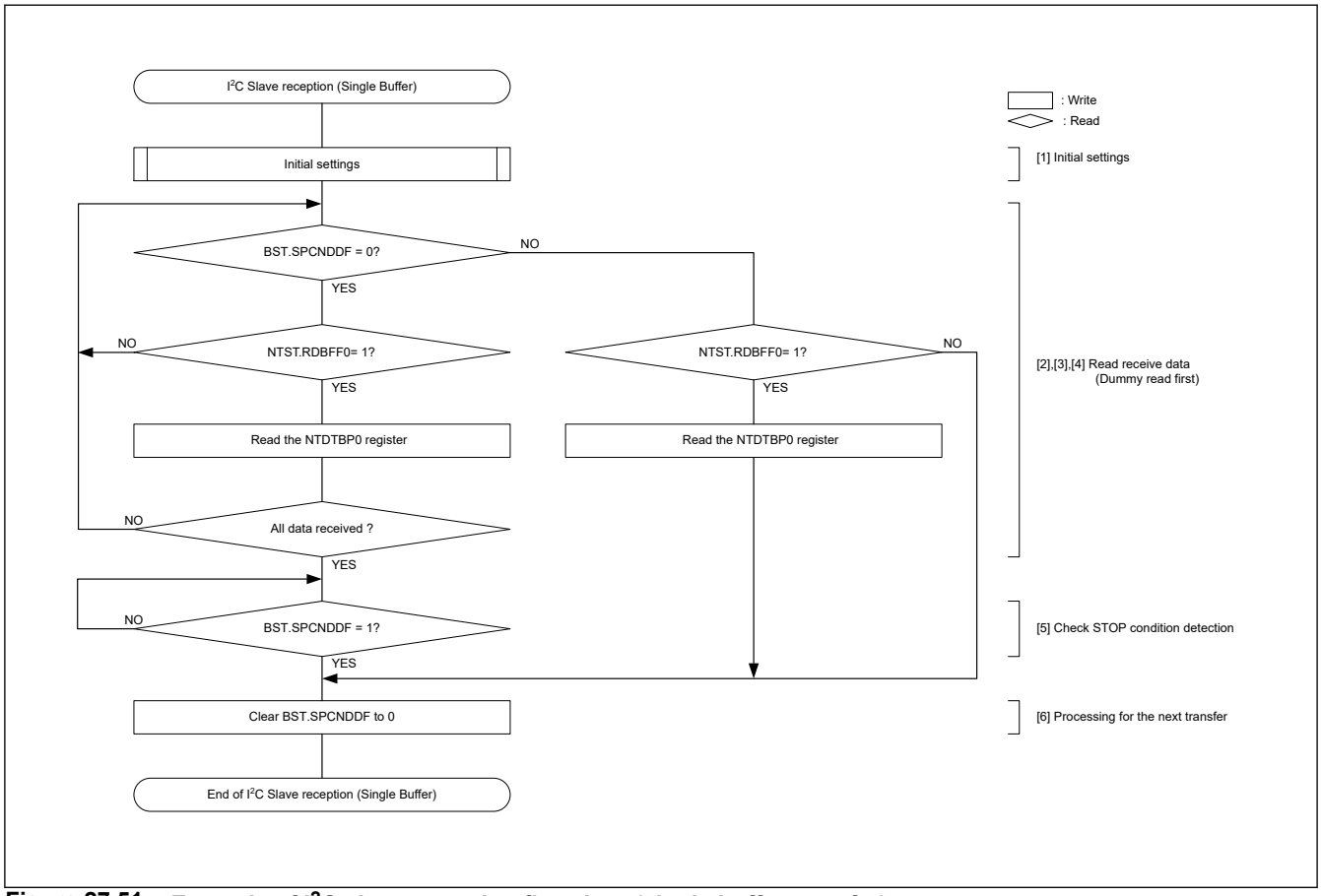


Figure 27.51 Example of I²C slave reception flowchart (single buffer transfer)

27.4 Interrupt Sources

IIC can generate the following interrupt requests:

27.4.1 Overview

The IIC has the interrupt factors shown in Table 27.10.

The interrupt indicated by Possible in the DMAC/DTC Activation column are capable of activating data transfer by the DTC or DMAC.

Table 27.10 Interrupt Generation

Symbol	Interrupt source		Interrupt flag	DTC/DMAC Activation
IICn_RX	Normal receive data buffer full		NTST.RDBEF0	Possible
IICn_TX	Normal transmit data buffer empty		NTST.TDBEF0	Possible
IICn_TEND	Transmit end		BST.TENDF	Impossible
IICn_EEI	Transfer error or event occurrence	Start condition detection interrupt	BST.STCNDDF	Impossible
		STOP condition detection interrupt	BST.SPCNDDF	
		NACK detection interrupt	BST.NACKDF	
		Arbitration lost interrupt	BST.ALF	
		Timeout detection interrupt	BST.TODF	
IIC0_WU	Wake-up condition detection		BST.WUCNDDF	Impossible

Note: n = 0, 1

27.4.2 Buffer Operation for Buffer Full/Empty Interrupts

If the conditions for generating the each buffer full/empty interrupts are satisfied while the corresponding IR flag is 1, the interrupt request is not output for the ICU but retained internally (the capacity for internal retention is one request per source).

An interrupt request that was being retained within the ICU is output when the value of the ICU.IRn.IR flag becomes 0. Internally retained interrupt requests are automatically cleared under normal conditions of usage. Internally retained interrupt requests can also be cleared by writing 0 to the interrupt enable bit within the given peripheral module.

27.5 Event Link Output

IIC handles event output for the event link controller (ELC) corresponding to the following sources.

(1) Communication event

When a Communication event (arbitration-lost detection, detection of NACK, detection of timeout, detection of a START condition, or detection of a STOP condition) occurs, the corresponding event signal can be output for another module via the ELC.

(2) Receive data full

When a receive data register becomes full, the corresponding event signal can be output for another module via the ELC.

(3) Transmit data empty

When a transmit data register becomes empty, the corresponding event signal can be output for another module via the ELC.

(4) Transmit end

On completion of transfer, the corresponding event signal can be output for another module via the ELC.

27.5.1 Interrupt Handling and Event Linking

IIC module produces four kinds of interrupt: communication event (arbitration-lost detection, detection of NACK, detection of timeout, detection of a START condition, or detection of a STOP condition), receive data full, transmit data empty, and transmit end interrupts. Each of these has an enable bit to control enabling and disabling of the interrupt signal. An interrupt request signal is output for the CPU when an interrupt source condition is satisfied while the setting of the corresponding enable bit is enabled.

The corresponding event link output signals are sent to other modules as event signals via the ELC when the interrupt source conditions are satisfied, regardless of the settings of the interrupt enable bits. For details on interrupt sources, see section 27.4.1. Overview.

27.6 Reset Descriptions

Table 27.11 Register states when issuing each condition (1) (1 of 2)

Register symbol	Register bit name	System reset	RSTCTL Register	
			RI2CRST	INTLRST
BCTL	BUSE	In reset	In reset	Saved
RSTCTL	INTLRST	In reset	In reset	Saved
	RI2CRST	In reset	Saved	Saved
PRSST	PRSSTWP	In reset	In reset	In reset
	TRMD	In reset	In reset	In reset
	CRMS	In reset	In reset	In reset

Table 27.11 Register states when issuing each condition (1) (2 of 2)

Register symbol	Register bit name	System reset	RSTCTL Register	
			RI2CRST	INTLRST
BFCTL	HSME	In reset	In reset	Saved
	FMPE	In reset	In reset	Saved
	SMBS	In reset	In reset	Saved
	SCSYNE	In reset	In reset	Saved
	SALE	In reset	In reset	Saved
	NALE	In reset	In reset	Saved
	MALE	In reset	In reset	Saved

Table 27.12 Register states when issuing each condition (2) (1 of 2)

Register symbol	Register bit name	System reset	RSTCTL Register	
			RI2CRST	INTLRST
SVCTL	SVAE2	In reset	In reset	Saved
	SVAE1	In reset	In reset	Saved
	SVAE0	In reset	In reset	Saved
	HOAE	In reset	In reset	Saved
	DVIDE	In reset	In reset	Saved
	HSMCE	In reset	In reset	Saved
	GCAE	In reset	In reset	Saved
REFCKCTL	IREFCKS[2:0]	In reset	In reset	Saved
STDBR	DSBRPO	In reset	In reset	Saved
	SBRHP[5:0]	In reset	In reset	Saved
	SBRLO[7:0]	In reset	In reset	Saved
EXTBR	EBRHO[7:0]	In reset	In reset	Saved
	EBRLO[7:0]	In reset	In reset	Saved
BFRECDT	FRECYC[8:0]	In reset	In reset	Saved
OUTCTL	SDODCS	In reset	In reset	Saved
	SDOD[2:0]	In reset	In reset	Saved
	EXCYC	In reset	In reset	Saved
	SOCWP	In reset	In reset	In reset
	SCOC	In reset	In reset	Saved
	SDOC	In reset	In reset	Saved
INCTL	SDID[1:0]	In reset	In reset	Saved
	DNFE	In reset	In reset	Saved
	DNFS[3:0]	In reset	In reset	Saved
TMOCTL	TOMDS[1:0]	In reset	In reset	Saved
	TOHCTL	In reset	In reset	Saved
	TOLCTL	In reset	In reset	Saved
	TODTS[1:0]	In reset	In reset	Saved

Table 27.12 Register states when issuing each condition (2) (2 of 2)

Register symbol	Register bit name	System reset	RSTCTL Register	
			RI2CRST	INTLRST
WUCTL	WUFE	In reset	In reset	Saved
	WUFSYNE	In reset	In reset	Saved
	WUANFS	In reset	In reset	Saved
	WUACKS	In reset	In reset	Saved
ACKCTL	ACKTWP	In reset	In reset	In reset
	ACKT	In reset	In reset	In reset
	ACKR	In reset	In reset	In reset
SCSTRCTL	RWE	In reset	In reset	Saved
	ACKTWE	In reset	In reset	Saved

Table 27.13 Register states when issuing each condition (3)

Register symbol	Register bit name	System reset	RSTCTL Register	
			RI2CRST	INTLRST
CNDCTL	SPCND	In reset	In reset	In reset
	SRCND	In reset	In reset	In reset
	STCND	In reset	In reset	In reset
NTDTBP0	NTDTBP0[31:0]	In reset	In reset	In reset
BST	WUCNDDF	In reset	In reset	Saved
	TODF	In reset	In reset	In reset
	ALF	In reset	In reset	In reset
	TENDF	In reset	In reset	In reset
	NACKDF	In reset	In reset	In reset
	SPCNDDF	In reset	In reset	In reset
	STCNDDF	In reset	In reset	In reset
BSTE	WUCNDDE	In reset	In reset	Saved
	TODE	In reset	In reset	Saved
	ALE	In reset	In reset	Saved
	TENDE	In reset	In reset	Saved
	NACKDE	In reset	In reset	Saved
	SPCNDDE	In reset	In reset	Saved
	STCNDDE	In reset	In reset	Saved

Table 27.14 Register states when issuing each condition (4)

Register symbol	Register bit name	System reset	RSTCTL Register	
			RI2CRST	INTLRST
BIE	WUCNDDIE	In reset	In reset	Saved
	TODIE	In reset	In reset	Saved
	ALIE	In reset	In reset	Saved
	TENDIE	In reset	In reset	Saved
	NACKDIE	In reset	In reset	Saved
	SPCNDDIE	In reset	In reset	Saved
	STCNDDIE	In reset	In reset	Saved
BSTFC	WUCNDDFC	In reset	In reset	Saved
	TODFC	In reset	In reset	Saved
	ALFC	In reset	In reset	Saved
	TENDFC	In reset	In reset	Saved
	NACKDFC	In reset	In reset	Saved
	SPCNDDFC	In reset	In reset	Saved
	STCNDDFC	In reset	In reset	Saved
NTST	RDBFF0	In reset	In reset	In reset
	TDBEF0	In reset	In reset	In reset
NTSTE	RDBFE0	In reset	In reset	Saved
	TDBEE0	In reset	In reset	Saved
NTIE	RDBFIE0	In reset	In reset	Saved
	TDBEIE0	In reset	In reset	Saved

Table 27.15 Register states when issuing each condition (5)

Register symbol	Register bit name	System reset	RSTCTL Register	
			RI2CRST	INTLRST
NTSTFC	RDBFFC0	In reset	In reset	Saved
	TDBEFC0	In reset	In reset	Saved

Table 27.16 Register states when issuing each condition (6)

Register symbol	Register bit name	System reset	RSTCTL Register	
			RI2CRST	INTLRST
BCST	BFREF	In reset	In reset	Saved
SVST	SVAF2	In reset	In reset	In reset
	SVAF1	In reset	In reset	In reset
	SVAF0	In reset	In reset	In reset
	HOAF	In reset	In reset	In reset
	DVIDF	In reset	In reset	In reset
	HSMCF	In reset	In reset	In reset
	GCAF	In reset	In reset	In reset
WUST	WUASYNF	In reset	In reset	Saved

Table 27.17 Register states when issuing each condition (7)

Register symbol	Register bit name	System reset	RSTCTL Register	
			RI2CRST	INTLRST
SDATBASy (y = 0 to 2)	SDADLS	In reset	In reset	Saved
	SDSTAD[9:0]	In reset	In reset	Saved

Table 27.18 Register states when issuing each condition (8)

Register symbol	Register bit name	System reset	RSTCTL Register	
			RI2CRST	INTLRST
SVDVADy (y = 0 to 2)	SSTADV	In reset	In reset	Saved
	SADLG	In reset	In reset	Saved
	SVAD[9:0]	In reset	In reset	Saved

Table 27.19 Register states when issuing each condition (9)

Register symbol	Register bit name	System reset	RSTCTL Register	
			RI2CRST	INTLRST
BITCNT	BCNT[4:0]	In reset	In reset	In reset
PRSTDBG	SDOLV	In reset	In reset	In reset
	SCOLV	In reset	In reset	In reset
	SDILV	In reset	In reset	Saved
	SCILV	In reset	In reset	Saved

27.7 Usage Notes

27.7.1 Settings for the Operating Clock

The following relation is required between the frequencies of the bus clock (PCLKA) and transfer clock (IICCLK):

$IICCLK/2 \leq PCLKA \leq IICCLK$

28. CAN with Flexible Data-rate (CANFD)

This is the CANFD_B version of the CANFD peripheral module.

CANFD_B is referred to as CANFD in this chapter.

28.1 Overview

The CAN with Flexible Data-rate (CANFD) supports the following functions:

- CAN with Flexible Data rate.*1

Note 1. This feature is not available in the classical CAN function.

The CANFD module has a flexible message buffer and FIFO structure that meet the requirements of various applications. It also provides test modes to achieve high testability of the module that can be useful for power-on testing.

This specification describes of the CANFD module.

The CANFD mode is only available in certain products that support it.

28.1.1 CANFD Module

Table 28.1 CANFD module specifications (1 of 2)

Parameter		Specifications
Communication		CAN functionality conforms to CANFD ISO 11898-1 (2015)
Protocol engine version		RS-CANFD_PE V3.0
Data transfer rate	CANFD*1	Up to 1 Mbps for arbitration phase and up to 5 Mbps for data phase
	Classical CAN	Up to 1 Mbps
Operation frequency Peripheral clock		60 MHz (PCLKB) RAM clock: 120 MHz (PCLKA)
Data Link Layer (DLL) clock		Max ≤ 40 MHz
Input/Output pins		CTX0/CRX0
CAN channels		1 channel
Selectable ID type		11-bit Standard ID
		11-bit Standard ID + 18-bit Extended ID
Selectable frame type		Data frame (RTR = 0) (CAN and CANFD frames)
		Remote frame (RTR = 1) (only CAN frames)
Variable data byte count for data frames		DLC range: 0 to F
Message buffer		Up to 32 reception message buffers
		4 transmit message buffers
		1 transmission queue Automatic message transfer into transmission queues supported
FIFO number		2 reception FIFO buffers 1 COMMON FIFOs individually configurable as: ● Reception FIFO ● Transmission FIFO
Automatic delay interval timer for transmission		The delay timer can be applied to: ● Transmission FIFO

Table 28.1 CANFD module specifications (2 of 2)

Parameter	Specifications
Enhanced reception filtering	Support of 11 bits and 29 bits CAN identifier
	Programmable 29 bits CAN identifier acceptance filter mask for each entry
	Programmable routing capability for each FIFO and reception message buffers (up to 2 routing destinations)
	RTR and IDE masking
	Data Length Code (DLC) filter
	Message buffer payload overload protection
	Updating Acceptance Filter List (AFL) entry during communication
General software support	Automatic label information added to receive message (for upper software layer support)
Timer	TX and RX Time Stamp function
Power down function	Module start stop function for CAN node (Channel and Global Sleep mode)
RAM	RAM ECC protected (2 bits error detection, 1-bit error correction)
TrustZone Filter	One security attribution can be set

Note 1. The CANFD mode is available only for CANFD supported product.

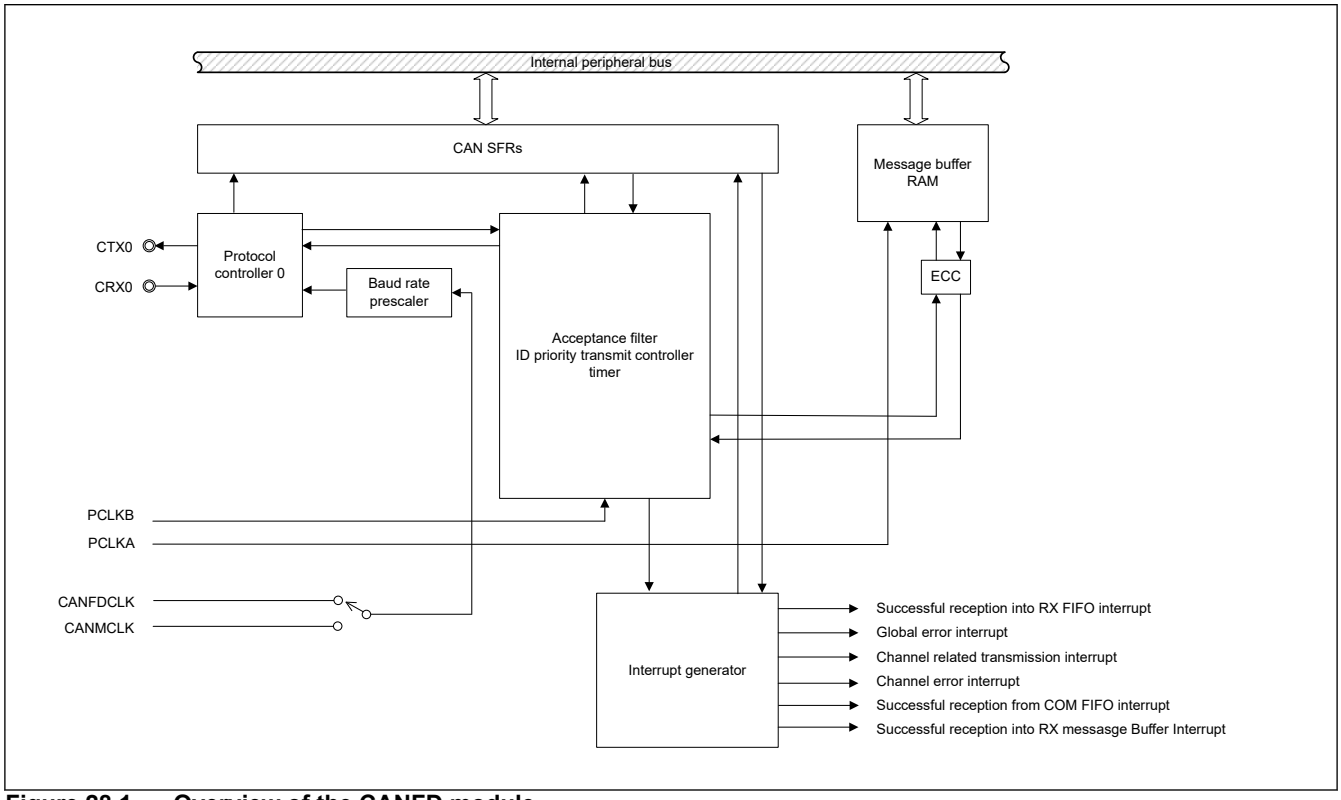


Figure 28.1 Overview of the CANFD module

- CTX0/CRX0:
  Input/Output pins of the CANFD module
- Protocol controller:
  Handles CAN protocol processing such as bus arbitration, bit timing at transmission and reception, stuffing, error handling
- Message buffer RAM:

This RAM is used to store messages after reception or for transmission using a normal message buffer or a FIFO. Each message entry has an individual ID, data length code, data field, message pointer for upper layer application usage and a time stamp.
This RAM is used to store the message acceptance filtering entries. Each acceptance filter entry has an individual ID, data length code, data field, message pointer for upper layer application usage and message direction pointer.

- Acceptance filter:
  Performs filtering of received messages. The entries in the Acceptance filter list RAM are used for the filtering process.
- Two timers:
  - Reception Timestamp function
  - Transmission separation time for FIFO buffers
- Interrupt generator:
  Generates several types of global and channel interrupts
- CAN Special Function Registers (SFRs):
  Registers associated with CAN. See section 28.2. Register Descriptions.

28.1.2 Clock restriction

For the CAN communication the following restriction for the clocks should be satisfied:

- PCLKA / 2 = PCLKB ≥ CANFDCLK
- PCLKA / 2 = PCLKB ≥ CANMCLK

To avoid missing events the CAN engine clock (CANFDCLK or CANMCLK) frequency must be less than the PCLKB clock frequency.

To avoid loss of CAN message, the PCLKB should be set to a clock with a frequacy depend on the CAN communication Baud Rate. The constraint of a baud rate and a PCLKB clock is shown in Table 28.2.

Table 28.2 Clock restriction

	Baud rate	PCLKB
CANFD	1Mbps Nominal 5Mbps Data	PCLKB ≥ 40MHz
	500Kbps Nominall 5Mbps Data	PCLKB ≥ 32MHz
Classical CAN	1Mbps Data	PCLKB ≥ 32MHz

The frequency of CANFD and CANMCLK depend on the required baud rate. For information how to configure the baud rate, refer to section 28.4.1.3. Baud Rate.

28.2 Register Descriptions

28.2.1 Register Table

The reset value shown for the RAM area, consisting of CFDGAFLIDr, CFDGAFLMr, CFDGAFLP0r, CFDGAFLP1r, CFDRMBCPb, CFDRFMBCPb, CFDCFMBCP0, CFDTMBCPb, CFDTHLACC0, CFDTHLACC1 and CFDRPGACCk is valid after initialization of a hardware reset. See section 28.4.2. CAN Module Configuration after Hardware Reset for details of the initialization process.

If a write access with a size of 8 or 16 bits is performed for the RAM area, then the CANFD module does a read-modify write-access to the RAM location, because the RAM requires a 32-bit access through the ECC module.

For single bit error, the correct data is written back. For multiple bit errors, unknown data is written back.

Do not access the space where the register is not assigned.

The read data from the space where the register is not assigned is unknown.

28.2.2 Legend

For all repetitive registers and bits, a lowercase index is used to indicate which slice is being referenced. If an index is being used, it is defined and described in the Register table it is being used in.

There is one global index used across all the registers and bits that need it.

Table 28.3 CANFD Registers (1 of 3)

Register name	Symbol	Value after Reset	Offset Address	Access size
Channel 0 Nominal Bitrate Configuration Register	CFDC0NCFG	0x00000000	0x0000	8, 16, 32
Channel 0 Control Register	CFDC0CTR	0x00000005	0x0004	8, 16, 32
Channel 0 Status Register	CFDC0STS	0x00000005	0x0008	8, 16, 32
Channel 0 Error Flag Register	CFDC0ERFL	0x00000000	0x000C	8, 16, 32
Global Configuration Register	CFDGCFG	0x00000000	0x0014	8, 16, 32
Global Control Register	CFDGCTR	0x00000005	0x0018	8, 16, 32
Global Status Register	CFDGSTS	0x0000000D	0x001C	8, 16, 32
Global Error Flag Register	CFDGERFL	0x00000000	0x0020	8, 16, 32
Global Timestamp Counter Register	CFDGTSC	0x00000000	0x0024	16, 32
Global Acceptance Filter List Entry Control Register	CFDGAFLECTR	0x00000000	0x0028	8, 16, 32
Global Acceptance Filter List Configuration Register	CFDGAFLCFG	0x00000000	0x002C	8, 16, 32
RX Message Buffer Number Register	CFDRMNB	0x00000000	0x0030	8, 16, 32
RX Message Buffer New Data Register	CFDRMND	0x00000000	0x0034	8, 16, 32
RX Message Buffer Interrupt Enable Configuration Register	CFDRMIEC	0x00000000	0x0038	8, 16, 32
RX FIFO Configuration / Control Registers a = [0:1]	CFDRFCCa	0x00000000	0x003C + a × 0x0004	8, 16, 32
RX FIFO Status Registers a = [0:1]	CFDRFSTSa	0x00000001	0x0044 + a × 0x0004	8, 16, 32
RX FIFO Pointer Control Registers a = [0:1]	CFDRFPCTRa	0x00000000	0x004C + a × 0x0004	8, 16, 32
Common FIFO Configuration / Control Register	CFDCFCC	0x00000000	0x0054	8, 16, 32
Common FIFO Status Register	CFDCFSTS	0x00000001	0x0058	8, 16, 32
Common FIFO Pointer Control Register	CFDCFPCTR	0x00000000	0x005C	8, 16, 32
FIFO Empty Status Register	CFDFESTS	0x00000103	0x0060	8, 16, 32
FIFO Full Status Register	CFDFFSTS	0x00000000	0x0064	8, 16, 32
FIFO Message Lost Status Register	CFDFMSTS	0x00000000	0x0068	8, 16, 32
RX FIFO Interrupt Flag Status Register	CFDRFISTS	0x00000000	0x006C	8, 16, 32
TX Message Buffer Control Registers i = [0:3]	CFDTMCi	0x00	0x0070 + i × 0x0001	8
TX Message Buffer Status Registers j = [0:3]	CFDTMSTSj	0x00	0x0074 + j × 0x0001	8
TX Message Buffer Transmission Request Status Register	CFDTMTRSTS	0x00000000	0x0078	8, 16, 32
TX Message Buffer Transmission Abort Request Status Register	CFDTMTARSTS	0x00000000	0x007C	8, 16, 32
TX Message Buffer Transmission Completion Status Register	CFDTMTCSTS	0x00000000	0x0080	8, 16, 32

Table 28.3 CANFD Registers (2 of 3)

Register name	Symbol	Value after Reset	Offset Address	Access size
TX Message Buffer Transmission Abort Status Register	CFDTMTASTS	0x00000000	0x0084	8, 16, 32
TX Message Buffer Interrupt Enable Configuration Register	CFDTMIEC	0x00000000	0x0088	8, 16, 32
TX Queue Configuration / Control Register	CFDTXQCC	0x00000000	0x008C	8, 16, 32
TX Queue Status Register	CFDTXQSTS	0x00000001	0x0090	8, 16, 32
TX Queue Pointer Control Register	CFDTXQPCTR	0x00000000	0x0094	8, 16, 32
TX History List Configuration / Control Register	CFDTHLCC	0x00000000	0x0098	8, 16, 32
TX History List Status Register	CFDTHLSTS	0x00000001	0x009C	8, 16, 32
TX History List Pointer Control Register	CFDTHLPCTR	0x00000000	0x00A0	8, 16, 32
Global TX Interrupt Status Register	CFDGTINTSTS	0x00000000	0x00A4	8, 16, 32
Global Test Configuration Register	CFDGTSTCFG	0x00000000	0x00A8	8, 16, 32
Global Test Control Register	CFDGTSTCTR	0x00000000	0x00AC	8, 16, 32
Global FD Configuration register	CFDGFDCFG	0x00000000	0x00B0	8, 16, 32
Global Lock Key Register	CFDGLOCKK	0x00000000	0x00B8	16, 32
Global AFL Ignore Entry Register	CFDGAFLIGNENT	0x00000000	0x00C0	8, 16, 32
Global AFL Ignore Control Register	CFDGAFLIGNCTR	0x00000000	0x00C4	16, 32
DMA Transfer Control Register	CFDCDTCT	0x00000000	0x00C8	8, 16, 32
DMA Transfer Status Register	CFDCDTSTS	0x00000000	0x00CC	8, 16, 32
Global SW reset Register	CFDGRSTC	0x00000000	0x00D8	16, 32
Channel 0 Data Bitrate Configuration Register	CFDC0DCFG	0x00000000	0x0100	8, 16, 32
Channel 0 CANFD Configuration Register	CFDC0FDCFG	0x00000000	0x0104	8, 16, 32
Channel 0 CANFD Control Register	CFDC0FDCTR	0x00000000	0x0108	8, 16, 32
Channel 0 CANFD Status Register	CFDC0FDSTS	0x00000000	0x010C	8, 16, 32
Channel 0 CANFD CRC Register	CFDC0FDCRC	0x00000000	0x0110	8, 16, 32
Global Acceptance Filter List ID Registers r = [1…16]	CFDGAFLIDr	0x00000000*1	0x0120 + (r-1) × 0x0010	8, 16, 32
Global Acceptance Filter List Mask Registers r = [1…16]	CFDGAFLMr	0x00000000*1	0x0124 + (r-1) × 0x0010	8, 16, 32
Global Acceptance Filter List Pointer 0 Registers r = [1…16]	CFDGAFLP0r	0x00000000*1	0x0128 + (r-1) × 0x0010	8, 16, 32
Global Acceptance Filter List Pointer 1 Registers r = [1…16]	CFDGAFLP1r	0x00000000*1	0x012C + (r-1) × 0x0010	8, 16, 32
RAM Test Page Access Registers k = [0…63]	CFDRPGACCk	0x00000000*1	0x0280 + k × 0x0004	8, 16, 32
RX FIFO Access ID Registers b = [0…1]	CFDRFIDb	0x00000000*1	0x0520 + b × 0x004C	8, 16, 32
RX FIFO Access Pointer Registers b = [0…1]	CFDRFPTRb	0x00000000*1	0x0524 + b × 0x004C	8, 16, 32
RX FIFO Access CANFD Status Registers b = [0…1]	CFDRFFDSTSb	0x00000000*1	0x0528 + b × 0x004C	8, 16, 32
RX FIFO Access Data Field p Registers b = [0…1] p = [0…15]	CFDRFDFbp	0x00000000*1	0x052C + p × 0x0004 + b × 0x004C	8, 16, 32
Common FIFO Access ID Register	CFDCFID	0x00000000*1	0x05B8	8, 16, 32

Table 28.3 CANFD Registers (3 of 3)

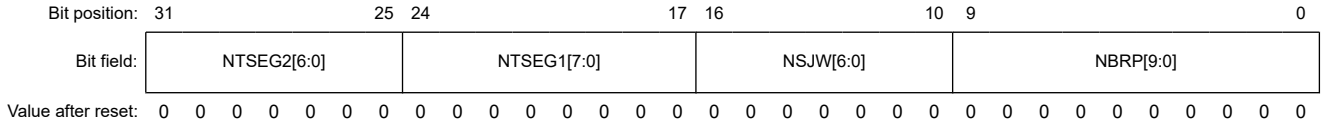
Register name	Symbol	Value after Reset	Offset Address	Access size
Common FIFO Access Pointer Register	CFDCFPTR	0x00000000*1	0x05BC	8, 16, 32
Common FIFO Access CANFD Control/ Status Register	CFDCFFDCSTS	0x00000000*1	0x05C0	8, 16, 32
Common FIFO Access Data Field p Registers p = [0…15]	CFDCFDFp	0x00000000*1	0x05C4 + p × 0x0004	8, 16, 32
TX Message Buffer ID Registers b = [0…3]	CFDTMIDb	0x00000000*1	0x0604 + b × 0x004C	8, 16, 32
TX Message Buffer Pointer Registers b = [0…3]	CFDTMPTRb	0x00000000*1	0x0608 + b × 0x004C	8, 16, 32
TX Message Buffer CANFD Control Registers b = [0…3]	CFDTMFDCTRb	0x00000000*1	0x060C + b × 0x004C	8, 16, 32
TX Message Buffer Data Field p Registers b = [0…3] p = [0…15]	CFDTMDFbp	0x00000000*1	0x0610 + p × 0x0004 + b × 0x004C	8, 16, 32
Channel 0 TX History List Access Registers 0	CFDTHLACC0	0x00000000*1	0x0740	8, 16, 32
Channel 0 TX History List Access Registers 1	CFDTHLACC1	0x00000000*1	0x0744	8, 16, 32
RX Message Buffer ID Registers b = [0…7]	CFDRMIDb	0x00000000*1	0x0920 + b × 0x004C	8, 16, 32
RX Message Buffer Pointer Registers b = [0…7]	CFDRMPTRb	0x00000000*1	0x0924 + b × 0x004C	8, 16, 32
RX Message Buffer CANFD Status Registers b = [0…7]	CFDRMFDSTSb	0x00000000*1	0x0928 + b × 0x004C	8, 16, 32
RX Message Buffer Data Field p Registers b = [0…7] p = [0…15]	CFDRMDFbp	0x00000000*1	0x092C + p × 0x0004 + b × 0x004C	8, 16, 32
RX Message Buffer ID Registers b = [8…15]	CFDRMIDb	0x00000000*1	0x0D20 + (b - 8) × 0x004C	8, 16, 32
RX Message Buffer Pointer Registers b = [8…15]	CFDRMPTRb	0x00000000*1	0x0D24 + (b - 8) × 0x004C	8, 16, 32
RX Message Buffer CANFD Status Registers b = [8…15]	CFDRMFDSTSb	0x00000000*1	0x0D28 + (b - 8) × 0x004C	8, 16, 32
RX Message Buffer Data Field p Registers b = [8…15] p = [0…15]	CFDRMDFbp	0x00000000*1	0x0D2C + p × 0x0004 + (b - 8) × 0x004C	8, 16, 32
RX Message Buffer ID Registers b = [16…23]	CFDRMIDb	0x00000000*1	0x1120 + (b - 16) × 0x004C	8, 16, 32
RX Message Buffer Pointer Registers b = [16…23]	CFDRMPTRb	0x00000000*1	0x1124 + (b - 16) × 0x004C	8, 16, 32
RX Message Buffer CANFD Status Registers b = [16…23]	CFDRMFDSTSb	0x00000000*1	0x1128 + (b - 16) × 0x004C	8, 16, 32
RX Message Buffer Data Field p Registers b = [16…23] p = [0…15]	CFDRMDFbp	0x00000000*1	0x112C + p × 0004 + (b - 16) × 0x004C	8, 16, 32
RX Message Buffer ID Registers b = [24…31]	CFDRMIDb	0x00000000*1	0x1520 + (b - 24) × 0x004C	8, 16, 32
RX Message Buffer Pointer Registers b = [24…31]	CFDRMPTRb	0x00000000*1	0x1524 + (b - 24) × 0x004C	8, 16, 32
RX Message Buffer CANFD Status Registers b = [24…31]	CFDRMFDSTSb	0x00000000*1	0x1528 + (b - 24) × 0x004C	8, 16, 32
RX Message Buffer Data Field p Registers b = [24…31] p = [0…15]	CFDRMDFbp	0x00000000*1	0x152C + p × 0x0004 + (b - 24) × 0x004C	8, 16, 32

Note 1. The RAM area is initialized after a hardware reset, see section 28.4.2. CAN Module Configuration after Hardware Reset.

28.2.3 CFDC0NCFG : Channel 0 Nominal Bitrate Configuration Register

Base address: CANFD_B = 0x400B_0000

Offset address: 0x0000



Bit	Symbol	Function	R/W
9:0	NBRP[9:0]	Channel Nominal Baud Rate Prescaler Nominal baud rate prescaler division ratio	R/W
16:10	NSJW[6:0]	Resynchronization Jump Width 0x00: 1 Tq 0x01: 2 Tq ⋮ 0x7E: 127 Tq 0x7F: 128 Tq	R/W
24:17	NTSEG1[7:0]	Timing Segment 1 0x00: Reserved 0x01: 2 Tq 0x02: 3 Tq 0x03: 4 Tq ⋮ 0xFE: 255 Tq 0xFF: 256 Tq	R/W
31:25	NTSEG2[6:0]	Timing Segment 2 0x00: Reserved 0x01: 2 Tq ⋮ 0x7E: 127 Tq 0x7F: 128 Tq	R/W

Note: Tq means time quantum.

This register configures the transmission/reception nominal baud rate parameters of the channels.

NBRP[9:0] bits (Channel Nominal Baud Rate Prescaler)

The NBRP[9:0] bits are used to define the peripheral bus clock periods contained in a time quantum.

Do not write to these bits in CH_OPERATION or CH_SLEEP mode.

Only write to these bits when the CANFD channel is in CH_RESET or CH_HALT mode.

NSJW[6:0] bits (Resynchronization Jump Width)

The NSJW[6:0] bits set the synchronization jump width. A value from 1 to 128 time quanta can be set.

Do not write to these bits in CH_OPERATION or CH_SLEEP mode.

Only write to these bits when the CANFD channel is in CH_RESET or CH_HALT mode.

NTSEG1[7:0] bits (Timing Segment 1)

The NTSEG1[7:0] bits set the segment TSEG1 to compensate for edges on the CAN bus with a positive phase error. These bits contain the propagation segment.

Do not write to these bits in CH_OPERATION or CH_SLEEP mode.

Only write to these bits when the CANFD channel is in CH_RESET or CH_HALT mode.

Additionally, configure a Tq value only between 2 and 256, inclusive. See section 28.4.1.2. CAN Bit Timing for more details.

NTSEG2[6:0] bits (Timing Segment 2)

The NTSEG2[6:0] bits set the segment TSEG2 to compensate for edges on the CAN bus with a negative phase error.

Do not write to these bits in CH_OPERATION or CH_SLEEP mode.

Only write to these bits when the CANFD channel is in CH_RESET or CH_HALT mode.

Additionally, configure a Tq value only between 2 and 128, inclusive.

28.2.4 CFDC0CTR : Channel 0 Control Register

Base address: CANFD_B = 0x400B_0000

Offset address: 0x0004

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	ROM	BFT	—	—	—	CTMS[1:0]		CTME	ERRD	BOM[1:0]		—	TDCVFIE	SOCOIE	EOCOIE	TAIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ALIE	BLIE	OLIE	BORIE	BOEIE	EPIE	EWIE	BEIE	—	—	—	—	RTBO	CSLPR	CHMDC[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Bit	Symbol	Function	R/W
1:0	CHMDC[1:0]	Channel Mode Control 0 0: Channel operation mode request 0 1: Channel reset request 1 0: Channel halt request 1 1: Keep current value	R/W
2	CSLPR	Channel Sleep Request 0: Channel sleep request disabled 1: Channel sleep request enabled	R/W
3	RTBO	Return from Bus-Off 0: Channel is not forced to return from bus-off 1: Channel is forced to return from bus-off	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
8	BEIE	Bus Error Interrupt Enable 0: Bus error interrupt disabled 1: Bus error interrupt enabled	R/W
9	EWIE	Error Warning Interrupt Enable 0: Error warning interrupt disabled 1: Error warning interrupt enabled	R/W
10	EPIE	Error Passive Interrupt Enable 0: Error passive interrupt disabled 1: Error passive interrupt enabled	R/W
11	BOEIE	Bus-Off Entry Interrupt Enable 0: Bus-off entry interrupt disabled 1: Bus-off entry interrupt enabled	R/W
12	BORIE	Bus-Off Recovery Interrupt Enable 0: Bus-off recovery interrupt disabled 1: Bus-off recovery interrupt enabled	R/W
13	OLIE	Overload Interrupt Enable 0: Overload interrupt disabled 1: Overload interrupt enabled	R/W
14	BLIE	Bus Lock Interrupt Enable 0: Bus lock interrupt disabled 1: Bus lock interrupt enabled	R/W

Bit	Symbol	Function	R/W
15	ALIE	Arbitration Lost Interrupt Enable 0: Arbitration lost interrupt disabled 1: Arbitration lost interrupt enabled	R/W
16	TAIE	Transmission Abort Interrupt Enable 0: TX abort interrupt disabled 1: TX abort interrupt enabled	R/W
17	EOCOIE	Error Occurrence Counter Overflow Interrupt Enable 0: Error occurrence counter overflow interrupt disabled 1: Error occurrence counter overflow interrupt enabled	R/W
18	SOCOIE	Successful Occurrence Counter Overflow Interrupt Enable 0: Successful occurrence counter overflow interrupt disabled 1: Successful occurrence counter overflow interrupt enabled	R/W
19	TDCVFIE*1	Transceiver Delay Compensation Violation Interrupt Enable 0: Transceiver delay compensation violation interrupt disabled 1: Transceiver delay compensation violation interrupt enabled	R/W
20	—	This bit is read as 0. The write value should be 0.	R/W
22:21	BOM[1:0]	Channel Bus-Off Mode 0 0: Normal mode (comply with ISO 11898-1) 0 1: Entry to Halt mode automatically at bus-off start 1 0: Entry to Halt mode automatically at bus-off end 1 1: Entry to Halt mode (during bus-off recovery period) by software	R/W
23	ERRD	Channel Error Display 0: Only the first set of error codes displayed 1: Accumulated error codes displayed	R/W
24	CTME	Channel Test Mode Enable 0: Channel test mode disabled 1: Channel test mode enabled	R/W
26:25	CTMS[1:0]	Channel Test Mode Select 0 0: Basic test mode 0 1: Listen-only mode 1 0: Self-test mode 0 (External loopback mode) 1 1: Self-test mode 1 (Internal loopback mode)	R/W
29:27	—	These bits are read as 0. The write value should be 0.	R/W
30	BFT	Bit Flip Test 0: First data bit of reception stream not inverted 1: First data bit of reception stream inverted	R/W
31	ROM*1	Restricted Operation Mode 0: Restricted operation mode disabled 1: Restricted operation mode enabled	R/W

Note 1. These bits are not available in the classical CAN function.

Channel Control register controls the modes of the related channel. It is used to enable generation of interrupts if errors are detected on the CAN bus connected to this channel. It is also used to configure the channel in test mode.

CHMDC[1:0] bits (Channel Mode Control)

The CHMDC[1:0] bits can be used to configure modes of the CAN channel.

CAN mode transitions are described in more details in section 28.3.3. Channel Modes.

Setting CHMDC[1:0] bits to 11b has no effect. When the CANFD module is in GL_HALT mode, these bits can only be set to 10b or 01b. These bits cannot be set in CH_SLEEP mode.

These bits can change automatically when transitioning to Halt mode by the CFDC0CTR.BOM settings.

If CPU write access to CFDC0CTR.CHMDC occurs at the same time when the CAN channel enters Halt mode (at the start of bus-off when CFDC0CTR.BOM = 01b, or at the end of bus-off when CFDC0CTR.BOM = 10b), then the CPU write access has the highest priority.

The CAN channel changes the value of CFDC0CTR.CHMDC within the Channel Control Registers for the specified cases only if the CFDC0CTR.CHMDC value is 00b (Operation mode).

CSLPR bit (Channel Sleep Request)

When the CSLPR bit is 1, a Sleep mode request is generated for the corresponding CAN channel

When this bit is 0, a request to exit Sleep mode is generated for the related CANFD channel.

Only write to this bit when the related CANFD channel is in CH_RESET or CH_SLEEP mode.

RTBO bit (Return from Bus-Off)

When the protocol controller of the CAN channel enters bus-off state, you can force a recovery from bus-off state by setting the RTBO bit in the Channel Control Register to 1.

The error state changes from bus-off state to integrating with a maximum delay of 1 CAN bit time.

When the RTBO bit is set to 1, the REC and TEC registers are initialized and the Bus-Off Status bit (Channel Bus-off Status, CFDC0STS.BOSTS) is set to 0.

Registers other than the REC and TEC registers are not initialized by this command. Even if CFDC0CTR.BORIE is set, a bus-off recovery interrupt is not generated by this recovery from the bus-off state.

The RTBO bit cannot be set in CH_SLEEP mode. Setting this bit in any state other than bus-off state has no effect and the bit is cleared immediately. The read value is always 0.

Return from the Bus-Off command should be used only when CFDC0CTR.BOM is set to 00b.

Only write to this bit when the related CANFD channel is in CH_OPERATION mode. This bit is automatically cleared when set by software.

BEIE bit (Bus Error Interrupt Enable)

When the BEIE and the CFDC0ERFL.BEF bits are both 1, an error interrupt request is generated.

This bit cannot be set in CH_SLEEP mode. Only write to this bit when the related CANFD channel is in CH_RESET mode.

EWIE bit (Error Warning Interrupt Enable)

When the EWIE and the CFDC0ERFL.EWF bits are both 1, an error interrupt request is generated.

The EWIE bit cannot be set in CH_SLEEP mode. Only write to this bit when the related CANFD channel is in CH_RESET mode.

EPIE bit (Error Passive Interrupt Enable)

An error interrupt request is generated when the EPIE bit and the CFDC0ERFL.EPF are both 1.

The EPIE bit cannot be set in CH_SLEEP mode. Only write to this bit when the related CANFD channel is in CH_RESET mode.

BOEIE bit (Bus-Off Entry Interrupt Enable)

When the BOEIE and the CFDC0ERFL.BOEF bits are both 1, an error interrupt request is generated.

The BOEIE bit cannot be set in CH_SLEEP mode. Only write to this bit when the related CANFD channel is in CH_RESET mode.

BORIE bit (Bus-Off Recovery Interrupt Enable)

When the BORIE and the CFDC0ERFL.BORF bits are both 1, an error interrupt request is generated.

The BORIE bit cannot be set in CH_SLEEP mode. Only write to this bit when the related CANFD channel is in CH_RESET mode.

OLIE bit (Overload Interrupt Enable)

When the OLIE and the CFDC0ERFL.OVLF bits are both 1, an error interrupt request is generated.

Do not write to this bit in CH_SLEEP mode. Only write to this bit when the related CANFD channel is in CH_RESET mode.

BLIE bit (Bus Lock Interrupt Enable)

When the BLIE and the CFDC0ERFL.BLF bits are both 1, an error interrupt request is generated.

Do not write to this bit in CH_SLEEP mode. Only write to this bit when the related CANFD channel is in CH_RESET mode.

ALIE bit (Arbitration Lost Interrupt Enable)

When the ALIE and the CFDC0ERFL.ALF bits are both 1, an error interrupt request is generated.

Do not write to this bit in CH_SLEEP mode. Only write to this bit when the related CANFD channel is in CH_RESET mode.

TAIE bit (Transmission Abort Interrupt Enable)

When the TAIE bit is 1 and a transmission is successfully aborted from a TX MB belonging to the corresponding CAN channel, an interrupt request is generated.

Do not write to this bit in CH_SLEEP mode. Only write to this bit when the related CANFD channel is in CH_RESET mode.

EOCOIE bit (Error Occurrence Counter Overflow Interrupt Enable)

When the EOCOIE bit is 1 and the CFDC0FDSTS.EOCO bit belonging to the corresponding CAN channel is 1, an error interrupt request is generated.

The EOCOIE bit cannot be set in CH_SLEEP mode. Only write to this bit when the related CANFD channel is in CH_RESET mode.

SOCOIE bit (Successful Occurrence Counter Overflow Interrupt Enable)

When the SOCOIE bit is 1 and the CFDC0FDSTS.SOCO bit belonging to the corresponding CAN channel is 1, an error interrupt request is generated.

The SOCOIE bit cannot be set in CH_SLEEP mode. Only write to this bit when the related CANFD channel is in CH_RESET mode.

TDCVFIE bit (Transceiver Delay Compensation Violation Interrupt Enable)

When the TDCVFIE bit is 1 and the CFDC0FDSTS.TDCVF bit belonging to the corresponding CAN channel is 1, an error interrupt request is generated.

The TDCVFIE bit cannot be set in CH_SLEEP mode.

Only write to this bit when the related CANFD channel is in CH_RESET mode. Do not set this bit when in Classical CAN mode.

Note: This bit is not available in the classical CAN function.

BOM[1:0] bits (Channel Bus-Off Mode)

The BOM[1:0]bits control the timing of the recovery from Bus-Off mode of the CANFD Channel.

Do not write to these bits in CH_SLEEP mode. Only write to these bits when the related CANFD channel is in CH_RESET mode.

Only write to these bits when the related CANFD channel is in CH_RESET mode.

ERRD bit (Channel Error Display)

The ERRD bit controls the display mode of the error flag bits [14:8] in the Channel Error Flag Register (CFDC0ERFL).

If the ERRD bit is 0 and more than one error occur at the same time, the error flag bits are set for all the errors that occurred at the same time. No further errors are flagged until CFDC0ERFL[14:8] is cleared.

Do not write to the ERRD bit in CH_SLEEP mode. Only write to this bit when the related CANFD channel is in CH_RESET or CH_HALT mode.

CTME bit (Channel Test Mode Enable)

The CTME bit enables the channel test modes.

Do not write to this bit in CH_SLEEP mode. Only write to this bit when the related CANFD channel is in CH_HALT mode.

CTMS[1:0] bits (Channel Test Mode Select)

The CTMS[1:0] bits are used to select the required test mode.

Do not write to these bits in CH_SLEEP or CH_RESET mode. Only write to these bits when the related CANFD channel is in CH_HALT mode.

These bits are cleared automatically when the related CANFD channel is in CH_RESET mode.

BFT bit (Bit Flip Test)

The BFT bit checks the internal CRC generator logic of the protocol controller.

It inverts the first bit (ID bit) of the CAN message data stream being received, so that the internal generated CRC result will not match the received CRC value of the frame. Refer to the bit stuffing rule, when using this feature, as there is the possibility of receiving a stuff error (due to the inversion) rather than a CRC error.

The internal generated CRC value is always observed in the following registers:

- CFDC0ERFL.CRCREG (Classical CAN frames)
- CFDC0FDCRC.CRCREG (CANFD frames).*1

Note 1. This feature is not available in the classical CAN function.

Some restriction exist when using this bit:

Other CAN node will send a reference message and the receiver node(s) can invert one bit of incoming bit stream.

Note: The transmitter and receiver modes share the same CRC generator, therefore it is not necessary to consider the modes separately when testing.

The Bit Flip test mode is enabled if the BFT (new control signal that inverts the first bit of the bit stream) and CTME bits are both 1 and CFDC0CTR.CTMS is 0x00.

If this function is used by a transmitting node, a bit error or an arbitration lost will occur.

Do not write to the BFT bit in CH_SLEEP mode. Users should not use this function when the Self test mode 1 (Internal Loop back mode). Only write to this bit when the related CANFD channel is in CH_HALT mode.

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode.

ROM bit (Restricted Operation Mode)

When the ROM and CTME bits are both 1, the restricted operation mode is enabled. This mode should only be used in basic test mode (CFDC0CTR.CTMS[1:0] = 00b).

The ROM bit cannot be set in CH_SLEEP mode. Only write to this bit when the related CANFD channel is in CH_HALT mode.

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode. Do not set this bit when in Classical CAN mode.

Note: This bit is not available in the classical CAN function.

28.2.5 CFDC0STS : Channel 0 Status Register

Base address: CANFD_B = 0x400B_0000

Offset address: 0x0008

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	TEC[7:0]								REC[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	ESIF	COMSTS	RECSTS	TRMSTS	BOSTS	EPSTS	CSLPSTS	CHLTSTS	CRSTSTS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Bit	Symbol	Function	R/W
0	CRSTSTS	Channel Reset Status 0: Channel not in Reset mode 1: Channel in Reset mode	R
1	CHLTSTS	Channel Halt Status 0: Channel not in Halt mode 1: Channel in Halt mode	R
2	CSLPSTS	Channel Sleep Status 0: Channel not in Sleep mode 1: Channel in Sleep mode	R
3	EPSTS	Channel Error Passive Status 0: Channel not in error passive state 1: Channel in error passive state	R
4	BOSTS	Channel Bus-Off Status 0: Channel not in bus-off state 1: Channel in bus-off state	R
5	TRMSTS	Channel Transmit Status 0: Channel is not transmitting 1: Channel is transmitting	R
6	RECSTS	Channel Receive Status 0: Channel is not receiving 1: Channel is receiving	R
7	COMSTS	Channel Communication Status 0: Channel is not ready for communication 1: Channel is ready for communication	R
8	ESIF*1	Error State Indication Flag 0: No CANFD message has been received when the ESI flag was set 1: At least one CANFD message was received when the ESI flag was set	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W
23:16	REC[7:0]	Reception Error Count These bits increment or decrement the counter value according to error status of the CAN channel during reception.	R
31:24	TEC[7:0]	Transmission Error Count These bits increment or decrement the counter value according to error status of the CAN channel during transmission.	R

Note 1. This bit is not available in the classical CAN function.

Channel Status Register shows the mode, error and transmission or reception status of the related channel together with its reception and transmission error count values.

CRSTSTS bit (Channel Reset Status)

The CRSTSTS bit indicates whether the related CAN channel is in Reset mode.

This bit is set automatically when the related CAN channel enters Channel Reset mode. When the mode is changed from Reset mode to Sleep mode, the CRSTSTS bit remains 1.

This bit is cleared automatically when the related CAN channel exits the Channel Reset mode, except when changing to Sleep mode.

CHLTSTS bit (Channel Halt Status)

The CHLTSTS bit indicates whether the related CAN channel is in Halt mode.

This bit is set automatically when the related CAN module enters Halt mode, and is cleared automatically when the related CAN module exits Halt mode.

CSLPSTS bit (Channel Sleep Status)

The CSLPSTS bit indicates whether the related CAN channel is in Sleep mode.

This bit is set automatically when the related CANFD channel enters Sleep mode, and is cleared automatically when the related CANFD channel exits Sleep mode.

EPSTS bit (Channel Error Passive Status)

The EPSTS bit indicates whether the related CANFD channel has entered the error passive state.

This bit is set automatically when the value of the CAN Transmission or Reception Counter Register exceeds the value of 0x7F.

This bit is cleared automatically when the related CANFD channel exits the error passive state or enters Reset mode.

BOSTS bit (Channel Bus-Off Status)

The BOSTS bit indicates whether the related CANFD channel has entered the error bus-off state.

This bit is set automatically when the value of the related CAN Transmission Error Count Register exceeds 0xFF and the related CANFD channel is in the bus-off state (CAN Transmission Error Count Register > 0xFF).

This bit is cleared automatically when the related CANFD channel exits bus-off state.

TRMSTS bit (Channel Transmit Status)

The TRMSTS bit indicates whether the related CANFD channel is transmitting a message.

This bit is set automatically when the related CANFD channel is operating as a transmitter node or is in the bus-off state.

This bit is cleared automatically when the related CANFD channel is in the bus-idle state or starts operating as a receiver node.

RECSTS bit (Channel Receive Status)

The RECSTS bit indicates whether the related CANFD channel is receiving a message.

This bit is set automatically when the related CANFD channel is operating as a receiver node.

This bit is cleared automatically when the related CANFD channel is in the bus-idle state or starts operating as a transmitter node.

COMSTS bit (Channel Communication Status)

The COMSTS bit indicates whether the related CANFD channel is ready for communication.

This bit is set automatically when the related CANFD channel is ready to perform communication following the detection of 11 consecutive recessive bits after exiting the Reset or Halt mode.

This bit is cleared automatically when the related CANFD channel is in CH_RESET or CD_HALT mode.

Note: This bit is 1 during bus-off state.

ESIF bit (Error State Indication Flag)

The ESIF bit is set when the ESI bit is sampled recessively for a reception CAN message without any error. When in Loopback or Mirror mode, the self-transmitted messages are considered reception messages.

If a set from the CANFD channel occurs simultaneously with a clear by a write access, then the bit is set.

This bit is cleared by writing 0 to it. This bit is cleared automatically when the related CANFD channel is in CH_RESET mode.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

Note: This bit is not available in the classical CAN function.

REC[7:0] bits (Reception Error Count)

The REC[7:0] bits increment or decrement the counter value according to error status of the CANFD channel during reception, and display the value of the REC error counter.

The value in bus-off state is indeterminate.

These bits are cleared automatically when the CANFD module enters GL_RESET or the CANFD channel is in CH_RESET mode.

TEC[7:0] bits (Transmission Error Count)

The TEC[7:0] bits increment or decrement the counter value according to error status of the CANFD channel during transmission, and display the value of the TEC error counter.

Only write to these bits when in test mode and CANFD channel is in CH_HALT mode.

These bits are cleared automatically when CANFD module is in GL_RESET or CANFD channel is in CH_RESET mode.

28.2.6 CFDC0ERFL : Channel 0 Error Flag Register

Base address: CANFD_B = 0x400B_0000

Offset address: 0x000C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	CRCREG[14:0]														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	ADERR	B0ERR	B1ERR	CERR	AERR	FERR	SERR	ALF	BLF	OVLF	BORF	BOEF	EPF	EWF	BEF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BEF	Bus Error Flag 0: Channel bus error not detected 1: Channel bus error detected	R/W
1	EWF	Error Warning Flag 0: Channel error warning not detected 1: Channel error warning detected	R/W
2	EPF	Error Passive Flag 0: Channel error passive not detected 1: Channel error passive detected	R/W
3	BOEF	Bus-Off Entry Flag 0: Channel bus-off entry not detected 1: Channel bus-off entry detected	R/W
4	BORF	Bus-Off Recovery Flag 0: Channel bus-off recovery not detected 1: Channel bus-off recovery detected	R/W
5	OVLF	Overload Flag 0: Channel overload not detected 1: Channel overload detected	R/W
6	BLF	Bus Lock Flag 0: Channel bus lock not detected 1: Channel bus lock detected	R/W
7	ALF	Arbitration Lost Flag 0: Channel arbitration lost not detected 1: Channel arbitration lost detected	R/W
8	SERR	Stuff Error 0: Channel stuff error not detected 1: Channel stuff error detected	R/W
9	FERR	Form Error 0: Channel form error not detected 1: Channel form error detected	R/W
10	AERR	Acknowledge Error 0: Channel acknowledge error not detected 1: Channel acknowledge error detected	R/W

Bit	Symbol	Function	R/W
11	CERR	CRC Error 0: Channel CRC error not detected 1: Channel CRC error detected	R/W
12	B1ERR	Bit 1 Error 0: Channel bit 1 error not detected 1: Channel bit 1 error detected	R/W
13	B0ERR	Bit 0 Error 0: Channel bit 0 error not detected 1: Channel bit 0 error detected	R/W
14	ADERR	Acknowledge Delimiter Error 0: Channel acknowledge delimiter error not detected 1: Channel acknowledge delimiter error detected	R/W
15	—	This bit is read as 0. The write value should be 0.	R/W
30:16	CRCREG[14:0]	CRC Register value These bits show the CRC value calculated for the CAN2.0 CAN frame.	R
31	—	This bit is read as 0. The write value should be 0.	R/W

Channel Error Flag register shows the status of various error conditions detectable regardless of the setting of the related CAN Channel Error Interrupt Enable Register. It also shows the status of the various bus errors detectable by the CAN channel. Refer to the CAN specification (ISO 11898-1) to check when each error condition occurs.

For this register, only a single bit can be cleared by software. Do not use the bit clear instruction to clear the bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

Example in assembler language to clear the CFDC0ERFL.BEF bit:

mov.b #0x0FE, CFDC0ERFL ;

BEF bit (Bus Error Flag)

The BEF bit indicates a detection of a CAN channel bus error state, flagged by bits [14:8] in this register.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

This bit is set automatically when a bus error is detected, and is cleared automatically when the related CANFD channel is in CH_RESET mode.

If a set from the CAN channel occurs simultaneously with a clear by a write access, then the bit is set.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

EWF bit (Error Warning Flag)

The EWF bit indicates whether an error warning condition has been detected for the CAN channel.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

This bit is set automatically when either TEC or REC exceeds 0x5F.

The setting of this bit only occurs when the TEC or REC initially exceeds 0x5F. Therefore, if the TEC or REC remains > 0x5F and the EWF bit is cleared by software, it is not set again until both the TEC and REC go below 0x60 and either TEC or REC crosses over again from a value 0x5F to a value > 0x5F.

If a set condition occurs simultaneously with a clear condition, then the bit is set. It is cleared automatically when the related CANFD channel is in CH_RESET mode.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

EPF bit (Error Passive Flag)

The EPF bit indicates a detection of a CAN channel error passive state.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

This bit is set automatically when the CAN error state becomes error passive state.

The setting of this bit only occurs when the TEC or REC initially exceeds 0x7F. Therefore, if the TEC or REC remains > 0x7F and the bit is cleared by software, it is not set again until both the TEC and REC go below 0x80 and either TEC or REC crosses over again from a value ≤ 0x7F to a value > 0x7F.

If a set condition occurs simultaneously with a clear condition, then the bit is set. It is cleared automatically when the related CANFD channel is in CH_RESET mode.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

BOEF bit (Bus-Off Entry Flag)

The BOEF bit indicates a detection of a CAN channel bus-off entry state.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

This bit is set automatically when the CAN error state enters the bus-off state.

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode. If a set condition occurs simultaneously with a clear condition, then the bit is set.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

BORF bit (Bus-Off Recovery Flag)

The BORF bit indicates a detection of a CAN channel bus-off recovery state.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

This bit is set automatically if CAN channel recovers from bus-off state in the following conditions:

- When CFDC0CTR.BOM is 00b and normal recovery (11 consecutive recessive bits x 128 times detected) occurs
- When CFDC0CTR.BOM is 10b and normal recovery (11 consecutive recessive bits x 128 times detected) occurs
- When CFDC0CTR.BOM is 11b and normal recovery (11 consecutive recessive bits x 128 times detected) occurs.

The bit is not set if CAN channel recovers from bus-off state in the following conditions:

- When CAN Reset mode is requested
- When CFDC0CTR.RTBO is set to 1 (the CAN channel returns to error active)
- When CFDC0CTR.BOM is 01b
- When CFDC0CTR.BOM is 11b and a halt request is asserted before the CAN channel reaches the end of the bus-off state.

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode. If a set condition occurs simultaneously with a clear condition, the flag is set.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

OVLF bit (Overload Flag)

The OVLF flag indicates a detection of a CAN channel overload state.

The OVLF bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

This bit is set automatically when an overload condition is detected. If a set condition occurs simultaneously with a clear condition, then the bit is set.

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

BLF bit (Bus Lock Flag)

The BLF bit indicates a detection of a CAN channel bus lock condition.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

This bit is set automatically when 32 consecutive dominant bits are detected on the CAN bus while the CAN channel is in Operation mode.

If a set condition occurs simultaneously with a clear condition, then the bit is set. It is cleared automatically when the related CANFD channel is in CH_RESET mode.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

It is cleared automatically when the related CANFD channel is in CH_RESET mode.

ALF bit (Arbitration Lost Flag)

The ALF bit indicates a detection of a CAN channel bus arbitration lost condition.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

The bit is set automatically when an arbitration lost condition is detected on the CAN bus while the CAN channel is in Operation mode.

If a set condition occurs simultaneously with a clear condition, then the bit is set. It is cleared automatically when the related CANFD channel is in CH_RESET mode.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

SERR bit (Stuff Error)

The SERR bit indicates a detection of a CAN stuff error.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

To clear this bit, use the following sequence:

1. Clear the corresponding flag bit.
2. Read if the flag bit is cleared.
3. If yes, continue, else go back to step 1.

This bit is set automatically when a stuff error is detected. If CFDC0CTR.ERRD bit is 1 and if the set and clear conditions occur at the same time for this bit, then this bit is set.

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode. If CFDC0CTR.ERRD bit is 0 and the set and clear conditions occur at the same time for this bit, then it is cleared if a bit at CFDC0ERFL[14:8] is already set. Otherwise, this bit is set if CFDC0ERFL[14:8] is 0000000b.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

FERR bit (Form Error)

The FERR bit indicates a detection of a CAN form error.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

To clear this bit, use the following sequence:

1. Clear the corresponding flag bit.
2. Read if the flag bit is cleared.
3. If yes, continue, else go back to step 1.

This bit is set automatically when a form error is detected. If CFDC0CTR.ERRD bit is 1 and if the set and clear conditions occur at the same time for this bit, then this bit is set.

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode. If CFDC0CTR.ERRD bit is 0 and the set and clear conditions occur at the same time for this bit, then it is cleared if a bit at CFDC0ERFL[14:8] is already set. Otherwise, this bit is set if CFDC0ERFL[14:8] is 0000000b.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

AERR bit (Acknowledge Error)

The AERR bit indicates a detection of a CAN acknowledge error.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

To clear this bit, use the following sequence:

1. Clear the corresponding flag bit.
2. Read if the flag bit is cleared.
3. If yes, continue, else go back to step 1.

This bit is set automatically when an acknowledge error is detected. If CFDC0CTR.ERRD bit is 1 and if the set and clear conditions occur at the same time for this bit, then this bit is set.

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode. If CFDC0CTR.ERRD bit is 0 and the set and clear conditions occur at the same time for this bit, then it is cleared if a bit at CFDC0ERFL[14:8] is already set. Otherwise, this bit is set if CFDC0ERFL[14:8] is 0000000b.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION.

CERR bit (CRC Error)

The CERR bit indicates a detection of a CAN CRC error.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

To clear this bit, use the following sequence:

1. Clear the corresponding flag bit.
2. Read if the flag bit is cleared.
3. If yes, continue, else go back to step 1.

This bit is set automatically when a CRC error is detected. If CFDC0CTR.ERRD bit is 1 and if the set and clear conditions occur at the same time for this bit, then this bit is set.

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode. If CFDC0CTR.ERRD bit is 0 and the set and clear conditions occur at the same time for this bit, then it is cleared if a bit at CFDC0ERFL[14:8] is already set. Otherwise, this bit is set if CFDC0ERFL[14:8] is 0000000b.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

B1ERR bit (Bit 1 Error)

The B1ERR bit indicates a detection of a recessive bit error.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

To clear this bit, use the following sequence:

1. Clear the corresponding flag bit.
2. Read if the flag bit is cleared.
3. If yes, continue, else go back to step 1.

This bit is set automatically when a recessive bit error (expected recessive bit, sampled as dominant bit) is detected. If CFDC0CTR.ERRD bit is 1 and if the set and clear conditions occur at the same time for this bit, then this bit is set.

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode. If CFDC0CTR.ERRD bit is 0 and the set and clear conditions occur at the same time for this bit, then it is cleared if a bit at CFDC0ERFL[14:8] is already set. Otherwise, this bit is set if CFDC0ERFL[14:8] is 0000000b.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

B0ERR bit (Bit 0 Error)

The B0ERR bit indicates a detection of a dominant bit error.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

To clear this bit, use the following sequence:

1. Clear the corresponding flag bit.
2. Read if the flag bit is cleared.
3. If yes, continue, else go back to step 1.

This bit is set automatically when a dominant bit error (expected dominant bit, sampled as recessive bit) is detected. If CFDC0CTR.ERRD bit is 1 and if the set and clear conditions occur at the same time for this bit, then this bit is set.

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode. If CFDC0CTR.ERRD bit is 0 and the set and clear conditions occur at the same time for this bit, then it is cleared if a bit at CFDC0ERFL[14:8] is already set. Otherwise, this bit is set if CFDC0ERFL[14:8] is 0000000b.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

ADERR bit (Acknowledge Delimiter Error)

The ADERR bit indicates a detection of an acknowledge delimiter bit error.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

To clear this bit, use the following sequence:

1. Clear the corresponding flag bit.
2. Read if the flag bit is cleared.
3. If yes, continue, else go back to step 1.

This bit is set automatically when a form error is detected during the acknowledge delimiter state of frame transmission. If CFDC0CTR.ERRD bit is 1 and if the set and clear conditions occur at the same time for this bit, then this bit is set.

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode. If CFDC0CTR.ERRD bit is 0 and the set and clear conditions occur at the same time for this bit, then it is cleared if a bit at CFDC0ERFL[14:8] is already set. Otherwise, this bit is set if CFDC0ERFL[14:8] is 0000000b.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

CRCREG[14:0] bits (CRC Register value)

The CRCREG[14:0] bits read the calculated CRC value when CFDC0CTR.CTME bit is 1 for the channel.

If CFDC0CTR.CTME bit is 0, then these bits are always read as 0.

These bits show the CAN2.0 CRC value calculated by the CANFD channel logic when the CTME bit is enabled.

The CFDC0ERFL.CRCREG value is updated in the first bit of the CRC field of the CAN frame (reception and transmission).

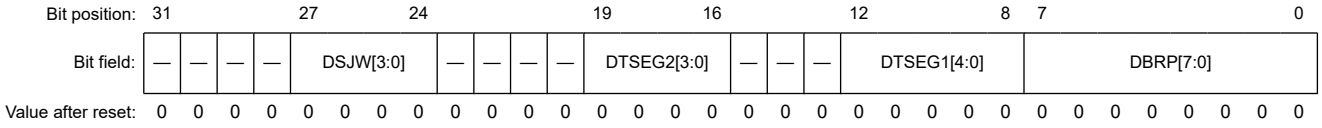
These bits are cleared automatically when the related CANFD channel is in CH_RESET mode.

28.2.7 CFDC0DCFG : Channel 0 Data Bitrate Configuration Register

This register is not available in the classical CAN function.

Base address: CANFD_B = 0x400B_0000

Offset address: 0x0100



Bit	Symbol	Function	R/W
7:0	DBRP[7:0]	Channel Data Baud Rate Prescaler Data Baud Rate Prescaler division ratio	R/W
12:8	DTSEG1[4:0]	Timing Segment 1 0x00: Reserved 0x01: 2 Tq 0x02: 3 Tq 0x03: 4 Tq ⋮ 0x1E: 31 Tq 0x1F: 32 Tq	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
19:16	DTSEG2[3:0]	Timing Segment 2 0x0: Reserved 0x1: 2 Tq ⋮ 0xE: 15 Tq 0xF: 16 Tq	R/W
23:20	—	These bits are read as 0. The write value should be 0.	R/W
27:24	DSJW[3:0]	Resynchronization Jump Width 0x0: 1 Tq 0x1: 2 Tq ⋮ 0xF: 16 Tq	R/W
31:28	—	These bits are read as 0. The write value should be 0.	R/W

Note: Tq means time quantum.

The Channel 0 Data Bitrate Configuration Register configures the transmission/reception data baud rate parameters of the channels.

The channel of Classical CAN mode does not perform configuration of this register.

DBRP[7:0] bits (Channel Data Baud Rate Prescaler)

The DBRP[7:0] bits define the peripheral bus clock periods contained in a time quantum.

Do not write to these bits in CH_OPERATION or CH_SLEEP mode.

Only write to these bits when the related CANFD channel is in CH_RESET or CH_HALT mode.

DTSEG1[4:0] bits (Timing Segment 1)

The DTSEG1[4:0] bits set the segment TSEG1 to compensate for edges on the CAN bus with a positive phase error. A value from 2 to 32 time quanta can be set.

The DTSEG1[4:0] bits are also used to set the propagation segment.

Do not write to these bits in CH_OPERATION or CH_SLEEP mode.

Only write to these bits when the related CANFD channel is in CH_RESET or CH_HALT mode. Do not write any other value to these bits. See section 28.4.1.2. CAN Bit Timing for more details.

DTSEG2[3:0] bits (Timing Segment 2)

The DTSEG2[3:0] bits set the segment TSEG2 to compensate for edges on the CAN bus with a negative phase error. A value from 2 to 16 time quanta can be set.

Do not write to these bits in CH_OPERATION or CH_SLEEP mode.

Only write to these bits when the related CANFD channel is in CH_RESET or CH_HALT mode. Do not write any other value to these bits.

DSJW[3:0] bits (Resynchronization Jump Width)

The DSJW[3:0] bits set the synchronization jump width. A value from 1 to 16 time quanta can be set.

Do not write to these bits in CH_OPERATION or CH_SLEEP mode.

Only write to these bits when the related CANFD channel is in CH_RESET or CH_HALT mode.

28.2.8 CFDC0FDCFG : Channel 0 CANFD Configuration Register

Base address: CANFD_B = 0x400B_0000

Offset address: 0x0104

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	CLOE	REFE	FDOE	—	—	—	—	TDCO[7:0]							
Value after reset:	0	0/1*1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
2:0	EOCCFG[2:0]	Error Occurrence Counter Configuration 0 0 0: All transmitter or receiver CAN frames 0 0 1: All transmitter CAN frames 0 1 0: All receiver CAN frames 0 1 1: Reserved 1 0 0: Only transmitter or receiver CANFD data-phase (fast bits) 1 0 1: Only transmitter CANFD data-phase (fast bits) 1 1 0: Only receiver CANFD data-phase (fast bits) 1 1 1: Reserved	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
8	TDCOC*2	Transceiver Delay Compensation Offset Configuration 0: Measured + offset 1: Offset-only	R/W
9	TDCE*2	Transceiver Delay Compensation Enable 0: Transceiver delay compensation disabled 1: Transceiver delay compensation enabled	R/W
10	ESIC*2	Error State Indication Configuration 0: The ESI bit in the frame represents the error state of the node itself 1: The ESI bit in the frame represents the error state of the message buffer if the node itself is not in error passive. If the node is in error passive, then the ESI bit is driven by the node itself.	R/W
11	—	This bit is read as 0. The write value should be 0.	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W
23:16	TDCO[7:0]*2	Transceiver Delay Compensation Offset	R/W
27:24	—	These bits are read as 0. The write value should be 0.	R/W
28	FDOE*2	FD-Only Enable 0: FD-only mode disabled 1: FD-only mode enabled	R/W
29	REFE	RX Edge Filter Enable 0: RX edge filter disabled 1: RX edge filter enabled	R/W
30	CLOE*2 *3	Classical CAN Enable 0: Classical CAN mode disabled 1: Classical CAN mode enabled	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

Note 1. The value after reset is 0 for products that support the CAN-FD protocol, and 1 for products that support only classical CAN protocol.
Note 2. These bits are not available in the classical CAN function.
Note 3. This bit can only be written for products that support the CAN-FD protocol. For products that support only classical CAN protocol, this bit is reserved and fixed to 1.

The Channel 0 CANFD Configuration Register configures which communication direction (transmitter/receiver) errors are counted.

EOCCFG[2:0] bits (Error Occurrence Counter Configuration)

The EOCCFG[2:0] bits select which type of CAN frame configuration and direction, including protocol errors are counted.

Do not write to these bits in CH_OPERATION or CH_SLEEP mode.

Only write to these bits when the related CANFD channel is in CH_RESET or CH_HALT mode.

TDCOC bit (Transceiver Delay Compensation Offset Configuration)*1

The TDCOC bit selects which offset is used when defining the position of the secondary sample point (SSP) for the CANFD channel. If the bit is set to 0, the position of the SSP is the measured transceiver delay plus the fixed offset. If the bit is 1, the position of the SSP is defined only by the offset.

Do not write to this bit in CH_OPERATION or CH_SLEEP mode.

Only write to this bit when the related CANFD channel is in CH_RESET or CH_HALT mode. Do not set this bit when in Classical CAN mode.

TDCE bit (Transceiver Delay Compensation Enable)*1

The TDCE bit enables the transceiver delay compensation for the CANFD channel.

Do not write to this bit in CH_OPERATION or CH_SLEEP mode.

Only write to this bit when the related CANFD channel is in CH_RESET or CH_HALT mode. Do not set this bit when in Classical CAN mode.

ESIC bit (Error State Indication Configuration)*1

The ESIC bit controls the transmission of either the ESI flag information or the message of ESI flag information (CFDCFFDCSTS.CFESI or CFDTMFDCTRb.TMESI).

Do not write to this bit in CH_OPERATION or CH_SLEEP mode.

Only write to this bit when the related CANFD channel is in CH_RESET or CH_HALT mode. Do not set this bit when in Classical CAN mode.

TDCO[7:0] bits (Transceiver Delay Compensation Offset)*1

The TDCO[7:0] bits set the secondary sample point offset. How this value is used, depends on the CFDC0FDCFG.TDCOC setting.

If CFDC0FDCFG.TDCOC = 0, the transceiver delay compensation result is equal to the Trv_Delay (measured delay) + the value in CFDC0FDCFG.TDCO, rounded down to the nearest integer number of time quanta. Otherwise, the result is equal to the value in CFDC0FDCFG.TDCO. See section 28.4.1.5. Transmitter Delay Compensation for details on how CFDC0FDCFG.TDCO is used.

The actual offset value is interpreted as TDCO + 1. For example, if 4 is set in TDCO, the offset is 5 clock cycles. Clock cycle is 1 cycle of CAN channel DLL clock.

Do not write to the TDCO[7:0] bits in CH_OPERATION or CH_SLEEP mode.

Only write to these bits when the related CANFD channel is in CH_RESET or CH_HALT mode. Do not set this bit when in Classical CAN mode.

FDOE bit (FD-Only Enable)*1

The FDOE bit enables the reception and transmission of CANFD-only frames. If enabled, communication in Classical CAN frame format is disabled. Transmission of Classical CAN frames is not possible because the FDF bit of the message buffer is a don’t care (CFDCFFDCSTS.CFFDF/CFDTMFDCTRb.TMFDF).

If messages with Classical CAN frame format are received, the protocol controller treats them as invalid frames and response with error frames. When a Classical CAN frame is configured for transmitting, the FDF bit is sent as recessive, therefore an FD frame is sent. If the data length code (DLC) is configured of greater than 8 bytes, the remaining data bytes are padded with 0xCC.

The FDOE bit cannot be written in CH_OPERATION, CH_HALT or CH_SLEEP mode.

Do not set CFDC0FDCFG.FDOE and CFDC0FDCFG.CLOE simultaneously.

REFE bit (RX Edge Filter Enable)

The REFE bit enables the RX edge filter during the IDLE detection (bus integration). When the bit is enabled, two consecutive dominant time quanta are required to detect a synchronization edge.

The REFE bit cannot be written in CH_OPERATION, CH_HALT and CH_SLEEP mode. Do not set this bit when in Classical CAN mode.

CLOE bit (Classical CAN Enable)*1

The CLOE bit enables the Classical CAN mode. If this bit is 1, the protocol controller can only send classical frames and response with a form or CRC error on FD frames.

Do not set CFDC0FDCFG.CLOE and CFDC0FDCFG.FDOE simultaneously.

CFDC0FDCFG.CLOE	CFDC0FDCFG.FDOE	Channel mode
0	0	CANFD mode
0	1	FD-only mode
1	0	Classical CAN mode
1	1	Reserved

The CANFD mode is available only for CANFD supported product.

Do not write to this bit in CH_OPERATION, CH_HALT or CH_SLEEP mode.

Only write to these bits when the CANFD channel is in CH_RESET mode.

Note 1. These bits are not available in the classical CAN function.

28.2.9 CFDC0FDCTR : Channel 0 CANFD Control Register

Base address: CANFD_B = 0x400B_0000

Offset address: 0x0108

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

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

Bit	Symbol	Function	R/W
0	EOCCLR	Error Occurrence Counter Clear 0: No error occurrence counter clear 1: Clear error occurrence counter	R/W
1	SOCCLR	Successful Occurrence Counter Clear 0: No successful occurrence counter clear 1: Clear successful occurrence counter	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

The Channel n CANFD Control Register (n = 0) controls the error and successful occurrence counters.

EOCCLR bit (Error Occurrence Counter Clear)

The EOCCLR bit is used to clear the error occurrence counter.

Do not write to this bit in CH_SLEEP or CH_RESET mode. The read value is always 0.

This bit is cleared automatically by the CANFD module logic and when the related CANFD channel is in CH_RESET mode.

SOCCLR bit (Successful Occurrence Counter Clear)

The SOCCLR bit is used to clear the successful occurrence counter.

Do not write to this bit in CH_SLEEP or CH_RESET mode. The read value is always 0.

This bit is cleared automatically by the CANFD module logic and when the related CANFD channel is in CH_RESET mode.

28.2.10 CFDC0FDSTS : Channel 0 CANFD Status Register

Base address: CANFD_B = 0x400B_0000

Offset address: 0x010C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	SOC[7:0]								EOC[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
7:0	TDCR[7:0]*1	Transceiver Delay Compensation Result	R
8	EOCO	Error Occurrence Counter Overflow 0: Error occurrence counter has not overflowed 1: Error occurrence counter has overflowed	R/W
9	SOCO	Successful Occurrence Counter Overflow 0: Successful occurrence counter has not overflowed 1: Successful occurrence counter has overflowed	R/W
14:10	—	These bits are read as 0. The write value should be 0.	R/W
15	TDCVF*1	Transceiver Delay Compensation Violation Flag 0: Transceiver delay compensation violation has not occurred 1: Transceiver delay compensation violation has occurred	R/W
23:16	EOC[7:0]	Error Occurrence Counter These bits show the error occurrence counter value.	R
31:24	SOC[7:0]	Successful occurrence counter These bits show the successful occurrence counter value.	R

Note 1. These bits are not available in the classical CAN function.

The Channel 0 CANFD Status Register indicates the transceiver compensation delay result and its related FIFO message lost status.

TDCR[7:0] bits (Transceiver Delay Compensation Result)

The TDCR[7:0] bits are set when the transceiver delay has been measured.

The measured delay is a multiple of the CAN channel DLL clock. The result depends on the CFDC0FDCFG.TDCOC configuration and the offset value in CFDC0FDCFG.TDCO. See section 28.4.1.5. Transmitter Delay Compensation for details on how this value is derived.

The TDCR[7:0] bits are updated at the falling edge between FDF and the RES bit when CFDC0FDCFG.TDCOC = 0 and the transceiver delay compensation is enabled (CFDC0FDCFG.TDCE = 1).

These bits are cleared automatically when the related CANFD channel is in CH_RESET mode.

Note: These bits are not available in the classical CAN function.

EOCO bit (Error Occurrence Counter Overflow)

The EOCO bit indicates whether the related CAN channel error occurrence counter has overflowed. This bit is cleared by writing 0 to it. Writing 1 has no effect.

This bit is set automatically when CFDC0FDSTS.EOC is 0xFF and a CAN bus error is detected based on the configuration defined in CFDC0FDCFG.EOCCFG.

If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

SOCO bit (Successful Occurrence Counter Overflow)

The SOCO bit indicates whether the related CAN channel successful occurrence counter has overflowed. This bit is cleared by writing 0 to it. Writing 1 has no effect.

This bit is set automatically when CFDC0FDSTS.SOC is 0xFF and a successful message reception or successful message transmission occurs.

If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode.

Write to this bit only when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

TDCVF bit (Transceiver Delay Compensation Violation Flag)

The CANFD module captures internally the transmitted data bit-by-bit. This data is then compared against the received CAN bus level which is delayed by the transceiver loop delay.

The transceiver delay has some variations depending on the physical parameters such as temperature. The result bit CFDC0FDSTS.TDCR is updated by each message. However, temporary maximum delay violation could be missed. Therefore, the TDCVF bit captures this violation.

This bit is cleared by writing 0 to it. Writing 1 has no effect.

This bit is set automatically when the transceiver delay compensation is greater than the maximum delay compensation (6 data bit times - 2 clk_dlc) and the internal bit is overrun.

If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

Note: This bit is not available in the classical CAN function.

EOC[7:0] bits (Error Occurrence Counter)

The EOC[7:0] bits are used together with the SOC[7:0] bits to support an option for host-controlled fall-back to payload bit rate identical to arbitration bit rate when messages utilizing the reduced payload bit length have significant higher error rates compared to other messages.

This higher error rate can be detected depending on the configuration of the CFDC0FDCFG.EOCCFG bits.

The EOC[7:0] bits are set only by CANFD module logic. These bits are cleared by writing 1 to CFDC0FDCTR.EOCCLR. Writing any other value has no effect.

These bits are updated when an error occurs, according to the configuration of the CFDC0FDCFG.EOCCFG bits. When the counter reaches the value of 0xFF, the update stops.

These bits are cleared automatically when the related CANFD channel is in CH_RESET mode.

SOC[7:0] bits (Successful occurrence counter)

The SOC[7:0] bits are used together with the EOC[7:0] bits to support an option for host-controlled fall-back to payload bit rate identical to arbitration bit rate when messages utilizing the reduced payload bit length have significant higher error rates compared to other messages.

The SOC[7:0] bits are set only by CANFD module logic. Writing any other value has no effect.

These bits are updated when the occurrence of any error-free messages on the bus is detected through reception or transmission. When the counter reaches the value of 0xFF, the update stops.

Note: In Loopback mode, the counter is incremented twice.

These bits are cleared by writing 1 to CFDC0FDCTR.SOCCLR.

These bits are cleared automatically when the related CANFD channel is in CH_RESET mode.

28.2.11 CFDC0FDCRC : Channel 0 CANFD CRC Register

This register is not available in the classical CAN function.

Base address: CANFD_B = 0x400B_0000

Offset address: 0x0110

Bit position:	31				27			24			20																				0
Bit field:	—	—	—	—	SCNT[3:0]				—	—	—	CRCREG[20:0]																			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
20:0	CRCREG[20:0]	CRC Register value These bits show the CRC value calculated for the CANFD frame.	R
23:21	—	These bits are read as 0. The write value should be 0.	R/W
27:24	SCNT[3:0]	Stuff bit count These bits shows the stuff bit count (mod 8) for the CANFD frame.	R
31:28	—	These bits are read as 0. The write value should be 0.	R/W

The Channel 0 CANFD CRC Register holds the CRC value calculated for the CANFD frame.

CRCREG[20:0] bits (CRC Register value)

The CRCREG[20:0] bits contain the CRC value calculated by the CANFD channel logic when the CFDC0CTR.CTME bit is enabled.

The CFDC0FDCRC.CRCREG value is updated in the first bit of the CRC field of the CANFD frame (reception and transmission).

When the CFDC0CTR.CTME bit is 0, the CRCREG[20:0] bits are always read as 0.

When bit 17th of the CRC field is used, CRCREG[20:17] are always read as 0.

These bits are cleared automatically when the related CANFD channel is in CH_RESET mode.

SCNT[3:0] bits (Stuff bit count)

The SCNT[3:0] bits contain the stuff count value of the CANFD frame. These bits indicate the number of inserted stuff bits (modulo 8, Graycoded) for a CANFD frame when the CFDC0CTR.CTME bit is enabled in CFDC0FDCRC.SCNT[3:1]. SCNT[0] is the parity bit.

When the CFDC0CTR.CTME bit is 0, the SCNT[3:0] bits are always read as 0.

The SCNT value is updated in the first bit of CRC field of the CANFD frame (reception and transmission).

These bits are cleared automatically when the related CANFD channel is in CH_RESET mode.

28.2.12 CFDGCFG : Global Configuration Register

Base address: CANFD_B = 0x400B_0000

Offset address: 0x0014

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	ITRCP[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	TSSS	TSP[3:0]				—	—	CMPOC	DCS	MME	DRE	DCE	TPRI
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TPRI	Transmission Priority 0: ID priority 1: Message buffer number priority	R/W
1	DCE	DLC Check Enable 0: DLC check disabled 1: DLC check enabled	R/W
2	DRE	DLC Replacement Enable 0: DLC replacement disabled 1: DLC replacement enabled	R/W
3	MME	Mirror Mode Enable 0: Mirror mode disabled 1: Mirror mode enabled	R/W
4	DCS	Data Link Controller Clock Select 0: Internal clean clock 1: External clock source connected to CANMCLK pin	R/W
5	CMPOC*1	CANFD Message Payload Overflow Configuration 0: Message is rejected 1: Message payload is cut to fit to configured message size	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
11:8	TSP[3:0]	Timestamp Prescaler 0x0: Timestamp prescaler = 1 0x1: Timestamp prescaler = 2 0x2: Timestamp prescaler = 4 0x3: Timestamp prescaler = 8 ⋮ 0xD: Timestamp prescaler = 8192 0xE: Timestamp prescaler = 16384 0xF: Timestamp prescaler = 32768	R/W
12	TSSS	Timestamp Source Select 0: Source clock for timestamp counter is peripheral clock 1: Source clock for timestamp counter is bit time clock	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
31:16	ITRCP[15:0]	Interval Timer Reference Clock Prescaler FIFO interval timer prescaler value	R/W

Note 1. This bit is not available in the classical CAN function.

The Global Configuration Register is used to select the transmission priority to be used for all the TX message buffers and the clock source for the CAN protocol engine of CAN channel. The CFDGCFG register is also used to select the source for the timestamp clock and to configure the frequency for the timestamp clock and interval timer reference clock.

TPRI bit (Transmission Priority)

The TPRI bit selects the transmission priority for CAN channel.

Do not write to this bit in GL_SLEEP mode. Only write to this bit when CANFD module is in GL_RESET mode.

Message buffer number priority should not be used together with TX queue transmission.

DCE bit (DLC Check Enable)

The DCE bit enables data length code (DLC) check for CAN channel.

Do not write to this bit in GL_SLEEP mode. Only write to this bit when CANFD module is in GL_RESET mode.

DRE bit (DLC Replacement Enable)

When the DRE bit is 1 and the DCE is 1, the CANFD stores the configured value (CFDGAFLP0r.GAFLDLC) of the DLC in the destination RX message buffer or FIFO buffer if the DLC check passes. Otherwise, the DLC value in the destination RX message buffer or FIFO buffer is unchanged.

Do not write to this bit in GL_SLEEP mode. Only write to this bit when CANFD module is in GL_RESET mode.

MME bit (Mirror Mode Enable)

The MME bit enables the Mirror mode for CAN channel.

Do not write to this bit in GL_SLEEP mode. Only write to this bit when CANFD module is in GL_RESET mode.

DCS bit (Data Link Controller Clock Select)

The DCS bit selects the clock source for CAN communication. Internal clean clock has a smaller clock jitter than the peripheral clock B (PCLKB).

Do not write to this bit in GL_SLEEP or GL_OPERATION mode. Only write to this bit when CANFD module is in GL_RESET mode.

CMPOC bit (CANFD Message Payload Overflow Configuration)

The CMPOC bit controls the message payload acceptance mechanism when the received payload is higher than the message buffer payload size CFDRMNB.RMPLS, CFDRFCCa.RFPLS, and CFDCFCC.CFPLS. The received message payload is always compared with the available message payload size in the message buffer.

Do not write to this bit in GL_SLEEP or GL_OPERATION mode. Only write to this bit when CANFD module is in GL_RESET mode.

When this bit is set and payload overflow occurs, the DLC value is stored in the RX message buffer or FIFO buffer unchanged.

Note: This bit is not available in the classical CAN function.

TSP[3:0] bits (Timestamp Prescaler)

The value configured in the TSP[3:0] bits defines the period of the clock source used for the timestamp counter.

Do not write to this bit in GL_SLEEP mode. Only write to this bit when CANFD module is in GL_RESET mode.

TSSS bit (Timestamp Source Select)

The TSSS bit allows the selection of the clock source for the timestamp counter.

Do not write to this bit in GL_SLEEP mode. Only write to this bit when CANFD module is in GL_RESET mode. Additionally, do not set this bit to 1 when CANFD communication is used.[*1]

Note: The bit time clock varies depending on the nominal and data rate bit configuration.

Note 1. This feature is not available in the classical CAN function.

ITRCP[15:0] bits (Interval Timer Reference Clock Prescaler)

The ITRCP[15:0] bits allow the definition of a reference clock for the FIFO interval timer source clock.

When these bits are 0x0000, the timer is disabled.

Do not write to this bit in GL_SLEEP mode. Only write to this bit when CANFD module is in GL_RESET mode.

28.2.13 CFDGCTR : Global Control Register

Base address: CANFD_B = 0x400B_0000

Offset address: 0x0018

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	CMPOFIE	THLEIE	MEIE	DEIE	—	—	—	—	—	GSLPR	GMDC[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Bit	Symbol	Function	R/W
1:0	GMDC[1:0]	Global Mode Control 0 0: Global operation mode request 0 1: Global reset mode request 1 0: Global halt mode request 1 1: Keep current value	R/W
2	GSLPR	Global Sleep Request 0: Global sleep request disabled 1: Global sleep request enabled	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
8	DEIE	DLC Check Interrupt Enable 0: DLC check interrupt disabled 1: DLC check interrupt enabled	R/W
9	MEIE	Message Lost Error Interrupt Enable 0: Message lost error interrupt disabled 1: Message lost error interrupt enabled	R/W
10	THLEIE	TX History List Entry Lost Interrupt Enable 0: TX history list entry lost interrupt disabled 1: TX history list entry lost interrupt enabled	R/W
11	CMPOFIE*1	CANFD Message Payload Overflow Flag Interrupt Enable 0: CANFD message payload overflow flag interrupt disabled 1: CANFD message payload overflow flag interrupt enabled	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W
16	TSRST	Timestamp Reset 0: Timestamp not reset 1: Timestamp reset	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. This bit is not available in the classical CAN function.

The Global Control Register controls the global mode of the CANFD module and the timestamp function. The register also enables and disables the global error interrupts.

GMDC bits (Global Mode Control)

The GMDC bits can be used to configure the modes for the CANFD module. Additionally, if CFDGCTR.GSLPR bit is 1 when the CANFD module is in Reset mode, the CANFD module enters Global Sleep mode.

Setting the GMDC bits to 11b has no effect. Mode transition is described in detail in section 28.3.2. Global Modes.

Do not write to this bit when the CANFD module is in GL_SLEEP mode.

GSLPR bit (Global Sleep Request)

The GSLPR bit globally selects the sleep request for CANFD module including CAN channels. Channel sleep request is set automatically for channels.

Only write to this bit when the CANFD module is in GL_RESET or GL_SLEEP mode.

DEIE bit (DLC Check Interrupt Enable)

When the DEIE bit is 1, an interrupt is generated if a DLC error is detected in the received frames.

Do not write to this bit when the CANFD module is in GL_SLEEP mode.

MEIE bit (Message Lost Error Interrupt Enable)

When the MEIE bit is 1, an interrupt is generated if a message lost condition occurs.

Do not write to this bit when the CANFD module is in GL_SLEEP mode.

THLEIE bit (TX History List Entry Lost Interrupt Enable)

When the THLEIE bit is 1, an interrupt is generated if a TX history list entry lost condition occurs.

Do not write to this bit when the CANFD module is in GL_SLEEP mode.

CMPOFIE bit (CANFD Message Payload Overflow Flag Interrupt Enable)

When the CMPOFIE bit is 1, an interrupt is generated when a CANFD message payload overflow condition occurs.

Do not write to this bit when the CANFD module is in GL_SLEEP mode.

Note: This bit is not available in the classical CAN function

TSRST bit (Timestamp Reset)

When the TSRST bit is 1, the Global Timestamp Register is reset to 0x0000.

Do not write to this bit when the CANFD module is in GL_SLEEP or GL_RESET mode.

Read value is always 0.

This bit is cleared automatically by the CANFD module logic.

28.2.14 CFDGSTS : Global Status Register

Base address: CANFD_B = 0x400B_0000

Offset address: 0x001C

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

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

Bit	Symbol	Function	R/W
0	GRSTSTS	Global Reset Status 0: Not in Reset mode 1: In Reset mode	R
1	GHLTSTS	Global Halt Status 0: Not in Halt mode 1: In Halt mode	R
2	GSLPSTS	Global Sleep Status 0: Not in Sleep mode 1: In Sleep mode	R
3	GRAMINIT	Global RAM Initialization 0: RAM initialization is complete 1: RAM initialization is ongoing	R

Bit	Symbol	Function	R/W
31:4	—	These bits are read as 0.	R

The Global Status Register indicates the global status of the CANFD module.

GRSTSTS bit (Global Reset Status)

The GRSTSTS bit indicates the status of Global CANFD module Reset mode.

This bit is set automatically when the CANFD module enters GL_RESET mode. When the mode changes from GL_RESET mode to GL_SLEEP mode, this bit remains set.

This bit is cleared automatically when the CANFD module exits the GL_RESET mode.

GHLTSTS bit (Global Halt Status)

The GHLTSTS bit indicates the status of Global CANFD module Halt mode.

This bit is set automatically when the CANFD module enters GL_HALT mode.

This bit is cleared automatically when the CANFD module exits the GL_HALT mode.

GSLPSTS bit (Global Sleep Status)

The GSLPSTS bit indicates the status of Global CANFD module Sleep mode.

This bit is set automatically when the CANFD module enters GL_SLEEP mode.

This bit is cleared automatically when the CANFD module exits the GL_SLEEP mode.

GRAMINIT bit (Global RAM Initialization)

The GRAMINIT bit indicates the status of Global CANFD module RAM initialization.

This bit is set automatically when the CANFD module enters GL_SLEEP mode after a hardware reset.

This bit is cleared automatically when the CANFD module completed RAM initialization.

This bit is cleared when the test_mode input port is set to 1.

28.2.15 CFDGERFL : Global Error Flag Register

Base address: CANFD_B = 0x400B_0000

Offset address: 0x0020

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	EEF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
0	DEF	DLC Error Flag 0: DLC error not detected 1: DLC error detected	R/W
1	MES	Message Lost Error Status 0: Message lost error not detected 1: Message lost error detected	R
2	THLES	TX History List Entry Lost Error Status 0: TX history list entry lost error not detected 1: TX history list entry lost error detected	R

Bit	Symbol	Function	R/W
3	CMPOF*1	CANFD Message Payload Overflow Flag 0: CANFD message payload overflow not detected 1: CANFD message payload overflow detected	R/W
4	—	This bit is read as 0. The write value should be 0.	R
5	—	This bit is read as 0. The write value should be 0.	R
6	—	This bit is read as 0. The write value should be 0.	R
15:7	—	These bits are read as 0. The write value should be 0.	R/W
16	EEF0	ECC Error Flag 0: ECC error not detected during TX-SCAN 1: ECC error detected during TX-SCAN	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. This bit is not available in the classical CAN function.

The Global Error Flag register indicates the detection of global errors.

DEF bit (DLC Error Flag)

The DEF bit indicates the error status of the DLC.

Do not write to this bit when the CANFD module is in GL_SLEEP or GL_RESET mode. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

This bit is set automatically when a DLC error is detected in a received frame.

If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set

The bit is cleared by writing 0 to it.

This bit is cleared automatically in GL_RESET mode.

MES bit (Message Lost Error Status)

The MES bit indicates status of the message lost error.

This bit is set automatically when a FIFO message lost error is detected.

This bit is cleared automatically when:

- All FIFO message lost flags are cleared
- The CANFD module is in GL_RESET mode.

THLES bit (TX History List Entry Lost Error Status)

The THLES bit indicates status of the TX history list entry lost error.

This bit is set automatically when a TX history list entry lost error is detected.

This bit is cleared automatically when:

- All TX history list entry lost flags are cleared
- The CANFD module is in GL_RESET mode.

CMPOF bit (CANFD Message Payload Overflow Flag)

The CMPOF bit is set automatically when a CANFD message payload overflow is detected on at least one channel.

Do not write to this bit when the CANFD module is in GL_SLEEP or GL_RESET mode.

This bit is cleared by writing 0 to it. Writing 1 to this bit has no effect.

If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

This bit is cleared automatically in GL_RESET mode.

Note: This bit is not available in the classical CAN function

EEF0 bit (ECC Error Flag)

The EEF0 bit specifies whether an ECC error has occurred on Channel 0.

Do not write to this bit when the CANFD module is in GL_SLEEP or GL_RESET mode. Writing 1 to this bit has no effect.

If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

The bit is cleared by writing 0 to it. This bit is cleared automatically in GL_RESET mode.

28.2.16 CFDGTINTSTS : Global TX Interrupt Status Register

Base address: CANFD_B = 0x400B_0000

Offset address: 0x00A4

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	THIF0	CFTIF0	TQIF0	TAI0	TSIF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TSIF0	TX Successful Interrupt Flag 0: Channel n TX Successful Interrupt flag not set 1: Channel n TX Successful Interrupt flag set	R
1	TAI0	TX Abort Interrupt Flag 0: Channel n TX Abort Interrupt flag not set 1: Channel n TX Abort Interrupt flag set	R
2	TQIF0	TX Queue Interrupt Flag 0: Channel n TX Queue Interrupt flag not set 1: Channel n TX Queue Interrupt flag set	R
3	CFTIF0	COM FIFO TX Mode Interrupt Flag 0: Channel n COM FIFO TX Mode Interrupt flag not set 1: Channel n COM FIFO TX Mode Interrupt flag set	R
4	THIF0	TX History List Interrupt 0: Channel n TX History List Interrupt flag not set 1: Channel n TX History List Interrupt flag set	R
31:5	—	These bits are read as 0.	R

The Global TX Interrupt Status register indicates the detection of transmit specific interrupts.

TSIF0 bit (TX Successful Interrupt Flag)

The TSIF0 bit is set to 1 when the TX Successful Interrupt flag of the related channel is set (when the interrupt is enabled). This bit is cleared automatically:

- When the related TX MB Result Status bits are cleared (when the interrupt enable is disabled)
- When in GL_RESET or CH_RESET mode.

TAI0 bit (TX Abort Interrupt Flag)

The TAIF0 bit is set to 1 when the TX Abort Interrupt flag of the related channel is set (when the interrupt is enabled).

This bit is cleared automatically:

- When the related TX MB Result Status bits are cleared (when the interrupt enable is disabled)
- When in GL_RESET or CH_RESET mode.

TQIF0 bit (TX Queue Interrupt Flag)

The TQIF0 bit is set to 1 when the TX Queue Interrupt flag of the related channel is set (when the interrupt is enabled). This bit is cleared automatically:

- When the related TX Queue Interrupt flag is cleared (when the interrupt is enable disabled)
- When in GL_RESET or CH_RESET mode.

CFTIF0 bit (COM FIFO TX Mode Interrupt Flag)

The CFTIF0 bit is set to 1 when the related COM TX FIFO Mode Interrupt flag (CFDCFSTS.CFTXIF) is set (when the interrupt is enabled).

This bit is cleared automatically:

- When the related COM TX FIFO Mode Interrupt flag (CFDCFSTS.CFTXIF) is cleared (when the interrupt enable is disabled)
- When in GL_RESET or CH_RESET mode.

THIF0 bit (TX History List Interrupt)

The THIF0 bit is set to 1 when the related TX History List Interrupt flag (CFDTHLSTS.THLIF) is set (when the interrupt is enabled).

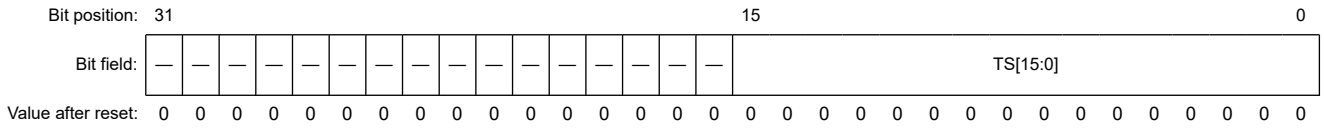
This bit is cleared automatically:

- When the related TX History List Interrupt flag (CFDTHLSTS.THLIF) is cleared (when the interrupt enable is disabled)
- When in GL_RESET or CH_RESET mode.

28.2.17 CFDGTSC : Global Timestamp Counter Register

Base address: CANFD_B = 0x400B_0000

Offset address: 0x0024



Bit	Symbol	Function	R/W
15:0	TS[15:0]	Timestamp value	R
31:16	—	These bits are read as 0.	R

The Global Timestamp Counter register stores the timestamp based on the selected configuration.

TS[15:0] bits (Timestamp value)

The Timestamp value is stored in the Global Timestamp Counter register based on the configuration of TSSS, TSBTCS and TSP. The accuracy of the timestamp counter cannot be guaranteed when transitioning to halt state.

The Timestamp value is stored in this register based on the configuration of TSSS, TSBTCS and TSP.

Do not write to bits TS[15:0] when the CANFD module is in GL_RESET or GL_SLEEP mode.

The TS[15:0] bits are cleared automatically in GL_RESET mode.

28.2.18 CFDGAFLECTR : Global Acceptance Filter List Entry Control Register

Base address: CANFD_B = 0x400B_0000

Offset address: 0x0028

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

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

Bit	Symbol	Function	R/W
0	AFLPN	Acceptance Filter List Page Number Select an Acceptance Filter List page	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
8	AFLDAE	Acceptance Filter List Data Access Enable 0: Acceptance Filter List data access disabled 1: Acceptance Filter List data access enabled	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

The Global Acceptance Filter List Entry Control Register is used to select the Global Acceptance Filter List page for reading or writing entries into the Global Acceptance Filter List.

AFLPN bit (Acceptance Filter List Page Number)

The AFLPN bit select the page number to access the desired RAM area of the Acceptance Filter List. Acceptance Filter List page consists of 16 Acceptance Filter List entries.

Read/write accesses to the Acceptance Filter List can only be performed through a fixed window.

Do not write to these bits when the CANFD module is in GL_SLEEP mode. Enter only the values between 0 and 1, inclusive.

AFLDAE bit (Acceptance Filter List Data Access Enable)

The AFLDAE bit prevents write access to the Acceptance Filter List when cleared after configuration of the Acceptance Filter List.

Data can be read from the Acceptance Filter List independent of the status of this bit.

Do not write to this bit when the CANFD module is in GL_SLEEP mode. Set this bit to enable write access for the Acceptance Filter List.

28.2.19 CFDGAFLCFG : Global Acceptance Filter List Configuration Register

Base address: CANFD_B = 0x400B_0000

Offset address: 0x002C

Bit position:	31										21										16									0
Bit field:	—	—	—	—	—	—	—	—	—	—	RNC0[5:0]						—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
21:16	RNC0[5:0]	Rule Number Number of rules dedicated to channel 0	R/W
31:22	—	These bits are read as 0. The write value should be 0.	R/W

The Global Acceptance Filter List Configuration Register is used to define the number of rules for entries in the Acceptance Filter List.

The total number of available entries in the Acceptance Filter List is 32.

RNC0[5:0] bits (Rule Number)

The RNC0[5:0] bits define the number of rules in the Acceptance Filter List for channel n.

Only write to these bits when the CANFD module is in GL_RESET mode. These bits can set to 6 bits for 32 rules.

28.2.20 CFDGAFLIDr : Global Acceptance Filter List ID Registers (r = 1 to 16)

Base address: CANFD_B = 0x400B_0000

Offset address: 0x0120 + 0x0010 × (r - 1)

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

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

Bit	Symbol	Function	R/W
28:0	GAFLID[28:0]	Global Acceptance Filter List Entry ID Field ID part of the Global Acceptance Filter List entry	R/W
29	GAFLLB	Global Acceptance Filter List Entry Loopback Configuration 0: Global Acceptance Filter List entry ID for acceptance filtering with attribute RX 1: Global Acceptance Filter List entry ID for acceptance filtering with attribute TX	R/W
30	GAFLRTR	Global Acceptance Filter List Entry RTR Field 0: Data frame 1: Remote frame	R/W
31	GAFLIDE	Global Acceptance Filter List Entry IDE Field 0: Standard identifier of rule entry ID is valid for acceptance filtering 1: Extended identifier of rule entry ID is valid for acceptance filtering	R/W

The Global Acceptance Filter List ID Registers are used to configure the ID field for the rules of entries in the Global Acceptance Filter List.

GAFLID[28:0] bits (Global Acceptance Filter List Entry ID Field)

The GAFLID[28:0] bits represent the CAN identifier (ID) field of each entry in the Global Acceptance Filter List.

Do not write to these bits when CFDGAFLECTR.AFLDAE bit is 0.

Only write to these bits when the related CANFD channel is in CH_RESET or CH_HALT mode.

GAFLLB bit (Global Acceptance Filter List Entry Loopback Configuration)

The GAFLLB bit selects whether entry in the Global Acceptance Filter List gets the attribute RX or TX.

This attribute determines the validity of the entry in Mirror mode, Loopback test mode, and during standard (non-loopback) reception. See section 28.5.5. Loopback Modes for detailed description of the validity of the Global Acceptance Filter List entry depending on transmitter/receiver case, the type of loopback mode, and RX/TX attribute.

Do not write to this bit when CFDGAFLECTR.AFLDAE bit is 0.

Only write to this bit when the related CANFD channel is in CH_RESET or CH_HALT mode.

GAFLRTR bit (Global Acceptance Filter List Entry RTR Field)

The GAFLRTR bit allows the configuration of the specified frame format (data frame or remote frame) for each entry of the Global Acceptance Filter List. For each rule entry in a CAN channel, the acceptance filter process compares this bit against the RTR bit of the received CAN message.

Do not write to this bit when CFDGAFLECTR.AFLDAE bit is 0.

Only write to this bit when the related CANFD channel is in CH_RESET or CH_HALT mode.

GAFLIDE bit (Global Acceptance Filter List Entry IDE Field)

The GAFLIDE bit allows the configuration of the ID format (standard ID or extended ID) for each entry in the Global Acceptance Filter List. For each rule entry in a CAN channel, the acceptance filter process compares this bit against the IDE bit of the received CAN message.

Do not write to this bit when CFDGAFLECTR.AFLDAE bit is 0.

Only write to this bit when the related CANFD channel is in CH_RESET or CH_HALT mode.

28.2.21 CFDGAFLMr : Global Acceptance Filter List Mask Registers (r = 1 to 16)

Base address: CANFD_B = 0x400B_0000

Offset address: 0x0124 + 0x0010 × (r - 1)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	GAFLIDEM	GAFLRTRM	GAFLIFL1	GAFLIDM[28:16]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
28:0	GAFLIDM[28:0]	Global Acceptance Filter List ID Mask Field Global Acceptance Filter List Mask field bits for ID field	R/W
29	GAFLIFL1	Global Acceptance Filter List Information Label 1 Global Acceptance Filter List information label bit 1	R/W
30	GAFLRTRM	Global Acceptance Filter List Entry RTR Mask 0: RTR bit is not used for ID matching 1: RTR bit is used for ID matching	R/W
31	GAFLIDEM	Global Acceptance Filter List IDE Mask 0: IDE bit is not used for ID matching 1: IDE bit is used for ID matching	R/W

The Global Acceptance Filter List Mask Registers are used to configure the Mask field of each rule for entries in the Global Acceptance Filter List.

GAFLIDM[28:0] bits (Global Acceptance Filter List ID Mask Field)

GAFLIDM[28:0] bits are the filter mask bits for the related bits in the CAN Identifier field of each Global Acceptance Filter List entry.

0	Corresponding STD-ID/EXT-ID bit is not used for ID matching
1	Corresponding STD-ID/EXT-ID bit is used for ID matching

Do not write to these bits when CFDGAFLECTR.AFLDAE bit is 0.

Only write to these bits when the related CANFD channel is in CH_RESET or CH_HALT mode.

GAFLIFL1 bit (Global Acceptance Filter List Information Label 1)

The GAFLIFL1 bit allows the configuration of a 2-bit information label to be attached to a received message accepted by the associated entry in the Global Acceptance Filter List. This bit is a MSB bit of an information label.

Do not write to this bit when CFDGAFLECTR.AFLDAE bit is 0.

Only write to this bit when the related CANFD channel is in CH_RESET or CH_HALT mode.

This bit is stored in the Information Label Field [1] (CFDRMFDSTSb.RMIFL [1], CFDRFFDSTSb.RFIFL [1], CFDCFFDCSTS.CFIFL [1]) of the storage location of an incoming message.

GAFLRTRM bit (Global Acceptance Filter List Entry RTR Mask)

The GAFLRTRM bit allows the configuration of the RTR mask bit for each entry in the Global Acceptance Filter List.

Do not write to this bit when CFDGAFLECTR.AFLDAE bit is 0.

Only write to this bit when the related CANFD channel is in CH_RESET or CH_HALT mode.

GAFLIDEM bit (Global Acceptance Filter List IDE Mask)

The GAFLIDEM bit allows the configuration of the IDE mask bit for each entry in the Global Acceptance Filter List.

When the IDE mask bit is 0, the ID comparison depends on the received IDE bit.

If the received IDE bit is 0, the STD-ID comparison takes place.

If the received IDE bit is 1, the EXT-ID comparison takes place.

Do not write to this bit when CFDGAFLECTR.AFLDAE bit is 0.

Only write to this bit when the related CANFD channel is in CH_RESET or CH_HALT mode.

28.2.22 CFDGAFLP0r : Global Acceptance Filter List Pointer 0 Registers (r = 1 to 16)

Base address: CANFD_B = 0x400B_0000

Offset address: 0x0128 + 0x0010 × (r - 1)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	GAFLPTR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	GAFLRMV	—	—	GAFLRMDP[4:0]					GAFLIFL0	—	—	—	GAFLDLC[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	GAFLDLC[3:0]	Global Acceptance Filter List DLC Field Minimum number of data bytes in a data frame required for acceptance	R/W
6:4	—	These bits are read as 0. The write value should be 0.	R/W
7	GAFLIFL0	Global Acceptance Filter List Information Label 0	R/W
12:8	GAFLRMDP[4:0]	Global Acceptance Filter List RX Message Buffer Direction Pointer RX message buffer number for storage of received messages	R/W
14:13	—	These bits are read as 0. The write value should be 0.	R/W
15	GAFLRMV	Global Acceptance Filter List RX Message Buffer Valid 0: Single message buffer direction pointer is invalid 1: Single message buffer direction pointer is valid	R/W
31:16	GAFLPTR[15:0]	Global Acceptance Filter List Pointer	R/W

The Global Acceptance Filter List Pointer 0 Registers are used to configure the data length code (DLC), software pointer, single message buffer select, and message buffer direction pointer for each rule entry in the Global Acceptance Filter List.

GAFLDLC[3:0] bits (Global Acceptance Filter List DLC Field)

The GAFLDLC[3:0] bits allow the configuration of a minimum data length code (DLC) value for a message to be accepted by the associated entry in the Global Acceptance Filter List (automatic DLC filter function).

DLC filter process is only passed if the DLC value of the message accepted by an entry in the Global Acceptance Filter List is equal to or higher than the DLC value configured for this associated Global Acceptance Filter List entry. Automatic DLC filter function is disabled for the corresponding rule entry when this field is set to 0.

Table 28.4 shows DLC value that can be configured.

Table 28.4 Configuration of DLC value

Format	DLC[3]	DLC[2]	DLC[1]	DLC[0]	Description
CAN and CANFD	0	0	0	0	DLC of received message = 0 or more (DLC filter check is disabled)
CAN and CANFD	0	0	0	1	DLC of received message = 1 or more
CAN and CANFD	0	0	1	0	DLC of received message = 2 or more
CAN and CANFD	0	0	1	1	DLC of received message = 3 or more
CAN and CANFD	0	1	0	0	DLC of received message = 4 or more
CAN and CANFD	0	1	0	1	DLC of received message = 5 or more
CAN and CANFD	0	1	1	0	DLC of received message = 6 or more
CAN and CANFD	0	1	1	1	DLC of received message = 7 or more
CAN	1	x	x	x	DLC of received message = 8 or more
CANFD	1	0	0	0	DLC of received message = 8 or more*1
CANFD	1	0	0	1	DLC of received message = 12 or more*1
CANFD	1	0	1	0	DLC of received message = 16 or more*1
CANFD	1	0	1	1	DLC of received message = 20 or more*1
CANFD	1	1	0	0	DLC of received message = 24 or more*1
CANFD	1	1	0	1	DLC of received message = 32 or more*1
CANFD	1	1	1	0	DLC of received message = 48 or more*1
CANFD	1	1	1	1	DLC of received message = 64*1

Note 1. This setting is not available in the classical CAN function.

Do not write to these bits when CFDGAFLECTR.AFLDAE bit is 0.

Only write to these bits when the related CANFD channel is in CH_RESET or CH_HALT mode.

GAFLIFL0 bit (Global Acceptance Filter List Information Label 0)

The GAFLIFL0 bit allows the configuration of a 2-bit information label that can be attached to a received message accepted by the related Global Acceptance Filter List entry. This bit is a LSB bit of an information label.

You cannot write to this bit when CFDGAFLECTR.AFLDAE bit is 0.

Only write to the bit when the related CANFD channel is in CH_RESET or CH_HALT mode.

This bit is stored in Information Label Field[0] (CFDRMFDSTSb.RMIFL[0], CFDRFFDSTSb.RFIFL[0], CFDCFFDCSTS.CFIFL[0]) of the storage location of an incoming message.

GAFLRMDP[4:0] bits (Global Acceptance Filter List RX Message Buffer Direction Pointer)

The GAFLRMDP[4:0] bits allow the configuration of a single reception message buffer as the destination target for a received message that passes the acceptance check of the related Global Acceptance Filter List entry. The value entered is the single destination message buffer number.

Do not write to these bits when CFDGAFLECTR.AFLDAE bit is 0.

Only write to these bits when the related CANFD channel is in CH_RESET or CH_HALT mode.

CFDRMNB.NRXMB[5:0] is the value entered in the RX Message Buffer Number Register to configure the number of RX message buffers. The value to be entered in CFDGAFLP0r.GAFLRMDP[4:0] bits should only be between 0x00 and CFDMNB.NMB[5:0] to 1 less.

If CFDRMNB.NRXMB[5:0] = 0x00, the GAFLRMV bit should be configured as 0.

GAFLRMV bit (Global Acceptance Filter List RX Message Buffer Valid)

The GAFLRMV bit allows the enabling or disabling of a single reception message buffer as the target for a received message that passes the acceptance check of the related Global Acceptance Filter List entry.

Do not write to these bits when CFDGAFLECTR.AFLDAE bit is 0.

Only write to these bits when the related CANFD channel is in CH_RESET or CH_HALT mode.

GAFLPTR[15:0] bits (Global Acceptance Filter List Pointer)

The GAFLPTR[15:0] bits allow the configuration of a 16-bit pointer to be attached to a received message accepted by the related Global Acceptance Filter List entry. The pointer is added during message storage in the Message Buffer area and can be used by the application as a support function. The pointer information can be used for example, to support PDU Identifier allocation for the received message in AUTOSAR systems.

Do not write to these bits when CFDGAFLECTR.AFLDAE bit is 0.

Only write to these bits when the related CANFD channel is in CH_RESET or CH_HALT mode.

28.2.23 CFDGAFLP1r : Global Acceptance Filter List Pointer 1 Registers (r = 1 to 16)

Base address: CANFD_B = 0x400B_0000

Offset address: 0x012C + 0x0010 × (r - 1)

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	GAFL FDP8	—	—	—	—	—	—	GAFL FDP1	GAFL FDP0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	GAFLFDP0	Global Acceptance Filter List FIFO Direction Pointer FIFO direction pointer bits for received message storage 0: Disable RX FIFO 0 as target for reception 1: Enable RX FIFO 0 as target for reception	R/W
1	GAFLFDP1	Global Acceptance Filter List FIFO Direction Pointer FIFO direction pointer bits for received message storage 0: Disable RX FIFO 1 as target for reception 1: Enable RX FIFO 1 as target for reception	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
8	GAFLFDP8	Global Acceptance Filter List FIFO Direction Pointer FIFO direction pointer bits for received message storage 0: Disable Common FIFO as target for reception 1: Enable Common FIFO as target for reception	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

The Global Acceptance Filter List Pointer 1 registers are used to configure the FIFO direction pointer fields in each Rule Entry of the Global Acceptance Filter List.

GAFLFDP8, GAFLFDP1, GAFLFDP0 bits (Global Acceptance Filter List FIFO Direction Pointer)

These bits allow the configuration of FIFO Buffers as the target for a received message passing the acceptance check of the related Global Acceptance Filter List entry. Each bit of the GAFLFDP8, GAFLFDP1, GAFLFDP0 is configuring a dedicated FIFO.

Users cannot write to these bits when CFDGAFLECTR.AFLDAE bit is 0.

For storage in Common FIFO, target for reception can only be those Common FIFO Buffers that are configured as RX FIFO.

Only write to these bits when the related CANFD channel is in CH_RESET or CH_HALT mode.

Users should only configure up to 2 destination FIFO Buffers or 1 destination FIFO Buffers plus one RX Message Buffer.

28.2.24 CFDRMNB : RX Message Buffer Number Register

Base address: CANFD_B = 0x400B_0000

Offset address: 0x0030

Bit position:	31																				10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RMPLS[2:0]			—	—	NRXMB[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	NRXMB[5:0]	Number of RX Message Buffers	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
10:8	RMPLS[2:0]	Reception Message Buffer Payload Data Size 0 0 0: 8 bytes 0 0 1: 12 bytes 0 1 0: 16 bytes 0 1 1: 20 bytes 1 0 0: 24 bytes 1 0 1: 32 bytes 1 1 0: 48 bytes 1 1 1: 64 bytes	R/W
31:11	—	These bits are read as 0. The write value should be 0.	R/W

The RX Message Buffer Number register is used to configure the total number of RX message buffers allocated to channels.

NRXMB[5:0] bits (Number of RX Message Buffers)

The NRXMB[5:0] bits are used to configure the number of RX message buffers.

Only write to these bits when the CANFD module is in GL_RESET mode.

Enter only values between 0 and 32 inclusive, with 0x00 indicating that no RX message buffer is allocated.

RMPLS[2:0] bits (Reception Message Buffer Payload Data Size)

The RMPLS[2:0] bits are used to configure the message buffer payload data size.

Only write to these bits when the CANFD module is in GL_RESET mode.

28.2.25 CFDRMND : RX Message Buffer New Data Register

Base address: CANFD_B = 0x400B_0000

Offset address: 0x0034

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

Bit	Symbol	Function	R/W
31:0	RMNS[31:0]	RX Message Buffer New Data Status 0: New data not stored in corresponding RX message buffer 1: New data stored in corresponding RX message buffer	R/W

The RX Message Buffer New Data Status Register specifies the new data storage status of the RX message buffers.

RMNS[31:0] bits (RX Message Buffer New Data Status)

The RMNS[31:0] bits indicate the status of new data for the corresponding RX message buffer. RMNS bit [0] corresponds to RX message buffer [0] and so on.

The bit position of CFDRMND corresponds to the buffer number of RXMB.

Do not write to these bits when the CANFD module is in GL_RESET or GL_SLEEP mode. Writing 1 has no effect.

These bits cannot be cleared when message storage in the corresponding RX message buffer is in progress.
Do not use the bit clear instruction to clear these bits. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

These bits are set automatically when storage of new messages are in the corresponding RX message buffer. These bits are cleared by writing 0. These bits are cleared automatically when the CANFD module is in GL_RESET mode.

When CFDRMNB.RMPLS = 000b (maximum 8 bytes payload), the duration of message storage is 6 PCLKB cycles.

When CFDRMNB.RMPLS > 000b, the duration of message storage is 6 PCLKB cycles + 1 for each 4 bytes (maximum of 20 PCLKB cycles for 64 bytes).

Note: This feature is not available in the classical CAN function.

28.2.26 CFDRFCCa : RX FIFO Configuration/Control Registers a (a = 0 to 1)

Base address: CANFD_B = 0x400B_0000

Offset address: 0x003C + 0x04 × a

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RFIGCV[2:0]			RFIM	—	RFDC[2:0]			—	RFPLS[2:0]			—	—	RFIE	RFE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RFE	RX FIFO Enable 0: FIFO disabled 1: FIFO enabled	R/W

Bit	Symbol	Function	R/W
1	RFIE	RX FIFO Interrupt Enable 0: FIFO interrupt generation disabled 1: FIFO interrupt generation enabled	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
6:4	RFPLS[2:0]*1	Rx FIFO Payload Data Size Configuration 0 0 0: 8 bytes 0 0 1: 12 bytes 0 1 0: 16 bytes 0 1 1: 20 bytes 1 0 0: 24 bytes 1 0 1: 32 bytes 1 1 0: 48 bytes 1 1 1: 64 bytes	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
10:8	RFDC[2:0]	RX FIFO Depth Configuration 0 0 0: FIFO Depth = 0 message 0 0 1: FIFO Depth = 4 messages 0 1 0: FIFO Depth = 8 messages 0 1 1: FIFO Depth = 16 messages 1 0 0: FIFO Depth = 32 essages 1 0 1: FIFO Depth = 48 messages 1 1 0: Reserved 1 1 1: Reserved	R/W
11	—	This bit is read as 0. The write value should be 0.	R
12	RFIM	RX FIFO Interrupt Mode 0: Interrupt generated when RX FIFO counter reaches RFIGCV value from values smaller than RFIGCV 1: Interrupt generated at the end of every received message storage	R/W
15:13	RFIGCV[2:0]	RX FIFO Interrupt Generation Counter Value 0 0 0: Interrupt generated when FIFO is 1/8th full 0 0 1: Interrupt generated when FIFO is 1/4th full 0 1 0: Interrupt generated when FIFO is 3/8th full 0 1 1: Interrupt generated when FIFO is 1/2 full 1 0 0: Interrupt generated when FIFO is 5/8th full 1 0 1: Interrupt generated when FIFO is 3/4th full 1 1 0: Interrupt generated when FIFO is 7/8th full 1 1 1: Interrupt generated when FIFO is full	R/W
16	—	These bits are read as 0. The write value should be 0.	R
31:17	—	These bits are read as 0. The write value should be 0.	R

Note 1. These bits are not available in the classical CAN function.

The RX FIFO Configuration/Control Registers are used to configure and control the two RX FIFOs.

RFE bit (RX FIFO Enable)

The RFE bit enables the FIFO. When this bit is set to 0, the RX FIFO is cleared to empty.

Only write to this bit when the CANFD module is in GL_HALT or GL_OPERATION mode.

This bit can only be set if the configured FIFO depth is greater than 0x000 (CFDRFCCa.RFDC > 0x000) and less than 0x110.

Set the RFE bit with a separate write access to the CFDRFCCa register, after all the other bits in the CFDRFCCa register are set.

This bit is cleared automatically when the CANFD module is in GL_RESET mode.

RFIE bit (RX FIFO Interrupt Enable)

The RFIE bit enables generation of the FIFO interrupt.

Do not write to this bit when the CANFD module is in GL_SLEEP mode.

RFPLS[2:0] bits (Rx FIFO Payload Data Size Configuration)

The RFPLS[2:0] bits define the message data payload allocation in the RAM.

This is the maximum number of bytes which can be received by this FIFO.

Only write to these bits when the CANFD module is in GL_RESET mode.

Note: These bits are not available in the classical CAN function.

RFDC[2:0] bits (RX FIFO Depth Configuration)

The RFDC[2:0] bits select the depth of the FIFO in terms of the number of messages. If the FIFO depth is configured to 0 messages, the FIFO cannot be used.

Only write to these bits when the CANFD module is in GL_RESET mode.

RFIM bit (RX FIFO Interrupt Mode)

The RFIM bit selects the interrupt generation condition for the FIFO.

Do not write to this bit when the CANFD module is in GL_SLEEP mode.

Only write to this bit when the CANFD module is in GL_RESET mode.

RFIGCV[2:0] bits (RX FIFO Interrupt Generation Counter Value)

The RFIGCV[2:0] bits select the counter value of the FIFO for generation of FIFO interrupts. These values represent fractions of the FIFO depth for which an interrupt is generated.

Do not write to these bits when the CANFD module is in GL_SLEEP mode.

The setting of the RFIGCV[2:0] bits should be synchronized with the RFDC[2:0] bits.

Only write to these bits when the CANFD module is in GL_RESET mode.

28.2.27 CFDRFSTSa : RX FIFO Status Registers a (a = 0 to 1)

Base address: CANFD_B = 0x400B_0000

Offset address: 0x0044 + 0x04 × a

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

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

Bit	Symbol	Function	R/W
0	RFEMP	RX FIFO Empty 0: FIFO not empty 1: FIFO empty	R
1	RFFLL	RX FIFO Full 0: FIFO not full 1: FIFO full	R
2	RFMLT	RX FIFO Message Lost 0: No message lost in FIFO 1: FIFO message lost	R/W
3	RFIF	RX FIFO Interrupt Flag 0: FIFO interrupt condition not satisfied 1: FIFO interrupt condition satisfied	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
13:8	RFMC[5:0]	RX FIFO Message Count Number of messages stored in FIFO	R
31:14	—	These bits are read as 0. The write value should be 0.	R/W

The RX FIFO Status Registers show the status of messages stored in the corresponding FIFO buffers.

RFEMP bit (RX FIFO Empty)

The RFEMP bit is set automatically when:

- The RFMC bit is 0
- RX FIFO is disabled by setting the CFDRFCCa.RFE bit to 0
- The CANFD module is in GL_RESET mode.

The RFEMP bit is cleared automatically when the first message is stored in the RX FIFO buffer.

RFFLL bit (RX FIFO Full)

The RFFLL bit is set automatically when the number of CAN messages stored in the FIFO buffer matches the configured FIFO depth.

The RFFLL is cleared automatically when:

- The number of CAN messages stored in the FIFO buffer is less than the configured FIFO depth
- RX FIFO is disabled by setting the CFDRFCCa.RFE bit to 0
- The CANFD module is in GL_RESET mode.

RFMLT bit (RX FIFO Message Lost)

Only write to the RFMLT bit when CANFD module is in GL_HALT or GL_OPERATION mode. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

This bit is set automatically whenever a message is lost due to attempted storage when the FIFO buffer is already full. If a set from the CAN channel occurs simultaneously with a clear by a write access, then the bit is set.

The bit is cleared:

- By writing 0 to it
- When the CANFD module is in GL_RESET mode.

RFIF bit (RX FIFO Interrupt Flag)

The RFIF bit is set automatically when the configured interrupt condition is satisfied. This bit is not automatically cleared when the RX FIFO buffer is disabled.

Only write to this bit when the CANFD module is in GL_HALT or GL_OPERATION mode. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

If a set from the CAN channel occurs simultaneously with a clear by a write access, then this bit is set.

The bit is cleared by writing 0 to it. The bit is also cleared when CANFD module is in GL_RESET mode.

RFMC[5:0] bits (RX FIFO Message Count)

The RFMC[5:0] bits indicate the number of CAN messages stored in the RX FIFO buffer that can be read by the CPU.

These bits are cleared automatically when the FIFO is disabled and when the CANFD module is in GL_RESET mode.

28.2.28 CFDRFPCTRa : RX FIFO Pointer Control Registers a (a = 0 to 1)

Base address: CANFD_B = 0x400B_0000

Offset address: 0x004C + 0x04 × a

Bit position:	31																						7							0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RFPC[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	RFPC[7:0]	RX FIFO Pointer Control Increments read pointer of the corresponding RX FIFO buffers	W
31:8	—	The write value should be 0.	W

The RX FIFO Pointer Control Registers can be used to increment the read pointer of the corresponding RX FIFO buffers.

RFPC bits (RX FIFO Pointer Control)

When the value 0xFF is written to the RFPC bits, the pointer of the corresponding RX FIFO buffer is moved to the next FIFO entry. Only write 0xFF to these registers when the corresponding RX FIFO buffer is enabled and not empty.

The read value from these bits is always 0x00.

Only write to these bits when the CANFD module is in GL_HALT or GL_OPERATION mode.

Do not write to the RX FIFO Pointer Control registers when DMA is enabled.

28.2.29 CFDCFCC : Common FIFO Configuration/Control Register

Base address: CANFD_B = 0x400B_0000

Offset address: 0x0054

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CFITT[7:0]								CFDC[2:0]			—	—	—	CFTML[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CFIGCV[2:0]			CFIM	CFITR	CFITSS	—	CFM	—	CFPLS[2:0]			—	CFTXIE	CFRXIE	CFE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CFE	Common FIFO Enable 0: FIFO disabled 1: FIFO enabled	R/W
1	CFRXIE	Common FIFO RX Interrupt Enable 0: FIFO interrupt generation disabled for Frame RX 1: FIFO interrupt generation enabled for Frame RX	R/W
2	CFTXIE	Common FIFO TX Interrupt Enable 0: FIFO interrupt generation disabled for Frame TX 1: FIFO interrupt generation enabled for Frame TX	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
6:4	CFPLS[2:0]*1	Common FIFO Payload Data Size Configuration 0 0 0: 8 bytes 0 0 1: 12 bytes 0 1 0: 16 bytes 0 1 1: 20 bytes 1 0 0: 24 bytes 1 0 1: 32 bytes 1 1 0: 48 bytes 1 1 1: 64 bytes	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
8	CFM	Common FIFO Mode 0: RX FIFO mode 1: TX FIFO mode	R/W
9	—	This bit is read as 0. The write value should be 0.	R/W
10	CFITSS	Common FIFO Interval Timer Source Select 0: Reference clock (× 1 / × 10 period) 1: Bit time clock of related channel (FIFO is linked to fixed channel)	R/W
11	CFITR	Common FIFO Interval Timer Resolution 0: Reference clock period × 1 1: Reference clock period × 10	R/W
12	CFIM	Common FIFO Interrupt Mode 0: RX FIFO mode: RX interrupt generated when Common FIFO counter reaches CFIGCV value from a lower value TX FIFO mode: TX interrupt generated when Common FIFO transmits the last message successfully 1: RX FIFO mode: RX interrupt generated at the end of every received message storage TX FIFO mode: interrupt generated for every successfully transmitted message	R/W
15:13	CFIGCV[2:0]	Common FIFO Interrupt Generation Counter Value 0 0 0: Interrupt generated when FIFO is 1/8th full 0 0 1: Interrupt generated when FIFO is 1/4th full 0 1 0: Interrupt generated when FIFO is 3/8th full 0 1 1: Interrupt generated when FIFO is 1/2 full 1 0 0: Interrupt generated when FIFO is 5/8th full 1 0 1: Interrupt generated when FIFO is 3/4th full 1 1 0: Interrupt generated when FIFO is 7/8th full 1 1 1: Interrupt generated when FIFO is full	R/W
17:16	CFTML[1:0]	Common FIFO TX Message Buffer Link Transmission scan link position of the corresponding channel	R/W
20:18	—	These bits are read as 0. The write value should be 0.	R/W
23:21	CFDC[2:0]	Common FIFO Depth Configuration 0 0 0: FIFO Depth = 0 message 0 0 1: FIFO Depth = 4 messages 0 1 0: FIFO Depth = 8 messages 0 1 1: FIFO Depth = 16 messages 1 0 0: FIFO Depth = 32 messages 1 0 1: FIFO Depth = 48 messages 1 1 0: FIFO Depth = Reserved 1 1 1: FIFO Depth = Reserved	R/W
31:24	CFITT[7:0]	Common FIFO Interval Transmission Time Delay the start of transmission from the FIFO if configured in TX mode, delay is a multiple of basic Interval Timer Clock Source unit	R/W

Note 1. These bits are not available in the classical CAN function.

CFE bit (Common FIFO Enable)

The CFE bit enables the FIFO when set. FIFO is disabled when this bit is cleared.

This bit can also be used, by clearing it, to abort transmission from Common FIFO when configured in TX mode, or to stop reception into the Common FIFO in RX mode.

Only write to this bit when the CANFD module is in GL_HALT or GL_OPERATION mode and the related CANFD channel is not in CH_RESET mode for FIFOs configured as TX FIFO.

This bit can only be set if the configured FIFO depth is greater than 0x000 (CFDCFCC.CFDC > 0x000) and less than 0x110 (0x110 > CFDCFCC.CFDC > 0x000).

Set the CFE bit with a separate write access to the CFDCFCC register, after all the other bits in this register are set.

This bit is cleared automatically when the CANFD module is in GL_RESET mode.

This bit is also cleared automatically when the related channel is in CH_RESET mode if the FIFO is configured in TX mode.

CFRXIE bit (Common FIFO RX Interrupt Enable)

The CFRXIE bit enables generation of FIFO interrupts when the interrupt flag is set after reception of a frame in the corresponding FIFO buffer.

Do not write to this bit when the CANFD module is in GL_SLEEP mode.

CFTXIE bit (Common FIFO TX Interrupt Enable)

The CFTXIE bit enables generation of common FIFO interrupts when the interrupt flag is set after transmission of a frame from the corresponding FIFO buffer.

Do not write to this bit when the CANFD module is in GL_SLEEP mode.

CFPLS[2:0] bits (Common FIFO Payload Data Size Configuration)

The CFPLS[2:0] bits define the message data payload allocation in the RAM. This is the maximum number of bytes which can be received or transmitted by the FIFO buffer.

For details, see section 28.6. FIFO Buffers and Normal Message Buffer Configuration.

Only write to this bit when the CANFD module is in GL_RESET mode.

Note: These bits are not available in the classical CAN function.

CFM bit (Common FIFO Mode)

The CFM bit selects the mode of the FIFO. When a hardware reset is applied, all the Common FIFO buffers are configured in RX FIFO mode.

Do not write to these bits in GL_OPERATION or GL_SLEEP mode.

Only write to these bits when the CANFD module is in GL_RESET mode.

CFITSS bit (Common FIFO Interval Timer Source Select)

The CFITSS bit selects the basic clock source for the Interval Transmission Timer.

Do not write to this bit when the CANFD module is in GL_SLEEP mode. In addition, do not write to this bit when the CFE bit is set to 1.

Do not write 1 to this bit when CANFD communication is used.*1

Note: The bit time clock can vary depending on the nominal and data rate bit configuration.

Note 1. This feature is not available in the classical CAN function.

CFITR bit (Common FIFO Interval Timer Resolution)

The CFITR bit selects the resolution of the reference clock for the Interval Transmission Timer (peripheral clock is the source for the reference clock).

Do not write to this bit when the CANFD module is in GL_SLEEP mode. Also, do not write to this bit when the CFE bit is set to 1.

CFIM bit (Common FIFO Interrupt Mode)

The CFIM bit selects the interrupt generation condition for the FIFO buffer.

Do not write to this bit in GL_SLEEP mode.

Only write to this bit when the CANFD module is in GL_RESET mode.

CFIGCV[2:0] bits (Common FIFO Interrupt Generation Counter Value)

The CFIGCV[2:0] bits select the message counter value for the generation of FIFO interrupts. These values represent fractions of the FIFO depth at which the interrupt is to be generated.

Do not write to these bits when the CANFD module is in GL_SLEEP mode.

The setting of these bits should be synchronized with the CFDC[2:0] bits.

Only write to these bits when the CANFD module is in GL_RESET mode.

CFTML[1:0] bits (Common FIFO TX Message Buffer Link)

The CFTML[1:0] bits select the normal transmit message buffer position where the TX FIFO is linked to, for transmission scanning.

Do not write to these bits in GL_OPERATION or GL_SLEEP mode.

Only write to this bit when the CANFD module is in GL_RESET mode.

CFDC[2:0] bits (Common FIFO Depth Configuration)

The CFDC[2:0] bits select the depth of the common FIFO in terms of the number of messages. If the FIFO depth is configured to 0 message, the FIFO cannot be used.

Only write to these bits when the CANFD module is in GL_RESET mode.

CFITT[7:0] bits (Common FIFO Interval Transmission Time)

The CFITT[7:0] bits select the delay in the start of transmission for all messages transmitted from this FIFO buffer when configured in TX mode. The delay is a multiple of the basic interval timer clock source period (reference clock × 1, reference clock × 10, or bit time clock of the related CAN channel).

Do not write to these bits when the CANFD module is in GL_SLEEP mode.

Do not write to these bits when the CFE bit is set to 1.

When CFDGCFG.ITRCP[15:0] = 0x0000, set the CFITT[7:0] bits to 0x0000.

28.2.30 CFDCFSTS : Common FIFO Status Register

Base address: CANFD_B = 0x400B_0000

Offset address: 0x0058

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	CFMC[5:0]						—	—	—	CFTXIF	CFRXIF	CFMLT	CFFLL	CFEMP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	CFEMP	Common FIFO Empty 0: FIFO not empty 1: FIFO empty	R
1	CFFLL	Common FIFO Full 0: FIFO not full 1: FIFO full	R
2	CFMLT	Common FIFO Message Lost 0: Number of message lost in FIFO 1: FIFO message lost	R/W

Bit	Symbol	Function	R/W
3	CFRXIF	Common RX FIFO Interrupt Flag 0: FIFO interrupt condition not satisfied after frame reception 1: FIFO interrupt condition satisfied after frame reception	R/W
4	CFTXIF	Common TX FIFO Interrupt Flag 0: FIFO interrupt condition not satisfied after frame transmission 1: FIFO Interrupt condition satisfied after frame transmission	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
13:8	CFMC[5:0]	Common FIFO Message Count Number of messages stored in FIFO	R
31:14	—	These bits are read as 0. The write value should be 0.	R/W

CFEMP bit (Common FIFO Empty)

The CFEMP bit is set automatically when:

- The CPU has read all messages from the FIFO configured in RX mode
- All messages have been transmitted from the FIFO configured in TX mode
- The FIFO is disabled by setting the CFE bit to 0
- The CANFD module is in GL_RESET mode
- The related CANFD channel is in CH_RESET when FIFO configured in TX mode.

The CFEMP bit is cleared automatically when:

- The first reception message is stored in the FIFO buffer when configured in RX mode
- The first message to be transmitted is stored in the FIFO buffer when configured in TX mode.

CFFLL bit (Common FIFO Full)

The CFFLL bit is set automatically when the number of CAN messages stored in the FIFO matches the configured FIFO depth.

The CFFLL bit is cleared automatically when:

- The number of CAN messages stored in the FIFO is less than the configured FIFO depth
- The FIFO is disabled by setting the CFE bit to 0
- The CANFD module is in GL_RESET mode
- The related CANFD channel is in CH_RESET mode when FIFO buffer is configured in TX mode.

CFMLT bit (Common FIFO Message Lost)

The CFMLT bit is set automatically whenever a message is lost due to attempted storage of a new message when FIFO is already full in RX mode.

If a set from the CAN channel occurs simultaneously with a clear by a write access, then this bit is set.

Only write to this bit when the CANFD module is in GL_HALT or GL_OPERATION mode and the related CANFD channel is not in CH_RESET mode for FIFO configured as TX FIFO. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

The CFMLT bit is cleared:

- By writing 0 to it
- When the CANFD module is in GL_RESET mode
- When the related CANFD channel is in CH_RESET mode if the FIFO buffer is configured in TX mode.

CFRXIF bit (Common RX FIFO Interrupt Flag)

The CFRXIF bit is not cleared automatically if the Common FIFO buffer is disabled.

Only write to this bit when the CANFD module is in GL_HALT or GL_OPERATION mode and the related CANFD channel is not in CH_RESET mode for FIFO configured as TX FIFO. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

This bit is set automatically when the configured interrupt condition is satisfied for Common FIFO buffers when configured in RX mode.

If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.

The CFRXIF bit is cleared:

- By writing 0 to it
- When the CANFD module is in GL_RESET mode

CFTXIF bit (Common TX FIFO Interrupt Flag)

The CFTXIF bit is not cleared automatically if the Common FIFO buffer is disabled.

Only write to this bit when the CANFD module is in GL_HALT or GL_OPERATION mode and the related CANFD channel is not in CH_RESET mode for FIFO buffer configured as TX FIFO. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

This bit is set automatically when the configured interrupt condition is satisfied for Common FIFO buffers configured in TX mode.

If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.

The CFTXIF bit is cleared:

- By writing 0 to it
- When the CANFD module is in GL_RESET mode
- When the related CANFD channel is in CH_RESET mode if the FIFO buffer is configured in TX mode.

CFMC[5:0] bits (Common FIFO Message Count)

The CFMC[5:0] bits indicate the following:

- Number of CAN messages stored by the CPU in the FIFO buffer configured in TX mode pending for transmission
- Number of CAN messages stored in the FIFO buffer configured in RX mode by CANFD module to be read by the CPU

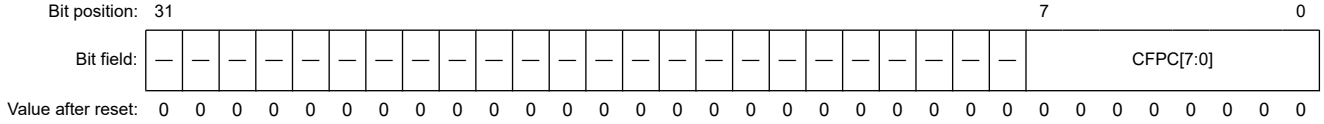
The CFMC[5:0] bits are cleared automatically when:

- The FIFO is disabled
- The CANFD module is in GL_RESET mode
- The related CANFD channel is in CH_RESET mode if the FIFO buffer is configured in TX mode.

28.2.31 CFDCFPCTR : Common FIFO Pointer Control Register

Base address: CANFD_B = 0x400B_0000

Offset address: 0x005C



Bit	Symbol	Function	R/W
7:0	CFPC[7:0]	Common FIFO Pointer Control Increments read or write pointer of the corresponding Common FIFO buffers depending on the mode configuration.	W

Bit	Symbol	Function	R/W
31:8	—	The write value should be 0.	W

The Common FIFO Pointer Control Registers can be used to increment the read or write pointer of the corresponding Common FIFO buffer.

CFPC[7:0] bits (Common FIFO Pointer Control)

When the value 0xFF is written into the CFPC[7:0] bits, the read pointer of the corresponding Common FIFO buffer (when configured in RX mode), or the write pointer of the corresponding Common FIFO buffer (when configured in TX mode) moves to the next FIFO entry.

The read value from these bits is always 0x00.

Only write to these bits when the CANFD module is in GL_HALT or GL_OPERATION mode.

Only write 0xFF to this register when:

- The Common FIFO buffer is enabled and is not empty if configured in RX mode
- The Common FIFO buffer is enabled and is not full if configured in TX mode

Do not write to the Common FIFO Pointer Control registers when DMA is enabled.

28.2.32 CFDFESTS : FIFO Empty Status Register

Base address: CANFD_B = 0x400B_0000

Offset address: 0x0060

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

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

Bit	Symbol	Function	R/W
1:0	RFXEMP[1:0]	RX FIFO Empty Status 0: Corresponding FIFO not empty 1: Corresponding FIFO empty	R
7:2	—	These bits are read as 0.	R
8	CFEMP	Common FIFO Empty Status 0: Corresponding FIFO not empty 1: Corresponding FIFO empty	R
31:9	—	These bits are read as 0.	R

The FIFO Empty Status register shows status of the empty bits of the FIFO buffers.

RFXEMP[1:0] bits (RX FIFO Empty Status)

The RFXEMP[1:0] bits are set when the CANFD module is in GL_RESET mode.

Each bit is set automatically when the corresponding bit is set in the RX FIFO Status Registers.

Each bit is cleared automatically when the corresponding bit is cleared in the RX FIFO Status Registers.

CFEMP bit (Common FIFO Empty Status)

The CFEMP bits are set when the CANFD module is in GL_RESET mode.

Each bit is set automatically when the corresponding bit is set in the Common FIFO Status Registers.

Each bit is cleared automatically when the corresponding bit is cleared in the Common FIFO Status Registers.

28.2.33 CFDFFSTS : FIFO Full Status Register

Base address: CANFD_B = 0x400B_0000

Offset address: 0x0064

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

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

Bit	Symbol	Function	R/W
1:0	RFXFLL[1:0]	RX FIF0 Full Status 0: Corresponding FIFO not full 1: Corresponding FIFO full	R
7:2	—	These bits are read as 0.	R
8	CFFLL	Common FIF0 Full Status 0: Corresponding FIFO not full 1: Corresponding FIFO full	R
31:9	—	These bits are read as 0.	R

The FIFO Full Status Register shows status of the full bits of the FIFO buffers.

RFXFLL[1:0] bits (RX FIF0 Full Status)

The RFXFLL[1:0] bits are cleared when CANFD module is in GL_RESET mode.

Each bit is set automatically when the corresponding bit is set in the RX FIFO Status Registers.

Each bit is cleared automatically when the corresponding bit is cleared in the RX FIFO Status Registers.

CFFLL bits (Common FIF0 Full Status)

The CFFLL bits are cleared when the CANFD module is in GL_RESET mode.

Each bit is set automatically when the corresponding bit is set in the Common FIFO Status Registers.

Each bit is cleared automatically when the corresponding bit is cleared in the Common FIFO Status Registers.

28.2.34 CFDFMSTS : FIFO Message Lost Status Register

Base address: CANFD_B = 0x400B_0000

Offset address: 0x0068

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

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

Bit	Symbol	Function	R/W
1:0	RFXMLT[1:0]	RX FIFO Message Lost Status 0: Corresponding FIFO Message Lost flag not set 1: Corresponding FIFO Message Lost flag set	R
7:2	—	These bits are read as 0.	R
8	CFMLT	Common FIFO Message Lost Status 0: Corresponding FIFO Message Lost flag not set 1: Corresponding FIFO Message Lost flag set	R
31:9	—	These bits are read as 0.	R

The FIFO Message Lost Status Register shows status of the Msg Lost bits of the FIFO buffers.

RFXMLT[1:0] bits (RX FIFO Message Lost Status)

The RFXMLT[1:0] bits are cleared when the CANFD module is in GL_RESET mode.

Each bit is set automatically when the corresponding bit is set in the RX FIFO Status Registers.

Each bit is cleared automatically when the corresponding bit is cleared in the RX FIFO Status Registers.

CFMLT bits (Common FIFO Message Lost Status)

The CFMLT bits are cleared when the CANFD module is in GL_RESET mode.

Each bit is set automatically when the corresponding bit is set in the Common FIFO Status Registers.

Each bit is cleared automatically when the corresponding bit is cleared in the Common FIFO Status Registers.

28.2.35 CFDRFISTS : RX FIFO Interrupt Flag Status Register

Base address: CANFD_B = 0x400B_0000

Offset address: 0x006C

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

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

Bit	Symbol	Function	R/W
1:0	RFXIF[1:0]	RX FIFO[x] Interrupt Flag Status 0: Corresponding RX FIFO Interrupt flag not set 1: Corresponding RX FIFO Interrupt flag set	R
31:2	—	These bits are read as 0.	R

The FIFO Interrupt Flag Status Register shows status of the interrupt flag bits of the RX FIFO buffers.

RFXIF[1:0] bits (RX FIFO[x] Interrupt Flag Status)

Each bit is set automatically when the corresponding interrupt flag bit is set in the RX FIFO Status Registers.

The RFXIF[1:0] bits are cleared when the CANFD module is in GL_RESET mode.

Each bit is cleared automatically when the corresponding interrupt flag bit is cleared in the RX FIFO Status Registers.

28.2.36 CFDCDTCT : DMA Transfer Control Register

Base address: CANFD_B = 0x400B_0000

Offset address: 0x00C8

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	CFDMAE	—	—	—	—	—	—	RFDMAE1	RFDMAE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RFDMAE0	DMA Transfer Enable for RXFIFO 0 0: DMA transfer request disabled 1: DMA transfer request enabled	R/W
1	RFDMAE1	DMA Transfer Enable for RXFIFO 1 0: DMA transfer request disabled 1: DMA transfer request enabled	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
8	CFDMAE	DMA Transfer Enable for Common FIFO 0 0: DMA transfer request disabled 1: DMA transfer request enabled	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

The DMA Transfer Control Register controls the start and stop of DMA transfer operation.

RFDMAEe (e = 0 to 1) bit (DMA Transfer Enable for RXFIFO e)

The RFDMAEe bit cannot be set in GL_SLEEP or GL_RESET mode.

This bit is cleared when the CANFD module is in GL_RESET mode.

CFDMAE bit (DMA Transfer Enable for Common FIFO)

The CFDMAE bit enables or disables DMA transfer request for common FIFO

The CFDMAE bit cannot be set in GL_SLEEP or GL_RESET mode.

Do not enable a DMA transfer for a Common FIFO that is configured as TX FIFO.

This bit is cleared when the CANFD module is in GL_RESET mode.

28.2.37 CFDCDTSTS : DMA Transfer Status Register

Base address: CANFD_B = 0x400B_0000

Offset address: 0x00CC

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	CFDMASTS	—	—	—	—	—	—	RFDMASTS1	RFDMASTS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RFDMASTS0	DMA Transfer Status for RX FIFO 0 0: DMA transfer stopped 1: DMA transfer on going	R
1	RFDMASTS1	DMA Transfer Status for RX FIFO 1 0: DMA transfer stopped 1: DMA transfer on going	R
7:2	—	These bits are read as 0.	R
8	CFDMASTS	DMA Transfer Status only for Common FIFO 0: DMA transfer stopped 1: DMA transfer on going	R
31:9	—	These bits are read as 0.	R

The DMA Transfer Status Register shows the status of the DMA transfer.

RFDMASTSe (e = 0 to 1) bit (DMA Transfer Status for RX FIFO e)

Each bit is set automatically when the corresponding DMA enable bit is set and the corresponding DMA FIFO is not empty.

Each bit is cleared automatically when the DMA transfer stops either because the DMA is disabled or the DMA FIFO is empty.

When CFDCDTCT.RFDMAEe (see CFDCDTCT.RFDMAEe bit in section 28.2.36. CFDCDTCT : DMA Transfer Control Register) is set to 0 while DMA transfer for the corresponding FIFO is on going, the RFDMASTSe bit becomes 0 when the DMA transfer is complete.

This bit is cleared when the CANFD module is in GL_RESET mode.

CFDMASTS bit (DMA Transfer Status only for Common FIFO)

Each bit is set automatically when the corresponding DMA enable bit is set and the corresponding DMA FIFO is not empty.

Each bit is cleared automatically when the DMA transfer stops either because the DMA is disabled or the DMA FIFO is empty.

When CFDCDTCT.CFDMAE (see CFDCDTCT.CFDMAE bit in section 28.2.36. CFDCDTCT : DMA Transfer Control Register) is set to 0 while DMA transfer for the corresponding FIFO is on going, the CFDMASTS bit becomes 0 when the DMA transfer is complete.

This bit is cleared when the CANFD module is in GL_RESET mode.

28.2.38 CFDTMCi : TX Message Buffer Control Registers i (i = 0 to 3)

Base address: CANFD_B = 0x400B_0000

Offset address: 0x0070 + 0x01 × i

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

Bit	Symbol	Function	R/W
0	TMTR	TX Message Buffer Transmission Request 0: TX Message buffer transmission not requested 1: TX message buffer transmission requested	R/W
1	TMTAR	TX Message Buffer Transmission Abort Request 0: TX message buffer transmission request abort not requested 1: TX message buffer transmission request abort requested	R/W
2	TMOM	TX Message Buffer One-shot Mode 0: TX message buffer not configured in one-shot mode 1: TX message buffer configured in one-shot mode	R/W

Bit	Symbol	Function	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

The TX Message Buffer Control Registers configure the TX message buffer functions.

TMTR bit (TX Message Buffer Transmission Request)

When the TMTR bit is set, the CANFD module logic tries to transmit the message stored in the corresponding message buffer.

Only write to this bit when the related CANFD module is in CH_HALT or CH_OPERATION mode.

Do not set this bit if the corresponding TX message buffer is linked to a COM FIFO in TX mode or is a part of TX Queue.

This bit cannot be directly cleared by a CPU write access.

This bit can only be set when the Transmission Result flag bits (CFDTMSTSj.TMTRF) in the CFDTMSTSj register corresponding to the message buffer are cleared to 00b.

The TMTR bit is automatically cleared by the:

- CANFD module logic at the end of a successful transmission
- CANFD module logic at the end of a transmission abort, requested by the corresponding CFDTMCi.TMTAR bit
- CANFD module logic when there is a detection of a CAN bus error or arbitration loss if CFDTMCi.TMOM bit is set for the message buffer
- CANFD module logic when the CANFD module is in GL_RESET mode or the related channel is in CH_RESET mode.

TMTAR bit (TX Message Buffer Transmission Abort Request)

When the TMTAR bit is set, the CANFD module logic tries to abort the transmission of the frame stored in the corresponding message buffer.

In most cases, transmission cannot be aborted if the internal scan for transmission is complete and the message buffer has already been selected for transmission. In this case, frame may be transmitted successfully from the message buffer. The message buffer selection is released by entering CH_HALT mode.

However, message buffer selected for transmission can be aborted by an abort request when the CAN node detects a new message on the bus (RX pin) before it starts transmission from the selected message buffer.

Only write to the TMTAR bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode. This bit can only be set when the related transmit request TMTR bit is set.

The TMTAR bit cannot be cleared by a CPU write access. Clearing of this bit by CANFD has priority over setting by a CPU write access.

The TMTAR bit is automatically cleared by:

- The CANFD module logic at the end of a successful transmission
- The CANFD module logic at the end of a transmission abort
- The CANFD module logic when there is detection of a CAN bus error or arbitration loss
- The CANFD module logic when the CANFD module is in GL_RESET mode or the related channel enters CH_RESET mode.

TMOM bit (TX Message Buffer One-shot Mode)

When the TMOM bit is set, the CANFD module logic tries to transmit the message only once.

If the transmission is successful, the CFDTMSTSj.TMTRF bits are set to 10b or 11b. Otherwise, the transmission is automatically aborted and CFDTMSTSj.TMTRF bits are set to 01b due to a bus error or a bus arbitration lost.

The TMOM bit remains set if the transmission has completed successfully or aborted due to an error or a loss of arbitration.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

Set this bit at the same time as the TMTR bit. Clear this bit with a write access.

If a message has already been requested for transmission, do not write to this bit until the message has been successfully transmitted or transmission has been aborted.

The TMOM bit is automatically cleared by the CANFD module logic when the CANFD module is in GL_RESET mode or the related channel is in CH_RESET mode.

28.2.39 CFDTMSTSj : TX Message Buffer Status Registers j (j = 0 to 3)

Base address: CANFD_B = 0x400B_0000

Offset address: 0x0074 + 0x01 × j

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	TMTARM	TMTRM	TMTRF[1:0]		TMTSTS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TMTSTS	TX Message Buffer Transmission Status 0: No on-going transmission 1: On-going transmission	R
2:1	TMTRF[1:0]	TX Message Buffer Transmission Result Flag 0 0: No result 0 1: Transmission aborted from the TX message buffer 1 0: Transmission successful from the TX message buffer and transmission abort was not requested 1 1: Transmission successful from the TX message buffer and transmission abort was requested	R/W
3	TMTRM	TX Message Buffer Transmission Request Mirrored 0: TX message buffer transmission not requested 1: TX message buffer transmission requested	R
4	TMTARM	TX Message Buffer Transmission Abort Request Mirrored 0: TX message buffer transmission request abort not requested 1: TX message buffer transmission request abort requested	R
7:5	—	These bits are read as 0. The write value should be 0.	R/W

The TX Message Buffer Status Registers show status of the transmission and transmission abort for the corresponding message buffers.

TMTSTS bit (TX Message Buffer Transmission Status)

The TMTSTS bit is set automatically at the start of the transmission from the corresponding TX message buffer.

This bit is cleared automatically when:

- Transmission stops
- The CANFD module is in GL_RESET mode
- The related CANFD channel is in CH_RESET mode.

TMTRF[1:0] bits (TX Message Buffer Transmission Result Flag)

The TMTRF[1:0] bits show the result for the corresponding TX message buffer. The status is as follows:

- 00: Transmission in progress or has not been requested
- 01: Transmission has been aborted from the corresponding TX message buffer
- 10: Transmission was successful from the corresponding TX message buffer and the CFDTMCi.TMTAR bit was not set for this TX message buffer
- 11: Transmission was successful from the corresponding TX message buffer, but the CFDTMCi.TMTAR bit was set for this TX message buffer.

Only write to these bits when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

The TMTRF[1:0] bits are cleared automatically when the CANFD module is in GL_RESET mode or the related channel is in CH_RESET mode.

If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.

TMTRM bit (TX Message Buffer Transmission Request Mirrored)

The TMTRM bit is set when the CFDTMCi.TMTR bit in the corresponding CFDTMCi register is set.

This bit is cleared when the CFDTMCi.TMTR bit in the corresponding CFDTMCi register is cleared.

TMTARM bit (TX Message Buffer Transmission Abort Request Mirrored)

The TMTARM bit is set when the CFDTMCi.TMTAR bit in the corresponding CFDTMCi register is set.

This bit is cleared when the CFDTMCi.TMTAR bit in the corresponding CFDTMCi register is cleared.

28.2.40 CFDTMTRSTS : TX Message Buffer Transmission Request Status Register

Base address: CANFD_B = 0x400B_0000

Offset address: 0x0078

Bit position:	31																										3	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CFDTMTRSTS[3:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	CFDTMTRSTS[3:0]	TX Message Buffer Transmission Request Status 0: Transmission not requested for corresponding TX message buffer 1: Transmission requested for corresponding TX message buffer	R
31:4	—	These bits are read as 0.	R

These bits show the TX Message Buffer Transmission Request Status for the corresponding TX Message Buffer. The bit 0 of a CFDTMTRSTS register corresponds to the TX message buffer 0.

The bit position of CFDTMTRSTS corresponds to the buffer number of TX message buffer.

CFDTMTRSTS[3:0] bits (TX Message Buffer Transmission Request Status)

The CFDTMTRSTS[3:0] bits show status of the CFDTMCi.TMTR bits of the TX Message Buffer Control Registers.

Each bit is set automatically when the corresponding bit is set in the TX Message Buffer Control Registers (CFDTMCi), and only when the message buffer does not belong to a TX Queue.

Each bit is cleared automatically when:

- The corresponding bit is cleared in the TX Message Buffer Control Registers
- The CANFD module is in GL_RESET mode
- The related CANFD channel is in CH_RESET mode.

28.2.41 CFDTMTARSTS : TX Message Buffer Transmission Abort Request Status Register

Base address: CANFD_B = 0x400B_0000

Offset address: 0x007C

Bit position:	31																										3	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CFDTMTARSTS[3:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	CFDTMTARSTS[3:0]	TX Message Buffer Transmission Abort Request Status 0: Transmission abort not requested for corresponding TX message buffer 1: Transmission abort requested for corresponding TX message buffer	R
31:4	—	These bits are read as 0.	R

These bits show the TX Message Buffer Transmission Abort Request Status for the corresponding TX Message Buffer. The bit 0 of a CFDTMTARSTS register corresponds to the TX message buffer 0.

The bit position of CFDTMTARSTS corresponds to the buffer number of TX message buffer.

CFDTMTARSTS[3:0] bits (TX Message Buffer Transmission Abort Request Status)

The CFDTMTARSTS[3:0] bits show status of the CFDTMCi.TMTAR bits of the TX Message Buffer Control Registers.

Each bit is set automatically when the corresponding bit is set in the TX Message Buffer Control Registers, and when the message buffer belongs to a TX Queue.

Each bit is cleared automatically when:

- The corresponding bit is cleared in the TX Message Buffer Control Registers
- The CANFD module is in GL_RESET mode
- The related CANFD channel is in CH_RESET mode.

28.2.42 CFDTMTCSTS : TX Message Buffer Transmission Completion Status Register

Base address: CANFD_B = 0x400B_0000

Offset address: 0x0080

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

Bit	Symbol	Function	R/W
3:0	CFDTMTCSTS[3:0]	TX Message Buffer Transmission Completion Status 0: Transmission not complete for corresponding TX message buffer 1: Transmission completed for corresponding TX message buffer	R
31:4	—	These bits are read as 0.	R

These bits show the TX Message Buffer Transmission Completion Status for the corresponding TX Message Buffer. The bit 0 of a CFDTMTCSTS register corresponds to the TX message buffer 0.

The bit position of CFDTMTCSTS corresponds to the buffer number of TX message buffer.

CFDTMTCSTS[3:0] bits (TX Message Buffer Transmission Completion Status)

The CFDTMTCSTS[3:0] bits show status of successful completion of the TX Message Buffer Status Registers.

Each bit is set automatically when the corresponding bit is set in the TX Message Buffer Status Registers.

Each bit is cleared automatically when:

- The corresponding bit is cleared in the TX Message Buffer Status Registers
- The CANFD module is in GL_RESET mode
- The related CANFD channel is in CH_RESET mode.

If a CAN channel enters CH_RESET mode, then the bits related to that channel are cleared.

28.2.43 CFDTMTASTS : TX Message Buffer Transmission Abort Status Register

Base address: CANFD_B = 0x400B_0000

Offset address: 0x0084

Bit position:	31																										3	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CFDTMTASTS[3:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	CFDTMTASTS[3:0]	TX Message Buffer Transmission Abort Status 0: Transmission not aborted for corresponding TX message buffer 1: Transmission aborted for corresponding TX message buffer	R
31:4	—	These bits are read as 0.	R

These bits show the TX Message Buffer Transmission abort Status for the corresponding TX Message Buffer. The bit 0 of a CFDTMTASTS register corresponds to the TX message buffer 0.

The bit position of CFDTMTASTS corresponds to the buffer number of TX message buffer.

CFDTMTASTS[3:0] bits (TX Message Buffer Transmission Abort Status)

The CFDTMTASTS[3:0] bits show status of the successful transmission abort of the corresponding TX message buffer.

Each bit is set automatically when the CFDTMSTSj.TMTRF bits are set to 01b in the corresponding TX Message Buffer Status Register.

Each bit is cleared automatically when:

- The CFDTMSTSj.TMTRF bits are cleared in the corresponding TX Message Buffer Status Register
- The CANFD module is in GL_RESET mode
- The related CANFD channel is in CH_RESET mode.

28.2.44 CFDTMIEC : TX Message Buffer Interrupt Enable Configuration Register

Base address: CANFD_B = 0x400B_0000

Offset address: 0x0088

Bit position:	31																										3	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TMIEg[3:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	TMIEg[3:0]	TX Message Buffer Interrupt Enable 0: TX message buffer interrupt disabled for corresponding TX message buffer 1: TX message buffer interrupt enabled for corresponding TX message buffer	R/W
31:4	—	These bits are read as 0. The write value should be 0.	R

These bits show the TX Message Buffer Interrupt Enable for the corresponding TX Message Buffer.

The bit 0 of a CFDTMIEC register corresponds to the TX message buffer 0.

The bit position of CFDTMIEC corresponds to the buffer number of TX message buffer.

g = [0…3]

TMIEg[3:0] bits (TX Message Buffer Interrupt Enable)

If the TMIEg[3:0] bits are set, an interrupt is generated at the end of a successful transmission from the corresponding message buffer.

See section 28.7. Interrupts and DMA for TX Message Buffer Interrupt specification.

Do not write to the TMIEg[7:0] bits when:

- The CANFD module is in GL_SLEEP mode
- The related CANFD channel is in CH_SLEEP mode
- The corresponding TX message buffer is part of a TX Queue
- The corresponding TX message buffer is linked to a Common FIFO with the CFDCFCC.CFTML bits.

28.2.45 CFDTXQCC : TX Queue Configuration/Control Register

Base address: CANFD_B = 0x400B_0000

Offset address: 0x008C

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

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

Bit	Symbol	Function	R/W
0	TXQE	TX Queue Enable 0: TX Queue disabled 1: TX Queue enabled	R/W
4:1	—	These bits are read as 0. The write value should be 0.	R/W
5	TXQTXIE	TX Queue TX Interrupt Enable 0: TX Queue TX interrupt disabled 1: TX Queue TX interrupt enabled	R/W
6	—	This bit is read as 0. The write value should be 0.	R/W
7	TXQIM	TX Queue Interrupt Mode 0: When the last message is successfully transmitted 1: At every successful transmission	R/W
9:8	TXQDC[1:0]	TX Queue Depth Configuration 0x00: 0 messages 0x01: Reserved 0x10: 3 messages 0x11: 4 messages	R/W
31:10	—	These bits are read as 0. The write value should be 0.	R/W

The TX Queue Configuration/Control Registers are used to configure the TX Queue transmission.

TXQ is composed of TXMB0 to TXMB3 (at the maximum) when TXQE is enabled.

TXQE bit (TX Queue Enable)

The TXQE bit cannot be set if the configured TX Queue depth is 0x00 (CFDTXQCC.TXQDC == 0x00).

You cannot write to this bit when the CANFD module is in GL_SLEEP mode.

Do not write to this bit when the related CANFD channel is in CH_RESET or CH_SLEEP mode.

The TXQE bit is cleared automatically when the related CANFD channel is in CH_RESET mode.

TXQTXIE bit (TX Queue TX Interrupt Enable)

When the TXQTXIE bit is set, an interrupt is generated based on the setting of the TXQIM bit.

You cannot write to this bit when the CANFD module is in GL_SLEEP mode.

Do not write to this bit when the related CANFD channel is in CH_SLEEP mode.

TXQIM bit (TX Queue Interrupt Mode)

The TXQIM bit selects the interrupt generation condition for the TX Queue.

You cannot write to this bit when the CANFD module is in GL_SLEEP mode.

Do not write to this bit when the related CANFD channel is in any of the following modes:

- CH_SLEEP
- CH_HALT
- CH_OPERATION.

TXQDC[1:0] bits (TX Queue Depth Configuration)

The TXQDC[1:0] bits select the depth of the transmission queue. The message buffer selection starts from MB[0] up to MB[3] depending on the configured depth.

You cannot write to this bit when the CANFD module is in GL_SLEEP mode.

Do not write to this bit when the related CANFD channel is in any of the following modes:

- CH_SLEEP
- CH_HALT
- CH_OPERATION.

28.2.46 CFDTXQSTS : TX Queue Status Register

Base address: CANFD_B = 0x400B_0000

Offset address: 0x0090

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

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

Bit	Symbol	Function	R/W
0	TXQEMP	TX Queue Empty 0: TX Queue not empty 1: TX Queue empty	R
1	TXQFLL	TX Queue Full 0: TX Queue not full 1: TX Queue full	R
2	TXQTXIF	TX Queue TX Interrupt Flag 0: TX Queue interrupt condition not satisfied after a frame TX 1: TX Queue interrupt condition satisfied after a frame TX	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
10:8	TXQMC[2:0]	TX Queue Message Count Number of messages in the TX Queue.	R
31:11	—	These bits are read as 0. The write value should be 0.	R/W

The TX Queue Status Registers show the status of the TX Queue of corresponding CAN channel.

TXQEMP bit (TX Queue Empty)

The TXQEMP bit is set automatically when the TX Queue is disabled or no messages are stored in the TX Queue.

This bit is set automatically when:

- The last message is transmitted from the TX Queue
- The related CANFD channel is in CH_RESET mode.

The bit is cleared automatically when the first message to be transmitted is stored in the TX Queue.

TXQFLL bit (TX Queue Full)

The TXQFLL bit is set automatically when the number of CAN messages stored in the TX Queue matches the configured TX Queue depth.

This bit is cleared automatically when:

- The number of CAN messages stored in the TX Queue is less than the configured TX Queue depth
- The related CANFD channel is in CH_RESET mode.

TXQTXIF bit (TX Queue TX Interrupt Flag)

The TXQTXIF bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1. Writing 1 has no effect.

This bit is set automatically when the configured interrupt condition is satisfied for the TX Queue.

If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.

You cannot write to this bit when the related CANFD channel is in CH_SLEEP or CH_RESET mode.

The bit is cleared:

- By writing 0 to it
- When the related CANFD channel is in CH_RESET mode.

TXQMC[2:0][13:8] bits (TX Queue Message Count)

The TXQMC[2:0] bits show the number of CAN messages in the TX Queue.

These bits are cleared automatically when the related CANFD channel is in CH_RESET mode.

28.2.47 CFDTXQPCTR : TX Queue Pointer Control Register

Base address: CANFD_B = 0x400B_0000

Offset address: 0x0094

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

Bit	Symbol	Function	R/W
7:0	TXQPC[7:0]	TX Queue Pointer Control Increments the write pointer to the TX Queue buffer in the corresponding channel	W
31:8	—	The write value should be 0.	W

The TX Queue Pointer Control Registers are used to confirm storage of a full message in the corresponding TX Queue buffers.

TXQPC[7:0] bits (TX Queue Pointer Control)

When the value 0xFF is written to the TXQPC[7:0] bits, the write pointer of the corresponding TX Queue buffer is updated and a transmit request is initiated for this message.

The read value from these bits is always 0x00. Do not write to the FIFO control registers when DMA is enabled.

You cannot write to these bits when the related CANFD channel is in CH_SLEEP or CH_RESET mode.

Only write 0xFF to this register when:

- The corresponding TX Queue is enabled and not full
- The Common FIFO is enabled.

28.2.48 CFDTHLCC : TX History List Configuration/Control Register

Base address: CANFD_B = 0x400B_0000

Offset address: 0x0098

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

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

Bit	Symbol	Function	R/W
0	THLE	TX History List Enable 0: TX History List disabled 1: TX History List enabled	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
8	THLIE	TX History List Interrupt Enable 0: TX History List Interrupt disabled 1: TX History List Interrupt enabled	R/W
9	THLIM	TX History List Interrupt Mode 0: Interrupt generated if TX History List level reaches ¾ of the TX History List depth 1: Interrupt generated for every successfully stored entry	R/W
10	THLDTE	TX History List Dedicated TX Enable 0: TX FIFO + TX Queue 1: Flat TX MB + TX FIFO + TX Queue	R/W
31:11	—	These bits are read as 0. The write value should be 0.	R/W

The TX History List Configuration/Control Register configures the TX History List functions.

THLE bit (TX History List Enable)

The THLE bit enables the TX History List buffer when it is set.

You cannot write to this bit when the related CANFD channel is in CH_RESET or CH_SLEEP mode.

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode.

THLIE bit (TX History List Interrupt Enable)

The THLIE bit enables the generation of the TX History List interrupt when it is set.

You cannot write to this bit when the CANFD module is in GL_SLEEP mode.

THLIM bit (TX History List Interrupt Mode)

The THLIM bit selects the interrupt generation condition for the FIFO.

You cannot write to this bit when the CANFD module is in GL_SLEEP mode.

Do not write to this bit when the CANFD module is in GL_HALT or GL_OPERATION mode.

THLDTE bit (TX History List Dedicated TX Enable)

The THLDTE bit selects the condition for storing an entry in the TX History List after successful transmission.

You cannot write to this bit when the CANFD module is in GL_SLEEP mode.

Do not write to this bit when the CANFD module is in GL_HALT or GL_OPERATION mode.

28.2.49 CFDTHLSTS : TX History List Status Register

Base address: CANFD_B = 0x400B_0000

Offset address: 0x009C

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

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

Bit	Symbol	Function	R/W
0	THLEMP	TX History List Empty 0: TX History List not empty 1: TX History List empty	R
1	THLFLL	TX History List Full 0: TX History List not full 1: TX History List full	R
2	THLELT	TX History List Entry Lost 0: No entry lost in TX History List 1: TX History List entry Lost	R/W
3	THLIF	TX History List Interrupt Flag 0: TX History List interrupt condition not satisfied 1: TX History List interrupt condition satisfied	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
11:8	THLMC[3:0]	TX History List Message Count Number of messages stored in TX History List	R
31:12	—	These bits are read as 0. The write value should be 0.	R/W

The TX History List Status register shows the status of data stored in the TX History List buffer.

THLEMP bit (TX History List Empty)

The THLEMP bit is set automatically when the CPU has read all the entries from the TX History List buffer.

This bit is cleared automatically when the first entry is stored to the TX History List.

This bit is set automatically when:

- TX History List is disabled
- The related CANFD channel is in CH_RESET mode.

THLFLL bit (TX History List Full)

The THLFLL bit is set automatically when the number of entries in the TX History List buffer matches the TX History List depth.

Each TX History List can store up to 8 entries.

This bit is cleared automatically when:

- The number of entries in the TX History List buffer is less than the TX History List depth
- The TX History List is disabled
- The related CANFD channel is in CH_RESET mode.

THLELT bit (TX History List Entry Lost)

The THLELT bit is set when a new entry cannot be stored because the related TX History List buffer is already full.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.

This bit is cleared:

- By writing 0 to it
- When the related CANFD channel is in CH_RESET mode.

THLIF bit (TX History List Interrupt Flag)

The THLIF bit is set when the configured interrupt condition is satisfied.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.

This bit is cleared:

- By writing 0 to it
- When the related CANFD channel is in CH_RESET mode.

The bit is cleared by writing 0 to it.

This bit is automatically cleared in CH_RESET mode.

THLMC[3:0] bits (TX History List Message Count)

The THLMC[3:0] bits show the number of transmitted messages stored in the TX History List.

These bits are cleared automatically when the related CANFD channel is in CH_RESET mode.

28.2.50 CFDTHLACC0 : TX History List Access Register 0

Base address: CANFD_B = 0x400B_0000

Offset address: 0x0740

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	TMTS[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
2:0	BT[2:0]	Buffer Type 0 0 1: Flat TX message buffer 0 1 0: TX FIFO message buffer number 1 0 0: TX Queue message buffer number	R
4:3	BN[1:0]	Buffer Number Number of the message buffer	R

Bit	Symbol	Function	R/W
15:5	—	These bits are read as 0.	R
31:16	TMTS[15:0]	Transmit Timestamp Transmit timestamp value for software drivers	R

The TX History List Access Registers 0 provide access to the entry in the TX History List based on the read timestamp value.

BT[2:0] bits (Buffer Type)

The BT[2:0] bits indicate whether data has been stored following a transmission from a FIFO buffer, a TX Queue or a TX message buffer.

BN[1:0] bits (Buffer Number)

The BN[1:0] bits show the message buffer from which transmission was successfully completed. If a message from a Common FIFO is transmitted, then these bits show the message buffer that is linked to the Common FIFO for transmission.

TMTS[15:0] bits (Transmit Timestamp)

The TMTS[15:0] bits indicate the timestamp for use by software drivers.

28.2.51 CFDTHLACC1 : TX History List Access Register 1

Base address: CANFD_B = 0x400B_0000

Offset address: 0x0744

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

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

Bit	Symbol	Function	R/W
15:0	TID[15:0]	Transmit ID These bits indicate that message buffer reference ID, TX FIFO reference ID, or AFL pointer field is stored for software drivers.	R
17:16	TIFL[1:0]	Transmit Information Label These bits indicate that message buffer information label, TX FIFO information label, or AFL information label is stored for software drivers.	R
31:18	—	These bits are read as 0.	R

The TX History List Access Registers 1 provide access to entry in the TX History List based on the read pointer value.

TID[15:0] bits (Transmit ID)

The TID[15:0] bits indicate whether the message buffer reference ID (CFDTMFDCTRb.TMPTR) or the TX FIFO reference ID (CFDCFFDCSTS.CFPTR) is for use by software drivers.

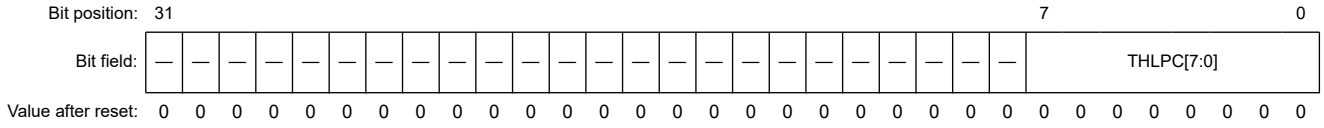
TIFL[1:0] bits (Transmit Information Label)

The TIFL[1:0] bits indicate whether the message buffer information label (CFDTMFDCTRb.TMIFL) or the TX FIFO information label (CFDCFFDCSTS.CFIFL) is for use by software drivers.

28.2.52 CFDTHLPCTR : TX History List Pointer Control Register

Base address: CANFD_B = 0x400B_0000

Offset address: 0x00A0



Bit	Symbol	Function	R/W
7:0	THLPC[7:0]	TX History List Pointer Control Increments the write pointer to the TX History List in the corresponding channel	W
31:8	—	The write value should be 0.	W

The TX History List Pointer Control Registers are used to increment the read pointer of the TX History List.

THLPC[7:0] bits (TX History List Pointer Control)

When 0xFF is written to the THLPC[7:0] bits, the read pointer of the TX History List is moved to the next TX History List entry address.

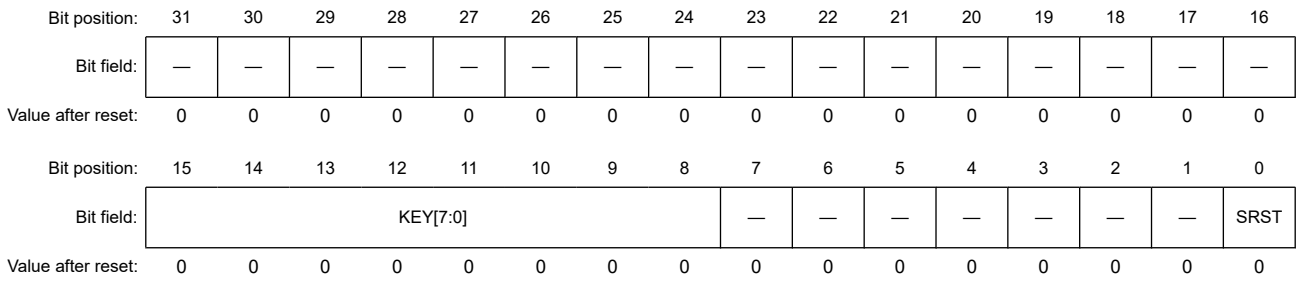
The read value from these bits is always 0x00. Only write to these bits when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

Only write 0xFF to these registers when the corresponding TX History List is enabled and not empty.

28.2.53 CFDGRSTC : Global SW reset Register

Base address: CANFD_B = 0x400B_0000

Offset address: 0x00D8



Bit	Symbol	Function	R/W
0	SRST	SW Reset 0: Normal state 1: SW reset state	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code These bits control the validity of rewriting of a SRST bit.	W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

SRST bit (SW Reset)

When the SRST bit is set, the CANFD module is in the same state as hardware reset. When a reset is required, write 1 then write 0 to this bit.

This bit is cleared when the CANFD module is in GL_SLEEP mode.

When this bit is cleared, the RAM initialization sequence does not operate. The configuration of RAM is performed by software.

The RAM is not initialized when software reset is performed during the initialization of RAM. Software must perform the initialization of RAM.

KEY[7:0] bits (Key Code)

When 0xC4 is written in the KEY[15:8] bits, a write to the SRST bit is valid.

The read value from these bits is always 0x00.

CFDGRSTC.SRST bit and the CFDGRSTC.KEY bit should be written simultaneously.

28.2.54 CFDGTSTCFG : Global Test Configuration Register

Base address: CANFD_B = 0x400B_0000

Offset address: 0x00A8

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

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

Bit	Symbol	Function	R/W
15:0	—	These bits are read as 0. The write value should be 0.	R/W
19:16	RTMPS[3:0]	RAM Test Mode Page Select Select a RAM test mode page	R/W
31:20	—	These bits are read as 0. The write value should be 0.	R/W

The Global Test Configuration Register is used to configure the RAM test mode page.

RTMPS[3:0] bits (RAM Test Mode Page Select)

The RTMPS[3:0] bits select the RAM page mode for CPU read/write access when the CANFD module is configured in RAM test mode.

See section 28.9.2.1. RAM Test Mode for the RAM test mode specification.

Do not write to these bits when the CANFD module is in GL_RESET or GL_SLEEP mode.

Only enter values from 0 to 9 (0x009) for the message buffer RAM.

Only write to these bits when the CANFD module is in GL_HALT mode.

These bits are cleared automatically when the related CANFD channel is in GL_RESET mode.

28.2.55 CFDGTSTCTR : Global Test Control Register

Base address: CANFD_B = 0x400B_0000

Offset address: 0x00AC

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

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

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
1	—	This bit is read as 0. The write value should be 0.	R/W
2	RTME	RAM Test Mode Enable 0: RAM test mode disabled 1: RAM test mode enabled	R/W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

The Global Test Control register is used to control the global test modes of the CANFD module.

RTME bit (RAM Test Mode Enable)

When the RTME bit is set, the CANFD module is configured in RAM test mode. See section 28.9.2.1. RAM Test Mode for RAM test mode specification.

Only write to this bit when the CANFD module is in GL_HALT mode.

Clear this bit when the CANFD module is in GL_HALT mode.

This bit is cleared automatically when the CANFD module is in GL_RESET mode.

28.2.56 CFDGFDCFG : Global FD Configuration Register

Base address: CANFD_B = 0x400B_0000

Offset address: 0x00B0

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

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

Bit	Symbol	Function	R/W
0	RPED	RES Bit Protocol Exception Disable 0: Protocol exception event detection enabled 1: Protocol exception event detection disabled	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
9:8	TSCCFG[1:0]	Timestamp Capture Configuration 0 0: Timestamp capture at the sample point of SOF (start of frame) 0 1: Timestamp capture at frame valid indication 1 0: Timestamp capture at the sample point of RES bit 1 1: Reserved	R/W
31:10	—	These bits are read as 0. The write value should be 0.	R/W

RPED bit (RES Bit Protocol Exception Disable)

The RPED bit configures the protocol exception event handling according to ISO 11898-1.

When this bit is enabled, the protocol exception event detection is disabled, and the protocol controller transmits an error frame when the protocol exception event is detected (RES bit is sampled recessive).

Only write to this bit when the CANFD module is in GL_RESET mode.

TSCCFG[1:0] bits (Timestamp Capture Configuration)

The TSCCFG[1:0] bits configure the different capture points of the timestamp for transmission and reception.

When CFDGFDCFG.TSCCFG[1:0] = 10b, the timestamp capture is performed for CANFD frames at RES bit and for Classical frames at the start of frame.

Only write to these bits when the CANFD module is in GL_RESET mode.

28.2.57 CFDGLOCKK : Global Lock Key Register

Base address: CANFD_B = 0x400B_0000

Offset address: 0x00B8

Bit position:	31																15														0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LOCK[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	LOCK[15:0]	Lock Key Key bits for unlocking the protection of test modes	W
31:16	—	The write value should be 0.	W

The Global Lock Key register is a write-only register that is used to unlock the protection for special test bits.

See section 28.9.2. Global Test Modes for Lock key specification.

LOCK[15:0] bits (Lock Key)

The unlock key sequence must be written in the LOCK[15:0] bits to configure the CANFD module in FIFO OTB disable and RAM test modes.

The read value from these bits is always 0x0000.

You cannot write to these bits when the CANFD module is in GL_SLEEP or GL_RESET mode.

Do not write to these bits when the CANFD module is in GL_OPERATION mode.

28.2.58 CFDRPGACCk : RAM Test Page Access Registers k (k = 0 to 63)

Base address: CANFD_B = 0x400B_0000

Offset address: 0x0280 + 0x0004 × k

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

Bit	Symbol	Function	R/W
31:0	RDTA[31:0]	RAM Data Test Access RAM data bytes	R/W

RDTA[31:0] bits (RAM Data Test Access)

Data can be read from or written into the RDTA[31:0] bits when the CANFD module is configured in RAM test mode.

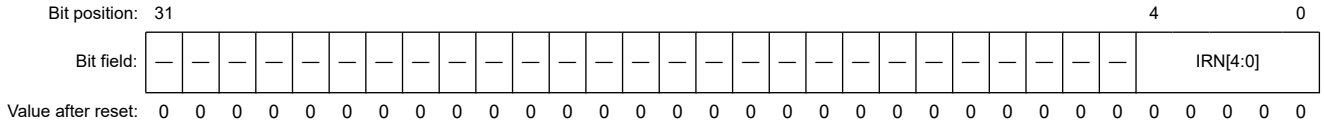
Only write to this bit when the CANFD module is in GL_HALT mode and RAM test mode is enabled.

Software data should be read/written in the RAM Test Page Access registers during RAM test mode.

28.2.59 CFDGAFLIGNENT : Global AFL Ignore Entry Register

Base address: CANFD_B = 0x400B_0000

Offset address: 0x00C0



Bit	Symbol	Function	R/W
4:0	IRN[4:0]	Ignore Rule Number Define rule number which ignores an AFL entry.	R/W
31:5	—	These bits are read as 0. The write value should be 0.	R/W

IRN[4:0] bits (Ignore Rule Number)

The IRN[4:0] bits define the rule number which updates an AFL entry.

Enter only values between 0 and 31 (0x1F) inclusive.

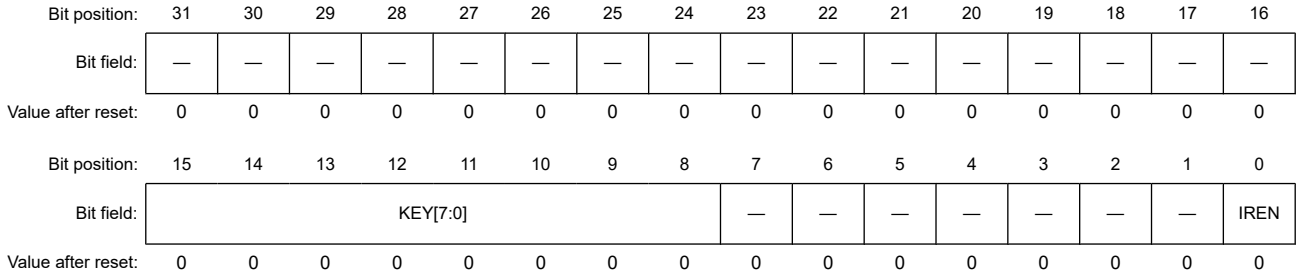
Only write to these bits when the CFDGAFLIGNCTR.IREN bit is 0.

You cannot write to these bits when the CANFD module is in GL_SLEEP mode.

28.2.60 CFDGAFLIGNCTR : Global AFL Ignore Control Register

Base address: CANFD_B = 0x400B_0000

Offset address: 0x00C4



Bit	Symbol	Function	R/W
0	IREN	Ignore Rule Enable 0: AFL entry number is not ignored 1: AFL entry number is ignored	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code These bits control the validity of rewriting the IREN bit.	W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

IREN bit (Ignore Rule Enable)

When the IREN bit is set, the entry number (selected by CFDGAFLIGNENT register) is ignored.

This bit is cleared automatically when the CANFD module is in GL_RESET mode.

KEY[7:0] bits (Key Code)

When 0xC4 is written in the KEY[7:0] bits, a write to the IREN bit is valid.

The read value from these bits is always 0x00.

CFDGAFLIGNCTR.IREN bit and the CFDGAFLIGNCTR.KEY bit should be written simultaneously

28.2.61 CFDRMIEC : RX Message Buffer Interrupt Enable Configuration Register

Base address: CANFD_B = 0x400B_0000

Offset address: 0x0038

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

Bit	Symbol	Function	R/W
31:0	RMIEg[31:0]	RX Message Buffer Interrupt Enable 0: RX Message Buffer Interrupt disabled for corresponding RX message buffer 1: RX Message Buffer Interrupt enabled for corresponding RX message buffer	R/W

These bits show the RX Message Buffer Interrupt Enable for the corresponding RX Message Buffer. CFDRMIEC bit 0 corresponds to RX Message Buffer 0 and so on.

The bit position of CFDRMIEC corresponds to the buffer number of RXMB.

RMIEg[31:0] bit (RX Message Buffer Interrupt Enable)

If this bit is set, then an interrupt will be generated at the end of a successful reception from the corresponding Message Buffer.

For details, see section 28.7.1. Interrupts.

Users cannot write to this bit when the CANFD module is in GL_SLEEP mode.

28.2.62 Message Buffer Component Structure

28.2.62.1 Start Addresses

The start address for each of the Message Buffer component is calculated using the number of related Message Buffer components.

The start addresses for each register in the Message Buffer component are depicted in Table 28.5.

Table 28.5 Message Buffer Component Register Start Addresses (1 of 2)

b = Message buffer component index	MBCP	p	Register	Start Address
[0…31] b = [0…7]	RMBCPb[0]	x	RMID	0x0920 + b × 0x004C
		x	RMPTR	0x0924 + b × 0x004C
		x	RMFDSTSb	0x0928 + b × 0x004C
		[1…15]	RMDFbp	0x092C + b × 0x004C + p × 0x0004
[0…31] b = [8…15]	RMBCPb[0]	x	RMIDb	0x0D20 + (b-8) × 0x004C
		x	RMPTRb	0x0D24 + (b-8) × 0x004C
		x	RMFDSTSb	0x0D28 + (b-8) × 0x004C
		[1…15]	RMDFbp	0x0D2C + (b-8) × 0x004C + p × 0x0004
[0…31] b = [16…23]	RMBCPb[0]	x	RMIDb	0x1120 + (b-16) × 0x004C
		x	RMPTRb	0x1124 + (b-16) × 0x004C
		x	RMFDSTSb	0x1128 + (b-16) × 0x004C
		[1…15]	RMDFbp	0x112C + (b-16) × 0x004C + p × 0x0004

Table 28.5 Message Buffer Component Register Start Addresses (2 of 2)

b = Message buffer component index	MBCP	p	Register	Start Address
[0…31] b = [24…31]	RMBCPb[0]	x	RMIDb	0x1520 + (b-24) × 0x004C
		x	RMPTRb	0x1524 + (b-24) × 0x004C
		x	RMFDSTSb	0x1528 + (b-24) × 0x004C
		[1…15]	RMDFbp	0x152C + (b-24) × 0x004C + p × 0x0004
[0…1]	RFMBCPb[0]	x	RFIDb	0x0520 + b × 0x004C
		x	RFPTRb	0x0524 + b × 0x004C
		x	RFFDSTSb	0x0528 + b × 0x004C
		[1…15]	RFDFbp	0x052C + b × 0x004C + p × 0x0004
[0]	CFMBCPb[0]	x	CFID	0x05B8
		x	CFPTR0	0x05BC
		x	CFFDCSTS0	0x05C0
		[1…15]	CFDFp0	0x05C4 + p × 0x0004
[0…3]	TMBCPb[0]	x	TMIDb	0x0604 + b × 0x004C
		x	TMPTRb	0x0608 + b × 0x004C
		x	TMFDCTRb	0x060C + b × 0x004C
		[1…15]	TMDFbp	0x0610 + b × 0x004C + p × 0x0004

The message buffer configuration consists of four types of Message Buffer components:

- RX Message Buffer Component (CFDRMBCPb[0])
- RX FIFO Access Message Buffer Component (CFDRFMBCPb[0])
- Common FIFO Access Message Buffer Component (CFDCFMBCP0[0])
- TX Message Buffer Component (CFDTMBCPb[0]).

Where b = the Message Buffer component index that has a range that varies based on the type of Message Buffer component.

For a summary of this configuration, see Figure 28.29. For a detailed description of the number of and the different types of message buffers, see section 28.6. FIFO Buffers and Normal Message Buffer Configuration.

As described in section 28.2. Register Descriptions, each Message Buffer component consists of the following registers:

- Identifier (ID)
- Pointer (PTR)
- Data Field (DFp).

Where p = the Data Field register index that has a range that varies based on the type of message buffer component.

Rc is the Message Buffer Component register where c = Message Buffer Component register index that has a range that varies based on the type of Message Buffer component.

A description of the registers, their associated bits and their accessibility are shown below the summary and detailed figures of each component.

In each of the figures, a cell that contains '-' means reserved and has the same behavio as reserved bits for registers in section 28.2.62. Message Buffer Component Structure.

28.2.62.2 CFDRMBCPb[0] : RX Message Buffer Component b (b = 0 to 31)

Base address: CANFD_B = 0x400B_0000

Offset address: See Table 28.5

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

Bit	Symbol	Function	R/W
31:0	Rc[31:0]	RX Message Buffer Component c Refer to Table 28.6, Table 28.7 and the descriptions that follow for a detailed description of each register and its related bits, contained within this message buffer component.	R/W

Where the total number of CFDRMBCPb = 32 as shown in Figure 28.29 (c = RX Message Buffer Component Register index = [0…18])

Rc[31:0] bit (RX Message Buffer Component c)

The RX Message Buffer Component is made up of the following registers: CFDRMIDb, CFDRMPTRb, CFDRMFDSTSb, and CFDRMDFbp. Refer to Table 28.7 for details of how to interpret the structure of this buffer component and how to access the respective registers.

Table 28.6 RX Message Buffer Component Summary

RX Message Buffer Component (RMBCP)	
Rc	**CANFD mode (CAN_FD_MODE = 1'b1)**
R0	RX Message Buffer (b) ID Registers
R1	RX Message Buffer (b) Pointer Registers
R2	RX Message Buffer (b) CANFD Status Registers
R3	RX Message Buffer (b) Data Field 0 Registers
R4	RX Message Buffer (b) Data Field 1 Registers
R5	RX Message Buffer (b) Data Field 2 Registers
R6	RX Message Buffer (b) Data Field 3 Registers
R7	RX Message Buffer (b) Data Field 4 Registers
R8	RX Message Buffer (b) Data Field 5 Registers
R9	RX Message Buffer (b) Data Field 6 Registers
R10	RX Message Buffer (b) Data Field 7 Registers
R11	RX Message Buffer (b) Data Field 8 Registers
R12	RX Message Buffer (b) Data Field 9 Registers
R13	RX Message Buffer (b) Data Field 10 Registers
R14	RX Message Buffer (b) Data Field 11 Registers
R15	RX Message Buffer (b) Data Field 12 Registers
R16	RX Message Buffer (b) Data Field 13 Registers
R17	RX Message Buffer (b) Data Field 14 Registers
R18	RX Message Buffer (b) Data Field 15 Registers
R[19…31]	—

Table 28.7 RX Message Buffer Component (RMBCP) Detailed

Rc	p	Symbol	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R0	x	CFDRMIDb	RMIDE	RMRTR	—	RMID																												
R1	x	CFDRMPTRb	RMDLC				—	—	—	—	—	—	—	—	—	—	—	—	RMTS															
R2	x	CFDRMFDSTSb	RMPTR																—	—	—	—	—	—	RMIFL		—	—	—	—	—	RMFDF	RMBRS	RMESI
R3	0	CFDRMDFbp	RMDB_HH								RMDB_HL								RMDB_LH								RMDB_LL							
R[4…18]	[1…15]	CFDRMDFbp	RMDB_HH								RMDB_HL								RMDB_LH								RMDB_LL							

28.2.62.3 CFDRMIDb : RX Message Buffer ID Registers (b = 0 to 31)

Base address: CANFD_B = 0x400B_0000

Offset address: 0x0920 + 0x004C × b (b = 0 to 7)
0x0D20 + 0x004C × (b - 8) (b = 8 to 15)
0x01120 + 0x004C × (b - 16) (b = 16 to 23)
0x01520 + 0x004C × (b - 24) (b = 24 to 31)

Bit position:	31	30	28	0
Bit field:	RMIDE	RMRTR	—	RMID[28:0]
Value after reset:	0	0	0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
28:0	RMID[28:0]	RX Message Buffer ID Field STD-ID/EXT-ID fields	R
29	—	This bit is read as 0. The write value should be 0.	R
30	RMRTR	RX Message Buffer RTR Bit 0: Data frame 1: Remote frame	R
31	RMIDE	RX Message Buffer IDE Bit 0: STD-ID is stored 1: EXT-ID is stored	R

The RX Message Buffer ID Register b (b = 0 to 31) store the ID field, IDE bit, and RTR bit of the received message.

RMID[28:0] bits (RX Message Buffer ID Field)

The RMID[28:0] are the bits of the STD-ID/EXT-ID fields of the message stored in the RX message buffer.

See section 28.2.62.1. Start Addresses for details on how to interpret the structure of this buffer component.

RMRTR bit (RX Message Buffer RTR Bit)

The RMRTR bit shows whether a data frame or a remote frame was stored in the RX message buffer.

Note: There are no remote frames in CANFD format. When a CANFD frame is received, the register reflects the state of the received value (the RRS bit in FD frame format).

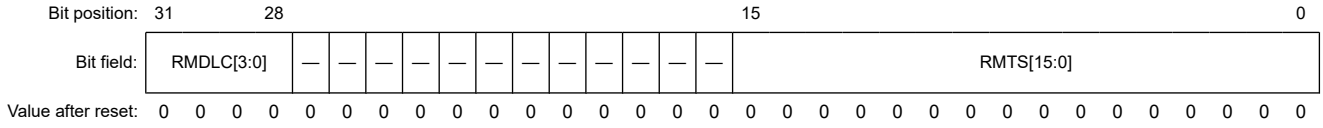
RMIDE bit (RX Message Buffer IDE Bit)

The RMIDE bit shows whether message with Standard Identifier or Extended Identifier was stored in the RX message buffer.

28.2.62.4 CFDRMPTRb : RX Message Buffer Pointer Registers (b = 0 to 31)

Base address: CANFD_B = 0x400B_0000

Offset address: 0x0924 + 0x004C × b (b = 0 to 7)
0x0D24 + 0x004C × (b - 8) (b = 8 to 15)
0x01124 + 0x004C × (b - 16) (b = 16 to 23)
0x01524 + 0x004C × (b - 24) (b = 24 to 31)



Bit	Symbol	Function	R/W
15:0	RMTS[15:0]	RX Message Buffer Timestamp Field Timestamp value stored for the message in the RX message buffer	R
27:16	—	These bits are read as 0. The write value should be 0.	R
31:28	RMDLC[3:0]	RX Message Buffer DLC Field Number of data bytes received in a CAN frame.	R

The RX Message Buffer Pointer Register b (b = 0 to 31) store the DLC and Timestamp fields for the received message.

RMTS[15:0] bits (RX Message Buffer Timestamp Field)

The RMTS[15:0] bits store the timestamp value taken at the capture point as configured by CFDGFDCFG.TSCCFG of the received message.

RMDLC[3:0] bits (RX Message Buffer DLC Field)

The RMDLC[3:0] bits store the number of data bytes that were received in the RX message buffer.

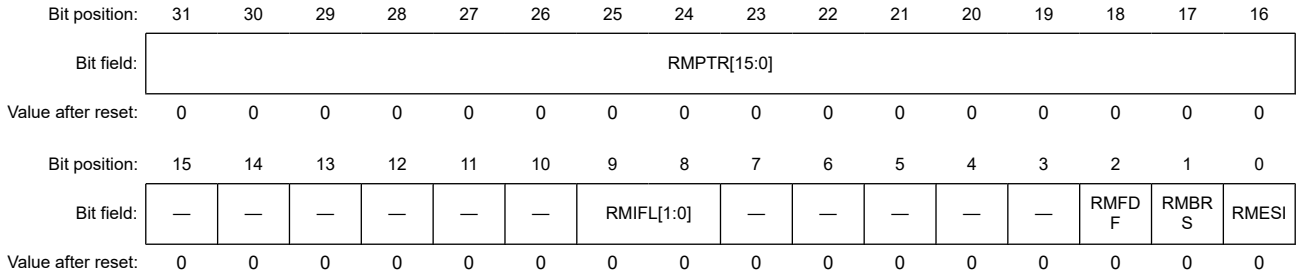
See Table 5 in ISO 11898-1 (2015) Specification for details in defining the number of data bytes that were received.

Note: The maximum capacity of the buffer belongs to CFDRMNB.RMPLS and this is not available in the classical CAN function.

28.2.62.5 CFDRMFDSTSb : RX Message Buffer CANFD Status Registers (b = 0 to 31)

Base address: CANFD_B = 0x400B_0000

Offset address: 0x0928 + 0x004C × b (b = 0 to 7)
0x0D28 + 0x004C × (b - 8) (b = 8 to 15)
0x01128 + 0x004C × (b - 16) (b = 16 to 23)
0x01528 + 0x004C × (b - 24) (b = 24 to 31)



Bit	Symbol	Function	R/W
0	RMESI*1	Error State Indicator bit 0: CANFD frame received from error active node 1: CANFD frame received from error passive node	R
1	RMBRS*1	Bit Rate Switch bit 0: CANFD frame received with no bit rate switch 1: CANFD frame received with bit rate switch	R

Bit	Symbol	Function	R/W
2	RMFDF*1	CAN FD Format bit 0: Non CANFD frame received 1: CANFD frame received	R
7:3	—	These bits are read as 0. The write value should be 0.	R
9:8	RMIFL[1:0]	RX Message Buffer Information Label Field	R
15:10	—	These bits are read as 0. The write value should be 0.	R
31:16	RMPTR[15:0]	RX Message Buffer Pointer Field	R

Note 1. This bit is not available in the classical CAN function.

The RX Message Buffer CANFD Status Register b (b = 0 to 31) show the status of the FDF, BRS and ESI bits, and pointer of the received CANFD frame.

RMESI bit (Error State Indicator bit)

The RMESI bit has the same value as the ESI bit of the received CANFD frame.

When the received FDF bit is 0, this means a CAN2.0 frame is received and 0 is stored to this bit.

Note: This bit is not available in the classical CAN function.

RMBRS bit (Bit Rate Switch bit)

The RMBRS bit has the same value as the BRS bit of the received CANFD frame.

When the received FDF bit is 0, this means a CAN2.0 frame is received and 0 is stored to this bit.

Note: This bit is not available in the classical CAN function.

RMFDF bit (CAN FD Format bit)

The RMFDF bit has the same value as the FDF bit of the received CANFD frame.

Note: This bit is not available in the classical CAN function.

RMIFL[1:0] bits (RX Message Buffer Information Label Field)

The RMIFL[1:0] bits store the information label value from the related Global Acceptance Filter List entry.

RMPTR[15:0] bits (RX Message Buffer Pointer Field)

The RMPTR[15:0] bits store the pointer value from the related Global Acceptance Filter List entry.

28.2.62.6 CFDRMDFb_p : RX Message Buffer Data Field p Registers (p = 0 to 15, b = 0 to 31)

Base address: CANFD_B = 0x400B_0000

Offset address: 0x092C + 0x004C × b + 0x0004 × p (b = 0 to 7 , p = 0 to 15)
0x0D2C + 0x004C × (b - 8) + 0x0004 × p (b = 8 to 15 , p = 0 to 15)
0x0112C + 0x004C × (b - 16) + 0x0004 × p (b = 16 to 23 , p = 0 to 15)
0x0152C + 0x004C × (b - 24) + 0x0004 × p (b = 24 to 31 , p = 0 to 15)

Bit position:	31 … 24	23 … 16	15 … 8	7 … 0
Bit field:	RMDB_HH[7:0]	RMDB_HL[7:0]	RMDB_LH[7:0]	RMDB_LL[7:0]
Value after reset:	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
7:0	RMDB_LL[7:0]	RX Message Buffer Data Byte (p × 4)	R
15:8	RMDB_LH[7:0]	RX Message Buffer Data Byte ((p × 4) + 1)	R
23:16	RMDB_HL[7:0]	RX Message Buffer Data Byte ((p × 4) + 2)	R
31:24	RMDB_HH[7:0]*1	RX Message Buffer Data Byte ((p × 4) + 3)	R

Note 1. These bits are not available in the classical CAN function.

The RX Message Buffer Data Field p Register b (p = 0 to 15, b = 0 to 31) store the data bytes (p × 4) to data bytes ((p × 4) + 3) of the received message.

RMDB_LL[7:0] bits (RX Message Buffer Data Byte (p × 4))

The RMDB_LL[7:0] bits store data bytes (p × 4) of the message in the RX message buffer.

Unused data bytes are filled with 0x00.

RMDB_LH[7:0] bits (RX Message Buffer Data Byte ((p × 4) + 1))

The RMDB_LH[7:0] bits store data bytes ((p × 4) + 1) of the message in the RX message buffer.

Unused Data Bytes will be filled with 0x00.

RMDB_HL[7:0] bits (RX Message Buffer Data Byte ((p × 4) + 2))

The RMDB_HL[7:0] bits store data bytes ((p × 4) + 2) of the message in the RX message buffer.

Unused data bytes are filled with 0x00.

RMDB_HH[7:0] bits (RX Message Buffer Data Byte ((p × 4) + 3))

The RMDB_HH[7:0] bits store data bytes ((p × 4) + 3) of the message in the RX message buffer.

Unused data bytes are filled with 0x00.

28.2.62.7 CFDRFMBCPb[0] : RX FIFO Access Message Buffer Component b (b = 0 to 1)

Base address: CANFD_B = 0x400B_0000

Offset address: see Table 28.5

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

Bit	Symbol	Function	R/W
31:0	Rc[31:0]	RX FIFO Access Message Buffer Component c See Table 28.8, Table 28.9 and the descriptions that follow for a detailed description of each register and its related bits, contained within this message buffer component.	R

Where the total number of CFDRFMBCPb = 2 as shown in Figure 28.29 (c = RX FIFO Access Message Buffer Component Register index = [0…18])

Rc[31:0] bits (RX FIFO Access Message Buffer Component c)

The RX FIFO Access Message Buffer component comprises of the following registers:

- CFDRFIDb
- CFDRFPTRb
- CFDRFFDSTSb
- CFDRFDFbp

See Table 28.9 for details on how to interpret the structure of this buffer component and how to access the respective registers.

Table 28.8 RX FIFO Access Message Buffer component summary (1 of 2)

Rc	
R0	RX FIFO Access ID Registers
R1	RX FIFO Access Pointer Register

Table 28.8 RX FIFO Access Message Buffer component summary (2 of 2)

Rc	
R2	RX FIFO Access CANFD Status Registers
R3	RX FIFO Access Data Field 0 Registers
R4	RX FIFO Access Data Field 1 Registers
R5	RX FIFO Access Data Field 2 Registers
R6	RX FIFO Access Data Field 3 Registers
R7	RX FIFO Access Data Field 4 Registers
R8	RX FIFO Access Data Field 5 Registers
R9	RX FIFO Access Data Field 6 Registers
R10	RX FIFO Access Data Field 7 Registers
R11	RX FIFO Access Data Field 8 Registers
R12	RX FIFO Access Data Field 9 Registers
R13	RX FIFO Access Data Field 10 Registers
R14	RX FIFO Access Data Field 11 Registers
R15	RX FIFO Access Data Field 12 Registers
R16	RX FIFO Access Data Field 13 Registers
R17	RX FIFO Access Data Field 14 Registers
R18	RX FIFO Access Data Field 15 Registers
R[19...31]	—

Table 28.9 RX Message Buffer Component (RMBCP) Detailed

Rc	p	Symbol	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R0	x	CFDRMIDb	RMIDE	RMRTR	—	RMID																												
R1	x	CFDRMPTRb	RMDLC				—	—	—	—	—	—	—	—	—	—	—	—	RMTS															
R2	x	CFDRMFDSTSb	RMPTR																—	—	—	—	—	—	RMIFL		—	—	—	—	—	RMFDF	RMBRS	RMESI
R3	0	CFDRMDFbp	RMDB_HH																RMDB_LH								RMDB_LL							
R[4…18]	[1…15]	CFDRMDFbp	RMDB_HH																RMDB_LH								RMDB_LL							

28.2.62.8 CFDRFIDb : RX FIFO Access ID Register b (b = 0 to 1)

Base address: CANFD_B = 0x400B_0000

Offset address: 0x0520 + 0x004C × b

Bit position:	31	30	28	0
Bit field:	RFIDE	RFRTR	—	RFID[28:0]
Value after reset:	0	0	0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
28:0	RFID[28:0]	RX FIFO Buffer ID Field STD-ID/EXT-ID fields	R
29	—	This bit is read as 0.	R
30	RFRTR	RX FIFO Buffer RTR bit 0: Data frame 1: Remote frame	R
31	RFIDE	RX FIFO Buffer IDE bit 0: STD-ID has been received 1: EXT-ID has been received	R

The RX FIFO Access ID Registers b (b = 0 to 1) store the ID field, IDE bit and RTR bit of the message.

RFID[28:0] bits (RX FIFO Buffer ID Field)

The RFID[28:0] bits are the bits of the STD-ID/EXT-ID fields of the message in the FIFO buffer.

For alignment of these bits in standard and extended frame format, see Identifier Bits Alignment.

RFRTR bit (RX FIFO Buffer RTR bit)

The RFRTR bit shows whether a data frame or a remote frame was stored in the FIFO buffer.

Note: There are no remote frames in CANFD format. When a CANFD frame was received, the register reflects the state of the received value (RRS bit in FD frame format).

RFIDE bit (RX FIFO Buffer IDE bit)

The RFIDE bit shows whether message with the Standard Identifier or Extended Identifier was received in the FIFO buffer.

28.2.62.9 CFDRFPTRb : RX FIFO Access Pointer Register b (b = 0 to 1)

Base address: CANFD_B = 0x400B_0000

Offset address: 0x0524 + 0x004C × b

Bit position:	31			28										15																0
Bit field:	RFDLC[3:0]				—	—	—	—	—	—	—	—	—	—	—	—	RFTS[15:0]														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	RFTS[15:0]	RX FIFO Timestamp Value Timestamp value of the received CAN frame	R
27:16	—	These bits are read as 0.	R
31:28	RFDLC[3:0]	RX FIFO Buffer DLC Field Number of data bytes received in a CAN frame	R

The FIFO Access Pointer Registers b (b = 0 to 1) store the DLC and Timestamp fields for the received message.

RFTS[15:0] bits (RX FIFO Timestamp Value)

The RFTS[15:0] bits store the timestamp value taken at the capture point as configured by the CFDGFDCFG.TSCCFG bit of the received message.

RFDLC[3:0] bits (RX FIFO Buffer DLC Field)

The RFDLC[3:0] bits store the number of data bytes that were received in the RX FIFO buffer.

See Table 5 in ISO 11898-1 (2015) Specification for details in defining the number of data bytes that were received.

28.2.62.10 CFDRFFDSTSb : RX FIFO Access CANFD Status Register b (b = 0 to 1)

Base address: CANFD_B = 0x400B_0000

Offset address: 0x0528 + 0x004C × b

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CFDRFPTR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
0	RFESI*1	Error State Indicator bit 0: CANFD frame received from error active node 1: CANFD frame received from error passive node	R
1	RFBRS*1	Bit Rate Switch bit 0: CANFD frame received with no bit rate switch 1: CANFD frame received with bit rate switch	R
2	RFFDF*1	CAN FD Format bit 0: Non CANFD frame received 1: CANFD frame received	R
7:3	—	These bits are read as 0.	R
9:8	RFIFL[1:0]	RX FIFO Buffer Information Label Field	R
15:10	—	These bits are read as 0.	R
31:16	CFDRFPTR[15:0]	RX FIFO Buffer Pointer Field	R

Note 1. This bit is not available in the classical CAN function.

The RX FIFO Access CANFD Status Registers b (b = 0 to 1) show the status of the FDF, BRS, and ESI bits, including the pointer of the received CANFD frame.

RFESI bit (Error State Indicator bit)

The RFESI bit has the same value as the ESI bit of the received CANFD frame.

When the received FDF bit is 0, this means a CAN2.0 frame is received and 0 is stored to this bit.

Note: This bit is not available in the classical CAN function.

RFBRS bit (Bit Rate Switch bit)

The RFBRS bit has the same value as the BRS bit of the received CANFD frame.

When the received FDF bit is 0, this means a CAN2.0 frame is received and 0 is stored to this bit.

Note: This bit is not available in the classical CAN function.

RFFDF bit (CAN FD Format bit)

The RFFDF bit has the same value as the FDF bit of the received CANFD frame.

Note: This bit is not available in the classical CAN function.

RFIFL[1:0] bits (RX FIFO Buffer Information Label Field)

The RFIFL[1:0] bits store the information label value from the related Global Acceptance Filter List entry.

CFDRFPTR[15:0] bits (RX FIFO Buffer Pointer Field)

The CFDRFPTR[15:0] bits store the pointer value from the related Global Acceptance Filter List entry.

28.2.62.11 CFDRFDFb_p : RX FIFO Access Data Field p Register b (p = 0 to 15, b = 0 to 1)

Base address: CANFD_B = 0x400B_0000

Offset address: 0x052C + 0x004 × p + 0x04C × b

Bit position:	31 … 24	23 … 16	15 … 8	7 … 0
Bit field:	RFDB_HH[7:0]	RFDB_HL[7:0]	RFDB_LH[7:0]	RFDB_LL[7:0]
Value after reset:	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
7:0	RFDB_LL[7:0]	RX FIFO Buffer Data Byte (p × 4)	R
15:8	RFDB_LH[7:0]	RX FIFO Buffer Data Byte ((p × 4) + 1)	R
23:16	RFDB_HL[7:0]	RX FIFO Buffer Data Byte ((p × 4) + 2)	R
31:24	RFDB_HH[7:0]	RX FIFO Buffer Data Byte ((p × 4) + 3)	R

The RX FIFO Access Data Field p Registers b (p = 0 to 15, b = 0 to 1) store data bytes ((p × 4) to data byte ((p × 4) + 3) of the received message.

RFDB_LL[7:0] bits (RX FIFO Buffer Data Byte (p × 4))

The RFDB_LL[7:0] bits store data bytes (p × 4) of the message present in the FIFO buffer.

Unused data bytes are filled with 0x00 according to the configured data payload size CFDRFCCa.RFPLS.

RFDB_LH[7:0] bits (RX FIFO Buffer Data Byte ((p × 4) + 1))

The RFDB_LH[7:0] bits store data bytes ((p × 4) + 1)) of the message present in the FIFO buffer.

Unused data bytes are filled with 0x00.

RFDB_HL[7:0] bits (RX FIFO Buffer Data Byte ((p × 4) + 2))

The RFDB_HL[7:0] bits store data bytes ((p × 4) + 2) of the message present in the FIFO buffer.

Unused data bytes are filled with 0x00.

RFDB_HH[7:0] bits (RX FIFO Buffer Data Byte ((p × 4) + 3))

The RFDB_HH[7:0] bits store data bytes ((p × 4) + 3) of the message present in the FIFO buffer.

Unused data bytes are filled with 0x00.

28.2.62.12 CFDCFMBCP0[0] : Common FIFO Access Message Buffer Component

Base address: CANFD_B = 0x400B_0000

Offset address: See Table 28.5

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

Bit	Symbol	Function	R/W
31:0	Rc[31:0]	Common FIFO Access Message Buffer Component c Refer to Table 28.10, Table 28.11 and the descriptions that follow for a detailed description of each register and its related bits, contained within this message buffer component.	R

Where the total number of CFDCFMBCP0 = 1 as shown in Figure 28.29 (c = Common FIFO Message Buffer Component Register index = [0…18])

Rc[31:0] bit (Common FIFO Access Message Buffer Component c)

The Common FIFO Access Message Buffer Component is made up of the following registers: CFDCFID, CFDCFPTR, CFFDSTS0, and CFDCFDFp. Refer to Table 28.11 for details of how to interpret the structure of this buffer component and how to access the respective registers.

Table 28.10 Common FIFO Access Message Buffer Component Summary

Common FIFO Access Message Buffer Component (CFMBCP)	
Rc	**CANFD mode (CAN_FD_MODE = 1'b1)**
R0	Common FIFO Access ID Registers
R1	Common FIFO Access Pointer Register
R2	Common FIFO Access CANFD Status Registers
R3	Common FIFO Access Data Field 0 Registers
R4	Common FIFO Access Data Field 1 Registers
R5	Common FIFO Access Data Field 2 Registers
R6	Common FIFO Access Data Field 3 Registers
R7	Common FIFO Access Data Field 4 Registers
R8	Common FIFO Access Data Field 5 Registers
R9	Common FIFO Access Data Field 6 Registers
R10	Common FIFO Access Data Field 7 Registers
R11	Common FIFO Access Data Field 8 Registers
R12	Common FIFO Access Data Field 9 Registers
R13	Common FIFO Access Data Field 10 Registers
R14	Common FIFO Access Data Field 11 Registers
R15	Common FIFO Access Data Field 12 Registers
R16	Common FIFO Access Data Field 13 Registers
R17	Common FIFO Access Data Field 14 Registers
R18	Common FIFO Access Data Field 15 Registers
R[19…31]	—

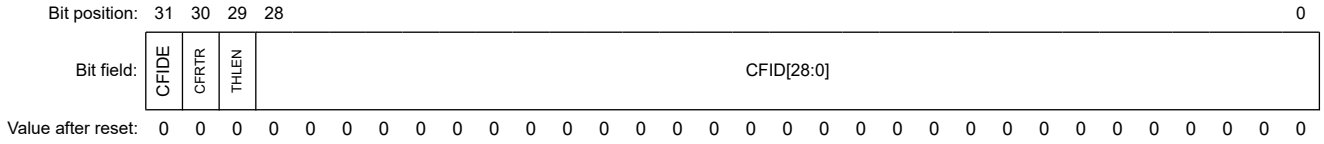
Table 28.11 Common FIFO Access Message Buffer Component (CFMBCP) Detailed

Rc	p	Symbol	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R0	x	CFDCFID	CFIDE	CFRTR	THLEN	CFID																												
R1	x	CFDCFPTR	CFDLC				—	—	—	—	—	—	—	—	—	—	—	—	CFTS															
R2	x	CFDCFFDCSTS	CFPTR																—	—	—	—	—	—	CFIFL	—	—	—	—	—	CFFDF	CFBRS	CFESI
R3	0	CFDCFDFp	CFDB_HH								CFDB_HL								CFDB_LH							CFDB_LL							
R[4…18]	[1…15]	CFDCFDFp	CFDB_HH								CFDB_HL								CFDB_LH							CFDB_LL							
R[19…31]	x	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

28.2.62.13 CFDCFID : Common FIFO Access ID Register

Base address: CANFD_B = 0x400B_0000

Offset address: 0x05B8



Bit	Symbol	Function	R/W
28:0	CFID[28:0]	Common FIFO Buffer ID Field STD-ID / EXT-ID fields	R/W
29	THLEN	THL Entry enable TX FIFO Mode: 0: Entry will not be stored in THL after successful TX. 1: Entry will be stored in THL after successful TX. RX FIFO Mode: Reserved, this bit is read as 0	R/W
30	CFRTR	Common FIFO Buffer RTR Bit 0: Data Frame 1: Remote Frame	R/W
31	CFIDE	Common FIFO Buffer IDE Bit 0: STD-ID will be transmitted or has been received 1: EXT-ID will be transmitted or has been received	R/W

The Common FIFO Access ID registers store the ID field, IDE bit and RTR bit of the message.

In TX mode, users can read data from the FIFO, only for the current entry based on the write pointer value, not for the other entries.

CFID[28:0] bit (Common FIFO Buffer ID Field)

These are the bits of the STD-ID / EXT-ID fields of the message in the FIFO Buffer.

In TX mode, users can write and read from FIFO buffers.

In RX mode, users can only read data from FIFO buffers.

THLEN bit (THL Entry enable)

This bit controls the storage of an entry corresponding to the transmitted message in the TX History list at the end of a successful transmission.

In TX mode, users can write and read from FIFO buffers.

In RX mode, users can only read data from FIFO buffers.

CFRTR bit (Common FIFO Buffer RTR Bit)

This bit selects whether a Data Frame or a Remote Frame will be transmitted from or was received in the FIFO Buffer.

Note: There are no remote frames in CANFD format. In case a CANFD frame was received (RX mode) the register reflects the state of the received value (RRS bit in FD frame format). In case of CANFD transmission (TX mode CFDCFID.CFFDF =1) the bit is always transmitted dominant (Data Frame).

In TX mode, users can write and read from FIFO buffers.

In RX mode, users can only read data from FIFO buffers.

CFIDE bit (Common FIFO Buffer IDE Bit)

This bit selects whether a message with EXT-ID or STD-ID will be transmitted from or was received in the FIFO Buffer.

In TX mode, users can write and read from FIFO buffers.

In RX mode, users can only read data from FIFO buffers.

28.2.62.14 CFDCFPTR : Common FIFO Access Pointer Register

Base address: CANFD_B = 0x400B_0000

Offset address: 0x05BC

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

Bit	Symbol	Function	R/W
15:0	CFTS[15:0]	Common FIFO Timestamp Value Timestamp value of the received CAN frame (FIFO in RX mode).	R/W
27:16	—	These bits are read as 0. The write value should be 0.	R/W
31:28	CFDLC[3:0]	Common FIFO Buffer DLC Field Number of data bytes received in a CAN frame, or to be transmitted in a CAN frame.	R/W

The Common FIFO Access Pointer Registers store the DLC and Timestamp fields.

In TX mode, you can read data from the FIFO buffer, only for the current entry based on the write pointer value, and not for the other entries.

CFTS[15:0] bits (Common FIFO Timestamp Value)

The CFTS[15:0] bits store the timestamp value taken at the capture point as configured by the CFDGFDCFG.TSCCFG bit of the received message (if FIFO is configured in RX mode).

In TX mode, you can read and write from FIFO buffers.

In RX mode, you can only read data from FIFO buffers.

CFDLC[3:0] bits (Common FIFO Buffer DLC Field)

The CFDLC[3:0] bits store the number of data bytes that were received in the FIFO buffer or are to be transmitted.

See Table 5 in ISO 11898-1 (2015) Specification for details in defining the number of data bytes.

In TX mode, you can read and write from the FIFO buffers. Do not read data for the other entries in the FIFO when configured in TX mode.

In RX mode, you can only read data from the FIFO buffers.

28.2.62.15 CFDCFFDCSTS : Common FIFO Access CANFD Control/Status Register

Base address: CANFD_B = 0x400B_0000

Offset address: 0x05C0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CFPTR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
0	CFESI*1	Error State Indicator bit 0: CANFD frame received or to transmit by error active node 1: CANFD frame received or to transmit by error passive node	R/W

Bit	Symbol	Function	R/W
1	CFBRS*1	Bit Rate Switch bit 0: CANFD frame received or to transmit with no bit rate switch 1: CANFD frame received or to transmit with bit rate switch	R/W
2	CFFDF*1	CAN FD Format bit 0: Non CANFD frame received or to transmit 1: CANFD frame received or to transmit	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
9:8	CFIFL[1:0]	COMMON FIFO Buffer Information Label Field	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
31:16	CFPTR[15:0]	Common FIFO Buffer Pointer Field	R/W

Note 1. This bit is not available in the classical CAN function.

The Common FIFO Access CANFD Control/Status Registers show the status of the FDF, BRS and ESI bits, including the pointer of the received CANFD frame or the CANFD frame to transmit.

In TX mode, you can read data from the FIFO, only for the current entry based on the write pointer value, and not for the other entries.

CFESI bit (Error State Indicator bit)

In TX mode, you can read and write from FIFO buffers. In this mode, when the CANFD module is not in error passive, the CFESI bit equals the write value. Otherwise, it is a don’t care and the bit is transmitted as 1 on the CAN bus, indicating this is an error passive node.

In RX mode, you can only read data from FIFO buffers.

In RX mode, the CFESI bit is updated with the ESI bit value of the CANFD frame when it has been received, indicating the error state of the transmitting node. In RX mode, 0 is stored to this bit when the received FDF bit is 0, this means a CAN 2.0 frame is received.

Note: This bit is not available in the classical CAN function.

CFBRS bit (Bit Rate Switch bit)

In TX mode, you can read and write from FIFO buffers. In this mode, the CANFD module either transmits a 0 to indicate no bit rate switch in the frame to be transmitted or a 1 to indicate a bit rate switch in the frame to be transmitted.

In RX mode, you can only read data from FIFO buffers.

In RX mode, the CFBRS bit is updated with the BRS bit value of the CANFD frame when it has been received, indicating whether there is a bit rate switch (1) or (0) on the CANFD frame.

In RX mode, 0 is stored to the CFBRS bit when the received FDF bit is 0, this means a CAN 2.0 frame is received.

Note: This bit is not available in the classical CAN function.

CFFDF bit (CAN FD Format bit)

In TX mode, you can read and write from FIFO buffers. In this mode, the CANFD module either transmits a 0 to indicate a CAN 2.0 frame is to be transmitted or a 1 to indicate a CANFD frame is to be transmitted.

In RX mode, you can only read data from FIFO buffers.

In RX mode, the CFFDF bit is updated with the FDF bit value of the CAN frame when it has been received, indicating whether it is a CAN 2.0 frame (0) or a CANFD frame (1).

Note: This bit is not available in the classical CAN function.

CFIFL[1:0] bits (COMMON FIFO Buffer Information Label Field)

If the Common FIFO is configured in TX mode, the value programmed in CFDCFFDCSTS.CFIFL[1:0] is stored together with additional message information, to the TX History List after successful transmission of the message.

The information label value from the related Global Acceptance Filter List entry is stored in these bits (if FIFO is configured in either RX mode).

In TX mode, you can read and write from FIFO buffers.

In RX mode, you can only read data from FIFO buffers.

CFPTR[15:0] bits (Common FIFO Buffer Pointer Field)

If the Common FIFO is configured in TX mode, the value programmed in CFDCFFDCSTS.CFPTR[15:0] is stored together with additional message information, to the TX History List after successful transmission of the message.

The pointer value from the related Global Acceptance Filter List entry is stored in these bits (if FIFO is configured in either RX mode).

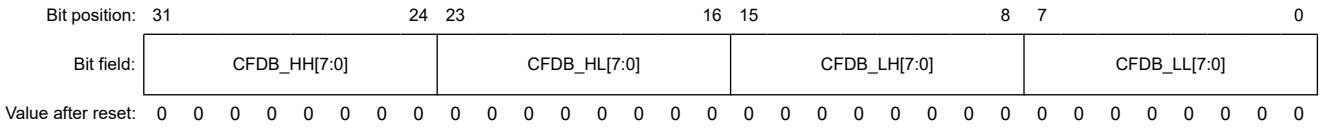
In TX mode, you can read and write from FIFO buffers.

In RX mode, you can only read data from FIFO buffers.

28.2.62.16 CFDCFDFp : Common FIFO Access Data Field p Registers (p = 0 to 15)

Base address: CANFD_B = 0x400B_0000

Offset address: 0x05C4 + 0x004 × p



Bit	Symbol	Function	R/W
7:0	CFDB_LL[7:0]	Common FIFO Buffer Data Bytes (p × 4)	R/W
15:8	CFDB_LH[7:0]	Common FIFO Buffer Data Bytes ((p × 4) + 1)	R/W
23:16	CFDB_HL[7:0]	Common FIFO Buffer Data Bytes ((p × 4) + 2)	R/W
31:24	CFDB_HH[7:0]	Common FIFO Buffer Data Bytes ((p × 4) + 3)	R/W

The FIFO Access Data Field p Registers (p = 0 to 15) store data bytes (p × 4) to data bytes ((p × 4) + 3) of the message.

In TX mode, you can read data from the FIFO, only for the current entry based on the write pointer value, and not for the other entries.

CFDB_LL[7:0] bits (Common FIFO Buffer Data Bytes (p × 4))

The CFDB_LL[7:0] bits store data bytes (p × 4) of the message present in the FIFO buffer.

In TX mode, you can read and write from the FIFO buffers.

In RX mode, you can only read data from the FIFO buffers.

In RX mode, unused data bytes are filled with 0x00, according to their configured data payload size CFDCFCC.CFPLS.*1

CFDB_LH[7:0] bits (Common FIFO Buffer Data Bytes ((p × 4) + 1))

The CFDB_LH[7:0] bits store data bytes ((p × 4) + 1) of the message present in the FIFO buffer.

In TX mode, you can read and write from the FIFO buffers.

In RX mode, you can only read data from the FIFO buffers.

In RX mode, unused data bytes are filled with 0x00, according to their configured data payload size CFDCFCC.CFPLS.*1

CFDB_HL[7:0] bits (Common FIFO Buffer Data Bytes ((p × 4) + 2))

The CFDB_HL[7:0] bits store data bytes ((p × 4) + 2) of the message present in the FIFO buffer.

In TX mode, you can read and write from the FIFO buffers.

In RX mode, you can only read data from the FIFO buffers.

In RX mode, unused data bytes are filled with 0x00, according to their configured data payload size CFDCFCC.CFPLS.*1

CFDB_HH[7:0] bits (Common FIFO Buffer Data Bytes ((p × 4) + 3))

The CFDB_HH[7:0] bits store data bytes ((p × 4) + 3) of the message present in the FIFO buffer.

In TX mode, you can read and write from the FIFO buffers.

In RX mode, you can only read data from the FIFO buffers.

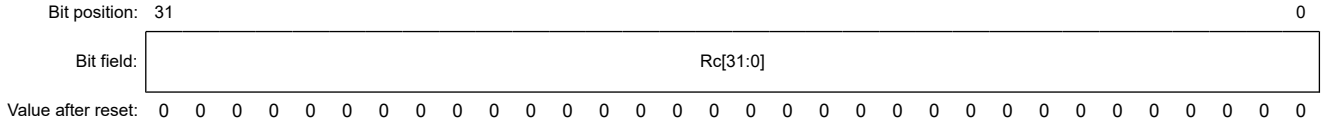
In RX mode, unused data bytes are filled with 0x00, according to their configured data payload size CFDCFCC.CFPLS.*1

Note 1. In RX mode, unused data bytes are filled with 0x00 according to the configured data payload size CFDCFCC.CFPLS, which is a CANFD feature not found in classical CAN.

28.2.62.17 CFDTMBCPb[0] : TX Message Buffer Component b (b = 0 to 3)

Base address: CANFD_B = 0x400B_0000

Offset address: See Table 28.5



Bit	Symbol	Function	R/W
31:0	Rc[31:0]	TX Message Buffer Component c Refer to Table 28.12, Table 28.13 and the descriptions that follow for a detailed description of each register and its related bits, contained within this message buffer component.	R

Where the total number of CFDTMBCPn = 4 as shown in Figure 28.29 (c = TX Message Buffer Component Register index = [0…18])

Rc[31:0] bit (TX Message Buffer Component c)

TX Message Buffer Component c

The TX Message Buffer Component is made up of the following registers: CFDTMIDb, CFDTMPTRb, CFDTMFDCTRb, and CFDTMDFbp. Refer to Table 28.13 for details of how to interpret the structure of this buffer component and how to access the respective registers.

Table 28.12 TX Message Buffer Component Summary (1 of 2)

TX Message Buffer Component (TMBCP)	
Rc	**CANFD mode (CAN_FD_MODE = 1'b1)**
R0	TX Message Buffer (b) ID Registers CHn
R1	TX Message Buffer (b) Pointer Registers CHn
R2	TX Message Buffer (b) CANFD Status Registers CHn
R3	TX Message Buffer (b) Data Field 0 Registers CHn
R4	TX Message Buffer (b) Data Field 1 Registers CHn
R5	TX Message Buffer (b) Data Field 2 Registers CHn
R6	TX Message Buffer (b) Data Field 3 Registers CHn
R7	TX Message Buffer (b) Data Field 4 Registers CHn
R8	TX Message Buffer (b) Data Field 5 Registers CHn
R9	TX Message Buffer (b) Data Field 6 Registers CHn
R10	TX Message Buffer (b) Data Field 7 Registers CHn
R11	TX Message Buffer (b) Data Field 8 Registers CHn
R12	TX Message Buffer (b) Data Field 9 Registers CHn
R13	TX Message Buffer (b) Data Field 10 Registers CHn

Table 28.12 TX Message Buffer Component Summary (2 of 2)

TX Message Buffer Component (TMBCP)	
Rc	**CANFD mode (CAN_FD_MODE = 1'b1)**
R14	TX Message Buffer (b) Data Field 11 Registers CHn
R15	TX Message Buffer (b) Data Field 12 Registers CHn
R16	TX Message Buffer (b) Data Field 13 Registers CHn
R17	TX Message Buffer (b) Data Field 14 Registers CHn
R18	TX Message Buffer (b) Data Field 15 Registers CHn
R[19…31]	—

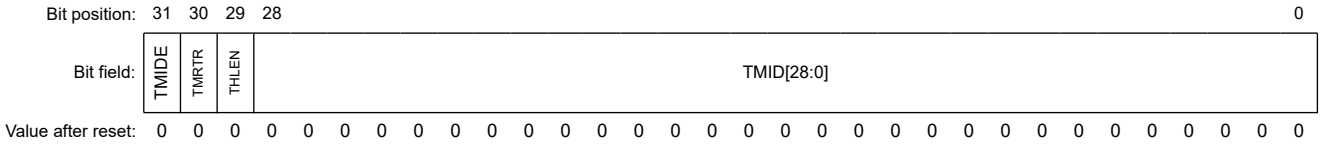
Table 28.13 TX Message Buffer Component (TMBCP) Detailed

Rc	p	Symbol	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R0	x	CFDTMIDb	TMIDE	TMRTR	THLEN	TMID																												
R1	x	CFDTMPTRb	TMDLC				—	—	—	—	—	—	—	—	—	—	—	—	CFTS															
R2	x	CFDTMFDCTRb	TMPTR																—	—	—	—	—	—	TMIFL						TMFDF	TMBRS	TMESI
R3	0	CFDTMDFbp	TMDB_HH								TMDB_HL								TMDB_LH								TMDB_LL						
R[4…18]	[1…15]	CFDTMDFbp	TMDB_HH								TMDB_HL								TMDB_LH								TMDB_LL						

28.2.62.18 CFDTMIDb : TX Message Buffer ID Registers (b = 0 to 3)

Base address: CANFD_B = 0x400B_0000

Offset address: 0x0604 + 0x004C × b



Bit	Symbol	Function	R/W
28:0	TMID[28:0]	TX Message Buffer ID Field STD-ID/EXT-ID fields	R/W
29	THLEN	Tx History List Entry 0: Entry not stored in THL after successful TX 1: Entry stored in THL after successful TX	R/W
30	TMRTR	TX Message Buffer RTR bit 0: Data frame 1: Remote frame	R/W
31	TMIDE	TX Message Buffer IDE bit 0: STD-ID is transmitted 1: EXT-ID is transmitted	R/W

Each TX Message Buffer ID Register b (b = 0 to 3) are used to store the ID, IDE, RTR fields and history configuration of the message to be transmitted from the associated buffer.

TMID[28:0] bits (TX Message Buffer ID Field)

The TMID[28:0] bits are bits of the STD-ID/EXT-ID fields of the message stored in this TX message buffer.

Do not write to these bits when the related CANFD channel is in CH_SLEEP mode.

THLEN bit (Tx History List Entry)

The THLEN bit controls the storage of an entry corresponding to the transmitted message in the TX History list at the end of a successful transmission.

Do not write to these bits when the related CANFD channel is in CH_SLEEP mode.

TMRTR bit (TX Message Buffer RTR bit)

The TMRTR bit selects whether a data frame or remote frame is to be transmitted from this TX message buffer.

Note: There are no remote frames in CANFD format. For a CANFD transmission (CFDTMFDCTRb.CFFDF = 1), this bit is always transmitted dominant (data frame).

Do not write to these bits when the related CANFD channel is in CH_SLEEP mode.

TMIDE bit (TX Message Buffer IDE bit)

The TMIDE bit selects whether a message with EXT-ID or STD-ID is to be transmitted from this TX message buffer.

Do not write to these bits when the related CANFD channel is in CH_SLEEP mode.

28.2.62.19 CFDTMPTRb : TX Message Buffer Pointer Register (b = 0 to 3)

Base address: CANFD_B = 0x400B_0000

Offset address: 0x0608 + 0x004C × b

Bit position:	31			28																										0
Bit field:	TMDLC[3:0]				—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
27:0	—	The read values are undefined. The write value should be 0.	R/W
31:28	TMDLC[3:0]	TX Message Buffer DLC Field Number of data bytes to be transmitted in a CAN frame.	R/W

Each TX Message Buffer Pointer Register b (b = 0 to 3) is used to store the DLC fields of the message to transmit from the associated buffer.

TMDLC[3:0] bits (TX Message Buffer DLC Field)

The TMDLC[3:0] bits select the number of data bytes to be transmitted from this TX message buffer when the corresponding TMRTR bit is configured as 0.

See Table 5 in ISO 11898-1 (2015) Specification for details in defining the number of data bytes to be transmitted.

Do not write to these bits when the related CANFD channel is in CH_SLEEP mode.

28.2.62.20 CFDTMFDCTRb : TX Message Buffer CANFD Control Register (b = 0 to 3)

Base address: CANFD_B = 0x400B_0000

Offset address: 0x060C + 0x004C × b

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	TMPTR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
0	TMESI*1	Error State Indicator bit 0: CANFD frame to transmit by error active node 1: CANFD frame to transmit by error passive node	R/W
1	TMBRS*1	Bit Rate Switch bit 0: CANFD frame to transmit with no bit rate switch 1: CANFD frame to transmit with bit rate switch	R/W
2	TMFDF*1	CAN FD Format bit 0: Non CANFD frame to transmit 1: CANFD frame to transmit	R/W
7:3	—	The read values are undefined. The write value should be 0.	R/W
9:8	TMIFL[1:0]	TX Message Buffer Information Label Field	R/W
15:10	—	The read values are undefined. The write value should be 0.	R/W
31:16	TMPTR[15:0]	TX Message Buffer Pointer Field	R/W

Note 1. This bit is not available in the classical CAN function.

The TX Message Buffer CANFD Control Registers b (b = 0 to 3) show the status of the FDF, BRS and ESI bits, including the pointer fields of the CANFD frame to be transmitted.

TMESI bit (Error State Indicator bit)

If the channel is not in error passive, then the TMESI bit equals the write value, otherwise it is a don't care and the bit is transmitted as 1 on the CAN bus, indicating this is an error passive node.

Do not write to the TMESI bit when the related CANFD channel is in CH_SLEEP mode.

Note: This bit is not available in the classical CAN function.

TMBRS bit (Bit Rate Switch bit)

Do not write to the TMBRS bit when the related CANFD channel is in CH_SLEEP mode.

Note: This bit is not available in the classical CAN function.

TMFDF bit (CAN FD Format bit)

Do not write to the TMFDF bit when the related CANFD channel is in CH_SLEEP mode.

Note: This bit is not available in the classical CAN function.

TMIFL[1:0] bits (TX Message Buffer Information Label Field)

The TMIFL[1:0] bits store the information label value to be copied, together with additional message information, in the TX History List after successful transmission of the message.

Do not write to these bits when the related CANFD channel is in CH_SLEEP mode.

TMPTR[15:0] bits (TX Message Buffer Pointer Field)

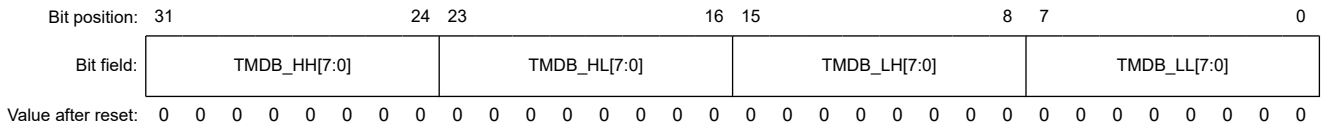
The TMPTR[15:0] bits store the pointer value to be copied, together with additional message information in the TX History List after successful transmission of the message.

Do not write to these bits when the related CANFD channel is in CH_SLEEP mode.

28.2.62.21 CFDTMDFb_p : TX Message Buffer Data Field Register (p= 0 to 15 , b= 0 to 3)

Base address: CANFD_B = 0x400B_0000

Offset address: 0x0610 + 0x004 × p + 0x004C × b



Bit	Symbol	Function	R/W
7:0	TMDB_LL[7:0]	TX Message Buffer Data Byte (p × 4)	R/W
15:8	TMDB_LH[7:0]	TX Message Buffer Data Byte ((p × 4) + 1)	R/W
23:16	TMDB_HL[7:0]	TX Message Buffer Data Byte ((p × 4) + 2)	R/W
31:24	TMDB_HH[7:0]	TX Message Buffer Data Byte ((p × 4) + 3)	R/W

Each TX Message Buffer Data Field p Register b (p = 0 to 15, b = 0 to 3) is used to store data bytes (p × 4) to data bytes ((p × 4) + 3) of the message to transmit from the associated buffer.

TMDB_LL[7:0] bits (TX Message Buffer Data Byte (p × 4))

TMDB_LL[7:0] bits store data bytes (p × 4) of the message in the TX message buffer.

Do not write to these bits when the related CANFD channel is in CH_SLEEP mode.

TMDB_LH[7:0] bits (TX Message Buffer Data Byte ((p × 4) + 1))

TMDB_LH[7:0] bits store data bytes ((p × 4) + 1) of the message in the TX message buffer.

Do not write to these bits when the related CANFD channel is in CH_SLEEP mode.

TMDB_HL[7:0] bits (TX Message Buffer Data Byte ((p × 4) + 2))

TMDB_HL[7:0] bits store data bytes ((p × 4) + 2) of the message in the TX message buffer.

Do not write to these bits when the related CANFD channel is in CH_SLEEP mode.

TMDB_HH[7:0] bits (TX Message Buffer Data Byte ((p × 4) + 3))

TMDB_HH[7:0] bits store data bytes ((p × 4) + 3) of the message in the TX message buffer.

Do not write to these bits when the related CANFD channel is in CH_SLEEP mode.

28.3 Modes of Operation

28.3.1 Overview

The modes of the CANFD module can be classified into 2 groups:

- Global modes
- Channel modes

28.3.2 Global Modes

These modes are applicable for the complete CANFD module and therefore are called Global modes. The global modes of the CANFD module are:

- Global Sleep
- Global Reset
- Global Halt
- Global Operation.

Figure 28.2 shows the possible transitions between the Global modes.

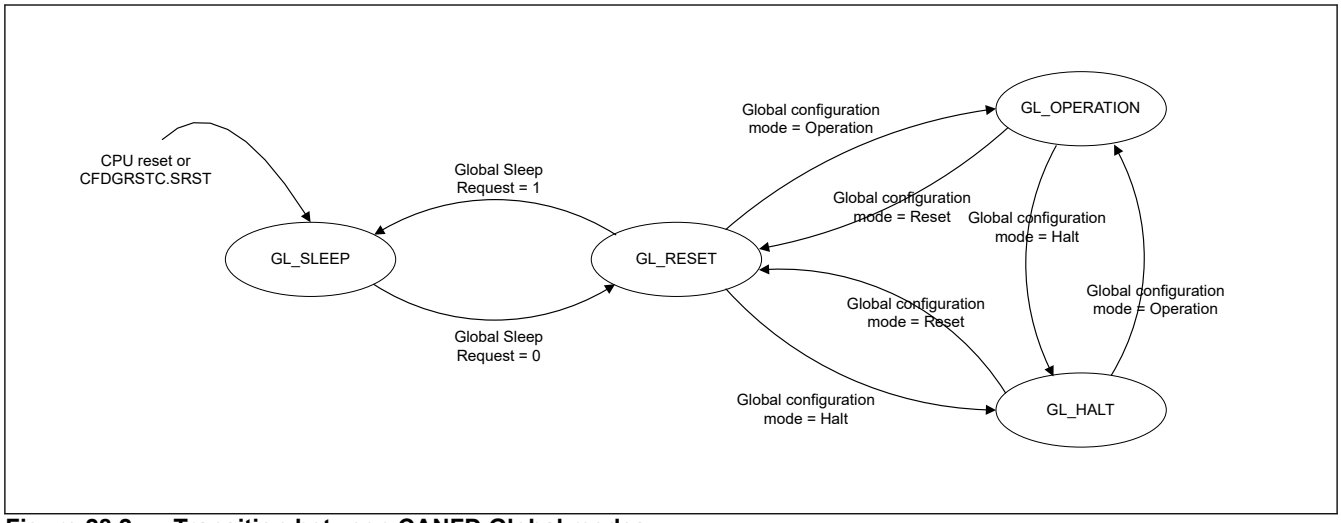


Figure 28.2 Transition between CANFD Global modes

Change in the Global mode can affect the Channel mode. Table 28.14 shows the effect of a Global mode transition on a Channel mode.

Table 28.14 Possible CANFD Channel modes and Global modes

Current Global mode	Target Global mode			
	Sleep	**Reset**	**Halt**	**Operation**
Sleep		Ch-Sleep: Keep Ch-Reset: N/A Ch-Halt: N/A Ch-Oper: N/A		
Reset	Ch-Sleep: Keep Ch-Reset: → Ch-Sleep Ch-Halt: N/A Ch-Oper: N/A		Ch-Sleep: Keep Ch-Reset: Keep Ch-Halt: N/A Ch-Oper: N/A	Ch-Sleep: Keep Ch-Reset: Keep Ch-Halt: N/A Ch-Oper: N/A
Halt		Ch-Sleep: Keep Ch-Reset: Keep Ch-Halt: → Ch-Reset Ch-Oper: N/A		Ch-Sleep:Keep Ch-Reset: Keep Ch-Halt: Keep Ch-Oper: N/A
Operation		Ch-Sleep: Keep Ch-Reset: Keep Ch-Halt: → Ch-Reset Ch-Oper: → Ch-Reset	Ch-Sleep: Keep Ch-Reset: Keep Ch-Halt: Keep Ch-Oper: → Ch-Halt	

28.3.2.1 Global Sleep Mode

After the release of a hardware reset or after setting and clearing a CFDGRSTC.SRST bit, the CANFD module automatically enters Global Sleep mode.

The CANFD module also enters the Global Sleep mode when the Global Sleep Request bit is set while it is in Global Reset mode. This control bit cannot be set in Global Halt mode or Global Operation mode.

Setting the Global Sleep Request bit sets Channel Sleep Request bit and forces the channel into the Channel Sleep mode.

Sleep mode is used for power saving purpose. When CANFD module is in Global Sleep mode, only the clock for CPU write access to the Global Sleep Mode Request bit is active. All other clocks are stopped and all other functions of the CANFD module are suspended.

Read access from all registers is still possible and all register values are preserved.

After setting the Global Sleep Request bit, it is necessary to confirm that the Global Sleep status has been updated, indicating successful transition to Global Sleep mode before the Global Sleep Request bit can be cleared again.

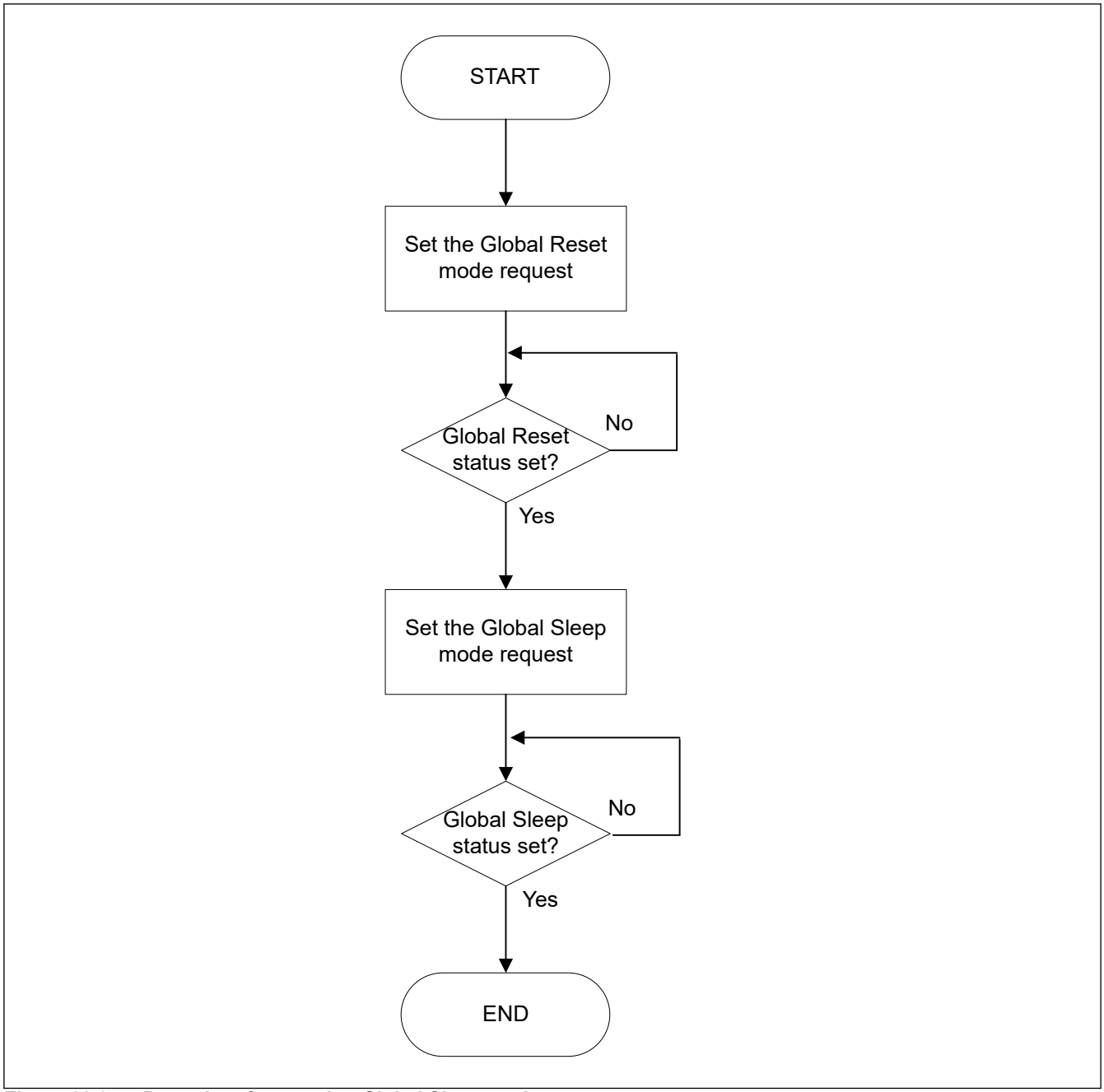


Figure 28.3 Procedure for entering Global Sleep mode

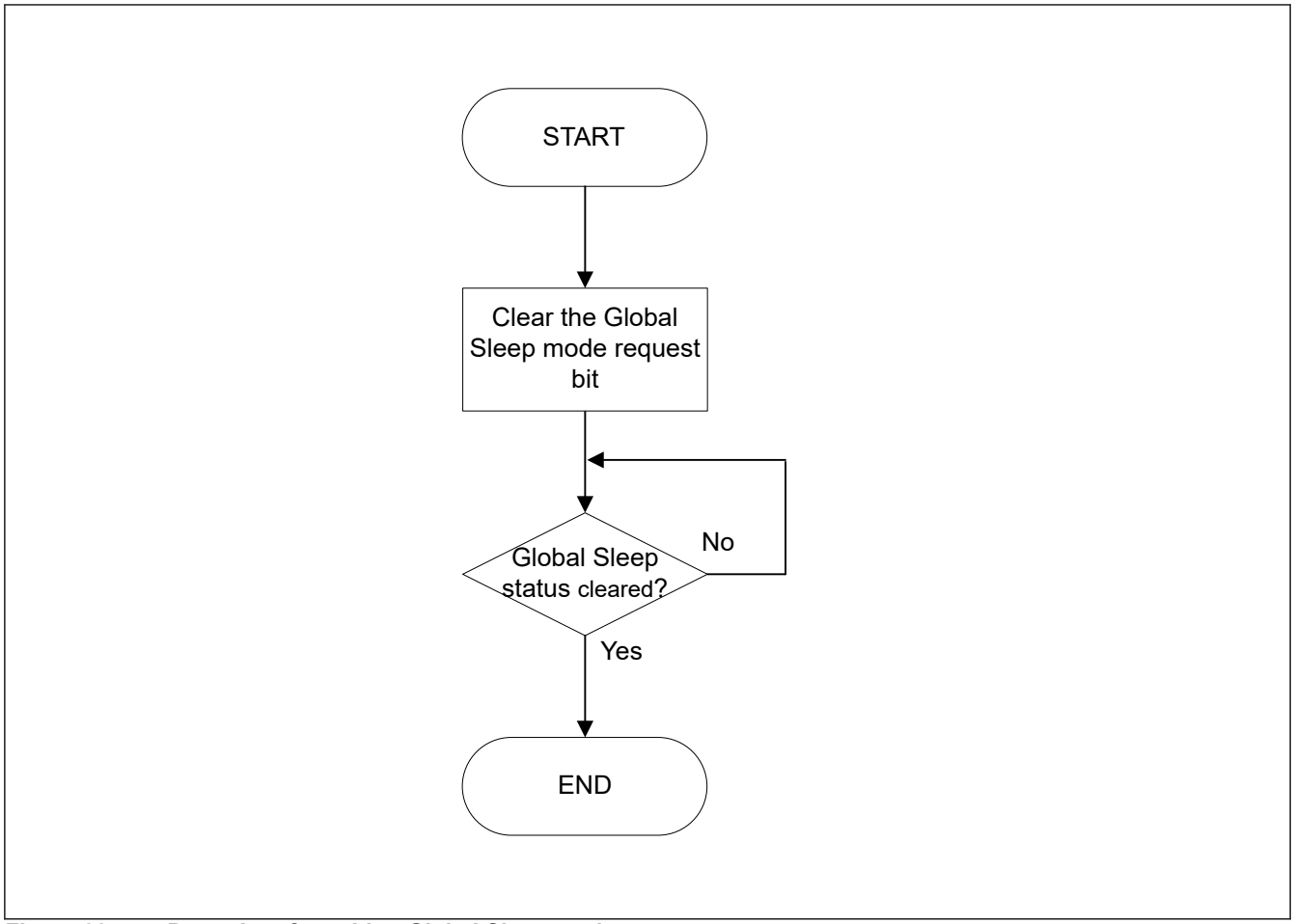


Figure 28.4 Procedure for exiting Global Sleep mode

28.3.2.2 Global Reset Mode

The CANFD module enters this mode in the following ways:

- Global Mode Control bit CFDGCTR.GMDC in the Global Control Register is configured for Global Reset mode while the CANFD module is in Global Halt or Global Operation mode
- Global Sleep Mode Request bit is cleared while CANFD module is in Global Sleep mode.

In Global Reset mode, all CANFD module functions are suspended and all status and flag registers are initialized.

Additionally all FIFOs and TX Queues are disabled and transmission control bits are cleared.

Configuration registers (except the test mode registers) are not initialized in this mode to their MCU reset values and the CANFD module can be configured.

See section 28.3.4. Global Mode and Channel Mode Transition Interactions for a detailed description of the behavior of all registers when transition to Global Reset mode is performed.

Setting the Global mode to Reset by setting the Global Mode Control bits CFDGCTR.GMDC in the Global Control Register to 01b sets Channel Mode Control bits CFDC0CTR.CHMDC in the Channel Control Registers to 01b and forces the channel into the Channel Reset mode.

For channels that are already in Channel Reset mode or Channel Sleep mode, this automatic transition is not performed (CFDC0CTR.CHMDC of related channel already set to 01b).

After setting Global Mode Control bit CFDGCTR.GMDC to Reset mode, it is necessary to confirm that the Reset Mode Status bit CFDGSTS.GRSTSTS in the Global Status Register has been updated, indicating successful transition to Global Reset mode before CFDGCTR.GMDC can be changed again.

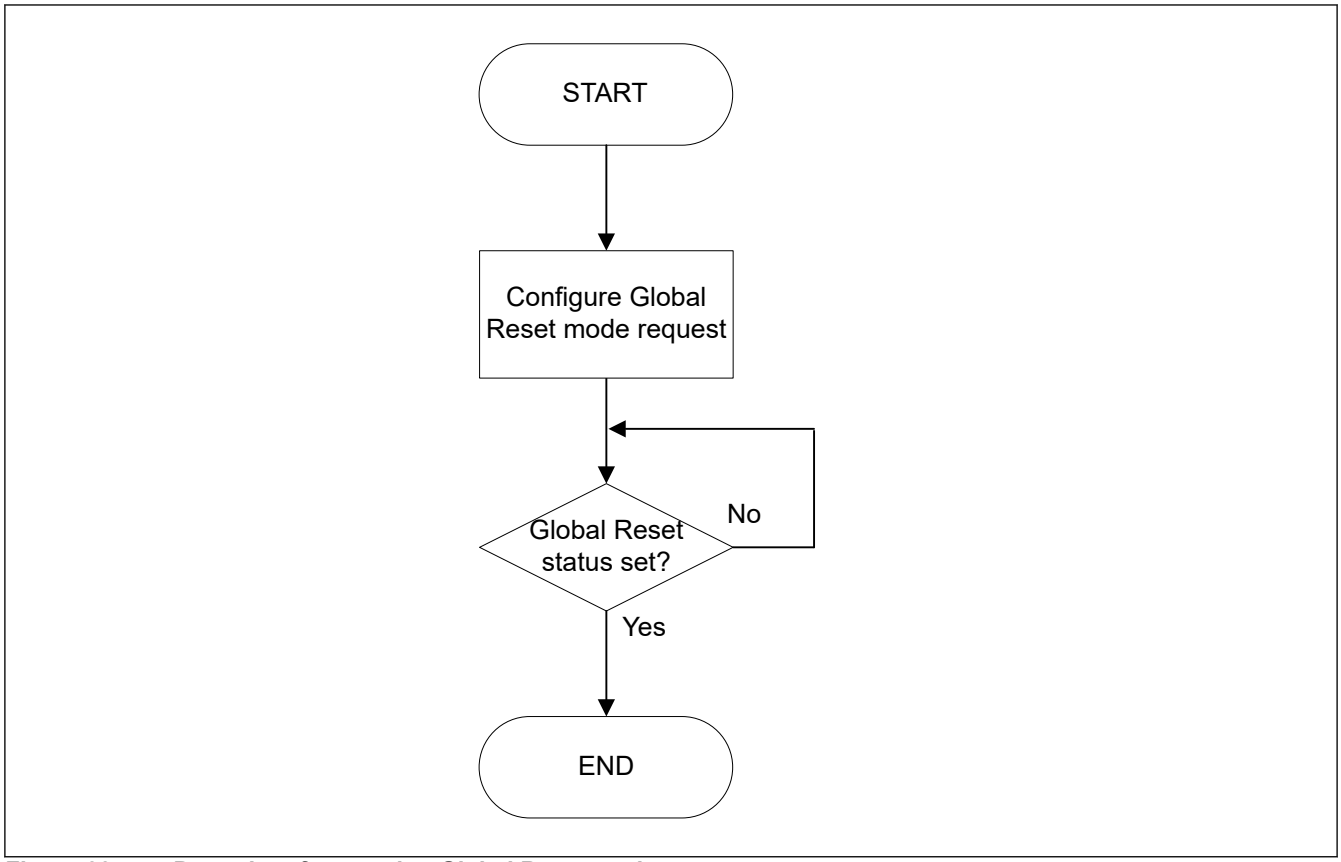


Figure 28.5 Procedure for entering Global Reset mode

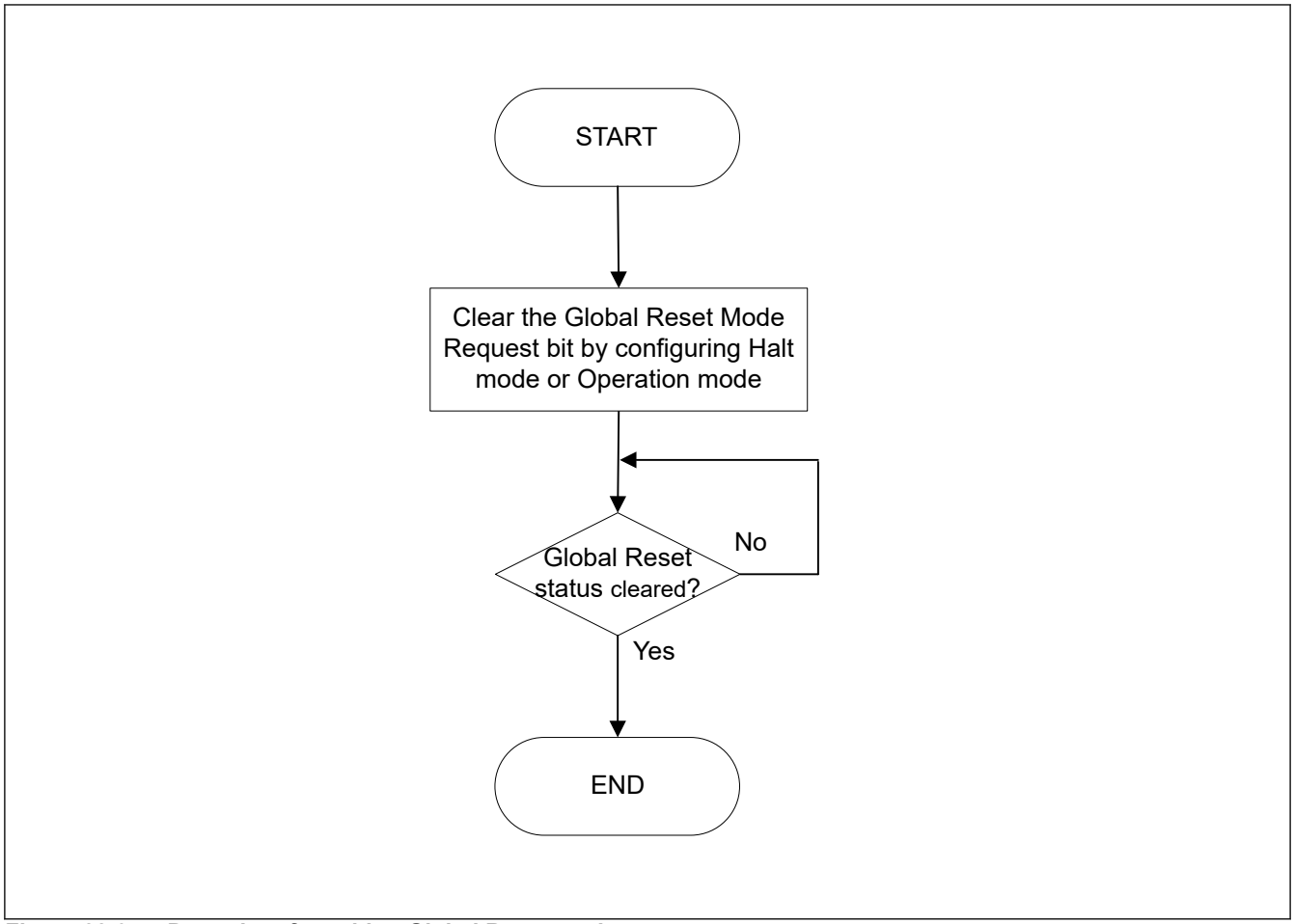


Figure 28.6 Procedure for exiting Global Reset mode

28.3.2.3 Global Halt Mode

The CANFD module enters this mode in the following ways:

- Global Mode Control bit CFDGCTR.GMDC in the Global Control Register is configured for Global Halt mode while the CANFD module is in Global Reset mode:
  - the channel in either Channel Reset or Channel Sleep mode remains in this mode
- Global Mode Control bit CFDGCTR.GMDC in the Global Control Register is configured for Global Halt mode while the CANFD module is in Global Operation mode:
  - the channel in Channel Reset, Channel Halt, or Channel Sleep mode remains in this mode
  - the channel in Channel Operation mode transitions to Channel Halt mode
  - Global Halt Mode Status bit is set when the channel has left Channel Operation mode.

If a transmission or reception is ongoing for a channel, the transition to Channel Halt mode is delayed until completion of the communication.

Similarly, if a channel is in bus-off, the full bus-off recovery sequence may be delayed depending on the channel configuration.

In Global Halt mode, all communications are suspended and CANFD logic does not cause any change to the Status and Flag registers (only when a channel is in the bus-off that its REC and TEC values are cleared). Additionally, the test mode configuration and control registers are not initialized in this mode.

The Global Halt mode should be used to configure global module test modes.

See section 28.3.4. Global Mode and Channel Mode Transition Interactions for a detailed description of the behavior of all registers when transition to Global Halt mode is performed.

Setting the Global mode to Halt by setting the Global Mode Control bit CFDGCTR.GMDC in the Global Control Register to 10b sets Channel Mode Control bits CFDC0CTR.CHMDC in the Channel Control Registers to 10b for the channel that are in Channel Operation mode and forces these channels into the Channel Halt mode.

For the channel that are already in Channel Reset, Channel Halt, or Channel Sleep mode, this automatic transition is not performed.

Therefore, the Global Halt mode request can be used to shut down all CANFD channel communications without loss of messages and disruption on the related CAN bus (no interruption of reception/transmission processes on the channels).

After setting the Global Mode Control bit CFDGCTR.GMDC to Halt mode, it is necessary to confirm that the Halt Mode Status bit CFDGSTS.GHLTSTS in the Global Status Register has been updated to indicate a successful transition to Global Halt mode. Do not specify any other SFR setting until confirming CFDGSTS.GHLTSTS is set.

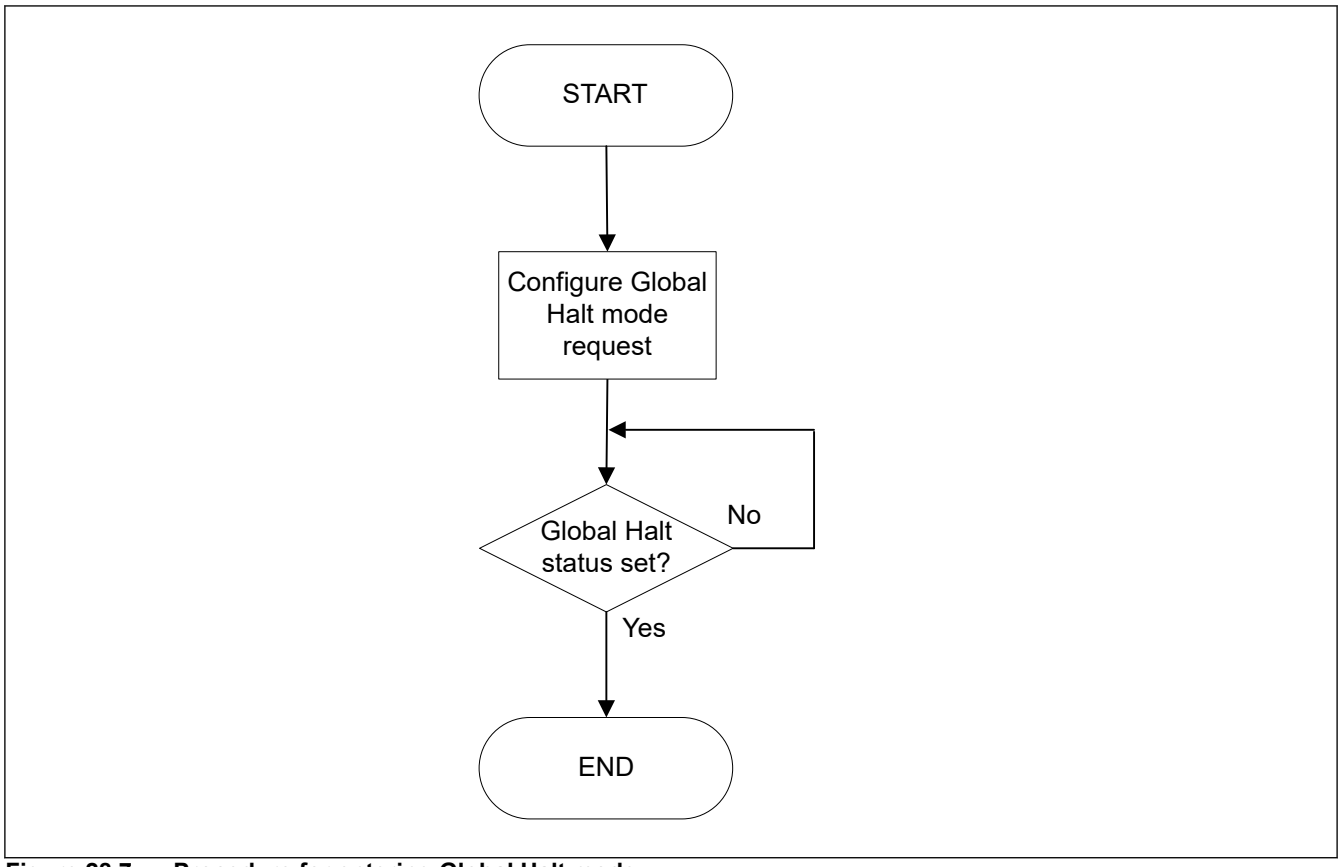


Figure 28.7 Procedure for entering Global Halt mode

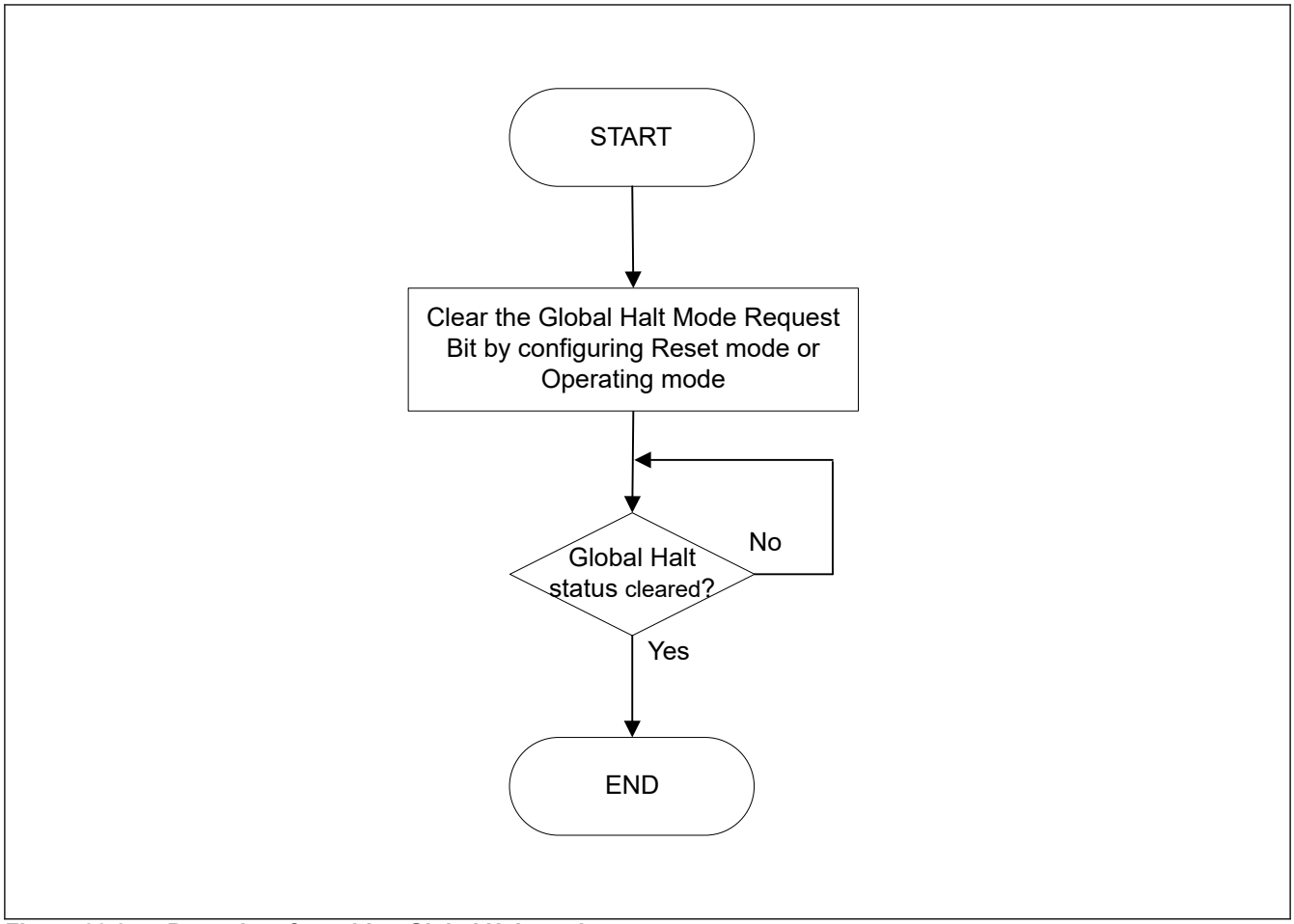


Figure 28.8 Procedure for exiting Global Halt mode

28.3.2.4 Global Operation Mode

The CANFD module enters this mode when the Global Mode Configuration bits are set to Global Operation mode.

The CANFD channel can only be set to Channel Operation mode and start CAN communication when CANFD is in Global Operation mode.

After setting the Global Mode Control bit CFDGCTR.GMDC to Global Operation mode, it is necessary to confirm that the Global Reset Mode Status bit CFDGSTS.GRSTSTS and the Global Halt Mode Status bit CFDGSTS.GHLTSTS in the Global Status Register have been cleared to indicate a successful transition to Global Operation mode before CFDGCTR.GMDC can be modified again.

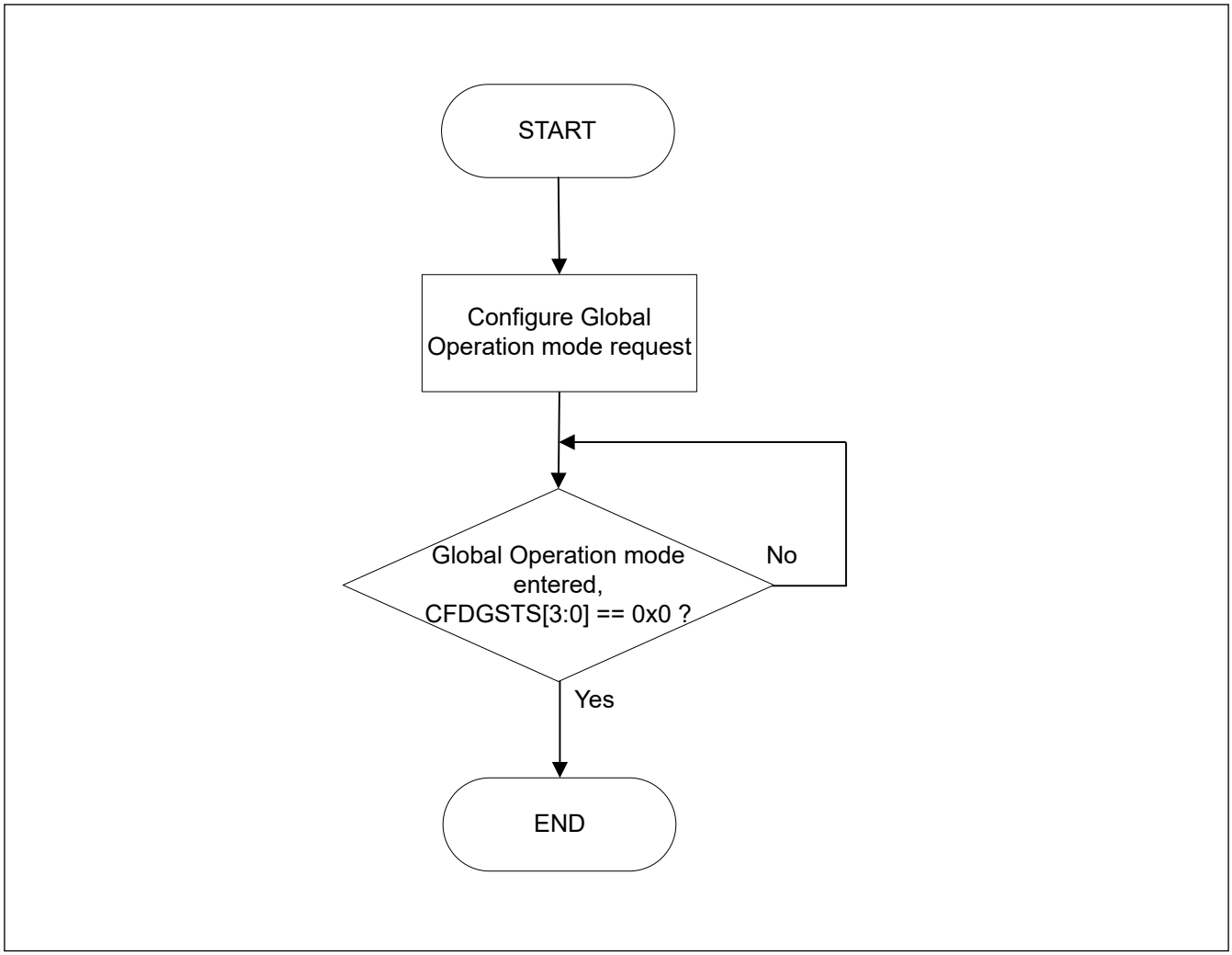


Figure 28.9 Procedure for entering Global Operation mode

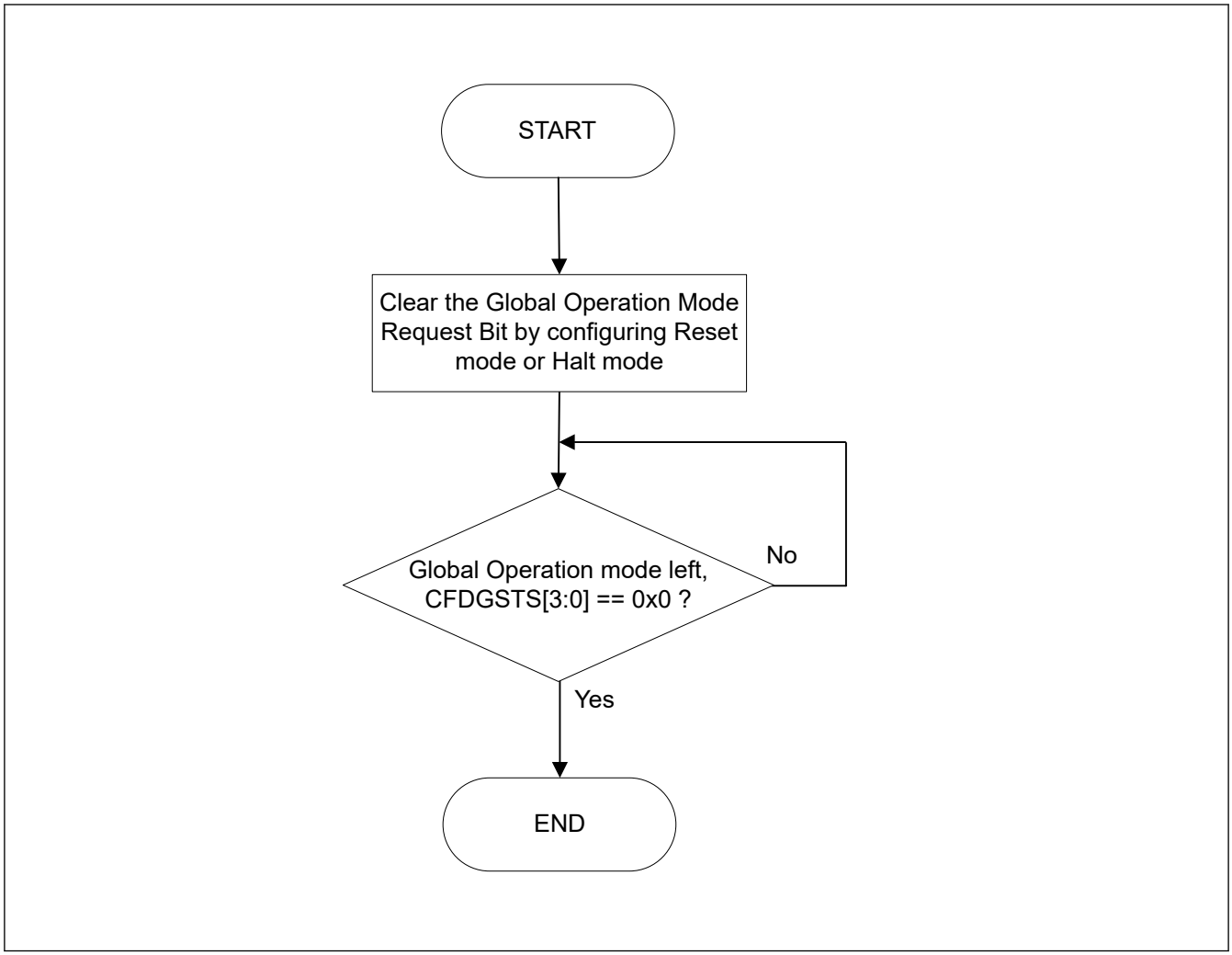


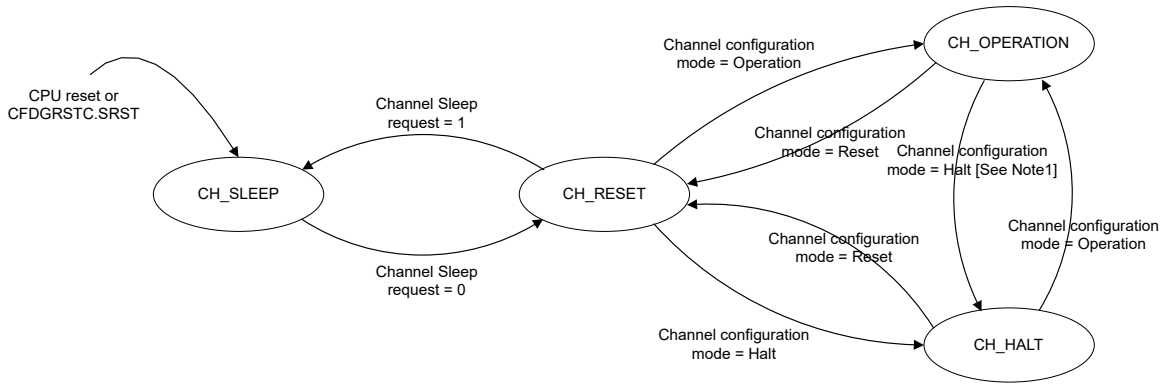
Figure 28.10 Procedure for exiting Global Operation mode

28.3.3 Channel Modes

A CAN channel can be in one of the following four channel modes:

- Reset
- Halt
- Operation
- Sleep.

Figure 28.11 shows the possible transitions between the channel modes.



Note 1. The transition timing depends on BOM setting in the Channel Configuration Register:
- when BOM = 01b, the state transition timing is immediate after entering bus-off state
- when BOM = 10b, the state transition timing is at the end of bus-off state
- when BOM = 11b, the state transition timing is matching with the setting of CHMDC to Halt mode.

Figure 28.11 Transition between CAN channel modes

28.3.3.1 CAN Channel Sleep Mode

After the release of a hardware reset or after setting and clearing the CFDGRSTC.SRST bit, a CAN channel of the CANFD module automatically enters Channel Sleep mode.

A CAN channel also enters Channel Sleep mode when the related Channel Sleep Mode Request bit is set while the CAN channel is in Channel Reset mode. Do not set this control bit in Channel Halt mode or Channel Operation mode.

Entering the CAN Channel Sleep mode instantly stops the clock supplied to the CAN channel unit and therefore reduces power consumption.

After setting the Channel Sleep Mode Request bit, it is necessary to confirm that the Channel Sleep mode status has been updated to indicate a successful transition to Channel Sleep mode before the Channel Sleep Mode Request bit can be cleared again.

During Channel Sleep mode, do not write to channel related registers. Read operation is still possible.

28.3.3.2 CAN Channel Reset Mode

A CANFD CAN channel enters this mode in the following ways:

- Channel Mode Control bit CFDC0CTR.CHMDC in the Channel Control Registers is configured for Channel Reset mode while the related CAN channel is in Channel Halt mode or Channel Operation mode
- Channel Sleep Mode Request bit is cleared while the related CAN channel is in Channel Sleep mode
- Global Mode Control bit CFDGCTR.GMDC is set to Global Reset mode and CAN channel is not in Channel Sleep mode or Channel Reset mode.

In Channel Reset mode, all CAN channel status and flag registers are initialized.

Additionally all channel related transmission control bits are cleared and the channel related TX Queue is disabled.

Configuration registers (except the Channel Test Mode registers) are not initialized in this mode and the CAN channel can be configured for communication.

See section 28.3.4. Global Mode and Channel Mode Transition Interactions for a detailed description of the behavior of all registers when transition to Channel Reset mode is performed.

After setting the Channel Mode Control bit CFDC0CTR.CHMDC to Channel Reset mode, it is necessary to confirm that the Reset Mode Status bit CFDC0STS.CRSTSTS in the related Channel Status Registers has been updated to indicate a successful transition to Channel Reset mode before the related CFDC0CTR.CHMDC bit can be modified again.

See Table 28.15 for the behavior of transitioning to Channel Reset mode while CAN communication is ongoing.

28.3.3.3 CAN Channel Halt Mode

A CANFD CAN channel enters this mode in the following ways:

- Channel Mode Control bit CFDC0CTR.CHMDC in the Channel Control Registers is configured for Channel Halt mode while the related CAN channel is in Channel Reset mode or Channel Operation mode
- Global Mode Control bit CFDGCTR.GMDC is set to Global Halt mode and CAN channel is in Channel Operation mode.

In Channel Halt mode, all channel CAN communication is suspended but all status and flag registers remain unchanged during Channel Halt mode entry (except for the bus-off case where REC and TEC values are cleared for this channel).

In addition, the Channel Test Mode Configuration and Control registers are not initialized in this mode.

The Channel Halt mode should be used to configure channel test modes.

See section 28.3.4. Global Mode and Channel Mode Transition Interactions for a detailed description of the behavior of all registers when transition to Channel Halt mode is performed.

After setting the Channel Mode Control bit CFDC0CTR.CHMDC to Channel Halt mode, it is necessary to confirm that the Halt Mode Status bit CFDC0STS.CHLTSTS in the related Channel Status Register has been updated to indicate a successful transition to Channel Halt mode before the related CFDC0CTR.CHMDC can be modified again.

See Table 28.15 for the transition behavior to Channel Halt mode while CAN communication is ongoing.

Table 28.15 Transition behavior in CAN Reset mode and Halt mode

Mode	State		
	Receiver	**Transmitter**	**Bus-Off**
CAN Channel Reset mode (CFDC0CTR.CHMDC = 01b)	The CAN channel enters Channel Reset mode without waiting for the completion of the ongoing reception.*1	The CAN channel enters Channel Reset mode without waiting for the completion of the ongoing transmission.*1	The CAN channel enters Channel Reset mode without waiting for the completion of the bus-off recovery.
CAN Channel Halt mode (CFDC0CTR.CHMDC = 10b)	CAN channel enters Channel Halt mode at the end of the ongoing reception or error.*2	CAN channel enters Channel Halt mode after completion of the ongoing transmission.	When CFDC0CTR.BOM is set to 00b, a Channel Halt mode request is accepted only after the completion of the full bus-off recovery sequence. When CFDC0CTR.BOM is set to 10b, the CAN channel transits automatically to Channel Halt mode after waiting for the completion of the bus-off recovery. When CFDC0CTR.BOM is set to 01b, the CAN channel transits automatically to Channel Halt mode without waiting for the completion of the bus-off recovery. When CFDC0CTR.BOM is set to 11b, the CAN channel enters Channel Halt mode as soon as Channel Halt mode is requested (without waiting for the completion of the bus-off recovery).

Note 1. If the entry to Channel Reset mode is required only at the end of an ongoing communication, then Channel Halt mode can be requested first to prevent interruption of CAN communication by direct transition to Channel Reset mode. After the CAN channel enters Channel Halt mode, the Channel Reset mode can be requested.

Note 2. If CAN communication is locked at dominant level after an error flag, software can detect this situation by monitoring the channel related BusLock flag and resolve lock condition by setting the CAN channel to Channel Reset mode.

28.3.3.4 CAN Channel Operation Mode

The Channel Operation mode is activated by setting the CFDC0CTR.CHMDC bits to 00b. If 11 consecutive recessive bits are detected after entering the CAN Operation mode, the CFDC0STS.COMSTS bit is set and the CAN channel:

- Enables the functions of the channel communication by allowing the channel to become an active node on the CAN network
- Releases the internal fault confinement logic including receive and transmit error counters

At this point, the CAN channel can start transmission and reception of CAN messages.

Within the CAN Channel Operation mode, the channel may be in four different sub-modes, depending on which type of communication functions are performed (see Figure 28.12):

- Channel idle: The CAN channel is neither receiving nor transmitting
- Channel receives: The channel is receiving a CAN message sent by another CAN node
- Channel transmits: The channel is transmitting a CAN message

  Note: The channel may receive its own message simultaneously when Self-test mode is enabled.
- Channel is in bus-off state: The CAN channel is cut-off from CAN bus communication.

After setting the Channel Mode Control bit CFDC0CTR.CHMDC to Channel Operation mode, it is necessary to confirm that the Channel Reset Mode Status bit CFDC0STS.CRSTSTS and the Channel Halt Mode Status bit CFDC0STS.CHLTSTS in the Channel Status Register have been updated to indicate a successful transition to Channel Operation mode before the related CFDC0CTR.CHMDC bit can be changed again.

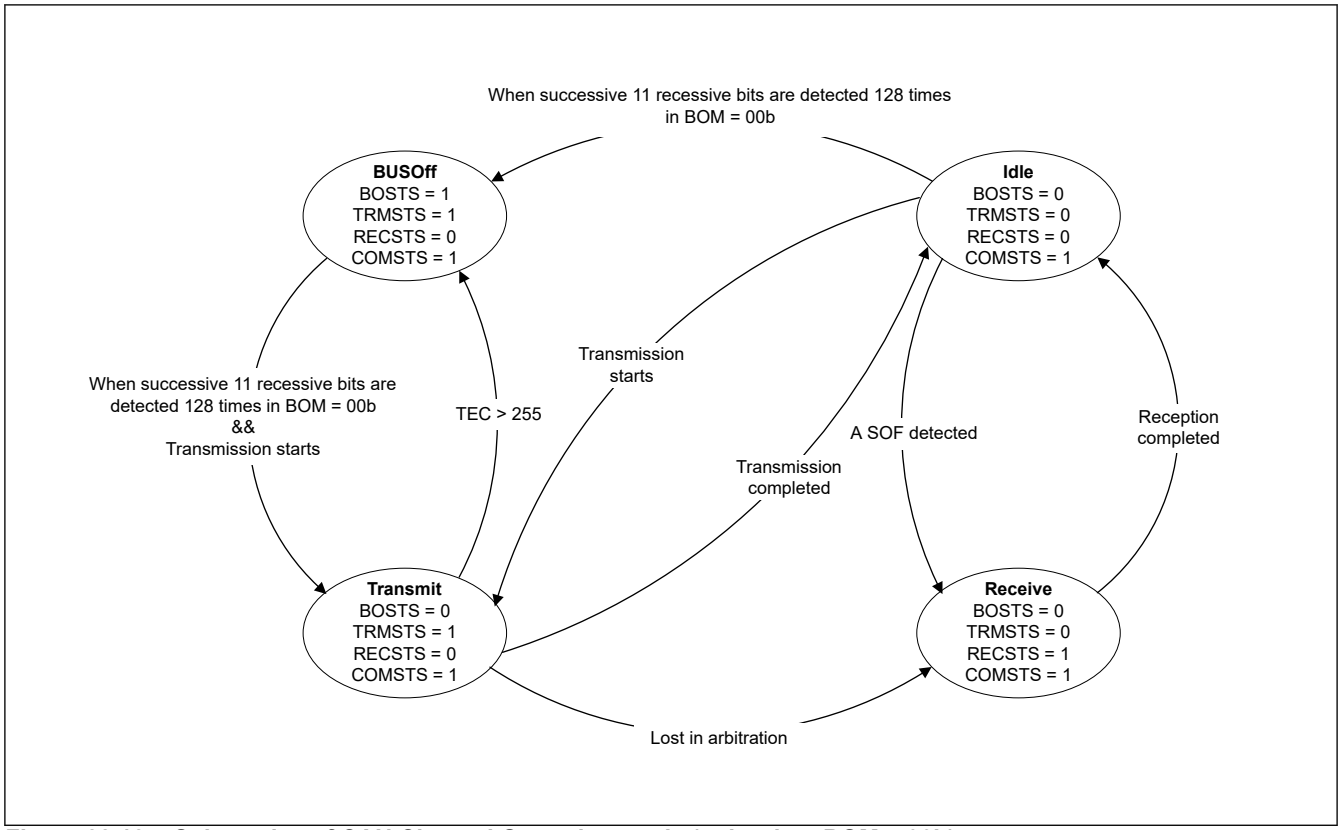


Figure 28.12 Sub-modes of CAN Channel Operation mode (only when BOM = 00b)

28.3.3.5 CAN Channel Bus-Off State

The CAN channel bus-off state is entered according to the fault confinement rules of the CAN specification. The following modes can be configured for returning to the CAN Channel Operation mode from the bus-off state:

- CFDC0CTR.BOM = 00b:
  Bus-Off recovery is compliant to ISO 11898-1, namely the CAN channel re-enters CAN communication (error active state) after 11 consecutive recessive bits are detected 128 times. TEC and REC counters are initialized to 0. The Bus-Off Recovery Flag CFDC0ERFL.BORF is set in this case.
- CFDC0CTR.BOM = 01b:

The CAN channel changes the value of the CFDC0CTR.CHMDC bits within the CAN Channel Control Register to 10b and switches immediately to Channel Halt mode automatically after entering bus-off state. TEC and REC counters are initialized to 0 and the Bus-Off Recovery Flag CFDC0ERFL.BORF is not set in this case.

- CFDC0CTR.BOM = 10b:
  The CAN channel changes the value of the CFDC0CTR.CHMDC bits within the CAN Channel Control Register to 10b as soon as it reaches bus-off state and enters Channel Halt mode automatically after the CAN channel has completed the bus-off recovery sequence (after 11 consecutive recessive bits are detected 128 times). TEC and REC counters are initialized to 0 and the Bus-Off Recovery Flag CFDC0ERFL.BORF is set in this case.
- CFDC0CTR.BOM = 11b:
  Bus-off recovery is initiated but CAN channel can immediately enter Channel Halt mode when still in bus-off state if a request is made to enter Channel Halt mode.
  TEC and REC counters are initialized to 0 and the Bus-Off Recovery Flag CFDC0ERFL.BORF is not set.
  Without setting CFDC0CTR.CHMDC [1:0] = 10b and when 11 recessive bits is detected 128 times continuously, transition conditions become the same as CFDC0CTR.BOM = 00b.

Note: If the recovery from bus-off occurs normally in this mode (after waiting for 128 sequences of 11 consecutive recessive bits), and no halt request has been generated during this period, then the Bus-Off Recovery flag CFDC0ERFL.BORF is set.

When software writes to the CFDC0CTR.CHMDC bit at the same time as the CAN channel enters Halt mode (at the start of bus-off when CFDC0CTR.BOM = 01b, or at the end of bus-off when CFDC0CTR.BOM = 10b), the software request has the highest priority.

Note: In the above case, the automatic setting of the CFDC0CTR.CHMDC bit to Channel Halt mode request is performed when the CFDC0CTR.CHMDC bit value is previously 00b (Channel Operation mode).

Additionally, it is possible to force the CAN channel to recover from the bus-off state by setting CFDC0CTR.RTBO to 1. The error state changes from bus-off state to integrating state with a maximum delay of 1 CAN bit time, and the CAN communication becomes possible again after 11 consecutive recessive bits are detected. The Bus-Off Recovery Flag is not set in this case, and the TEC and REC counters are initialized to 0.

Before setting CFDC0CTR.RTBO to 1, all pending transmissions from the TX message buffers, TX Queues and/or Common FIFO in TX mode should be disabled.

The disable of the pending transmission message buffer, TX Queue or FIFO must be confirmed by the corresponding acknowledge flags.

For the TX message buffer, the acknowledge flags are the Transmission Result Flags (CFDTMSTSj.TMTRF). For the TX Queue, it is the TX Queue Empty flag (CFDTXQSTS.TXQEMP). For the FIFO, it is the FIFO Empty flag (CFDCFSTS.CFEMP).

The CFDC0CTR.RTBO bit should be used for bus-off recovery only when CFDC0CTR.BOM is set to 00b.

Setting this bit in any state other than bus-off has no effect and the bit is cleared immediately.

Table 28.16 shows the settings for the Bus-Off Entry flag CFDC0ERFL.BOEF and the Bus-Off Recovery flag CFDC0ERFL.BORF for the different configurations of CFDC0CTR.BOM.

Table 28.16 Behavior of Bus-off Entry and Recovery flags

BOM	BOEF bit set	BORF bit set
00b	Always (on entry to bus-off)	Always (on exit from bus-off)
00b CFDC0CTR.RTBO set to 1	Always (on entry to bus-off)	Only if normal bus-off recovery occurs before software sets CFDC0CTR.RTBO to 1'
01b	Always (on entry to bus-off)	Never
10b	Always (on entry to bus-off)	Always (on exit from bus-off)
11b	Always (on entry to bus-off)	Only if normal bus-off recovery occurs before software issues a Halt request

For an efficient software procedure, it is not necessary to wait for the bus-off recovery sequence to end.

It is possible to perform a transmission re-initialization during bus-off recovery. To do this, follow the recommended software flow in Figure 28.13.

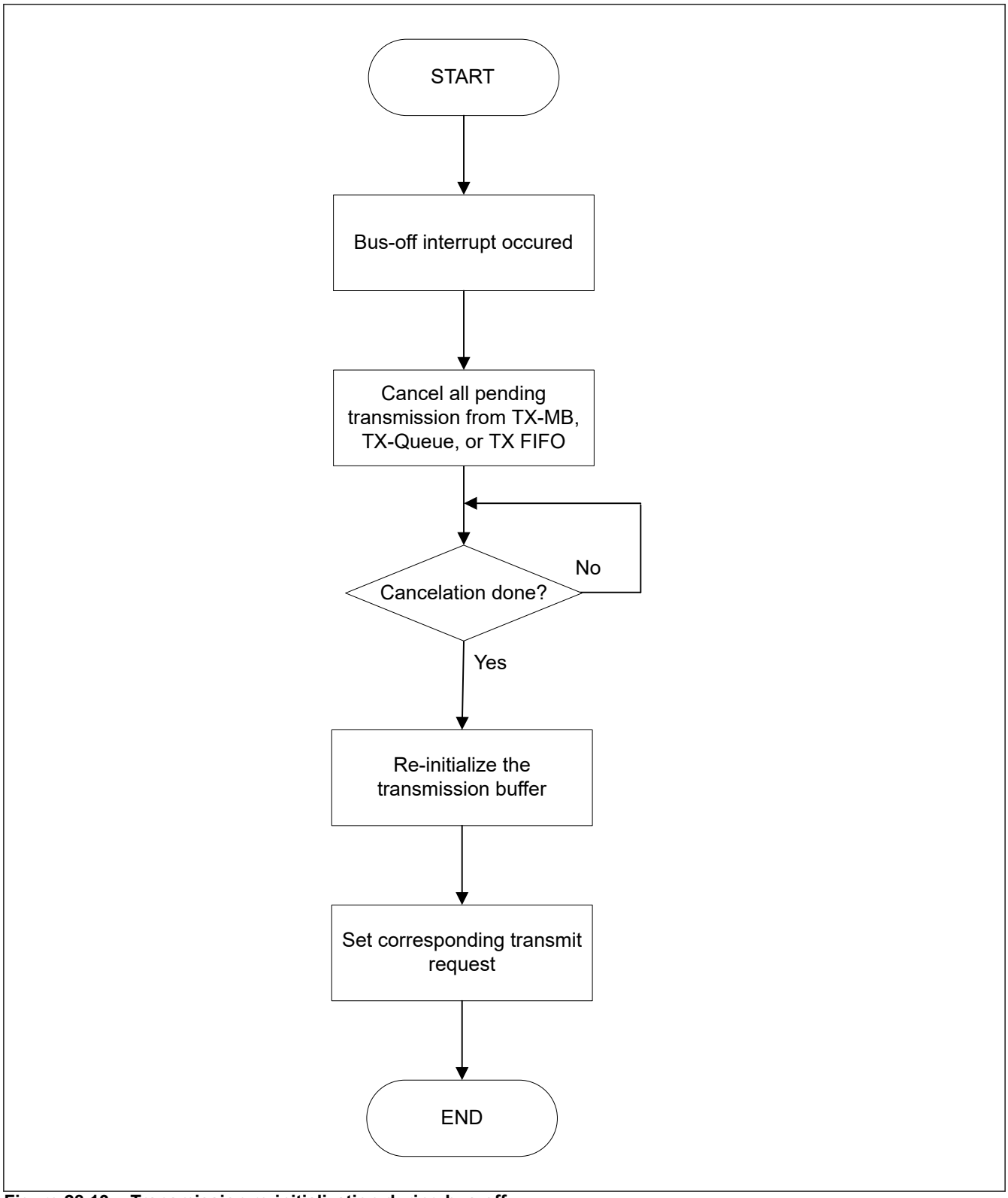


Figure 28.13 Transmission re-initialization during bus-off

28.3.4 Global Mode and Channel Mode Transition Interactions

The interaction between Global mode setting and Channel mode setting is as follows:

- Changing the Channel Mode Control bit CFDC0CTR.CHMDC in the Channel Control Registers does not affect the Global Mode Control bit CFDGCTR.GMDC.
- Changing the Global Mode Control bit CFDGCTR.GMDC affects the channel mode control as described in Table 28.17.

Table 28.17 Interaction between Global and Channel mode transition

Global mode change	Channel mode	Channel mode transition action
Sleep → Reset	Sleep	Channel remains in Sleep mode
Sleep → Halt	— (Global mode change not possible)	
Sleep → Operation	— (Global mode change not possible)	
Reset → Sleep	Sleep	Channel remains in Sleep mode
	Reset	Channel Sleep request bit is set automatically, channel enters Sleep Mode
Reset → Halt	Sleep	Channel remains in Sleep mode
	Reset	Channel remains in Reset mode
Reset → Operation	Sleep	Channel remains in Sleep mode
	Reset	Channel remains in Reset mode
Halt → Sleep	— (Global mode change not possible)	
Halt → Reset	Sleep	Channel remains in Sleep mode
	Reset	Channel remains in Reset mode
	Halt	Channel mode control is set to Reset mode, channel enters Reset mode
Halt → Operation	Sleep	Channel remains in Sleep mode
	Reset	Channel remains in Reset mode
	Halt	Channel remains in Halt mode
Operation → Sleep	— (Global mode change not possible)	
Operation → Reset	Sleep	Channel remains in Sleep mode
	Reset	Channel remains in Reset mode
	Halt	Channel mode control is set to Reset mode, channel enters Reset mode
	Operation	Channel mode control is set to Reset mode, channel enters Reset mode
Operation → Halt	Sleep	Channel remains in Sleep mode
	Reset	Channel remains in Reset mode
	Halt	Channel remains in Halt mode
	Operation	Channel mode control is set to Halt mode, channel enters Halt mode after communication finished

28.3.4.1 Timing of Global Mode Change

The transition time for the Global mode changes are shown in the following table.

Table 28.18 Maximum transition time for the global mode (1 of 2)

From	To	Maximum transition time
GL_SLEEP	GL_RESET	3 peripheral clock cycles*2
GL_RESET	GL_SLEEP	3 peripheral clock cycles
GL_RESET	GL_HALT	10 peripheral clock cycles
GL_RESET	GL_OPERATION	10 peripheral clock cycles

Table 28.18 Maximum transition time for the global mode (2 of 2)

From	To	Maximum transition time
GL_HALT	GL_RESET	2 CAN bit times
GL_HALT	GL_OPERATION	3 peripheral clock cycles
GL_OPERATION	GL_RESET	2 CAN bit times
GL_OPERATION	GL_HALT	3 CAN frames*1 *3

Note 1. The given transition time is the time without any errors on the bus. In case of an error condition, the transition time can lengthen to an uncalculated result. The transition time can also come to a stuck condition for locked RX lines or continued error conditions.
Note 2. Exit GL_SLEEP mode only when CFDGSTS.GRAMINIT is cleared.
Note 3. TQ, CAN frame and CAN bits are related to the individual channels. For the maximum transition time, the channel with the lowest baud rate must be used.

28.3.4.2 Timing of Channel Mode Change

The transition time for the Channel mode changes are shown in the following table.

Table 28.19 Maximum transition time for the channel mode

From	To	max. transition time
CH_SLEEP	CH_RESET	3 peripheral clock cycles
CH_RESET	CH_SLEEP	3 peripheral clock cycles
CH_RESET	CH_HALT	3 CAN bit times
CH_RESET	CH_OPERATION	4 CAN bit times
CH_HALT	CH_RESET	2 CAN bit times
CH_HALT	CH_OPERATION	4 CAN bit times*3
CH_OPERATION	CH_RESET	2 CAN bit times
CH_OPERATION	CH_HALT	2 CAN frames*1 *2

Note 1. The time specified for this transition does not include the case where channel enters bus-off state. For bus-off, the timing depends on the configuration of the CFDC0CTR.BOM[1:0] bits.
Note 2. The given transition time is the time without any errors on the bus. In case of an error condition, the transition time can lengthen to an uncalculated result. The transition time can also come to a stuck condition for locked RX lines or continued error conditions.
Note 3. In general, if the baudrate prescaler value CFDC0NCFG.NBRP is changed in CH_HALT mode, the transition time can deviate. The internal prescaler is a free running down counter that creates the TQ clock, and new BRP value is captured when the counter reaches the value 0.

28.4 Initialization

Before joining CAN communications, configure the following settings:

- Clock setting
- Bit timing setting (nominal and data rate)
- Baud Rate setting (nominal and data rate)
- CANFD setting
- Acceptance Filter setting (configuration of Global Acceptance Filter List)
- Reception, Transmission and GW-FIFO setting
- CAN Operation mode setting

28.4.1 Initialization of CAN Clock, Bit Timing and Baud Rate

28.4.1.1 Bit Timing Conditions

The following lines describe the composition of each segment and the restriction that apply to the segment setting.

1. Each segment setting
   SS = Fixed to 1 TQ

TSEG1 = See to (CFDC0NCFG) and (CFDC0DCFG)*1
TSEG2 = See to (CFDC0NCFG) and (CFDC0DCFG)*1
SJW = See to (CFDC0NCFG) and (CFDC0DCFG)*1
SS + TSEG1 + TSEG2 = 5 to 49 TQs for Data Bit Rate and 8 to 385 for Nominal Bit Rate

2. Restriction on TSEG1, TSEG2 and SJW
TSEG1(N) > TSEG2(N) ≥ SJW(N)
TSEG1(D) ≥ TSEG2(D) ≥ SJW(D)*1
When only classical frames are used, configure the bit fields TSEG1 and TSEG2 of CFDC0DCFG to valid values.

Note 1. This feature is not available in the classical CAN function.

Table 28.20 shows an example of how to set the bit timing to achieve the required Sample Point settings.

Table 28.20 Bit timing examples

1 bit	Set value (TQ)				Sample point*1(%)
	SS	TSEG1	TSEG2	SJW	
5TQ	1	2	2	1	60.00
8TQ	1	4	3	1	62.50
	1	5	2	1	75.00
10TQ	1	6	3	1	70.00
	1	7	2	1	80.00
12TQ	1	8	3	1	75.00
	1	9	2	1	83.33
15TQ	1	10	4	1	73.33
	1	11	3	1	80.00
16TQ	1	10	5	1	68.75
	1	11	4	1	75.00
20TQ	1	12	7	1	65.00
	1	13	6	1	70.00
24TQ	1	15	8	1	66.66
	1	16	7	1	70.83
50TQ	1	39	10	4	80.00

Note 1. Sample point (in case of 75%)

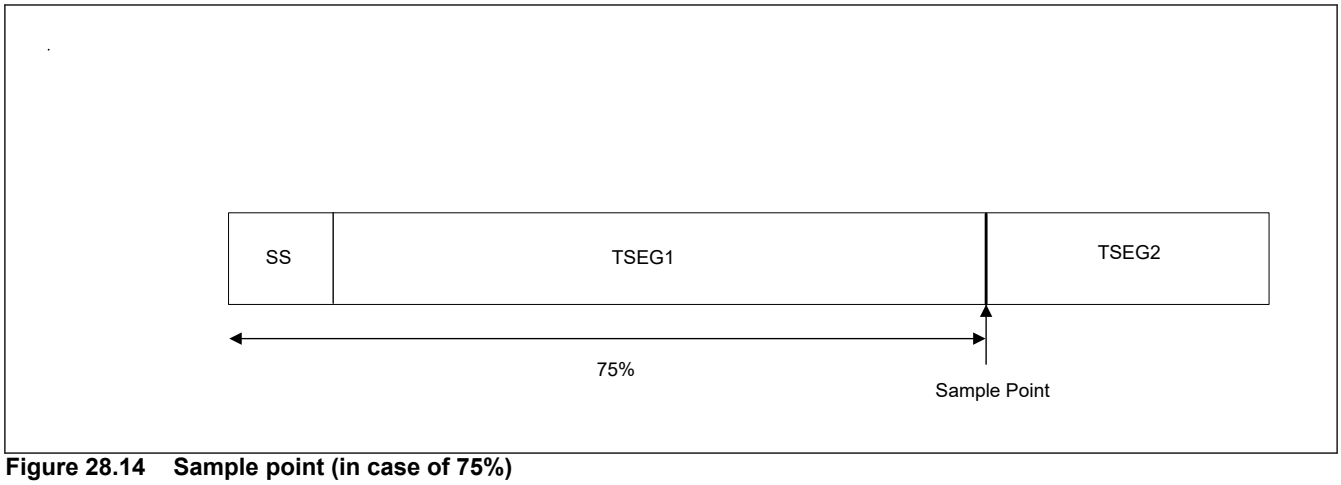


Figure 28.14 Sample point (in case of 75%)

28.4.1.2 CAN Bit Timing

In the CAN protocol, each bit in a communication frame is composed of three segments that can be configured individually for channel using the related CFDC0NCFG and CFDC0DCFG*1 registers.

Note 1. This register is not available in the classical CAN function.

Figure 28.15 shows the segment composition of a bit and the sample point in it.

Of these segments, the Time Segment 1 (TSEG1) and Time Segment 2 (TSEG2) are used to specify the position of the sample point, so that the timing at which each bit on the CAN bus is sampled can be altered by changing the values of these segments.

The minimum resolution for this timing is referred to as Time Quantum (TQ), which is determined by the clock frequency supplied to the CAN channel and the divide-by-N value of the baud rate prescaler (nominal and data rate).

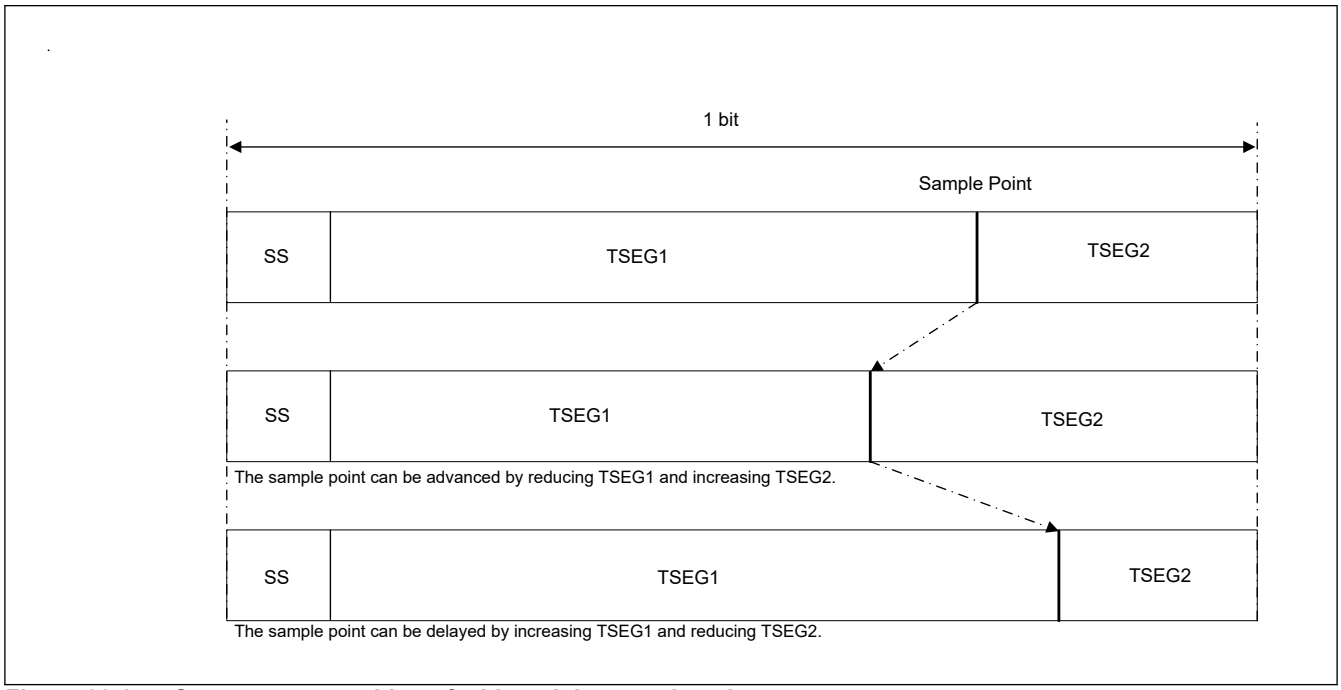


Figure 28.15 Segment composition of a bit and the sample point

1. SS: Synchronization Segment
   This segment is used to synchronize bits by monitoring a recessive-to-dominant edge during the interframe space. This comprises of intermission, suspend transmission, bus idle, during bus idle, and all nodes that can start transmission.
2. TSEG1: Time Segment 1
   This segment absorbs physical delays on the CAN network. A physical delay on the network is two times the total sum of a bus delay, input comparator delay, and output driver delay. It can be lengthened by SJW.
3. TSEG2: Time Segment 2
   This segment is used to correct a phase error by performing resynchronization. It can be shortened by SJW. While sending or receiving a message, communication frames between some nodes may get out of sync due to a drift in the oscillator frequency or a delay in the transmission path. This is referred to as a phase error.
4. SJW: Resynchronization Jump Width
   This is the maximum width by which bits that have become out of sync due to a phase error may be corrected.

Figure 28.15 shows only one symbolic sample point.

28.4.1.3 Baud Rate

Either the CAN channel system clock (clean clock) or the external oscillator clock can be selected globally as CAN communication clock.

The transfer speed is determined by the DLL clock, the divide-by-N value of the baud rate prescaler, and the number of TQs in one bit.

$$\text{baudrate} = \frac{\text{DLL\_Clock}}{(\text{number\_of\_time\_quanta\_per\_bit}) \times (\text{BRP} + 1)}$$

Figure 28.16 shows a block diagram of the circuit that generates the CAN channel system clock and Table 28.21 shows a baud rate examples.

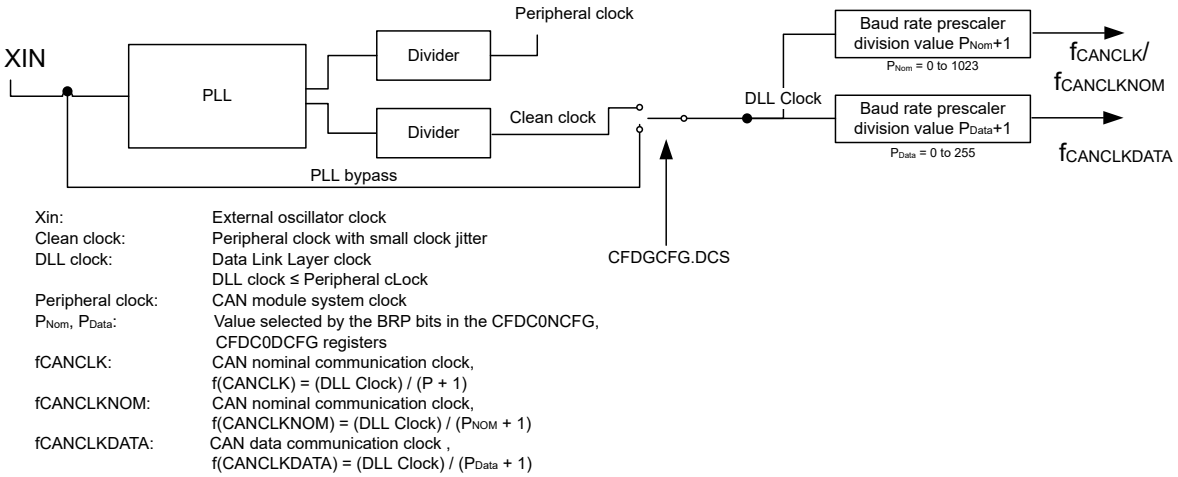


Figure 28.16 Block diagram of the circuit that generates the CAN channel communication clock

Table 28.21 Nominal baud rate calculation formula and example CAN communication configurations

Baud rate calculation formula	(DLL clock) (baud rate prescaler divide-by-N value*1) × (number of TQs in one bit)							
	40 MHz	32 MHz	30 MHz	24 MHz	20 MHz	16 MHz	10 MHz	8 MHz*2
1 Mbps	8TQ (5) 20TQ (2)	8TQ (4) 16TQ (2)	10TQ (3) 15TQ (2)	8TQ (3) 12TQ (2) 24TQ (1)	10TQ (2) 20TQ (1)	8TQ (2) 16TQ (1)	10TQ (1)	8TQ (1)
500 Kbps	8TQ (10) 20TQ (4)	8TQ (8) 16TQ (4)	10TQ (6) 15TQ (4) 20TQ (3)	8TQ (6) 12TQ (4) 24TQ (2)	10TQ (4) 20TQ (2)	8TQ (4) 16TQ (2)	10TQ (2) 20TQ (1)	8TQ (2) 16TQ (1)
250 Kbps	8TQ (20) 20TQ (8)	8TQ (16) 16TQ (8)	10TQ (12) 15TQ (8) 20TQ (6)	8TQ (12) 12TQ (8) 24TQ (4)	10TQ (8) 20TQ (4)	8TQ (8) 16TQ (4)	10TQ (4) 20TQ (2)	8TQ (4) 16TQ (2)
125 Kbps	8TQ (40) 20TQ (16)	8TQ (32) 16TQ (16)	10TQ (24) 15TQ (16) 20TQ (12)	8TQ (24) 12TQ (16) 24TQ (8)	10TQ (16) 20TQ (8)	8TQ (16) 16TQ (8)	10TQ (8) 20TQ (4)	8TQ (8) 16TQ (4)
83.3 Kbps	8TQ (60) 12TQ (40) 16TQ (30) 24TQ (20)	8TQ (48) 12TQ (32) 16TQ (24) 24TQ (16)	8TQ (45) 10TQ (36) 12TQ (30) 15TQ (24) 20TQ (18) 24TQ (15)	8TQ (36) 12TQ (24) 16TQ (18) 24TQ (12)	8TQ (30) 10TQ (24) 12TQ (20) 15TQ (16) 16TQ (15) 20TQ (12) 24TQ (10)	8TQ (24) 12TQ (16) 16TQ (12) 24TQ (8)	8TQ (15) 10TQ (12) 12TQ (10) 15TQ (8) 20TQ (6) 24TQ (5)	8TQ (12)
33.3 Kbps	8TQ (150) 12TQ (100) 16TQ (75) 20TQ (60) 24TQ (50)	8TQ (120) 10TQ (96) 12TQ (80) 15TQ (64) 16TQ (60) 20TQ (48) 24TQ (40)	10TQ (90) 12TQ (75) 15TQ (60) 20TQ (45)	8TQ (90) 10TQ (72) 12TQ (60) 15TQ (48) 16TQ (45) 20TQ (36) 24TQ (30)	8TQ (75) 10TQ (60) 12TQ (50) 15TQ (40) 20TQ (30) 24TQ (25)	8TQ (60) 10TQ (48) 12TQ (40) 15TQ (32) 16TQ (30) 20TQ (24) 24TQ (20)	10TQ (30) 12TQ (25) 15TQ (20) 20TQ (15)	8TQ (30)

Note: Shown in ( ) are the baud rate prescaler divide-by-N value.
Note 1. Baud rate prescaler divide-by-N value = P + 1 (P = 0 - 1023) P: value selected by the BRP bits in the Channel Configuration Registers.
Note 2. Minimum frequency to achieve maximum nominal baud rate of 1 Mbps.

Table 28.22 Baud rate calculation example for nominal and data bit rate CAN communication configurations

Baud rate calculation formula	(DLL clock) (baud rate prescaler divide-by-N value[*1]) × (number of TQs in one bit)	
	40 MHz	20 MHz
Nominal 1 Mbps Data 5 Mbps	40TQ (1)	20TQ (1)
	8TQ (1)	Not possible
Nominal 500 Kbps Data 2 Mbps	80TQ (1)	40TQ (1)
	20TQ (1)	10TQ (1)

Note: Shown in ( ) are the baud rate prescaler divide-by-N values and this table is not available in the classical CAN function.
Note 1. Baud rate prescaler divide-by-N value = P + 1 (P = 0 - 1023) P: value selected by the BRP bits in the Channel Configuration Registers.

For optimum clock tolerance in networks using the FD frame format, the length of the time quantum should be the same in nominal bit time and in data bit time. This means CFDC0NCFG.NBRP = CFDC0DCFG.DBRP.

Additionally, if transceiver delay compensation is used, do not program the CFDC0DCFG.DBRP bit to be greater than 1, as 1 means divide by 2.

28.4.1.4 Setting of CAN Clock, Bit Timing and Baud Rate

Figure 28.17 shows the procedure for setting the CAN clock and the baud rate for a channel.

These settings should be performed during Channel Reset mode (Configuration mode) for the CAN channels.

Before going to channel communication state, the baud rate must be configured, otherwise the mode does not switch correctly.

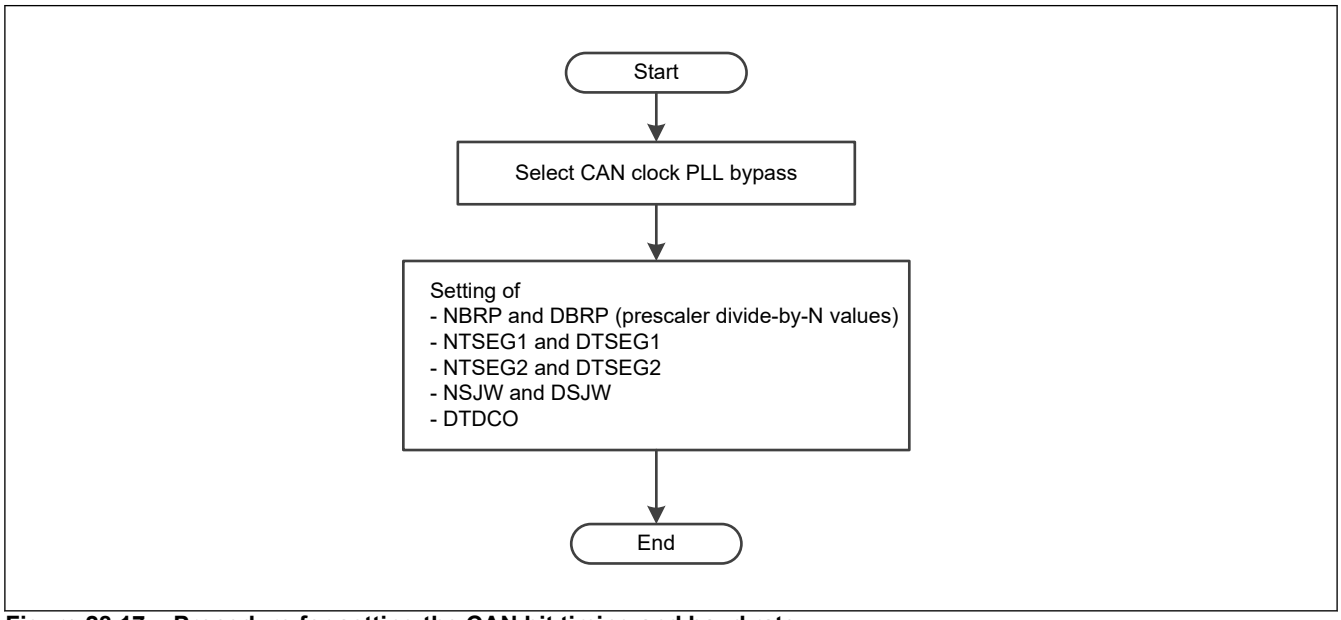


Figure 28.17 Procedure for setting the CAN bit timing and baud rate

28.4.1.5 Transmitter Delay Compensation

This chapter is not valid for classical CAN.

When a high baud rate is used such as 5 Mbps for the data phase, the transmitter delay can become greater than TSEG1. In this case, the transmitter always detects a bit-error in the data phase of the CANFD frame. The TDC compensates for the inability of the transmitter to receive its own transmitted bit at the sample point of that bit.

There is another symbolic sample point known as the Secondary Sample Point (SSP) that is used only during the data phase of CANFD frames. This is derived from the Transceiver Delay Compensation Result bit (CFDC0FDSTS.TDCR) as shown in Figure 28.18.

The resolution of the configuration, measured and offset values is based on the CAN channel DLL clock.

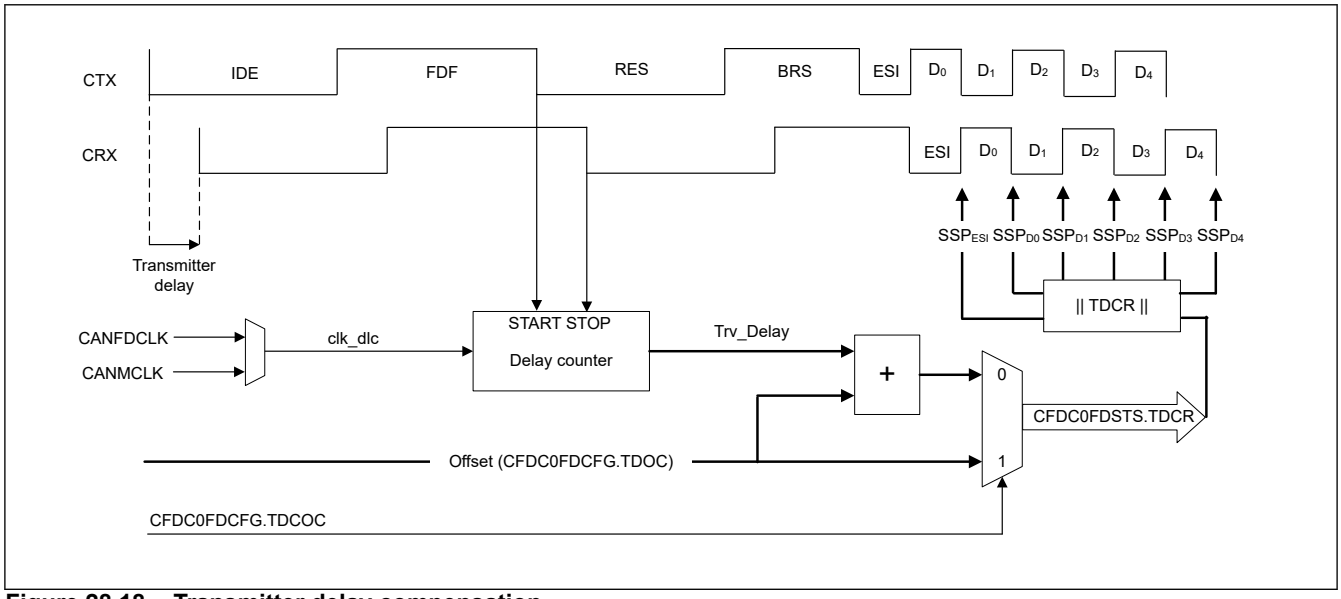


Figure 28.18 Transmitter delay compensation

The measured Trv_Delay is based on the number of clk_dlc clock cycles.The delay is counted up by one for each started clock until the dominant value is seen on CAN_RX. Figure 28.19 shows the measured result. Trv_Delay counted to maximum 127 with a clk_dlc clock.

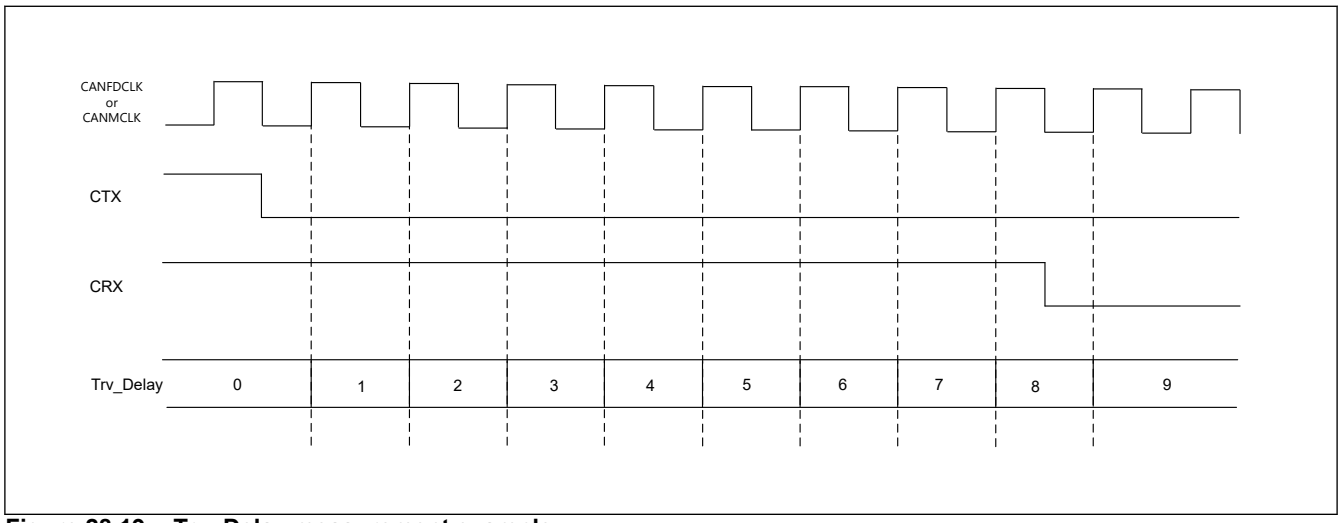


Figure 28.19 Trv_Delay measurement example

The SSP is calculated by taking the result from CFDC0FDSTS.TDCR and rounding the value down to the nearest integer number of data time quanta.

Figure 28.20 shows the positioning of the secondary sample point. When CFDC0FDCFG.TDCOC is equal to 0, the SSP is equal to the Trv_Delay (measured delay) + CFDC0FDCFG.TDCO, rounded down to the nearest integer number of time quanta. Usually, the TDCO value should have the size of (SyncSegmentdata + TSEG1data) to position the SSP to a theoretical location of the sample point.

If the CFDC0FDCFG.TDCOC is equal to 1, the SSP is defined by CFDC0FDCFG.TDCO. If CFDC0DCFG.DBRP is greater than 0, the value is also rounded down to the nearest integer number of time quanta.

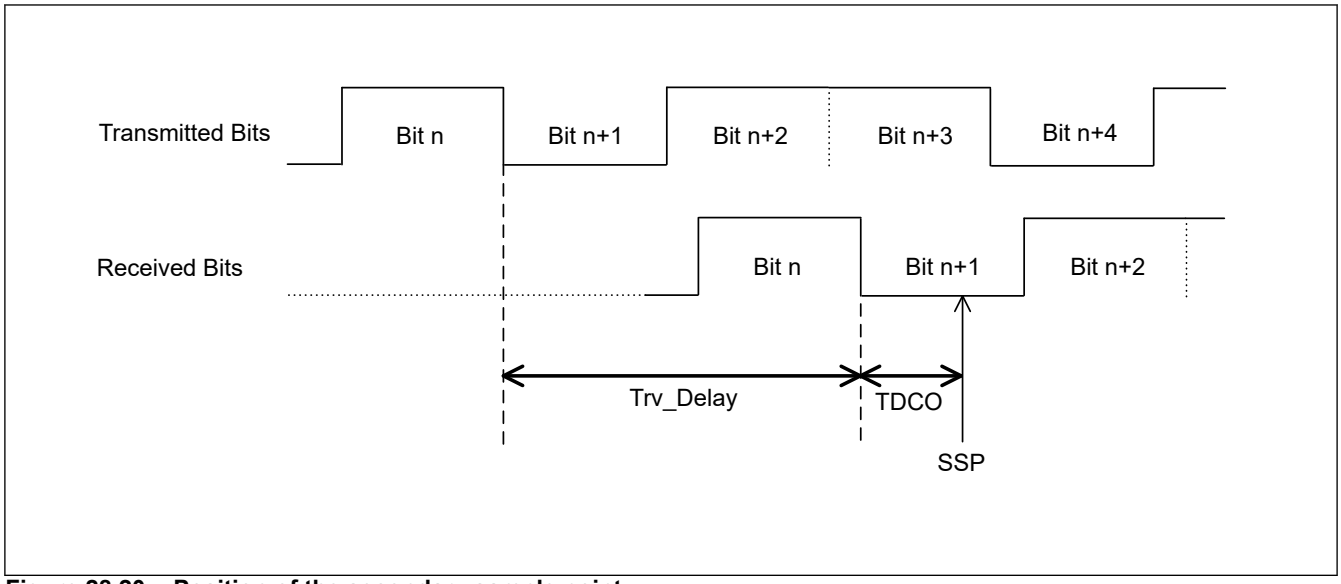


Figure 28.20 Position of the secondary sample point

The maximum delay (Trv_Delay + TDCO) which can be compensated by the CANFD module is (6 data bits - 2clk_dlc).

The ISO 11898-1 allows you to set different values for BRP_data and BRP_nom.

If different values are used for CFDC0NCFG.NBRP and CFDC0DCFG.DBRP, then two CAN nodes may be out of synchronization at the point when the bit rate changes from nominal bit rate to data bit rate after sample point of the BRS bit. This condition is shown in Figure 28.21.

The length of the time quantum should be the same in the nominal bit time and in the data bit time. This means CFDC0NCFG.NBRP = CFDC0DCFG.DBRP.

Different bit rates can be achieved by selecting different configuration values for the Time Segments. The nominal bit rate can be configured from 8 to 385 TQs and the data bit rate from 5 to 49 TQs.

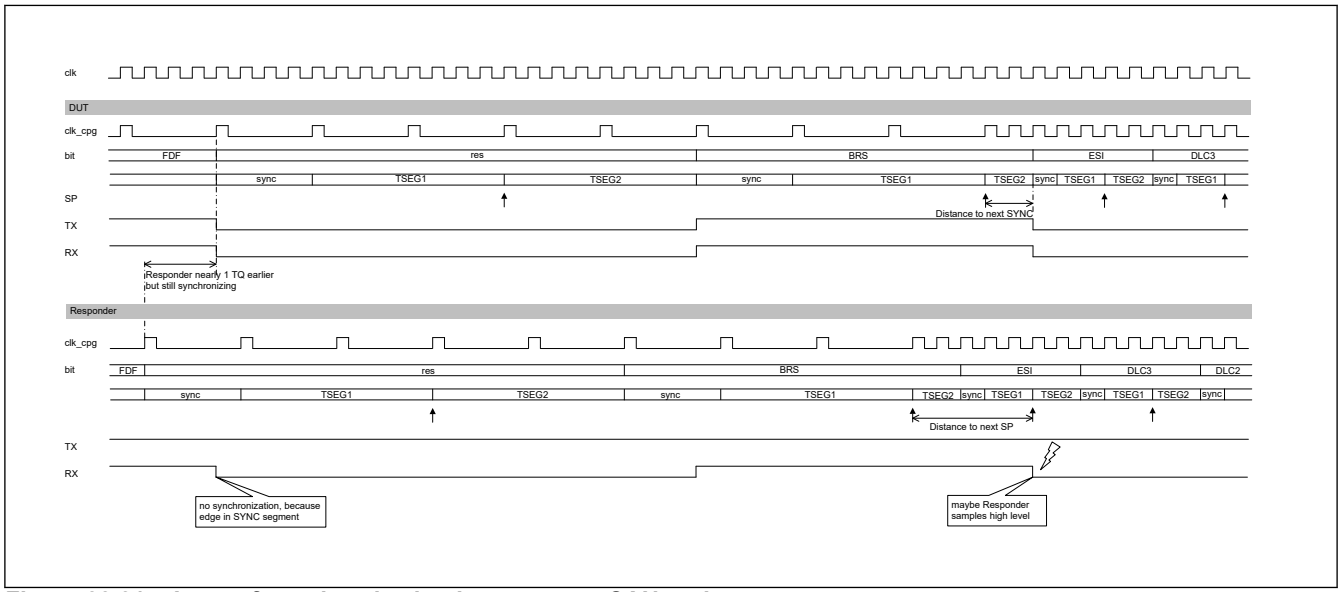


Figure 28.21 Loss of synchronization between two CAN nodes

The transmitter delay compensation measurement result is updated at the falling edge from FDF bit to RES bit when configured accordingly (CFDC0FDCFG.TDCE = 1, CFDC0FDCFG.TDCOC = 0).

Figure 28.22 shows the read flow to get the measured transmitter delay compensation result.

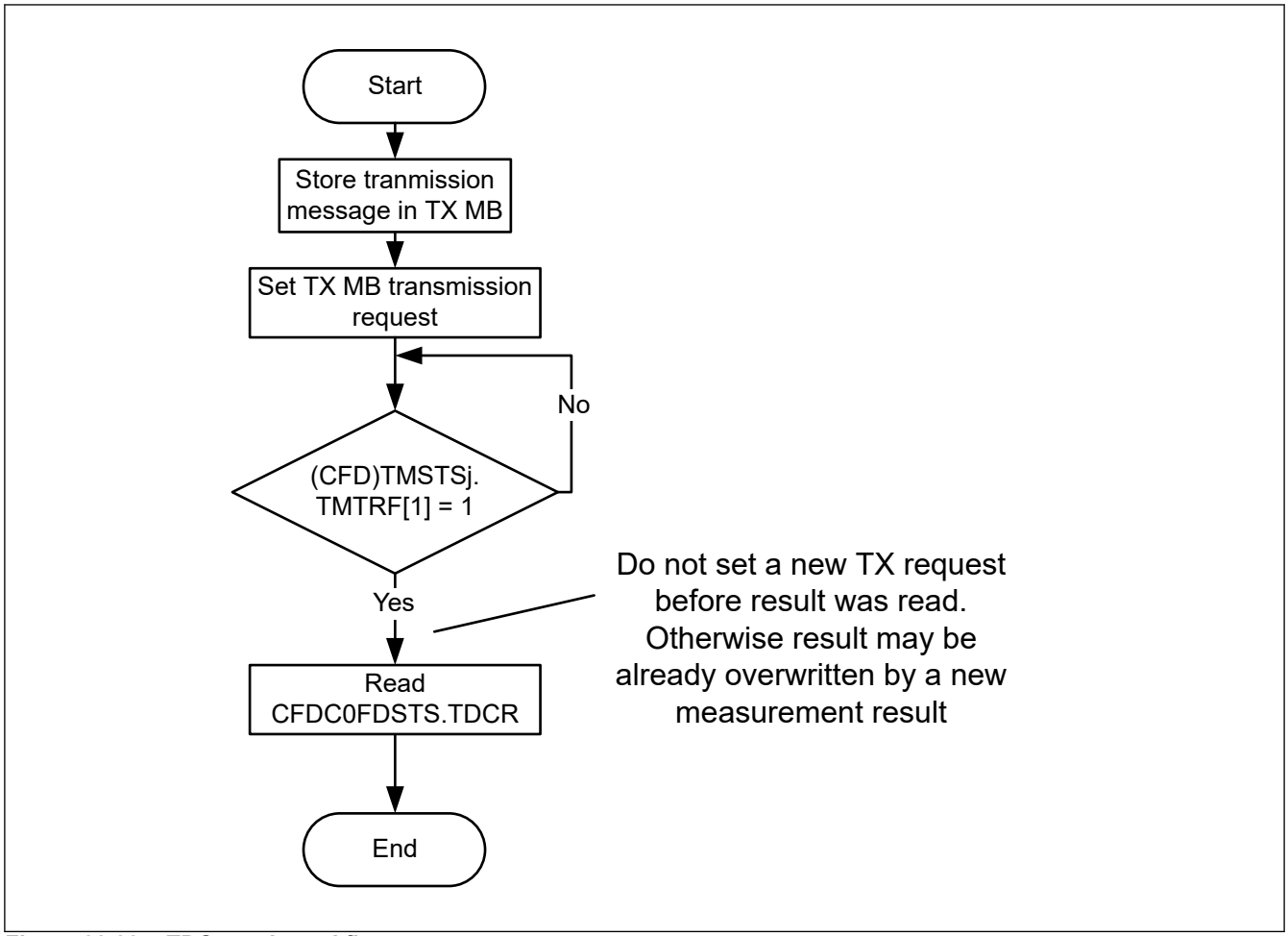


Figure 28.22 TDC result read flow

28.4.2 CAN Module Configuration after Hardware Reset

After a hardware reset (power on reset) or after setting and clearing a CFDGRSTC.SRST bit, the CANFD module enters Global Sleep mode automatically.

To enable configuration of the CANFD module, you must exit Sleep mode by clearing the Global Sleep Request bit CFDGCTR.GSLPR to 0.

After a hardware reset, the module starts RAM initialization, the CFDGSTS.GRAMINIT bit in the Global Status Register is set automatically to indicate that the CANFD logic is initializing the RAM.

After RAM initialization is complete, this bit is cleared automatically.

RAM initialization is necessary to avoid setting of false ECC error flag after HW reset the random data presented in the RAM.

Do not access registers of CANFD in either read or write until RAM initialization is complete and the CFDGSTS.GRAMINIT bit is cleared.

Before going to communication mode, the Global Acceptance Filter List and message FIFO buffers must be configured. In addition, CAN channel must be configured such as CAN bit timing. For this configuration, CAN channel must be released from Channel Sleep mode and must be configured for communication in Channel Reset mode (Configuration mode).

Figure 28.23 shows the configuration procedure. For details about each step, seesection 28.5. Acceptance Filtering Function using Global Acceptance Filter List (AFL), section 28.6. FIFO Buffers and Normal Message Buffer Configuration, section 28.7. Interrupts and DMA and section 28.4.1.3. Baud Rate.

The CANFD module does not perform the RAM initialization sequence after executing a software reset by setting CFDGRSTC.SRST.

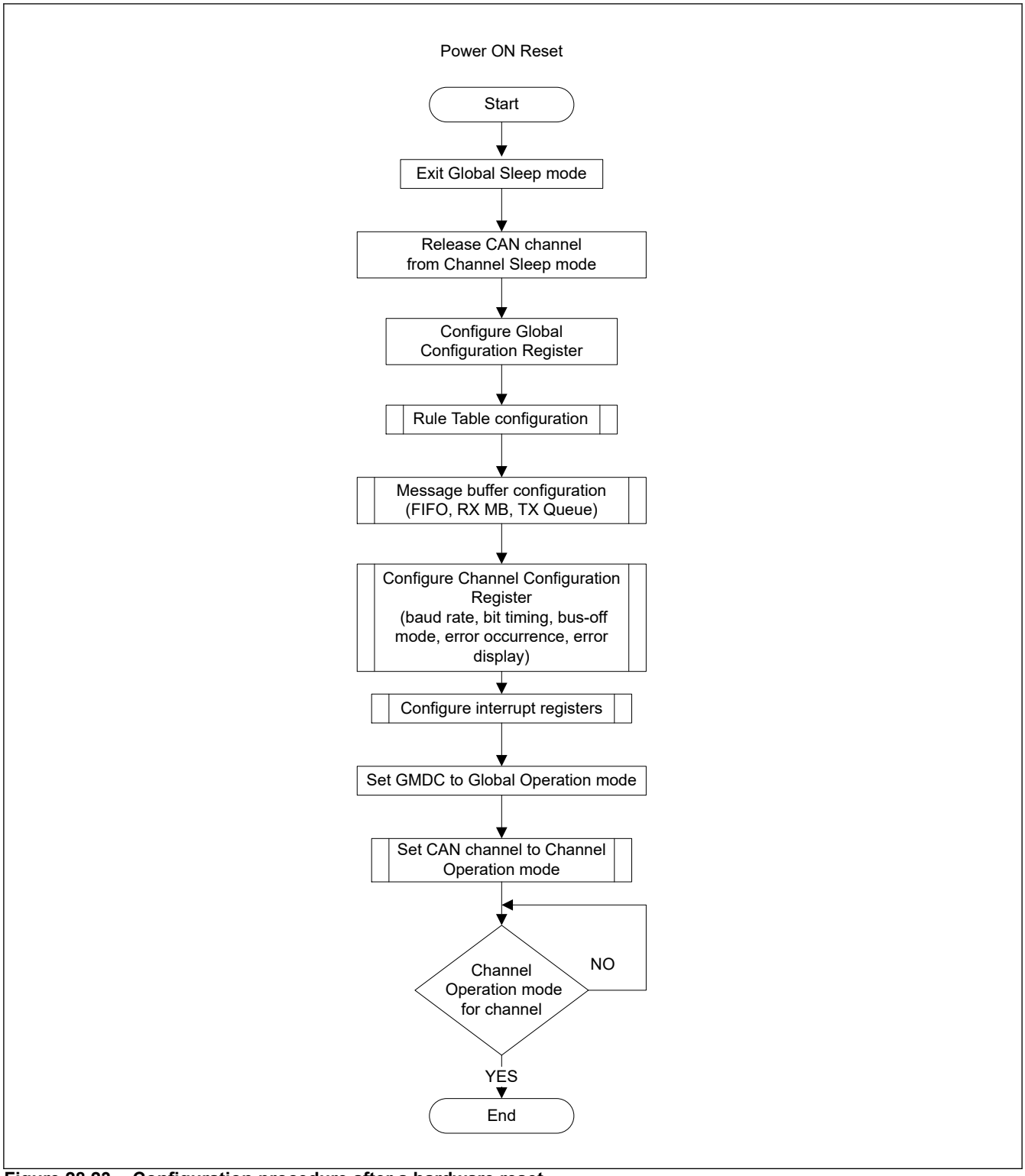


Figure 28.23 Configuration procedure after a hardware reset

28.5 Acceptance Filtering Function using Global Acceptance Filter List (AFL)

28.5.1 Overview

The CANFD module can handle message acceptance filtering with a global Acceptance Filter List (called AFL). Each element of the AFL defines a filter rule for messages received on a specific channel.

The following actions are performed based on the AFL entries:

- Acceptance filtering based on received CAN Identifier and masking

- DLC filtering based on received DLC value
- Message data payload according to the CFDGCFG.CMPOC bit*1
- Storage of accepted messages in the message buffer objects defined in the related AFL entry
- Attaching a 16-bit pointer to the stored messages defined in the related AFL entry, for example to support AUTOSAR applications
- Attaching a 2-bit information label to the stored messages defined in the related AFL entry

Note 1. This feature is not available in the classical CAN function.

The CANFD module allows a maximum of 32 AFL entries.

During acceptance filtering process, each AFL entry in a channel is checked against the received message by the acceptance filter unit. The check starts from the lowest AFL entry number for this channel.

AFL search stops when a match of the received identifier with a configured identifier/mask combination occurs or when the received identifier has been compared against all AFL entries defined for the related channel. If no match occurs, then the received message is rejected. No notification is given to the application in this case.

Additionally, an automatic DLC filtering is performed for each accepted message if DLC check is globally enabled. If the DLC value of the received message is equal to or higher than the configured DLC value in the matching AFL entry, the DLC check is passed.

If DLC replacement (CFDGCFG.DRE bit) is enabled, DLC value configured in the matching AFL entry is greater than 0x0 and DLC check passes, then the configured value of DLC in the matching AFL entry is stored in the destination RXMB or FIFO Buffer.

If the received value of DLC is greater than the configured DLC value in the matching AFL entry, then the additional data bytes received on the CAN Bus are not stored in the destination RXMB or FIFO Buffer. These additional data bytes are stored as 0x00 in the destination RXMB or FIFO Buffer.

If DLC replacement is enabled and DLC value of matching AFL entry is 0x0, then the received value of DLC is stored in the destination RX MB or FIFO Buffer.

If DLC replacement (the CFDGCFG.DRE bit) is disabled and DLC check passes, then the received value of DLC on the CAN bus is stored in the destination RXMB or FIFO buffer.

If the received value of DLC is greater than the configured DLC value in the matching AFL entry, then the additional data bytes received from the CAN bus are also stored in the destination RXMB or FIFO buffer.

If DLC value of the received message is less than the configured DLC value in the matching AFL entry, then DLC check fails. In this case, the received message is rejected and is not stored in any RXMB or FIFO buffer.

Additionally, DLC check failure is flagged by the DLC Error Flag in the Global Error Flag Register. If configured, an error interrupt is also generated. The DLC replacement configuration has no impact if the DLC check fails.

If a message has passed both acceptance filtering and DLC filtering, it is stored in a single reception message buffer and/or in FIFO buffers configured for reception function.

This message storage target information is also defined in the same AFL entry. Do not set a target at the AFL entry which is not configured.

Each accepted received message can be stored into a maximum of 2 different target destinations (single reception message buffer and/or FIFO buffers).

The programming of more than 2 target destinations is not allowed. If more destinations are programmed, then the internal timing might lead to a race condition that prevents the storage of received messages in the message RAM.. Correct configuration of the numbers of target destination is the responsibility of the application.

Additional protection mechanism is made for the case when a received message contains more data payload Bytes than possible to store in the target destination (CFDRMNB.RMPLS, CFDRFCCa.RFPLS or CFDCFCC.CFPLS).

If CFDGCFG.CMPOC = 0, the message is completely rejected and is stored in the target destination. When CFDGCFG.CMPOC = 0 and RX or Common FIFO full including the received message contains more data payload bytes than possible to store in the target destination (CFDRMNB.RMPLS, CFDRFCCa.RFPLS or CFDCFCC.CFPLS), the corresponding CFDFMSTS.RFxMLT or CFDFMSTS.CFxMLT bit is not set to 1, respectively.

When CFDGCFG.CMPOC = 1, the received data bytes greater than CFDRMNB.RMPLS is rejected. When CFDGCFG.CMPOC = 1 and RX or Common FIFO full including the received message contains more data payload bytes than possible to store in the target destination (CFDRMNB.RMPLS, CFDRFCCa.RFPLS or CFDCFCC.CFPLS), the corresponding CFDFMSTS.RFxMLT or CFDFMSTS.CFxMLT bit is set to 1, respectively.

Depending on the CFDGCFG.DRE bit, the original received DLC or the DLC value configured at the AFL entry is stored.

Regardless of the CFDGCFG.CMPOC bit setting, CFDGERFL.CMPOF is set to 1 if a payload overflow condition is detected.

The DLC filtering is performed before the payload overflow function. So for one reception frame, only one flag can be set at the same time with CFDGERFL.DEF or CFDGERFL.CMPOF*1.

Note 1. This bit is not available in the classical CAN function.

28.5.2 Allocation of AFL Entries

The number of AFL entries per channel can be configured using the dedicated field in the related Global Acceptance Filter Configuration Registers (see Figure 28.24).

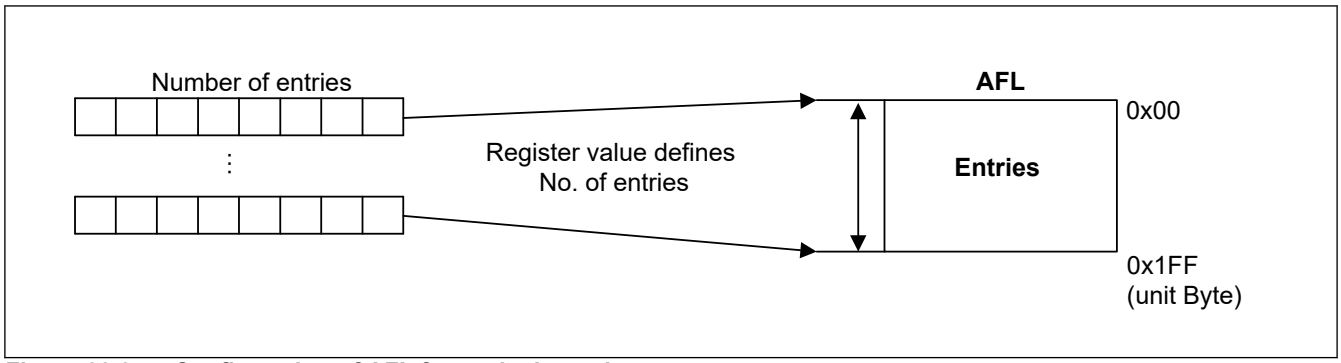


Figure 28.24 Configuration of AFL for each channel

The minimum number of entries for one channel is 0 (no entries defined for the channel) and the maximum number of entries is 32.

All entries are unique for a channel and overlapping or sharing of entries is not supported. Correct configuration of the AFL is the responsibility of the application.

The CANFD module does not flag errors related to the configuration of the AFL.

28.5.3 AFL Entry Description

Each AFL entry consists of 16 bytes. The fields in all entries are identical.

Each entry contains the following information for acceptance filtering and DLC filtering:

- Identifier (11 bits for Standard Frame format, 29 bits for Extended Frame format):
  Acceptance filter unit checks the identifier field of the received message against the identifier field of each AFL entry (full 29 bits masking of identifier bits is possible, see information that follows).
- IDE bit:
  Acceptance filter unit checks the IDE bit of the received message against this bit and selects the relevant part of the identifier field for acceptance filtering (masking of IDE bit is possible, see the information that follows).
- RTR bit:
  Acceptance filter unit only accepts data frames (RTR = 0) or remote frames (RTR = 1) according to the setting of this bit (masking of RTR bit is possible, see the information that follows).
- Loopback Configuration bit:
  This bit can enable or disable the AFL entry depending on the Loopback Configuration or Mirror mode condition.
- Mask for Identifier bits (29 bits):

Each bit in the identifier mask can mask the corresponding identifier bit in the AFL entry during acceptance filtering, see Figure 28.25.

- Mask for IDE bit:
  If this Mask bit masks the IDE bit of the AFL entry in both Standard Identifier and Extended Identifier format, messages can be accepted by this AFL entry. The identifier of the received message is compared against the Standard Identifier part of the AFL entry for Standard Identifier format messages and against the Extended Identifier part of the AFL entry for Extended Identifier format messages.

- Mask for RTR bit:
  If this Mask bit masks the RTR bit of the AFL entry in both frame formats, data frame and remote frame formats are accepted by this AFL entry.

- Pointer information (16 bits):
  This 16-bit pointer is attached to a received message accepted by the related AFL entry. The pointer is added during message storage in the message buffer area and can be used by application as support function. The pointer information can be used for example to support PDU identifier allocation for the received message in AUTOSAR systems.

- Information label (2 bits):
  This 2-bit label is attached to a received message accepted by the related AFL entry. The label is added during message storage in the message buffer area and can be used by application as support function.

- DLC value for automatic DLC filtering:
  If the DLC value of the received message is equal or higher than the configured DLC value, the DLC check is passed.

If the DLC value in this AFL entry is configured to 0, DLC filtering is effectively disabled for this entry (all accepted messages pass DLC filtering).

Each AFL entry contains the following information for the handling of received messages:

- Message buffer number of one single reception message buffer as target for received message storage
- Single reception message buffer enable bit to configure the single reception message buffer number to be valid or invalid, as target for received message storage
- FIFO direction pointer - each bit of the FIFO direction pointer configures a dedicated FIFO as possible target for a received message

There is no hardware protection against such storage of message. Therefore, the FIFO direction pointer must be configured carefully.

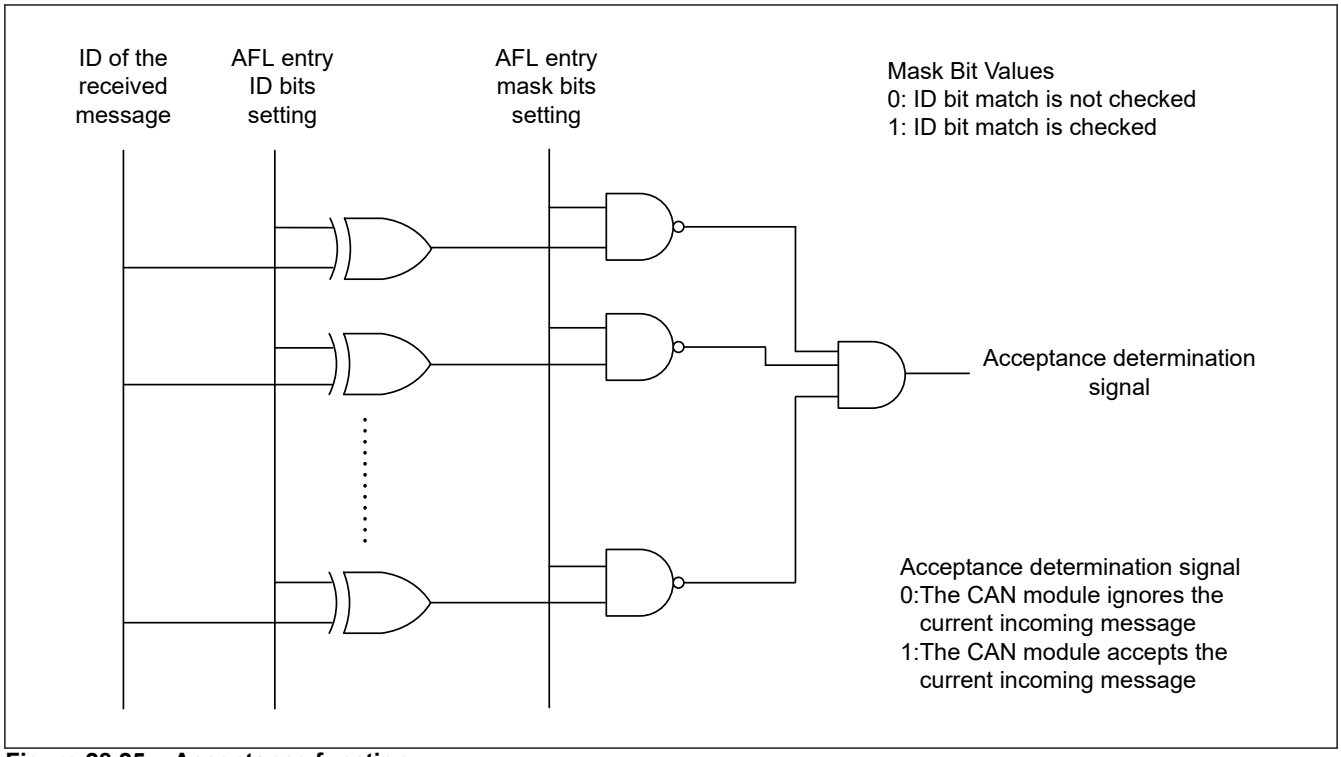


Figure 28.25 Acceptance function

28.5.4 Entering Entries in the AFL

Application software can enter one full entry into the AFL using the following registers:

- Global AFL ID Entry Register: Part 1 of the AFL entry
- Global AFL Mask Entry Register: Part 2 of the AFL entry
- Global AFL Pointer 0 Entry Register: Part 3 of the AFL entry
- Global AFL Pointer 1 Entry Register: Part 4 of the AFL entry.

16 sets of these registers form a group of AFL entries. Each group can be accessed through a page mechanism. For the CANFD module, 32 of these pages exist to allow access to the whole AFL range. The AFL should only be configured in CH_RESET or CH_HALT mode. Pages are linked to the AFL entries in the following way:

Page 0	Entry 0 — 15
Page 1	Entry 16 — 31

The selection of the AFL access page is done using the Global Acceptance Filter List Entry Control Register (CFDGAFLECTR) (Figure 28.26). This register has the following fields:

- 1 bit to select the AFL page number
- 1 bit to enable or disable the AFL data access to prevent unwanted write access to the AFL.

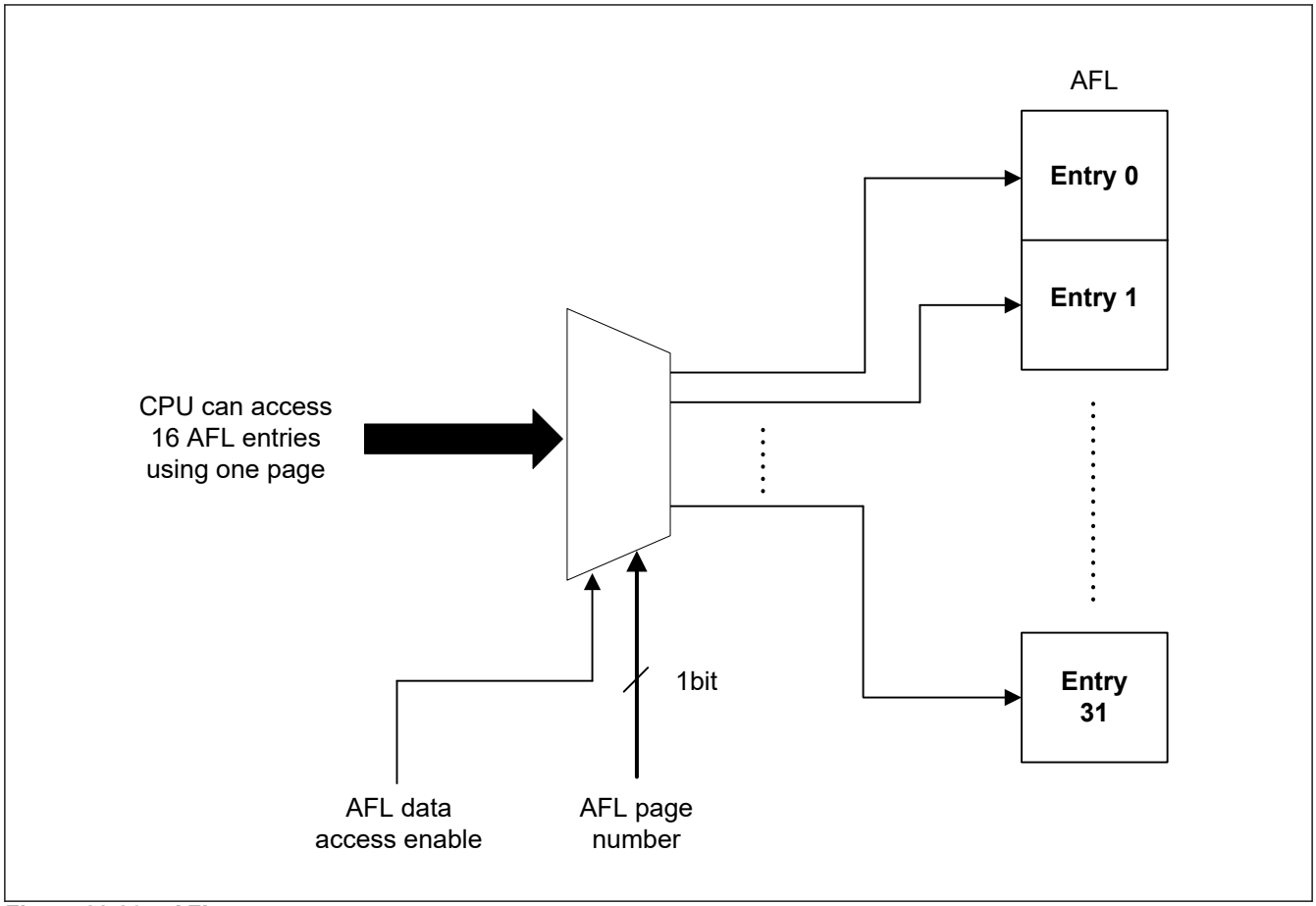


Figure 28.26 AFL page access

Application software should not write numbers higher than 0x1 for the AFL page number.

Follow the configuration shown in Figure 28.27 to program the AFL.

After entering all entries in Configuration mode, locking of the AFL access should be performed to protect unwanted write access to the AFL.

Write protection is active during all Global modes (GL_RESET, GL_HALT, and GL_OPERATION) if the lock bit is set.

Read access to AFL is still possible during all Global modes even when AFL data access is disabled (consistency check of AFL contents is possible during run time).

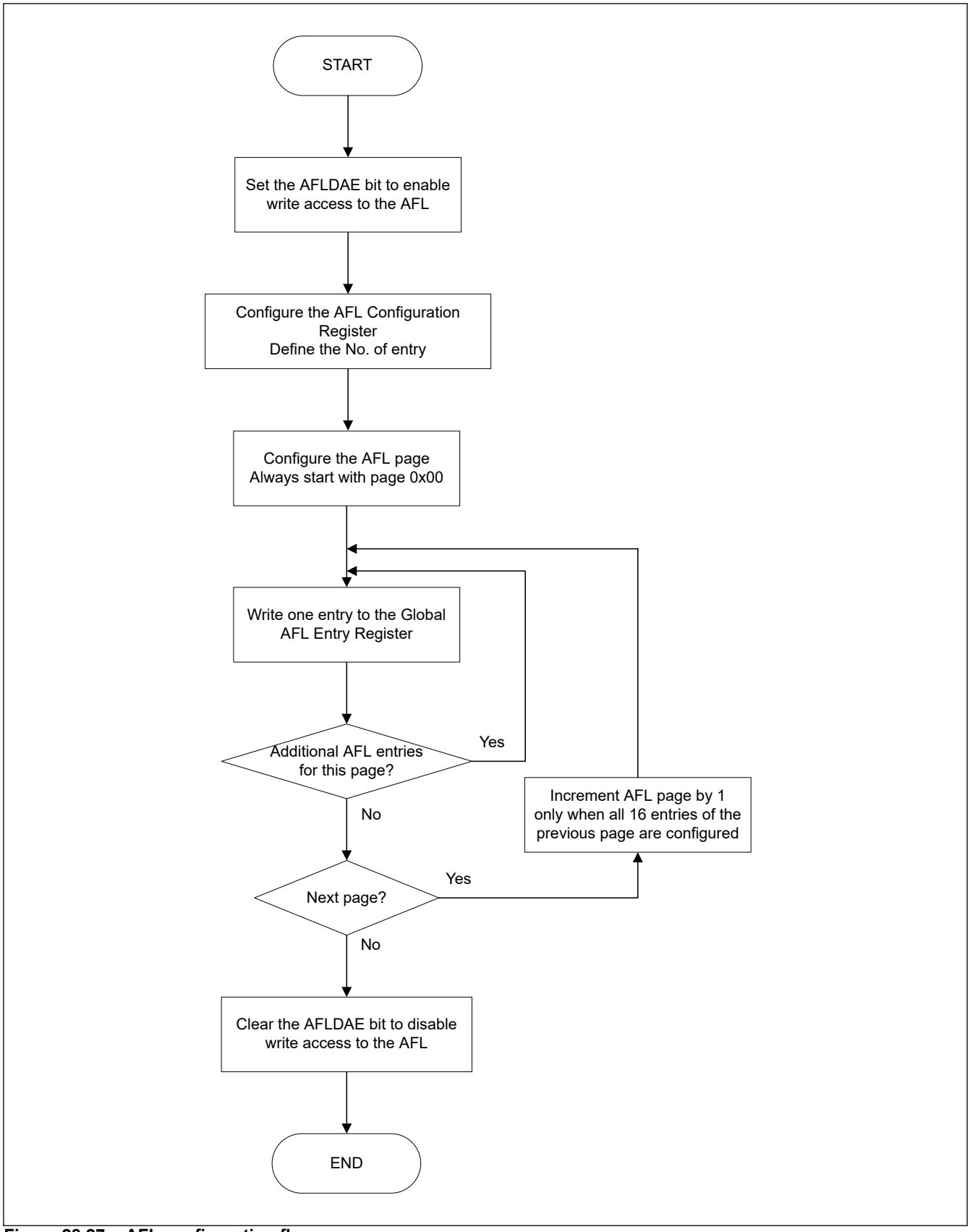


Figure 28.27 AFL configuration flow

28.5.5 Loopback Modes

If the Loopback Configuration bit is set, the AFL entry is only valid in Loopback test mode (Self-test mode 0 or Self-test mode 1) or in mirror mode when receiving messages that were transmitted by the respective CAN channel itself.

The AFL entry is not valid for received messages in loopback mode transmitted by other CAN nodes on the bus. The expression valid or invalid for the related entry means that this AFL entry is or is not compared against the received message ID respectively.

If the Loopback Configuration bit is 0, the AFL entry is only valid for:

- Received messages transmitted by other CAN nodes on the bus in normal (non-loopback mode) and mirror modes
- Received messages transmitted by other CAN nodes or the CAN channel itself in Loopback test mode.

The mirror mode can be enabled with the CFDGCFG.MME bit in the Global Configuration Register. If CFDGCFG.MME bit is set, then a successfully transmitted message can be stored back in an RX message buffer or FIFO buffer if a matching entry is configured in the AFL for that channel.

The Loopback Configuration bit in the matching AFL entry must be set to store this frame.

If Mirror mode and Loopback test mode are configured at the same time, the Loopback test mode behavior applies.

Table 28.23 shows the behavior of the acceptance filter unit depending on the setting of the related input signals.

Table 28.23 Behavior of acceptance filter based on the loopback configuration setting in AFL entry

Mirror Mode Enable (MME Configuration bit)	Loopback in test mode (Self-test mode 0 or Self-test mode 1)	Channel mode	Loopback Configuration bit in AFL entry	AFL entry
0	0	Receiver	0	Valid
			1	Invalid
		Transmitter	0	Invalid
			1	Invalid
	1	Receiver	0	Valid
			1	Invalid
		Transmitter	0	Valid
			1	Valid
1	0	Receiver	0	Valid
			1	Invalid
		Transmitter	0	Invalid
			1	Valid
	1	Receiver	0	Valid
			1	Invalid
		Transmitter	0	Valid
			1	Valid

Note: The expression valid or invalid for the related entry means that this AFL entry is or is not compared against the received message ID, respectively.

28.5.6 IDE Masking

When the GAFLIDEM bit is 0 in an AFL entry, the IDE bit configured in the AFL entry is not used for ID matching. In this case, the use of ID[10:0] or ID[28:0] matching is based on the received IDE bit.

Consider the following example:

- The ID and Mask fields of an AFL entry x is configured as follows:
  - CFDGAFLID [x] = 0xC0553A20 → IDE = 1, RTR = 1, LLB = 0, ID[10:0] = 0x220 / ID[28:0] = 0x00553A20
  - CFDGAFLMr = 0x0000FFFF → IDEM = 0. RTRM = 0, IDM[10:0] = 0x7FF / IDM[28:0] = 0x0000FFFF

- The comparison result for the four different received IDs with AFL entry x is described as follows:
  - If a frame with IDE = 0 and ID = 0x220 is received, this is considered as a match
  - If a frame with IDE = 0 and ID = 0x320 is received, this is not a match
  - If a frame with IDE = 1 and ID = 0x1FFF3A20 is received, this is considered as a match
  - If a frame with IDE = 1 and ID = 0x08803220 is received, this is not a match.

28.5.7 Updating AFL Entry during Communication

You can update the AFL entry without disabling all CAN communications. Choose the entry number to be updated by setting the AFL entry number, and ignore the enable bit.

This entry number is ignored from the AFL matching while the entry is being updated.

Figure 28.28 shows the update flow for an AFL entry.

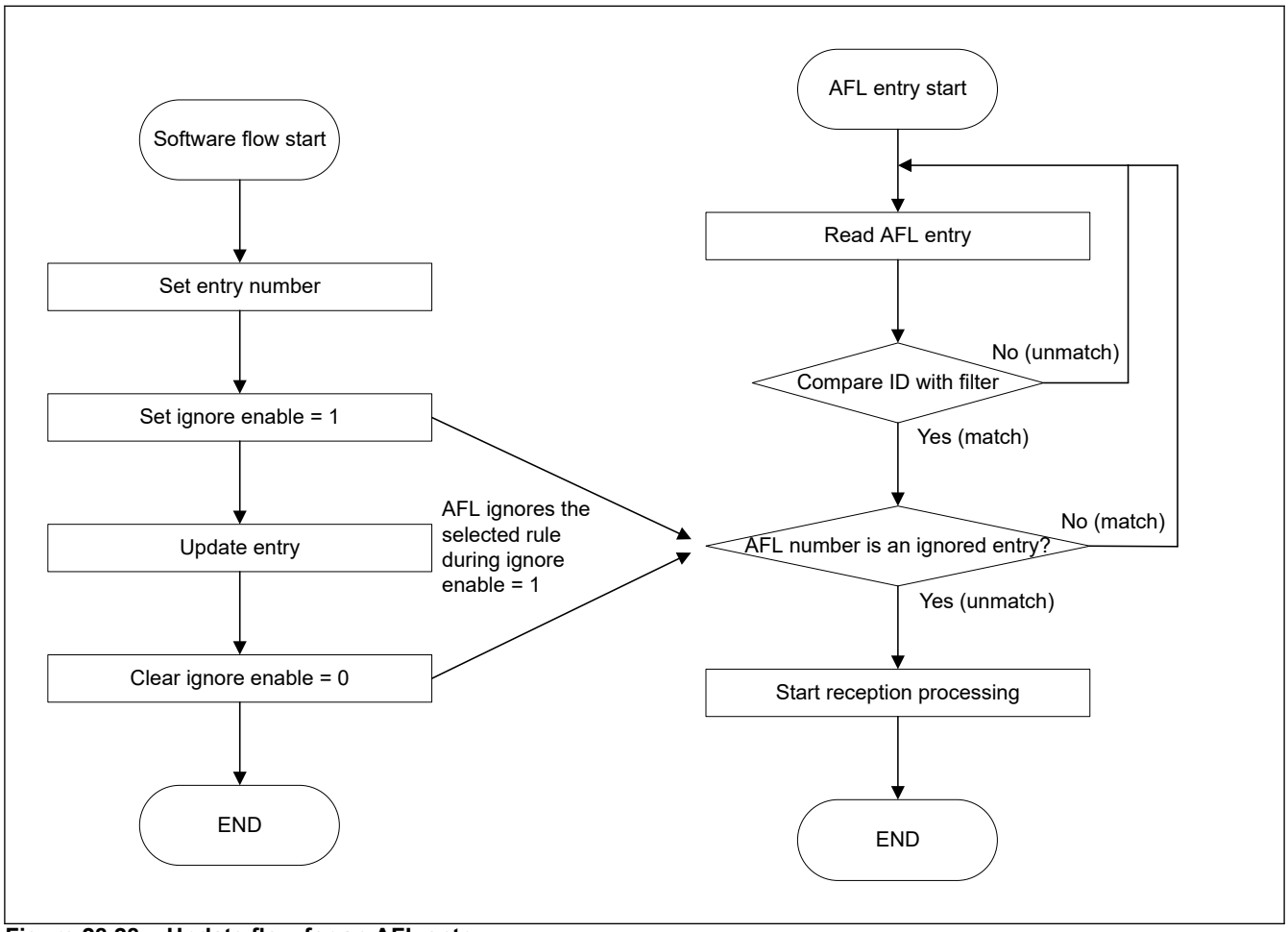


Figure 28.28 Update flow for an AFL entry

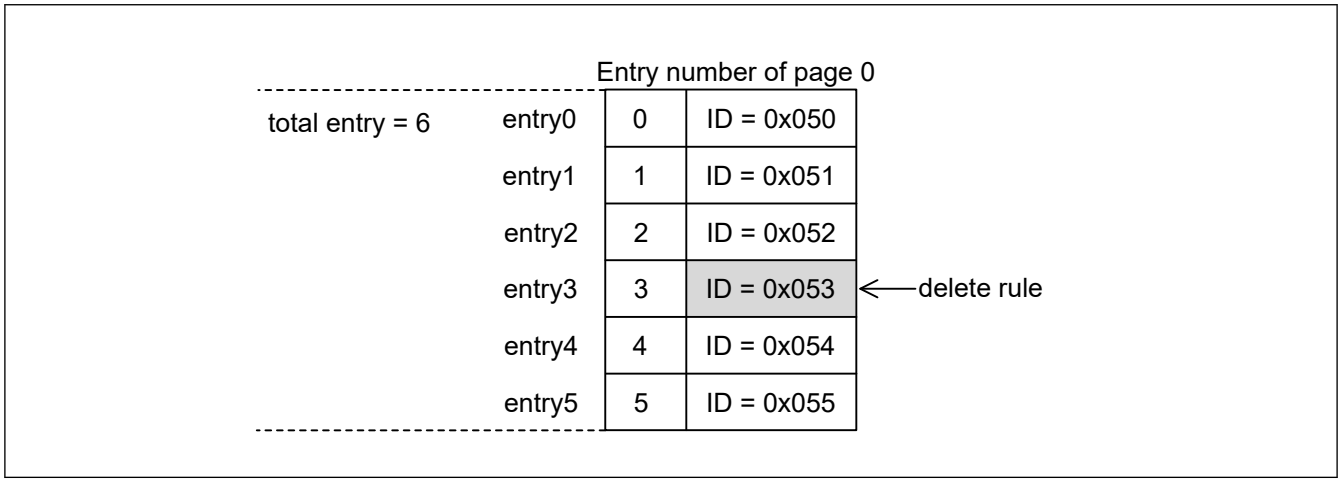
The method to update an AFL entry is as follows:

1. Set the entry number to CFDGAFLIGNENT register.
2. Set the value 0xC401 (key code and enable bit) to CFDGAFLIGNCTR register.
3. Set the entry page to CFDGAFLECTR register. This page includes the selected entry. CFDGAFLECTR.AFLDAE is set to 1.
4. Set the new rule to CFDGAFLIDr, CFDGAFLMr, CFDGAFLP0r, CFDGAFLP1r registers.
5. CFDGAFLECTR.AFLDAE is cleared to 0.
6. Set the value 0xC400 (key code and clear enable bit) to CFDGAFLIGNCTR register.

Note: This entry number is ignored during the periods from (2) to (5).

(1) Example 1: Deleting an entry

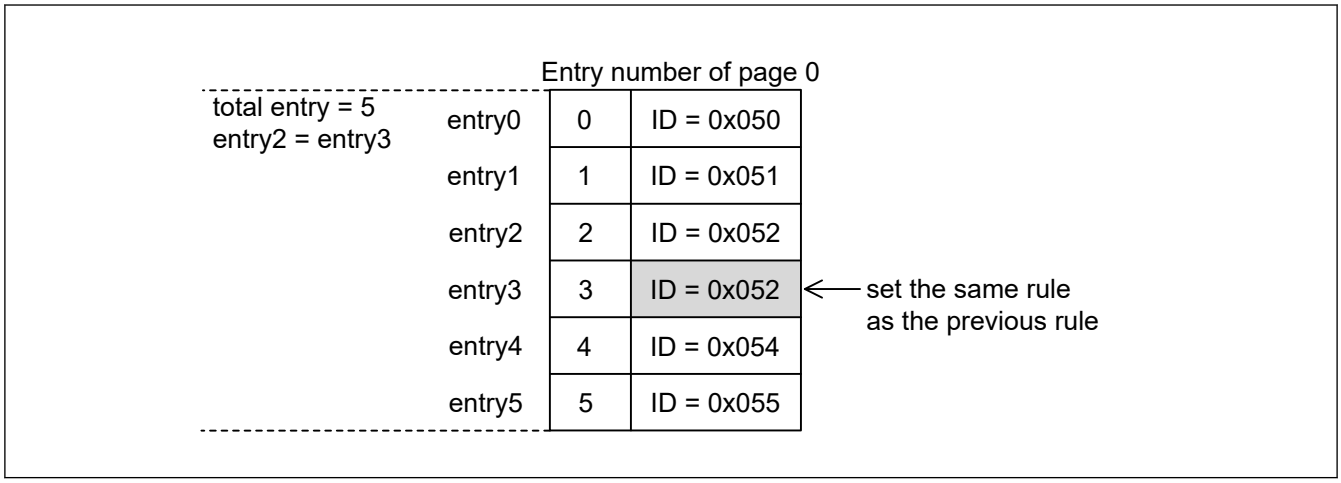
Deleting entry3 when the total number of entries is 6 channel.



How to delete an entry

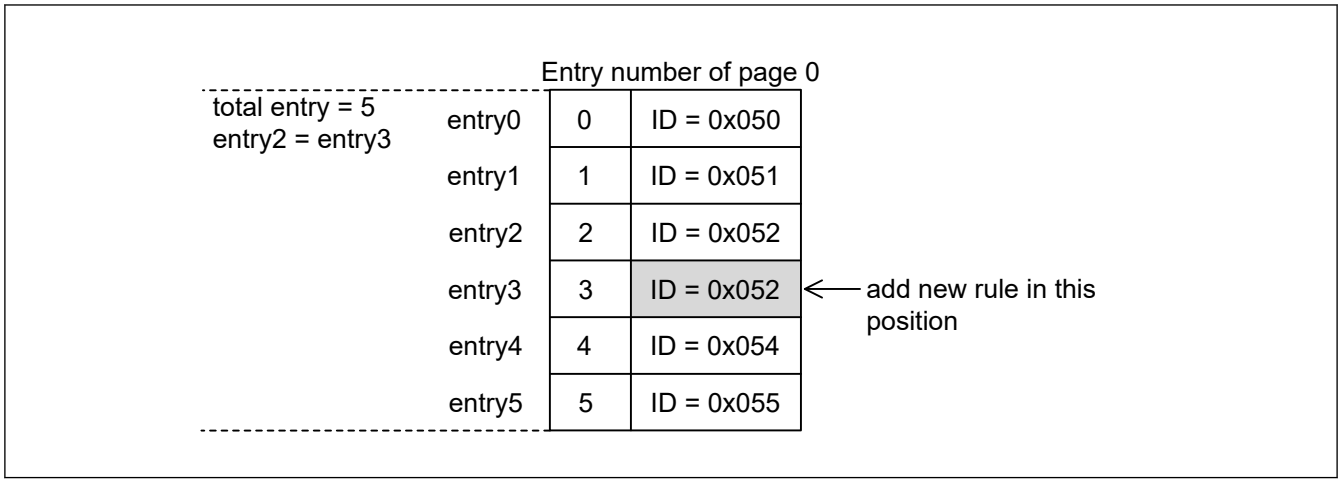
1. Set 0x00000003 to CFDGAFLIGNENT register.
2. Set 0x0000C401 to CFDGAFLIGNCTR register.
3. Set 0x00000100 to CFDGAFLECTR register.
4. Set the same rule as the previous rule by accessing CFDGAFLIDr, CFDGAFLMr, CFDGAFLP0r, CFDGAFLP1r (r = 3, this is entry3).
5. Set 0x00000000 to CFDGAFLECTR register.
6. Set 0x0000C400 to CFDGAFLIGNCTR register.

Entry3 is now deleted.



(2) Example 2: Adding an entry

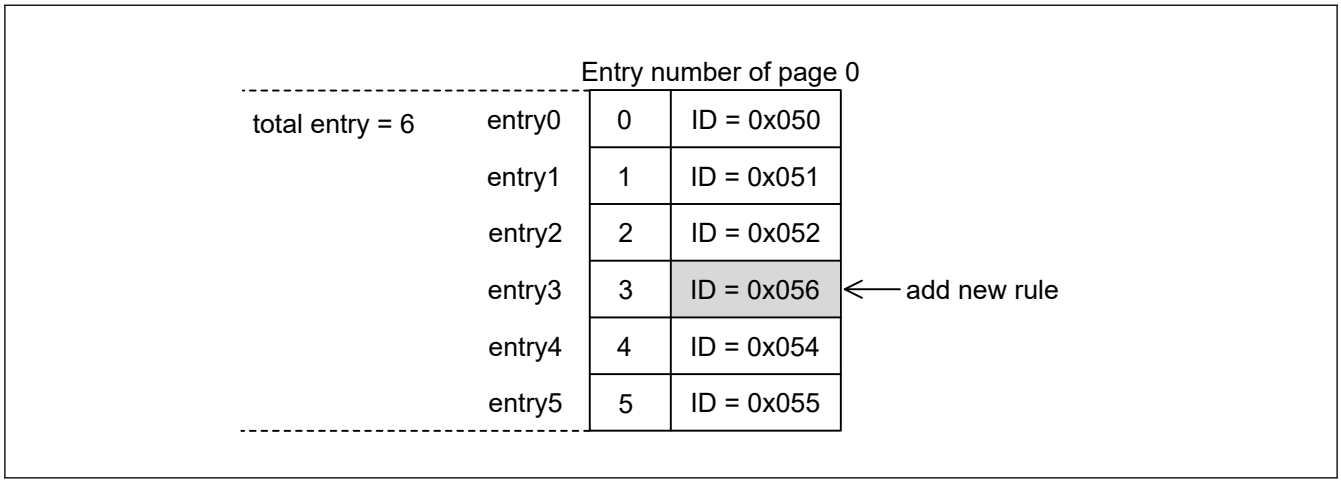
Adding a new entry to entry3 when the total number of entries is 6.



How to add an entry

1. Set 0x00000003 to CFDGAFLIGNENT register.
2. Set 0x0000C401 to CFDGAFLIGNCTR register.
3. Set 0x00000100 to CFDGAFLECTR register.
4. Set the new rule by accessing CFDGAFLIDr, CFDGAFLMr, CFDGAFLP0r, CFDGAFLP1r (r = 3, this is entry3).
5. Set 0x00000000 to CFDGAFLECTR register.
6. Set 0x0000C400 to CFDGAFLIGNCTR register.

The new entry is now added.



The AFL filter can be used to set CFDGAFLCFG, and addition/deletion of an entry is possible. Therefore, it is necessary to set the maximum number to be used to CFDGAFLCFG.

28.6 FIFO Buffers and Normal Message Buffer Configuration

This section describes the process for configuring the number of RX message buffers, the FIFO buffers, and the flat TX message buffers in the CANFD module. The message buffers are mapped as shown in Figure 28.29.

The RX message buffers can be accessed with the RX Message Buffer Registers.

The RX FIFO buffers and the common FIFO buffers configured in RX mode or TX mode can only be accessed with the FIFO Access Registers.

If the common FIFO is configured in TX mode, you can only write data into the FIFO buffer using the FIFO Access registers.

If the common FIFO is configured in RX mode, you can only read data from the FIFO Access Registers.

The TX message buffers can be accessed with the TX Message Buffer Registers.

If unused message buffer locations are read, the message buffer locations are read as unknown values.

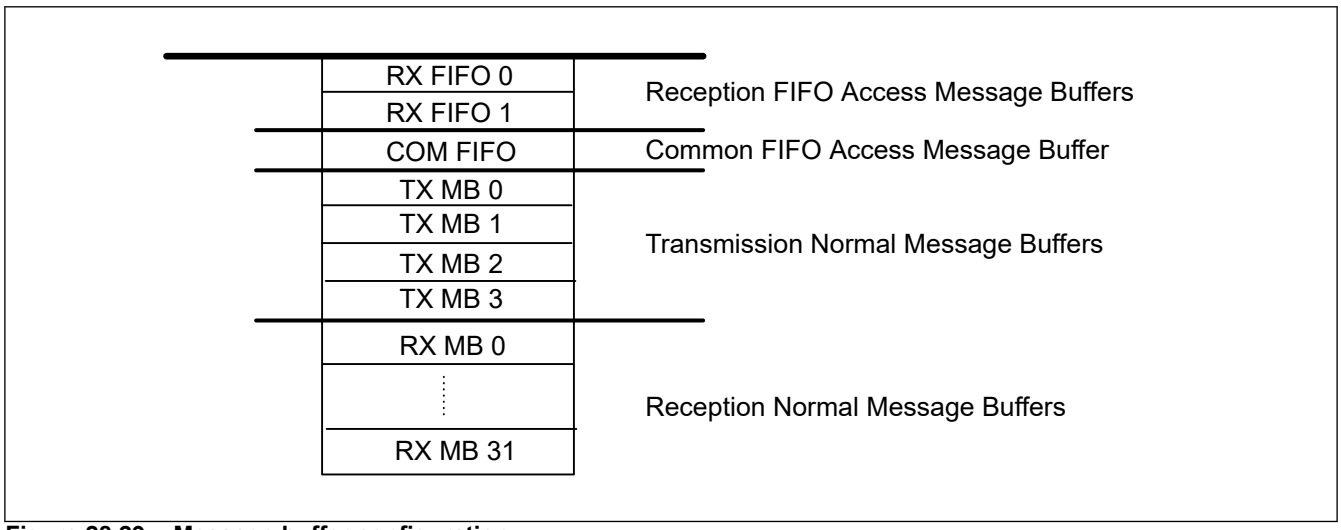


Figure 28.29 Message buffer configuration

28.6.1 Normal RX Message Buffers

In CANFD module, the frames received can be stored in normal RX message buffers based on the configuration of the AFL entries.

Additionally, the number of normal RX message buffers required in the system can be chosen up to a fixed maximum limit.

28.6.1.1 Normal RX Message Buffer Configuration

In CANFD module, the number of normal RX message buffers can be configured by writing to the RX Message Buffer Number Register.

The limiting values for the configuration of number of message buffers are:

- Minimum value = 0x00 (no normal RX MB)
- Maximum value = 0x20

Do not use values outside these limits.

The AFL entries for routing the received messages to normal RX message buffers must be configured to match the requirements of the system.

The AFL entries must also be configured properly, and an AFL entry for normal RX message buffers should not exceed the number of message buffers configured in the RX Message Buffer Number Register.

Note: There is no internal check procedure provided in CANFD module against wrong configuration of the AFL.

The data field size of the RX message buffer can be configured with the CFDRMNB.RMPLS bit. The default size is 8 bytes and the maximum data payload size is 64 bytes.

When the receiving frame exceeds the data field size, then the acceptance depends on the configuration of CFDGCFG.CMPOC (message rejecting or data payload cut).

Note: RMPLS and CMPOC bit is not available in the classical CAN function, so, these feature is not valid for classical CAN.

28.6.2 FIFO Buffers

The CANFD module provides a fixed number of FIFO buffers to support storage of frames for reception and transmission functions .

The number of reception-only FIFO buffers is fixed to 2. However, common FIFO buffer channel can be configured to store messages for transmission or reception function.

These FIFO buffers can be enabled or disabled, and the following parameters can be configured to match the system requirements:

- Size
- Interrupt structure
- Message lost mechanism
- Message overwrite mechanism of the FIFO buffers
- Location of the TX FIFO.

When the receiving frame exceeds the data field size, the acceptance depends on the configuration of the CFDGCFG.CMPOC bit (message rejecting or data payload cut).

28.6.2.1 FIFO Buffers Configuration

In CANFD module, the FIFO buffers can be configured to match the system requirements.

The total number of FIFO buffers = 2 RX FIFO buffers + 1 common FIFO buffer = 3 FIFO buffers.

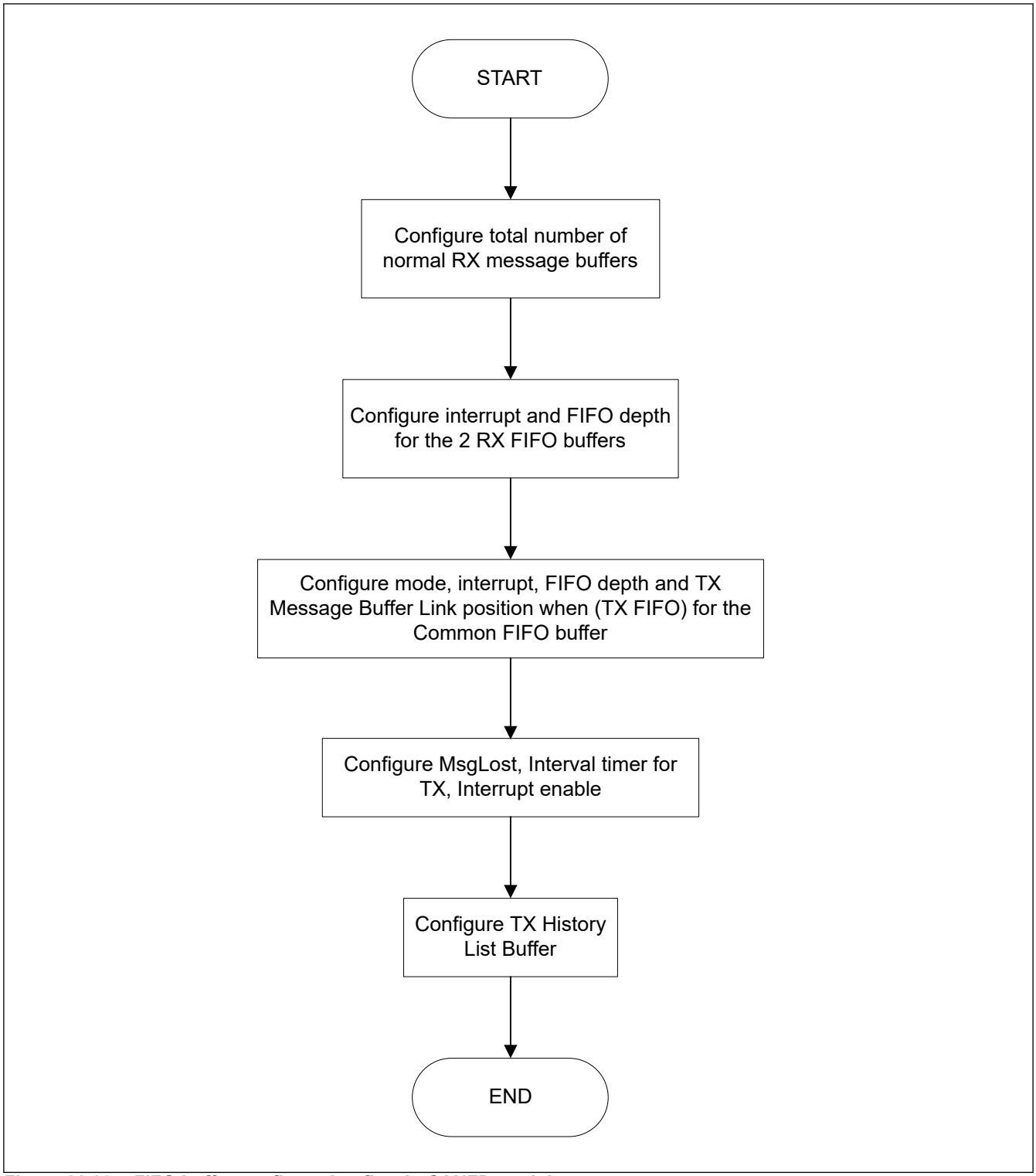


Figure 28.30 FIFO buffer configuration flow in CANFD module

As shown in Figure 28.30, the various FIFO buffers can be configured by writing to the RX FIFO Configuration/Control Registers and the Common FIFO Configuration/Control Registers.

For the 2 RX FIFO buffers, the following parameters can be configured:

- Interrupts
- FIFO depth
- FIFO payload data size.

For the common FIFO buffer, the following parameters can be configured:

- Mode
- Interrupts FIFO depth
- FIFO payload data size
- FIFO TX link position.

(1) FIFO mode configuration of Common FIFO buffer

The mode of the common FIFO buffer can be configured by writing to the CFDCFCC.CFM[1:0] bits in the Common FIFO Configuration/Control Register. The possible modes of configuration for Common FIFO buffer are:

- 0b RX mode (default mode after hardware reset)
- 1b TX mode

Messages can only be read from the RX FIFO buffers and the Common FIFO buffer configured in RX mode. Messages are stored by the CAN module in these FIFO buffers based on the AFL entries.

Messages can be read and written into the Common FIFO buffer configured in TX mode. These messages are transmitted on the appropriate CAN channel.

The pointers can only be incremented when a new message is stored in the FIFO buffer and decremented when a message is transmitted on the corresponding CAN channel by the CANFD module.

After a hardware reset, the Common FIFO buffer is configured in RX mode by default. Only enable the FIFO buffers after configuring the Common FIFO buffer in the required modes.

(2) FIFO TX message buffer link configuration

When the common FIFO is configured as TX FIFO, the FIFO buffer must be linked to a normal TX message buffer to participate in the transmission scan.

Do not write data into a TX message buffer that is linked to a Common FIFO buffer. Also, the TX message buffer linked to a Common FIFO buffer should not be a part of the TX Queue.

The TX message buffer link of each Common FIFO buffer can be configured by writing to the CFDCFCC.CFTML[1:0] bits in the Common FIFO Configuration/Control Registers. Available options for TX message buffer link configuration are:

- 0x00: TX Message Buffer 0
- 0x01: TX Message Buffer 1
- 0x10: TX Message Buffer 2
- 0x11: TX Message Buffer 3

(3) FIFO depth configuration

The depth of each FIFO buffer can be configured by writing to the CFDRFCCa.RFDC[2:0] bits and CFDCFCC.CFDC[2:0] bits in the RX FIFO Configuration/Control Registers and the Common FIFO Configuration/Control Registers. The 6 available options for depth configuration are:

- 0x000: 0 Message (FIFO buffer cannot be enabled)
- 0x001: 4 Messages
- 0x010: 8 Messages
- 0x011: 16 Messages
- 0x100: 32 Messages
- 0x101: 48 Messages

The RAM allocation for RX message buffers along with FIFO buffers is limited to 16 messages with 64 data bytes. Configuration of the RX message buffers, along with FIFO buffers, that exceeds this maximum limit should not be done.

CANFD module logic does not check the validity of the configuration.

Note: If the FIFO depth of a common FIFO is 4 messages or more (CFDCFCC.CFDC[2:0] > 000b), then the Common FIFO TX message buffer link is valid when the FIFO is disabled or enabled.

If FIFO depth is 0 messages, then the Common FIFO TX message buffer link is not valid when the FIFO is disabled or enabled.

(4) FIFO payload size configuration

The data size of each FIFO buffer can be configured by writing to the CFDRFCCa.RFPLS[2:0] bits and CFDCFCC.CFPLS[2:0] bits in the RX FIFO Configuration/Control Registers and the Common FIFO Configuration/Control Registers. The eight available options for depth configuration are:

- 000b: 8 bytes
- 001b: 12 bytes
- 010b: 16 bytes
- 011b: 20 bytes
- 100b: 24 bytes
- 101b: 32 bytes
- 110b: 48 bytes
- 111b: 64 bytes

The RAM allocation for RX message buffers along with FIFO buffers is limited to 16 messages with 64 data bytes. Configuration of the RX message buffers, along with FIFO buffers, that exceeds this maximum limit should not be done.

CANFD module logic does not check the validity of the configuration.

Note: This feature is not available in the classical CAN function.

(5) FIFO interrupt configuration

The Interrupt generation conditions for the FIFO buffers can be configured by writing to the CFDRFCCa.RFIM and CFDCFCC.CFIM bit in the RX FIFO Configuration/Control Registers and the Common FIFO Configuration/Control Registers. The two available options are:

- 0:
  - RX FIFO mode: Interrupt generated when the Common FIFO counter reaches CFDRFCCa.RFIGCV/ CFDCFCC.CFIGCV value
  - TX FIFO mode: Interrupt generated when the Common FIFO transmits the last message successfully
- 1:
  - RX FIFO mode: Interrupt generated at the end of storage of every received message
  - TX FIFO mode: Interrupt generated for every successfully transmitted message

If the Interrupt Mode bit is 0 for a RX FIFO, then interrupt is generated based on the configuration of the CFDRFCCa.RFIGCV[2:0] bits.

Similarly, if the Interrupt Mode bit is 0 for a Common FIFO configured in RX mode, then interrupt is generated based on the configuration of CFDCFCC.CFIGCV[2:0] bits.

The eight available options for configuring the FIFO counter value for generation of an interrupt are:

- 000b: Interrupt generated when FIFO is 1/8th Full
- 001b: Interrupt generated when FIFO is 1/4th Full
- 010b: Interrupt generated when FIFO is 3/8th Full
- 011b: Interrupt generated when FIFO is 1/2 Full
- 100b: Interrupt generated when FIFO is 5/8th Full
- 101b: Interrupt generated when FIFO is 3/4th Full
- 110b: Interrupt generated when FIFO is 7/8th Full
- 111b: Interrupt generated when FIFO is Full.

In this case, an interrupt is generated when the message count matches the configured value.

However, there are some limitations on the configuration of the CFDRFCCa.RFIGCV[2:0] and CFDCFCC.CFIGCV[2:0] bits depending on the FDC[2:0] bits (FIFO Depth Configuration), see Table 28.24.

Table 28.24 FIFO interrupt generation counter and FIFO depth configuration

RFDC[2:0] (CFDC[2:0])	RFIGCV[2:0] (CFIGCV[2:0])							
	111b	110b	101b	100b	011b	010b	001b	000b
000b	Don't care (FIFO cannot be enabled)							
001b	Allowed	Not allowed	Allowed	Not allowed	Allowed	Not allowed	Allowed	Not allowed
010b	Allowed							
011b	Allowed							
100b	Allowed							
101b	Allowed							
110b	Allowed							
111b	Allowed							

28.6.2.2 FIFO Buffers Control

The FIFO interrupt must be enabled by setting any one of the following bits in the RX FIFO Configuration/Control Registers:

- CFDRFCCa.RFIE

In addition, the FIFO interrupt must be enabled by setting any one of the following bits in the Common FIFO Configuration/Control Register:

- CFDCFCC.CFRXIE
- CFDCFCC.CFTXIE

After configuration is complete, each FIFO can be enabled by setting the CFDRFCCa.RFE and CFDCFCC.CFE bits in the RX FIFO Configuration/Control Registers and the Common FIFO Configuration/Control Register to allow transmission and reception of messages.

28.7 Interrupts and DMA

28.7.1 Interrupts

The CANFD module generates several interrupts. The interrupt output, which is connected to the Interrupt Controller Unit (ICU), can be controlled by the corresponding interrupt enable bit.

The status flag is set independent from this enable bit.

The channel transmission interrupt has an additional status flag register. The status bits are set when the corresponding interrupt enables are set.

The status flag register supports the identification of the interrupt source for the channel transmission, as this interrupt is driven by several trigger sources.

The interrupts in the CANFD module can be classified into two groups, global interrupts and channel interrupts:

- Global interrupts:
  The CANFD module can generate 3 global interrupts:
  - Global interrupt for successful reception into the 2 RX FIFO buffers
  - Global error interrupt.
  - Global Interrupt for successful reception into the 32 RX message buffers
- Channel interrupts:
  Channel of the CANFD module can generate 3 channel interrupts:

1. Channel transmission
   - Transmission completion from channel
   - Transmission abort from channel
   - Transmission from TX Queue for a channel
   - Channel THL interrupt
   - Successful transmission from a Common FIFO in TX mode for a channel.
2. Channel error interrupt
3. Successful reception in a Common FIFO in RX mode for a channel.

The interrupts are cleared when the corresponding flag bits are cleared or the Interrupt enable bits are cleared.

Table 28.25 gives an overview of interrupt sources for the different interrupt outputs. The interrupt outputs are active-high.

Table 28.25 Interrupt source overview

Parameter	Interrupt	Name	Interrupt source	Interrupt clearing
Global Interrupts	Successful reception into at least one RX FIFO	CAN_RXF	Interrupt flag of corresponding RX FIFO for which interrupt is enabled	Clear the interrupt flag of corresponding RX FIFO buffer for which interrupt is enabled
	Global Error	CAN_GLR	Any of the following: • DLC Error flag • Message Lost Status bit • TX History Entry Lost Status bit • CANFD Message Payload overflow flag	Clear all of : • DLC Error flag • Message Lost flags in all of the FIFO Status Registers • TX History List Entry Lost flag • CANFD Message Payload overflow flag
	Successful reception into at least one RXMB	CAN0_RXMB	Interrupt flag of corresponding RXMB for which interrupt is enabled	Clear the interrupt flag of corresponding RXMB buffer for which interrupt is enabled
Channel Transmission Interrupts	Channel successful transmission	CAN0_TX	Any channel related TXMB Successful flag when interrupt is enabled*1	Clear all channel related TXMB Result status bits for which the interrupt is enabled
	Channel Abort		Any channel related TXMB Abort flag when interrupt is enabled*1	Clear all channel related TXMB Result Status bits for which the interrupt is enabled globally
	Channel transmission from TX Queue		Related channel TX Queue Interrupt flag	Clear related channel TX Queue Interrupt flag
	Channel THL Interrupt		Channel THL Interrupt status flag	Clear the relevant THL Interrupt status flag
	Channel COM FIFO TX Interrupt		Interrupt Flag for Common FIFOs in TX mode belonging to the related channel	Clear the interrupt flags of Common FIFOs in TX mode belonging to the related channel
Channel Error Interrupt	Channel Error	CAN0_CHERR	Any channel related error flag in the Channel Error Flag Register for which interrupt is enabled in the Channel Error Interrupt Enable Register	Clear all channel related error flags in the Channel Error Flag Register for which interrupt is enabled in the Channel Error Interrupt Enable Register
Channel COM RX FIFO Interrupt	Channel COM FIFO RX Interrupt	CAN0_COMFRX	Interrupt flag for Common FIFOs in RX mode belonging to the related channel	Clear the interrupt flags of Common FIFOs in RX mode belonging to the related channel

Note 1. These interrupts are only set for TX Message Buffers that do not belong to anenabled TX Queue andare not pointing to a common FIFO.
Separate interrupts are provided for common FIFO buffers and TX Queue.

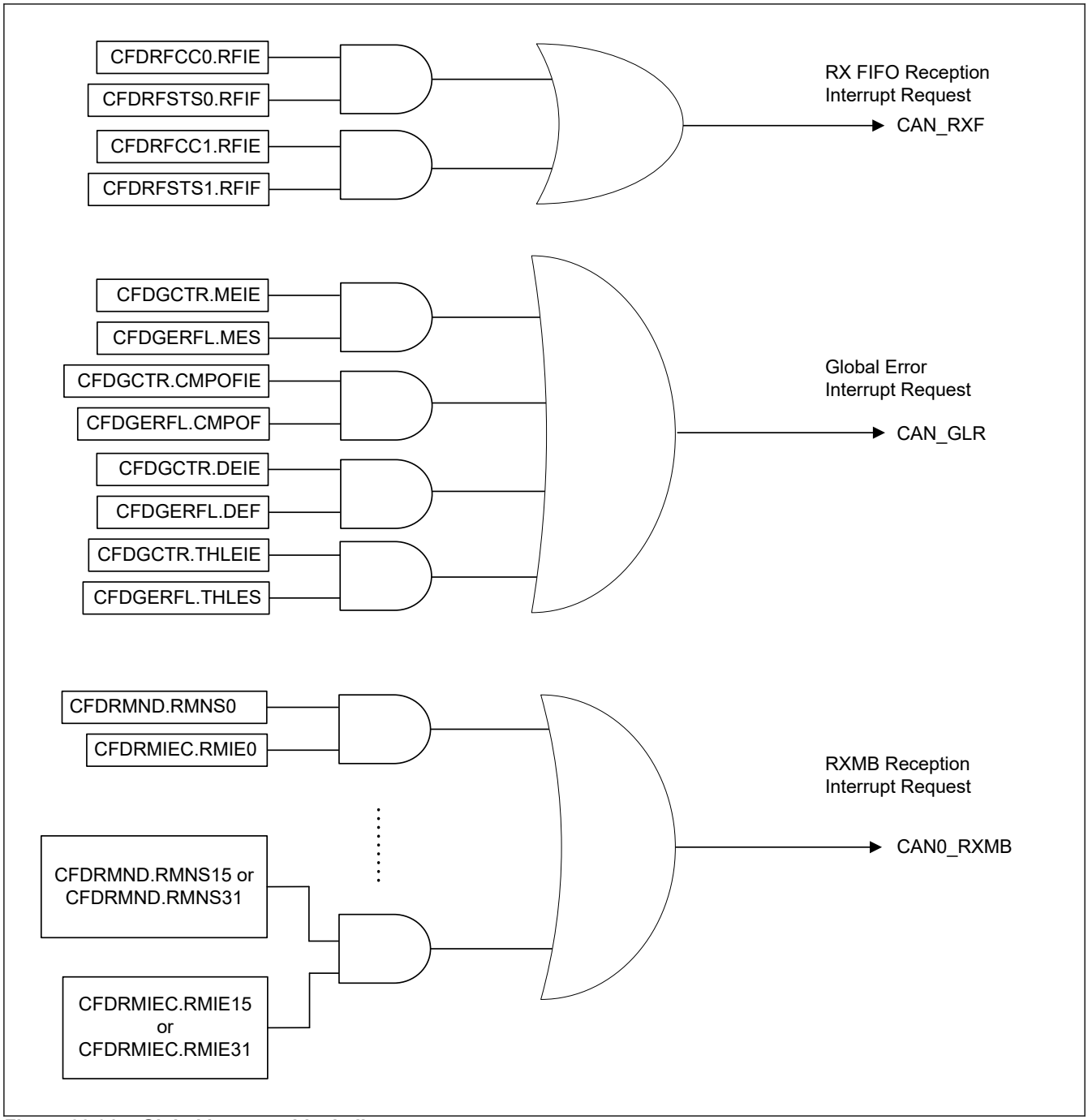


Figure 28.31 Global interrupt block diagram

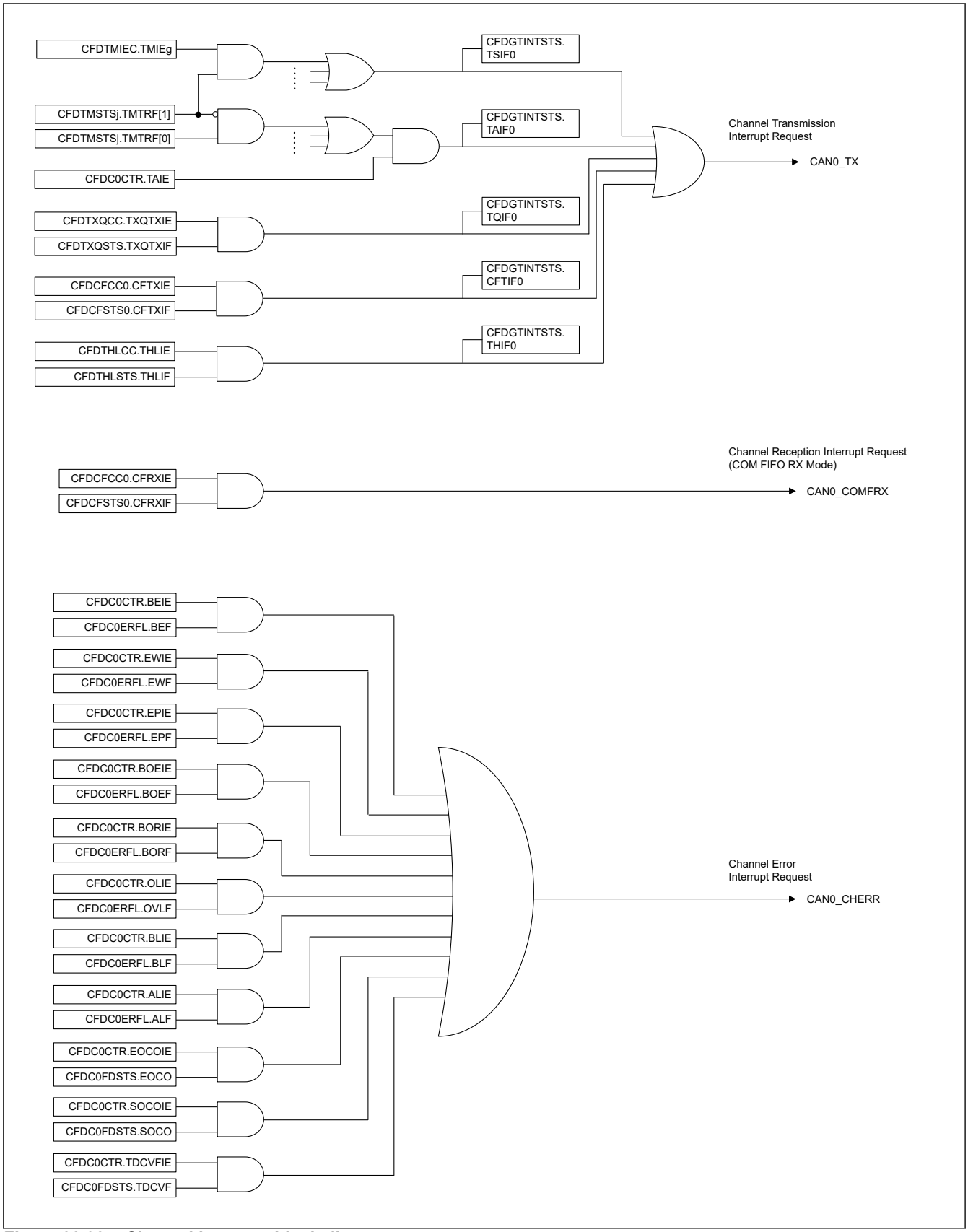


Figure 28.32 Channel interrupt block diagram

28.7.2 DMA Transfer

The CANFD module has message buffers that can be associated with a DMA channel:

- Reception DMA
  - 2 RX FIFO message buffers
  - Common FIFO Message Buffer

Figure 28.33 shows the potential DMA channels.

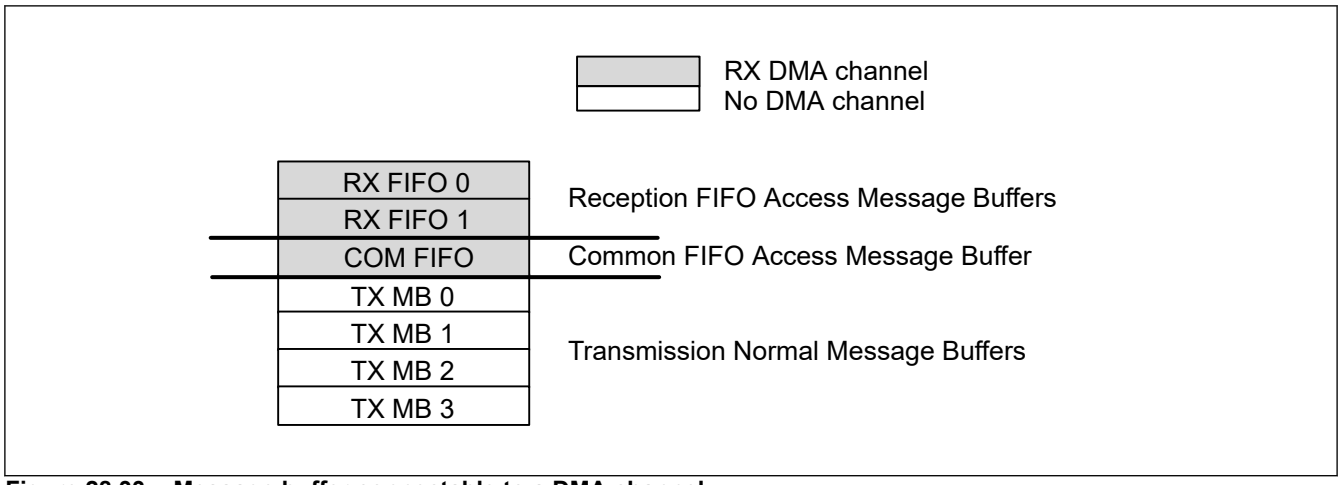


Figure 28.33 Message buffer connectable to a DMA channel

A DMA channel transfer request is generated for each FIFO entry to the DMAC when the related CFDCDTCT.RFDMAE or CFDCDTCT.CFDMAE is set to 1 and the belonging FIFO is not empty.

Reception FIFO Interrupt should be disabled for this particular FIFO (CFDRFCCa.RFIE or CFDCFCC.CFRXIE)

Use the regular start address for the DMA access window address. See Figure 28.34.

Table 28.26 DMA channel access window address

b = Message buffer component index	Message Buffer Component	Register	P	Regular Start Address
b = [0…1]	RFMBCPb[0]	CFDRFIDb	x	0x0520 + b × 0x004C
		CFDRFPTRb	x	0x0524 + b × 0x004C
		CFDRFFDSTSb	x	0x0528 + b × 0x004C
		CFDRFDFbp	[0…15]	0x052C + p × 0x0004 + b × 0x004C
—	CFMBCP0[0]	CFDCFID	x	0x05B8
		CFDCFPTR	x	0x05BC
		CFDCFFDCSTS	x	0x05C0
		CFDCFDFp	[0…15]	0x05C4 + p × 0x0004

DMA FIFO pointer decrement is done automatically by reading the last configured data payload byte (CFDRFCCa.RFPLS or CFDCFCC.CFPLS).

Note: The DMA must read the exact length of the configured data payload size (CFDRFCCa.RFPLS or CFDCFCC.CFPLS).

Note: This feature is not available for classical CAN function because CFDRFCCa.RFPLS and CFDCFCC.CFPLS are not in classical CAN.

Do not write to the FIFO control registers when DMA is enabled. The DMA enable of the particular DMA FIFO (CFDCDTCT.RFDMAE or CFDCDTCT.CFDMAE) can be set at any time. Figure 28.34 shows a configuration flow for an initial setup.

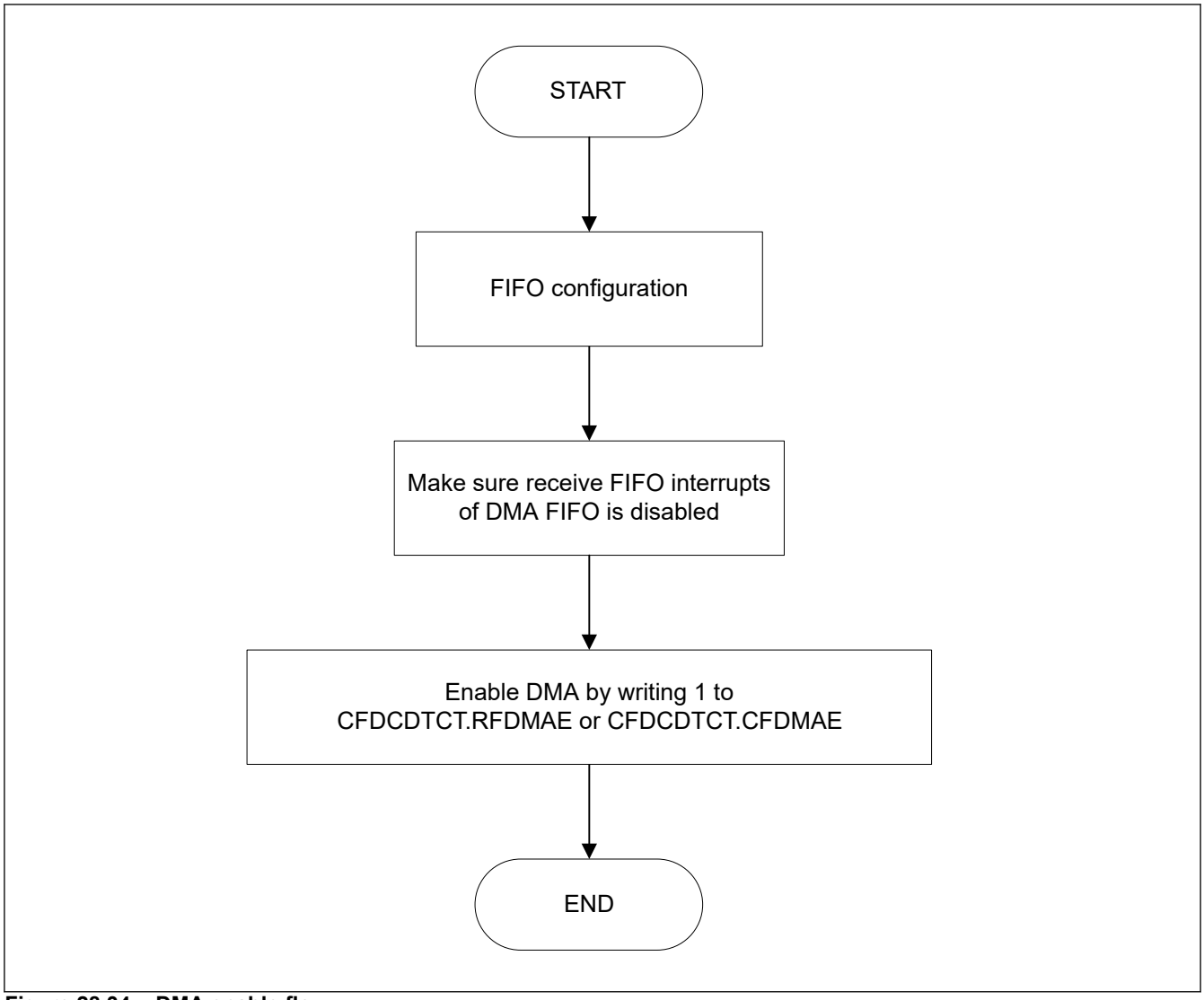


Figure 28.34 DMA enable flow

To disable a DMA transfer request, you must disable the particular DMA enable bit (CFDCDTCT.RFDMAE or CFDCDTCT.CFDMAE). If the disable is made during an ongoing transfer, then the transfer must be completed first before further action can be taken. The transfer status can be identified by the CFDCDTSTS.RFDMASTS or CFDCDTSTS.CFDMASTS bit. See Figure 28.35 for the DMA disable flow. When the DMA is disabled, consideration should be made for the remaining or new incoming messages to this particular reception FIFO.

When the FIFO is not disabled, reception to the FIFO continues.

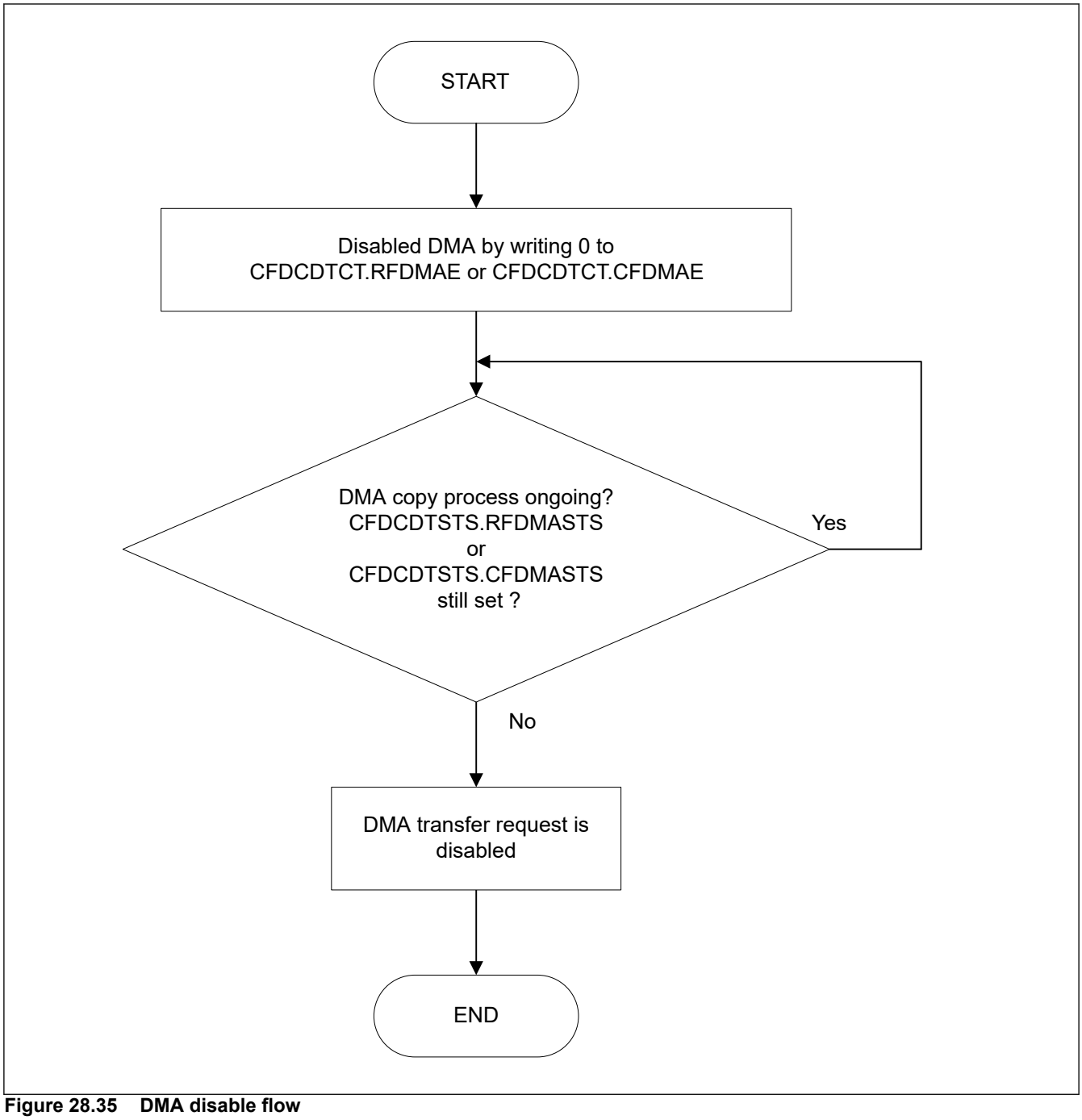


Figure 28.35 DMA disable flow

28.8 Reception and Transmission

28.8.1 Reception

In the CANFD module, CAN messages received on any of the channels are stored in RX message buffers, RX FIFO buffers, or Common FIFO buffers configured in RX mode depending on the Acceptance Filter List entries.

- Up to 32 RX message buffers can be configured
- 2 RX FIFO buffers available
- 1 Common FIFO Buffer can be configured in RX mode

28.8.1.1 Message Storage in RX Message Buffers

When a message is successfully received and stored in a RX message buffer, the corresponding New Data flag is set in the RX Message Buffer New Data Register.

The CAN message can be read from the corresponding RX message buffer.

If a new message is stored into a RX message buffer before the previous message in this message buffer can be read, then the original message is overwritten. There is no mechanism for preventing a new message from overwriting the current message in the RX message buffer. If such a loss of messages is not acceptable, then RX FIFO should be used for storing related messages.

Note: Users should do the same processing as the existing software flow also when using interrupt. (see Figure 28.37)

Note: Unused data bytes are filled with 0x00 depending on the DLC value.

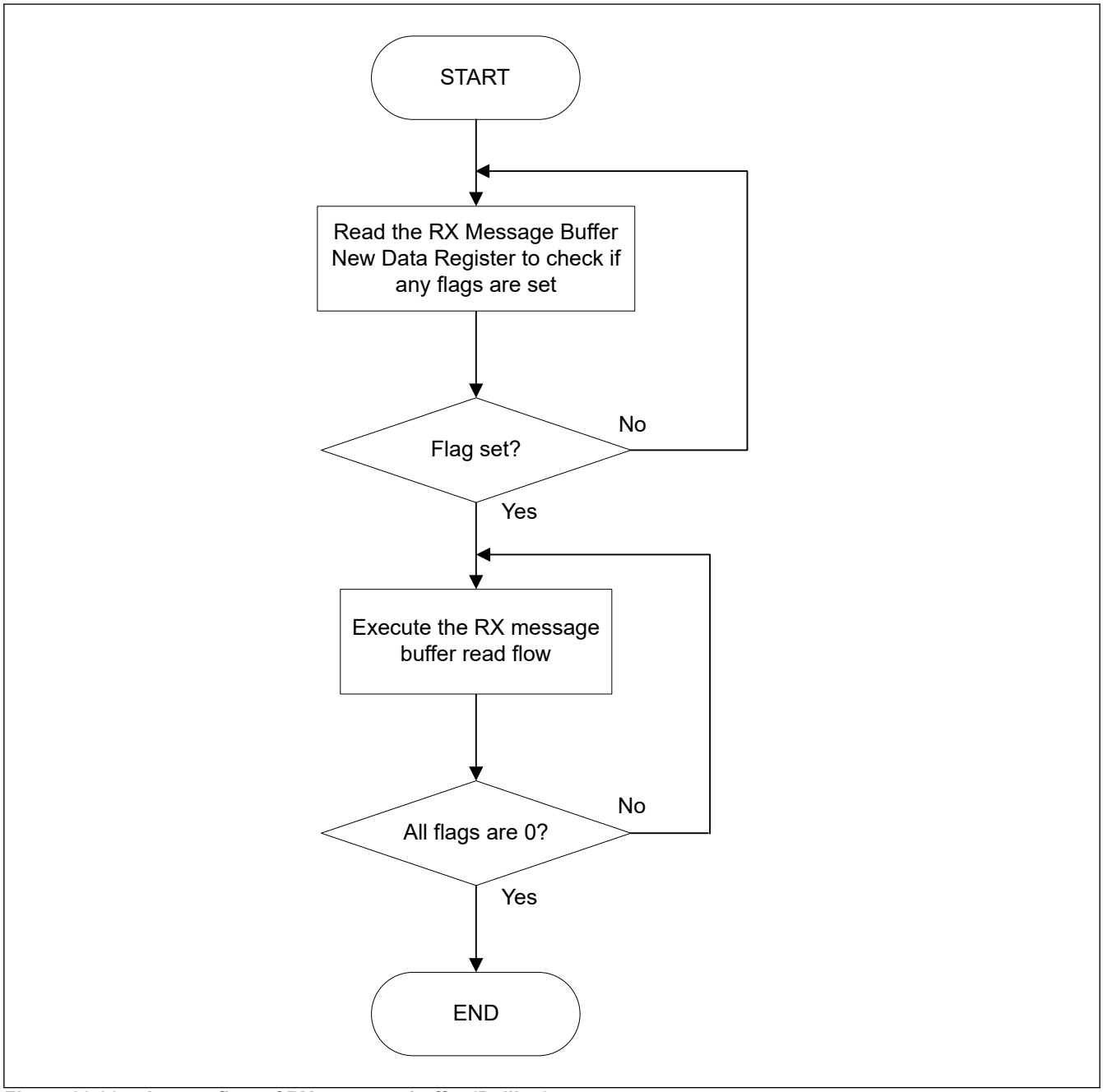


Figure 28.36 Access flow of RX message buffer (Polling)

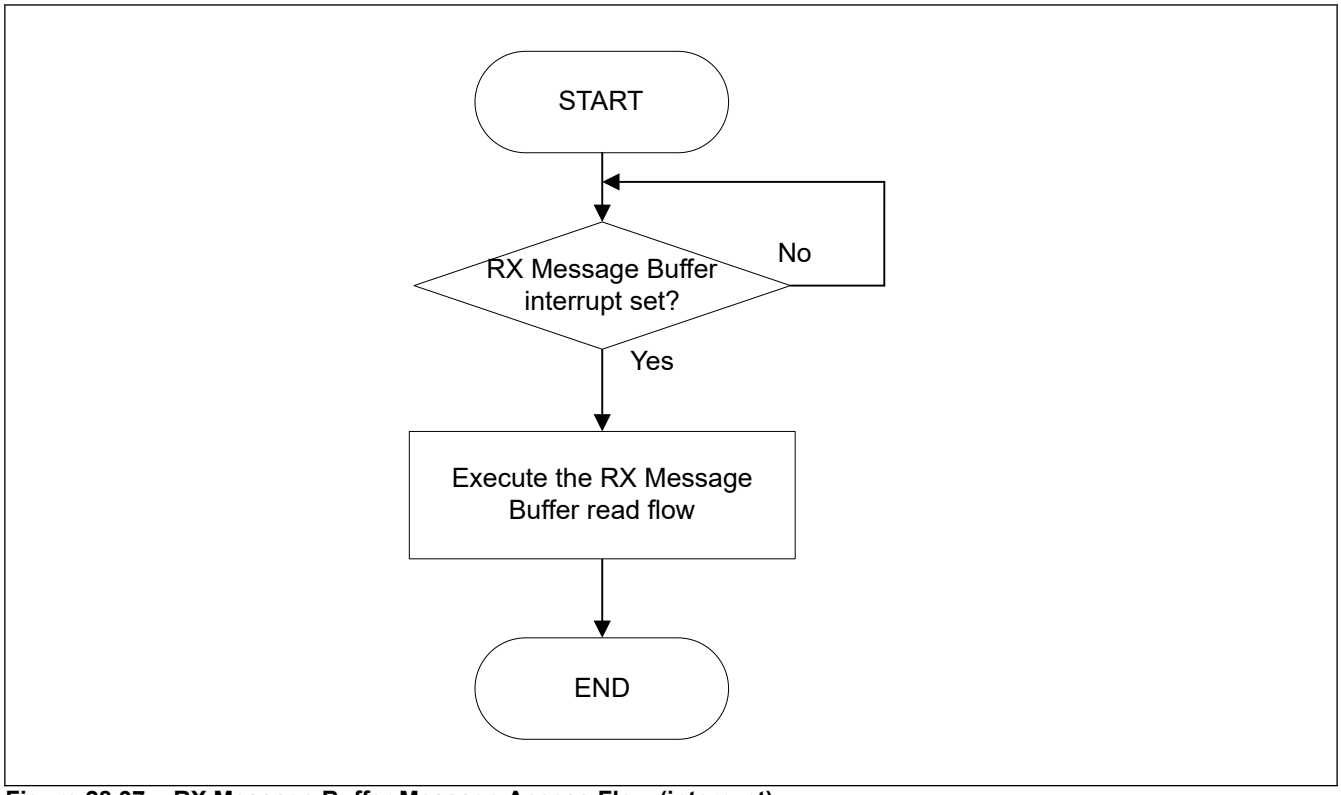


Figure 28.37 RX Message Buffer Message Access Flow (interrupt)

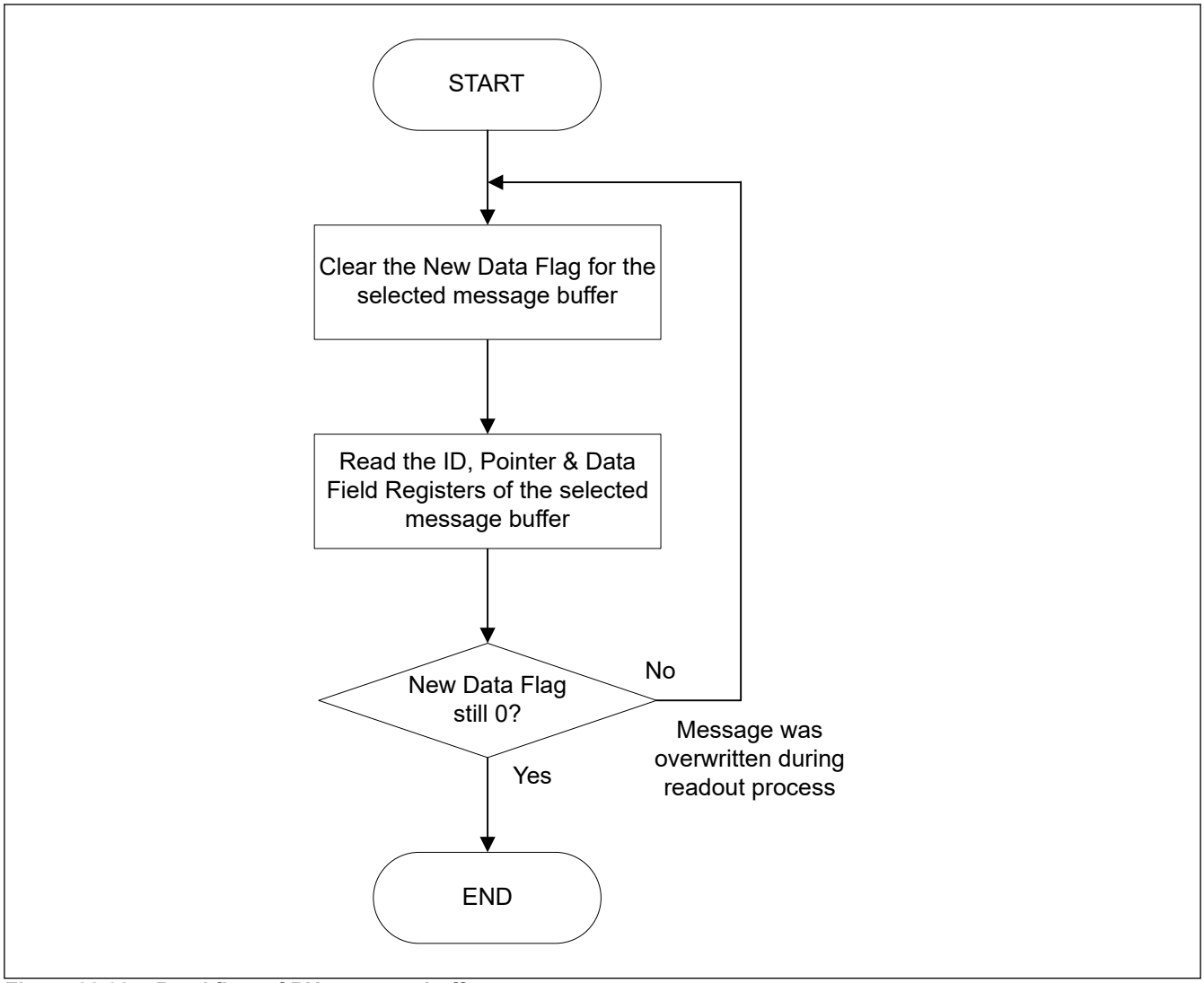


Figure 28.38 Read flow of RX message buffer

28.8.1.2 Message Storage in FIFO Buffers

The AFL entries for routing the received messages to RX FIFO buffers or Common FIFO buffer configured in RX mode should be configured based on system requirements.

The CFDGAFLP1r.GAFLFDP[8,1:0] field in the matching AFL entry selects the FIFO buffers to which the related reception message is stored.

When the received message is stored in one or more RX FIFO buffers or Common FIFO buffer configured in RX mode, the message counter value is incremented in the corresponding RX FIFO Status Registers or Common FIFO Status Register.

Depending on the configuration of the FIFO buffers, an interrupt might also be generated.

The message can be read from the corresponding FIFO Access registers.

Note: Because many messages can be stored in the FIFO buffers, reading more than one message may be required to read the latest message stored in a FIFO buffer.

If the message count value matches the FIFO depth, the FIFO Full flag is set.

When the value 0xFF is written to the corresponding FIFO Pointer Control Register, the message count is decremented by 1.

Only write 0xFF to the FIFO Pointer Control register after reading the complete message from the FIFO Access registers of the corresponding FIFO.

When all the messages stored in the FIFO are read, the FIFO Empty flag is set.

If a new message is stored into the FIFO when the FIFO message count matches the FIFO depth (FIFO full condition), the FIFO Message Lost flag is set and the new message is lost (no overwrite of already stored messages takes place).

An appropriate value can be configured as warning level to generate an interrupt before the FIFO full condition occurs to avoid loss of a message due to an overrun condition.

Note: The message lost can be set only in RX mode by CAN, and the flag is not set when the CPU is overloading the FIFO buffers.

The RX FIFO buffers and the Common FIFO buffers configured in RX mode can be disabled at any time by clearing the CFDRFCCa.RFE or CFDCFCC.CFE bit in the RX FIFO Configuration/Control Registers and the Common FIFO Configuration/Control Registers.

When the CFDRFCCa.RFE or CFDCFCC.CFE bit is cleared, the message read and write pointers of the FIFO are cleared and are no longer active. Therefore, all messages in the FIFO buffers are lost and no further messages can be stored into the FIFO.

When the RX FIFO buffers or Common FIFO buffer configured in RX mode is assigned as a DMA channel, software should not access the FIFO Access Register of this FIFO buffer or write 0xFF to the FIFO Pointer Control Register (CFDCFPCTR.CFPC or CFDRFPCTRa.RFPC). This can lead to unintended FIFO message decrement. The DMA channel controls the FIFO decrement automatically.

Note: If the interrupt flag is set for a FIFO buffer and then the FIFO is disabled, the interrupt flag is not cleared automatically. The interrupt flag should be cleared before disabling the FIFO.

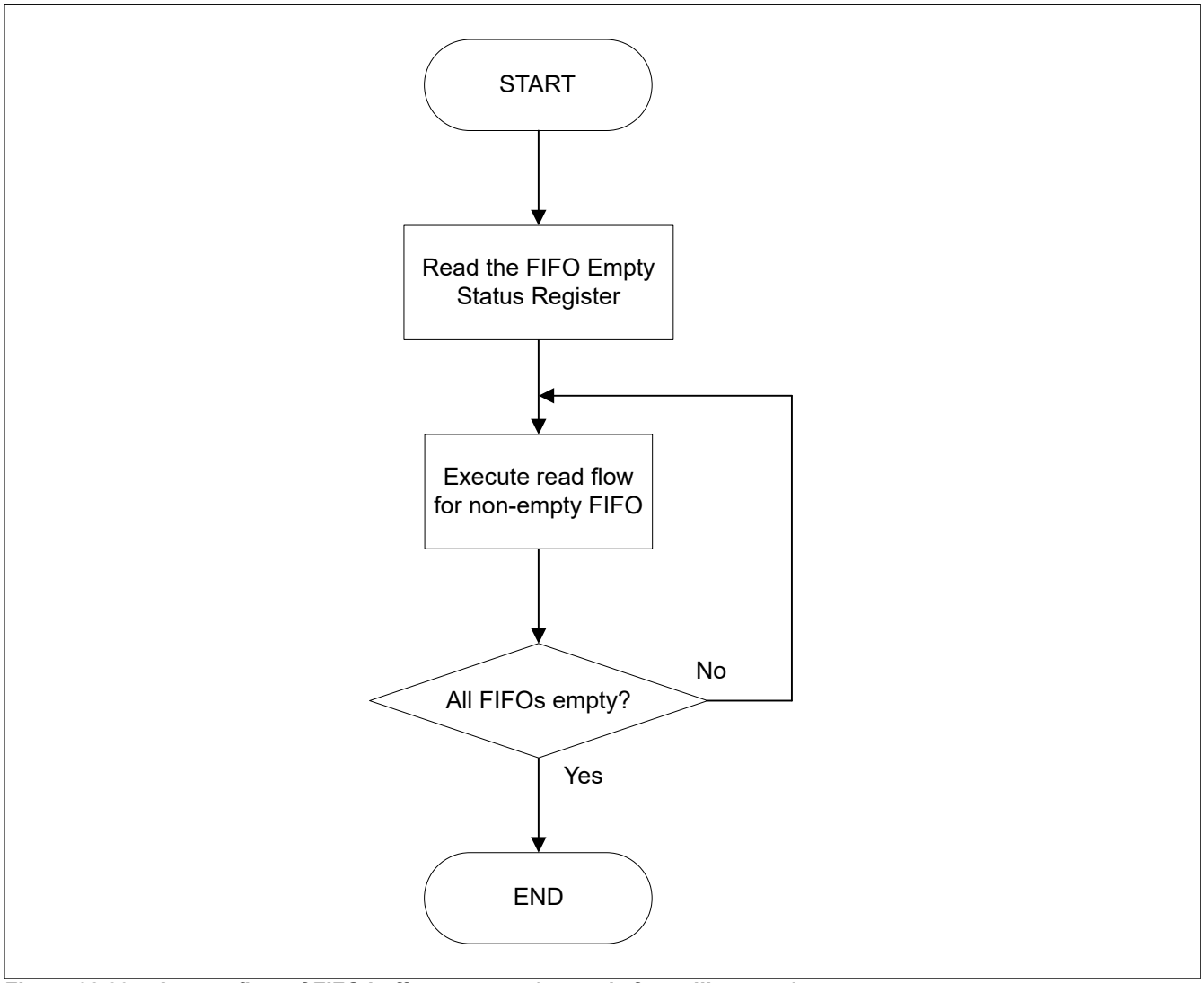


Figure 28.39 Access flow of FIFO buffer message (example for polling case)

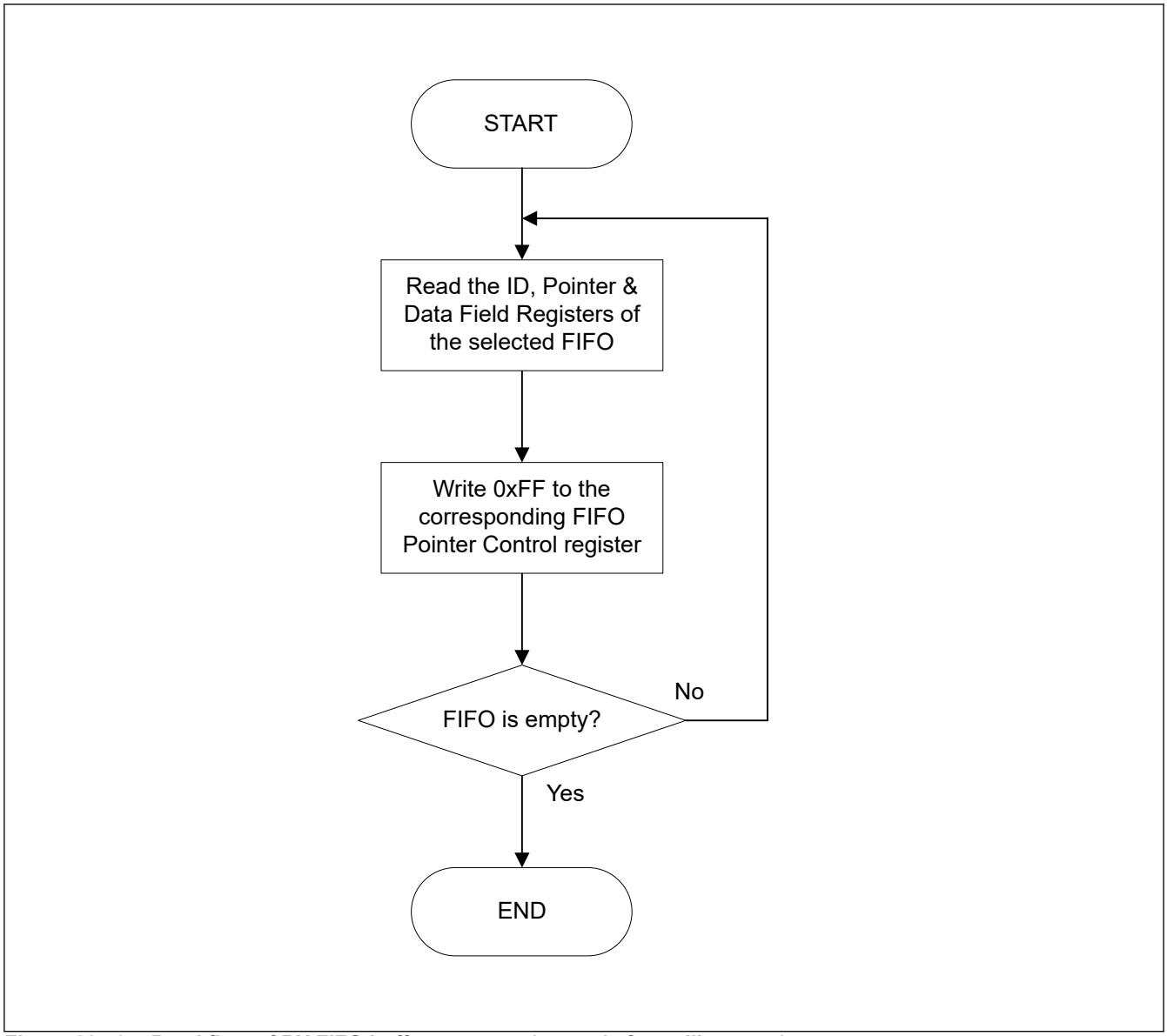


Figure 28.40 Read flow of RX FIFO buffer message (example for polling case)

Note: When the next frame is received before clearing the completion interrupt flag of reception, the completion interrupt of reception is not set again.

Even when an interruption flag is cleared after the completion processing of reception, the already received interrupt flag is not set.

It is necessary to perform the completion processing of reception even before the next completion of frame reception, and to clear an interruption flag.

When processing does not meet the condition, after checking that receiving data is empty, interrupt flag is cleared and it checks that receiving data is empty again.

28.8.1.3 Timestamp

The timestamp counter is a free-running counter that can be used to check reception time of an incoming message or transmission time of successful transmitted messages. The Timestamp counter value is captured based on the CFDGFDCFG.TSCCFG[1:0] configuration (at the sample point of start of frame, point in time when the frame is valid, or for CANFD frames also at the sample point of the RES bit). For reception, it is stored together with the message ID and data into the target RX message buffer or RX FIFO.

For transmit message, the timestamp counter value is stored as part of the TX History List entry.

The counter can be clocked with the peripheral clock or with the CAN channel bit timing clock. The counter source clock can be configured with the CFDGCFG.TSSS bit of the Global Configuration Register. If this bit is 0, the peripheral clock is used. If the bit is 1, the selected CAN channel bit time clock is used.

The channel selection is performed with the CFDGCFG.TSBTCS bit of the Global Configuration Register.

Care must be taken when using selected CAN channel bit time clock as the clock source. When entering Channel Halt mode or Channel Reset node, for this channel, the timestamp counter is stopped. For other CAN channels, the timestamp counter value is not updated.

If peripheral clock is selected as the timestamp counter clock source, Channel modes do not affect the timestamp counter function.

The source clock for the timestamp counter can be divided by a factor defined by the CFDGCFG.TSP bits (timestamp prescaler) in the Global Configuration Register.

The timestamp counter can be reset to 0x0000 with the CFDGCTR.TSRST bit (timestamp reset).

28.8.2 Transmission

There are several possible transmission configurations:

- Normal transmission
- FIFO transmission
- TX Queue transmission
  A fixed number of transmission message buffers (4 TX message buffers) are dedicated. These message buffers are only used for transmission and cannot be configured for reception.
  Additionally transmission from TX Queue or Common FIFO in TX mode can be configured in the following way (see Figure 28.41):
  – TX Queue: Up to four transmission message buffers can be grouped to form a TX Queue with a common access window.
    Upper transmission message buffers are used to form the TXQ.
    TXQ has an access window.
    • TXQ is transmission Message Buffer 0.
- Common FIFO (TX mode): Common FIFO in TX mode is linked to a dedicated channel.
  Channel has a fixed number of one Common FIFO assigned to it. Within the channel, a Common FIFO configured in TX mode, can be freely linked (assigned) between 0 and 3 transmission message buffers (only one FIFO to one transmission message buffer).
  The Common FIFO buffer then replaces the transmission message buffer linked to it.
  Transmission Control and Status registers of these transmission message buffers should not be used.

See Figure 28.29 for information about Common FIFO buffer assignment to related channel.

Note: Common FIFO buffer should not be linked to TX message buffers that are already part of a TX Queue.

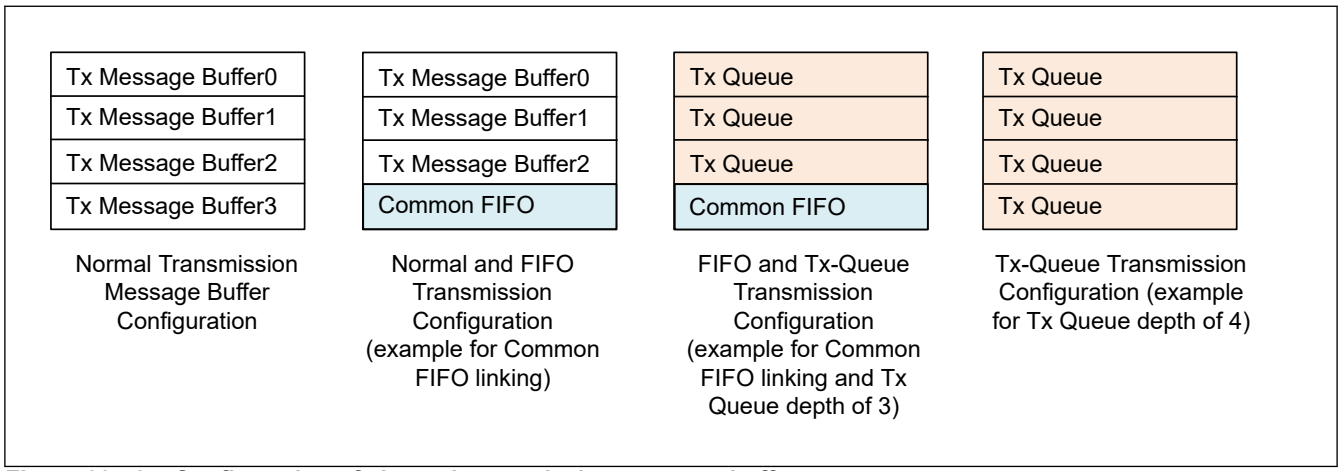


Figure 28.41 Configuration of channel transmission message buffer

28.8.2.1 Transmission Priority

If two or more transmission message buffers of a channel are configured for transmission, then the transmission priority in the CANFD module can be selected from the following two modes:

- CAN ID priority
- Message buffer number priority.

The transmission priority mode is common for all message buffers. It can be configured with the CFDGCFG.TPRI bit in the Global Configuration Register.

For message buffer number priority transmission, the smallest message buffer number with transmission request has the highest priority for transmission. This also includes the TX message buffers linked to the Common FIFO buffer configured in TX mode.

However, message buffer number priority should not be used if TX Queue is enabled.

For CAN ID priority transmission, ID priority complies with the CAN bus arbitration rule (as specified in ISO 11898-1 specification). All TX message buffers can enter the ID priority comparison for message buffers configured for transmission. This also includes the TX message buffers linked to the Common FIFO buffer configured in TX mode and includes the TX Queue message buffers.

If the ID of two or more message buffers is the same, then the smaller message buffer number has higher priority for transmission.

Note: For Common FIFO buffer configured in TX mode, only the message currently being pointed to by the FIFO read pointer can be included in the transmission arbitration.

If the message is being transmitted from the FIFO, then the next pending message within the same FIFO is considered in the transmission arbitration.

In contrast to this, all transmission message buffers of a TX Queue participate in internal transmission arbitration.

Figure 28.42 shows the transmission configuration flow.

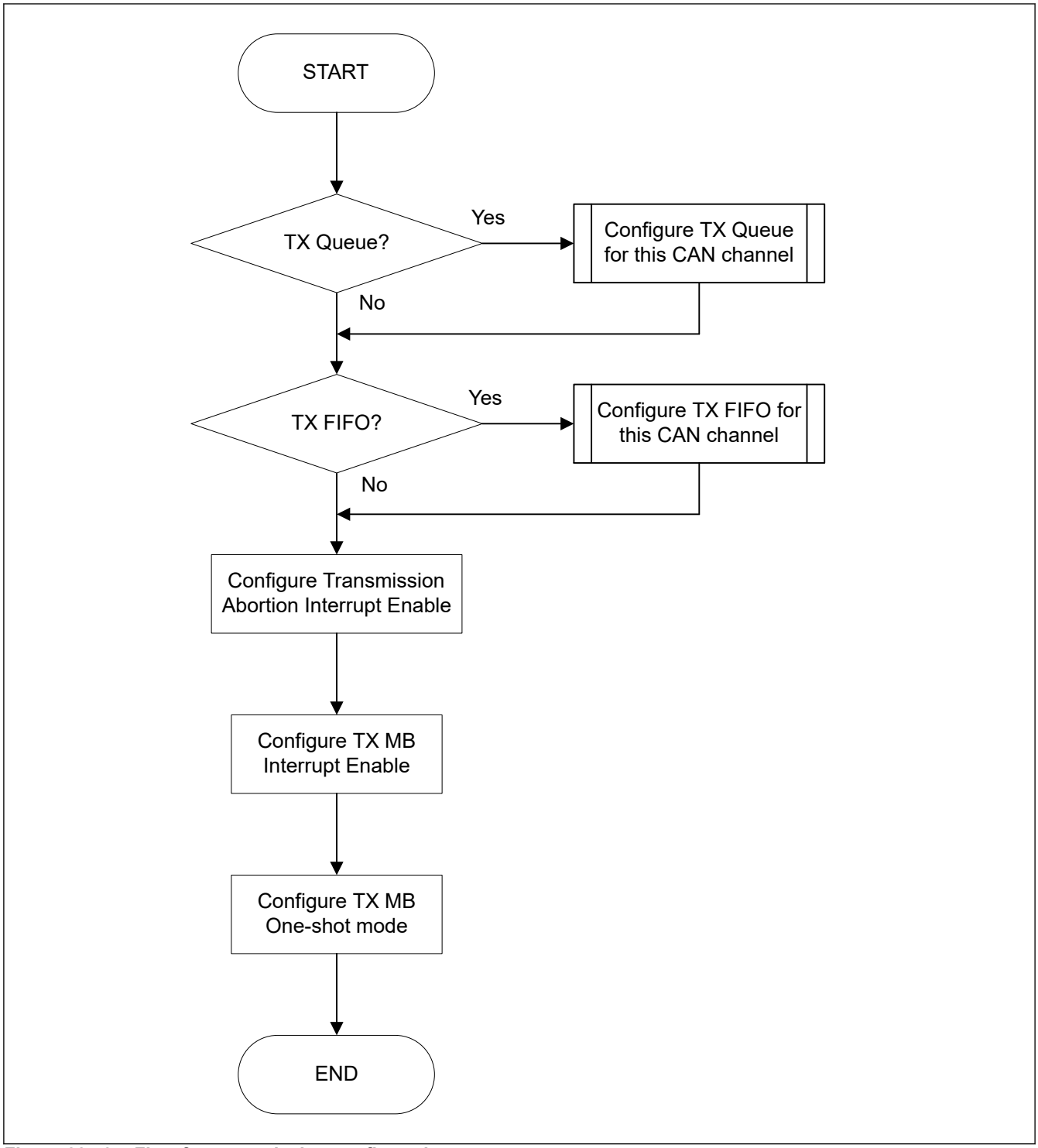


Figure 28.42 Flow for transmission configuration

28.8.2.2 Normal Transmission

Each transmission message buffer has two modes of message transmission:

1. Regular transmission mode
   If the message buffer is placed in regular transmission mode, the data frame or remote frame set in that message buffer can be transmitted.
   Completion of regular transmission can be checked through the related TX Message Buffer Transmission Result flag bits (CFDTMSTSj.TMTRF) in the TX Message Buffer Status Registers. These bits are set to 10b or 11b when the regular transmission is successful.

When arbitration is lost or an error occurs, message transmission is further attempted if no transmission abort request is set for this transmission message buffer.
New internal transmission arbitration for this channel is performed for all message buffers with transmission request.

2. One-shot transmission mode

When the CFDTMCi.TMOM bit of the TX Message Buffer Control Registers is set for a transmission message buffer, the message buffer is placed in One-shot transmission mode and attempts to transmit a message only once.

Completion of One shot transmission can be checked through the related TX Message Buffer Transmission Result Flag bits (CFDTMSTSj.TMTRF) in the TX Message Buffer Status Registers. The CFDTMSTSj.TMTRF bits are set to 10b or 11b when One-shot transmission is successful.

The CFDTMSTSj.TMTRF bits are set to 01b when arbitration is lost or an error occurs during transmission of the related message buffer.

Additional message transmission is not attempted in this case.

The regular transmission request procedure after a configuration is shown in Figure 28.43.

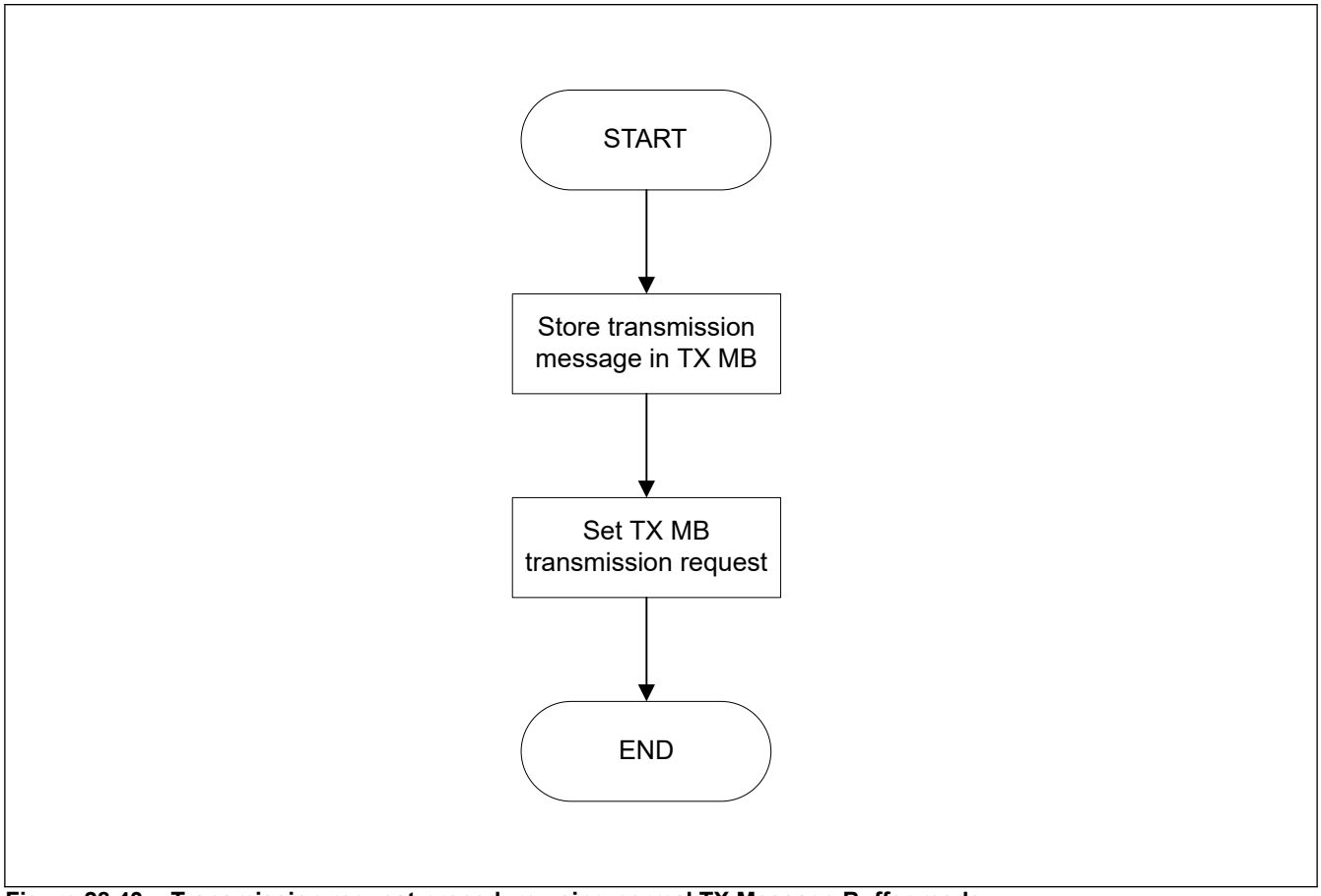


Figure 28.43 Transmission request procedure using normal TX Message Buffer mode

(1) Setting for TX Message Buffer Control Register

Table 28.27 shows configuration of a normal CAN transmission mode.

Table 28.27 Configuration of CAN transmission mode (1 of 2)

Transmission request CFDTMCi.TMTR	Transmission abort request CFDTMCi.TMTAR	One-shot enable CFDTMCi.TMOM	Communication activity
0	0	0	Message buffer disabled
0	0	1	Message buffer disabled

Table 28.27 Configuration of CAN transmission mode (2 of 2)

Transmission request CFDTMCi.TMTR	Transmission abort request CFDTMCi.TMTAR	One-shot enable CFDTMCi.TMOM	Communication activity
1	0	0	Configured as a transmission message buffer for a data frame or a remote frame
1	0	1	Configured as a one-shot transmission message buffer for a data frame or a remote frame
1	1	0	Transmission abort requested
1	1	1	One-shot transmission abort requested

The configuration bits can be configured in the TX Message Buffer Control Registers.

Figure 28.44 shows timings for successful transmission for two message buffers.

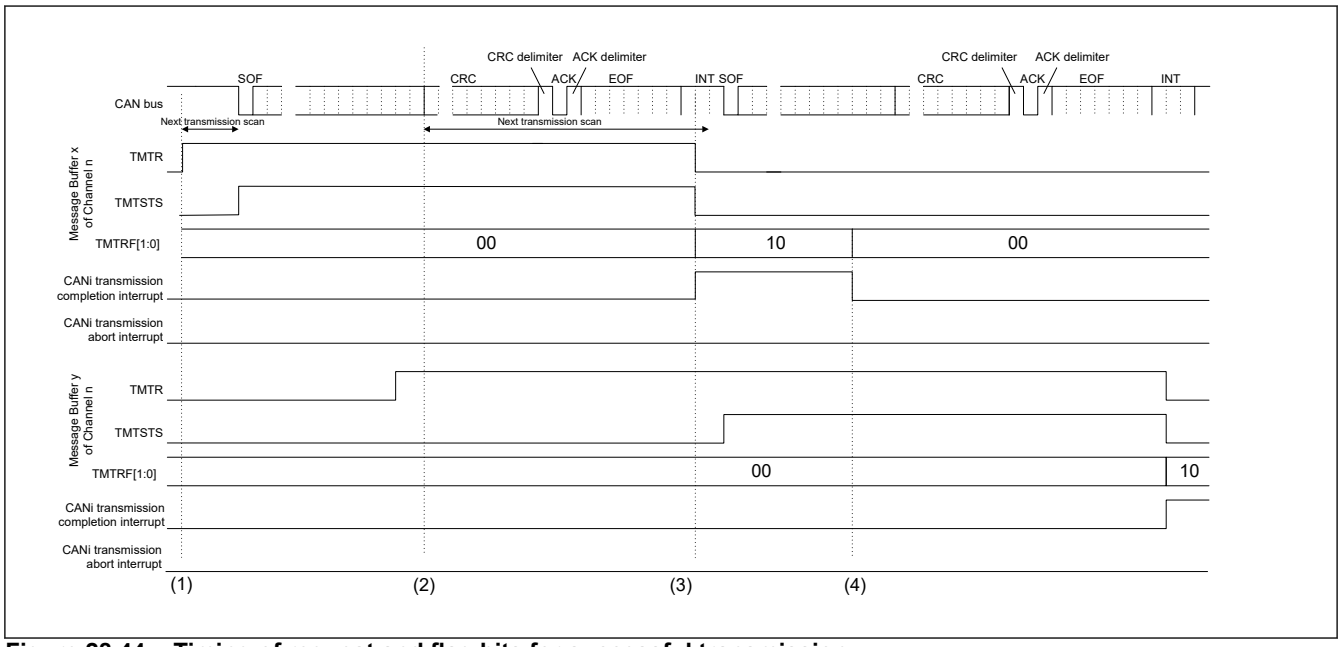


Figure 28.44 Timing of request and flag bits for successful transmission

1. If the CFDTMCi.TMTR bit in the TX Message Buffer Control Registers is set in the bus idle state, the message buffer scanning procedure determines the highest priority message buffer for transmission.
   When the transmission message buffer is determined, the CFDTMSTSj.TMTSTS bit in the related TX Message Buffer Status Registers is set (transmitting/transmitter), and CAN channel starts the transmission *1.
2. At the first bit of CRC, the transmission scanning procedure starts for the next possible transmission when pending transmission requests exist.
3. If the message has been successfully transmitted, the CFDTMSTSj.TMTRF[1:0] bits in the corresponding TX Message Buffer Status Registers are set to 10b and CFDTMSTSj.TMTSTS and the CFDTMCi.TMTR bits are cleared.
   When the TMIE bit in the TX Message Buffer Interrupt Enable Configuration Registers is set (interrupt enabled), the CAN successful transmission interrupt request is generated.
   To clear the related interrupt line, clear the CFDTMSTSj.TMTRF flag bits.
4. Before starting the next transmission, clear the CFDTMSTSj.TMTRF bits. Load the next message in the transmission message buffer and set the CFDTMCi.TMTR bit again.
   CFDTMCi.TMTR bit cannot be set again before CFDTMSTSj.TMTRF[1:0] bits are cleared.

Note 1. If arbitration is lost after the CAN channel starts the transmission, the CFDTMSTSj.TMTSTS bit is cleared.

The transmission scanning procedure is performed again to search for the highest priority transmission message buffer from the beginning of the first CRC bit.

If an error occurs either during transmission or following the loss of arbitration, then during the error frame, the transmission scanning procedure is performed again to search for the highest priority transmission message buffer.

Note: The setting point of CFDTMSTSj.TMTSTS is not always fixed at the start of the SOF. It may be delayed up to the start of the standard ID due to the synchronization logic implemented for the PLL bypass.

Figure 28.45 shows timings for transmission abort for two message buffers.

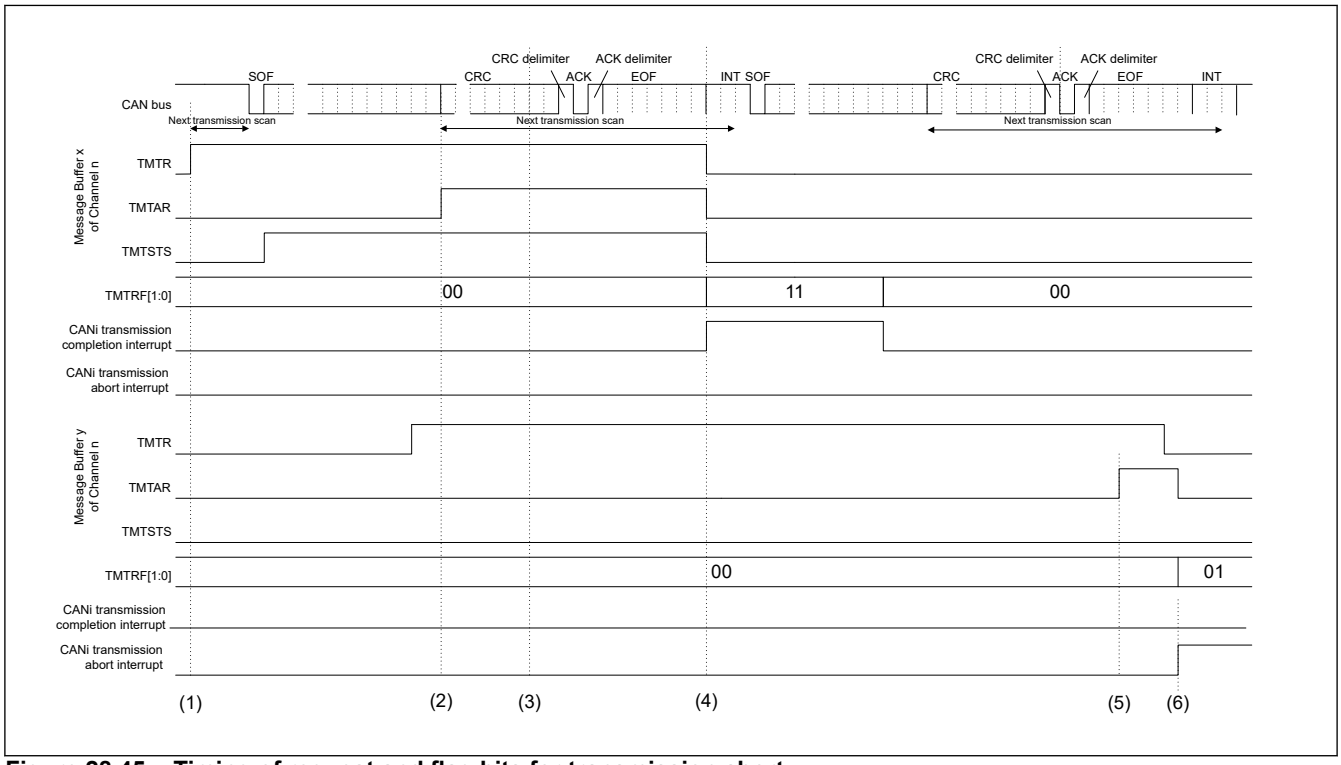


Figure 28.45 Timing of request and flag bits for transmission abort

1. If the CFDTMCi.TMTR bit in the TX Message Buffer Control Registers is set in the bus idle state, the message buffer scanning procedure determines the highest priority message buffer for transmission.
   When the transmission message buffer is determined, the CFDTMSTSj.TMTSTS bit in the TX Message Buffer Status Registers is set (transmitting/transmitter), and CAN channel starts the transmission*1.
2. If the CFDTMCi.TMTAR bit is set when the related message buffer is already selected for transmission or currently transmitting, the message is not aborted, if no error occurs or arbitration is lost.
3. At the first CRC bit, the transmission scanning procedure starts for the next transmission. In this example, timing chart message buffer y is not selected as the next transmission message buffer.
4. If the message has been successfully transmitted, the CFDTMSTSj.TMTRF[1:0] bits in the corresponding TX Message Buffer Status Registers are set to 11b and the CFDTMSTSj.TMTSTS and CFDTMCi.TMTR bits are cleared.
   When the TMIE bit in the TX Message Buffer Interrupt Enable Configuration Registers is set (interrupt enabled), the CAN successful transmission interrupt request is generated.
   To clear the related interrupt line, clear the CFDTMSTSj.TMTRF[1:0] bits.
5. Another CAN node is transmitting on the CAN bus (CFDTMSTSj.TMTSTS is not set). If the CFDTMCi.TMTAR bit is set when the related channel is under transmission scan, the transmission request cannot be cleared.
6. After internal processing time, the transmission is aborted and the CFDTMSTSj.TMTRF[1:0] bits are set to 01b.
   If the message buffer is not transmitting or selected as the next transmission message buffer or under transmit scan, then the abort is immediately accepted and the corresponding CFDTMSTSj.TMTRF[1:0] bits in the TX Message Buffer Status Registers are set to 01b.
   In addition, CFDTMCi.TMTR, and CFDTMCi.TMTAR bits are cleared automatically.
   When the transmission abort interrupt enable TAIE bit of the related Channel Control Register is set then an interrupt is generated for successful transmission abort.
   To clear the related interrupt line the CFDTMSTSj.TMTRF[1:0] bits have to be cleared.

Note 1. If arbitration is lost after the CAN channel starts the transmission, the CFDTMSTSj.TMTSTS bit is cleared.

The transmission scanning procedure is performed again to search for the highest priority transmission message buffer from the beginning of the first CRC bit.

If an error occurs, either during transmission, or following the loss of arbitration, then during the error frame, the transmission scanning procedure is performed again to search for the highest priority transmission message buffer.

28.8.2.3 TX FIFO Transmission

One common FIFO buffer is assigned to CANFD module. The FIFO buffer can be linked to any normal TX message buffer position for this channel with the CFDCFCC.CFTML bits in the Common FIFO Configuration/Control Register if configured in TX mode.

When the transmission scan starts and the FIFO buffer corresponding to this TX message buffer is enabled, the relevant message in the FIFO buffer participates in the transmission scan.

Configuration of a TX message buffer linked to a FIFO buffer configured in TX mode should not be done.

(1) TX FIFO Operation

CAN messages can be written into the TX FIFO by writing to the corresponding FIFO Access registers.

When the value 0xFF is written into the corresponding FIFO Pointer Control Register, the message count of the related FIFO is incremented by 1.

Only write to the FIFO Pointer Control register after writing the complete message to the corresponding FIFO Access registers. If the message count matches the FIFO depth, the FIFO Full flag is set.

The oldest message in the TX FIFO is included in the scan for transmission by the corresponding CANFD module channel logic.

When a message is successfully transmitted from the TX FIFO, the message count value is decremented by 1. When all the messages from the FIFO are transmitted, the FIFO Empty flag is set.

The interrupt generation conditions for the TX FIFO buffer can be configured by configuring the CFDCFCC.CFIM bit in the corresponding Common FIFO Configuration/Control Register.

If CFDCFCC.CFIM bit is 0, then interrupt is generated when the last message is successfully transmitted from the TX FIFO buffer.

If CFDCFCC.CFIM bit is 1, then interrupt is generated for every successfully transmitted message from the TX FIFO buffer.

The Common FIFO can set interrupt when CAN frame transmission is complete.

The Common FIFO buffer configured in TX Mode can be disabled by clearing the CFDCFCC.CFE bit in the Common FIFO Configuration/Control Register. If this bit is cleared to 0, the FIFO Empty flag is set as follows:

- Immediately if the message from the TX FIFO is neither scheduled for the next transmission nor in transmission
- Following the transmission completion, the detection of an error on the CAN bus, loss of arbitration or transition to Channel or Global Halt mode if the transmission from the TX FIFO is already scheduled for transmission or already in transmission.

Note: The Common FIFO buffer is considered as disabled after clearing the CFDCFCC.CFE bit only when the Empty flag is set for the corresponding Common FIFO buffer.

Other possible messages pending from the TX FIFO are lost and their transmission must be requested again. Before CFDCFCC.CFE is set again, ensure that CFDCFSTS.CFEMP bit is set and that there are no pending abort from the TX FIFO.

When the CFDCFCC.CFE bit is cleared, the message read and write pointers of the FIFO are cleared and are no longer active. Therefore, all messages in the FIFO buffers are lost and no further message can be stored into the FIFO.

The FIFO transmission request procedure after configuration is shown in Figure 28.46.

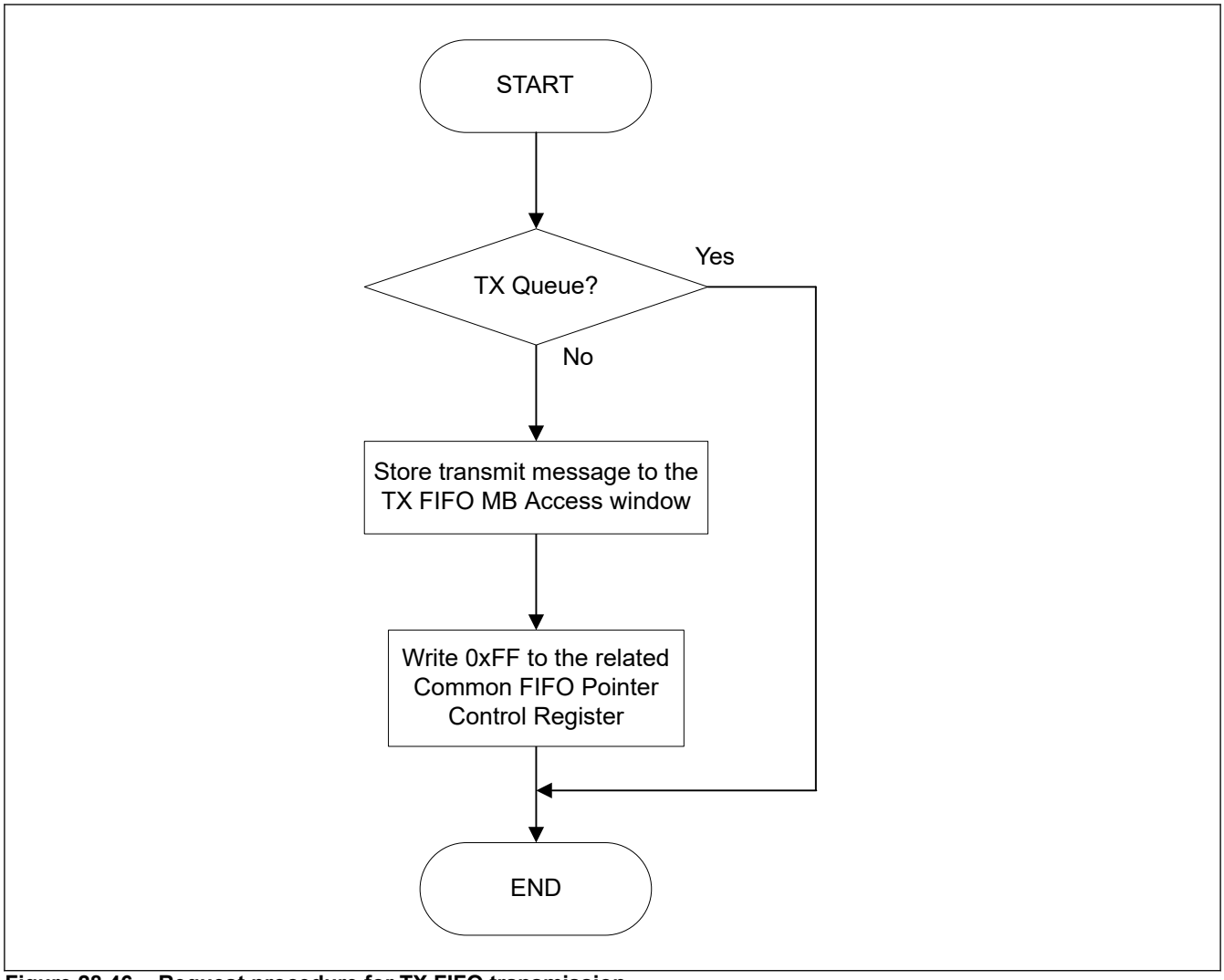


Figure 28.46 Request procedure for TX FIFO transmission

(2) Interval Timer for FIFO Transmission

For each Common FIFO in TX mode, it is possible to specify a delay between two consecutive messages that are configured for transmission from the same FIFO buffer. This delay is called interval time. This interval time starts after the first message has been successfully transmitted from the FIFO buffer after the CFDCFCC.CFE bit is set.

When the Common FIFO in TX mode is enabled, the first message is transmitted without considering this interval time.

The interval timer stops counting when:

- FIFO is disabled by clearing the CFDCFCC.CFE bit.
- CAN channel is in CH_RESET mode.

The interval time is specified by the CFDCFCC.CFITT value from 0 to 255 timer units in the Common FIFO Configuration/ Control Register.

The timer unit can be defined based on two different source clocks for the interval timer. To disable the interval timer for FIFO transmission, select a value of 0.

The timer source can be selected with the configuration bit CFITSS in the Common FIFO Configuration/Control Register.

If CAN channel bit time clock is configured as the clock source, and the CAN channel enters CH_HALT, CH_RESET, or CH_SLEEP mode, the interval timer is stopped for that channel.

If peripheral clock is selected as the interval timer clock source, the interval timer is stopped only when the CAN channel is in CH_RESET or CH_SLEEP mode.

The reference clock can be used to configure the interval time in fixed time units. It is based on the peripheral clock. The reference clock prescaler value CFDGCFG.ITRCP in the Global Configuration Register defines the relation between the peripheral clock frequency/period and the reference clock period.

See Table 28.28 for CFDGCFG.ITRCP configuration values to achieve different reference clock periods based on the peripheral clock frequency and period.

Table 28.28 Configuration example for the reference clock of the FIFO interval timer

Reference clock/Peripheral clock	1 µs	100 µs	500 µs
16 MHz/62.5 ns	16	1600	8000
20 MHz/50 ns	20	2000	10000
32 MHz/31.25 ns	32	3200	16000
50 MHz/20 ns	50	5000	25000

The reference clock resolution can be specified by the interval timer reference clock resolution value CFDCFCC.CFITR in the Common FIFO Configuration/Control Register.

The interval time is based on the reference clock period multiplied by the configured value (x1 or x10). The reference clock based interval timer can be used to satisfy the requirements of the ISO 15765-2 Separation Time. The whole range for the separation time from 100 µs to 127 ms can be covered.

The specified interval time starts after successful transmission event (after EOF7 state of the CAN protocol).

When the interval time has elapsed, the next transmission request is raised by the related TX FIFO. Therefore, the interval time defines the minimum time between two messages transmitted from one FIFO.

The next message is sent at earliest after this interval time. Figure 28.47 shows an example timing of the internal processing.

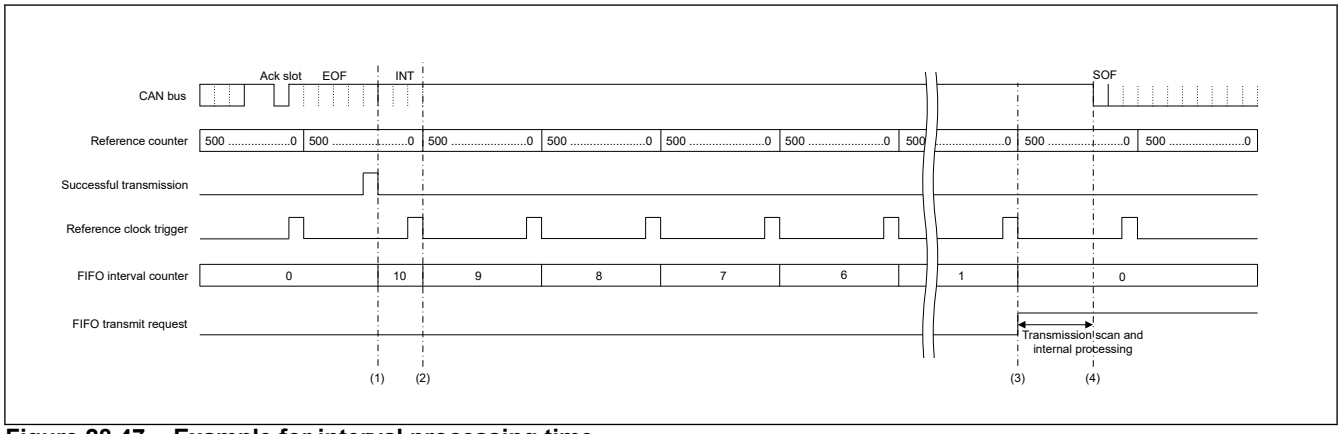


Figure 28.47 Example for interval processing time

The configuration for the timing in Figure 28.47 is as follows:

- Peripheral clock frequency = 50 MHz
- Interval timer reference clock (CFDGCFG.ITRCP) = 500 times
- Reference clock from the settings in Figure 28.47 = 10 µs
- Common FIFO interval timer source selection (CFDCFCC.CFITSS) = 0
- Common FIFO interval timer resolution (CFDCFCC.CFITR) = 0
- Common FIFO interval transmission time (CFDCFCC.CFITT) = 10 times
- Theoretical message separation interval = 100 µs

1. Internal FIFO interval timer is restarted with the occurrence of successful transmission result. This restart is not synchronized to the reference clock trigger. Therefore, the first interval is counting less or equal to 1 reference clock interval.
2. With the next reference clock trigger the FIFO interval timer is decremented.
3. When the FIFO interval timer reached the value 0, the FIFO transmit request is set.

4. When the FIFO is selected for transmission, the transmission starts. Due to internal processing, this usually takes less than 3 CAN bit time, between the internal FIFO transmit request set in step 3. and the actual transmission.

In the worst case when multiple events such as a reception scan, an internal message routing, a transmit scan on all channels occur, it can take up to 126 peripheral clock cycles.

As shown in Figure 28.47, it is not guaranteed that the minimum interval is always equal to the configured value. If a minimum time must never be breached, configure CFDCFCC.CFITT to the required minimum value plus 1.

If additional TX message buffers or TX FIFO are configured for transmission of the same channel, the real delay between two messages transmitted from a TX FIFO can be much longer than specified by the interval time. This is due to higher priority message transmission from these TX message buffers or TX FIFO.

Figure 28.48 shows a block diagram of the FIFO interval time generation circuit.

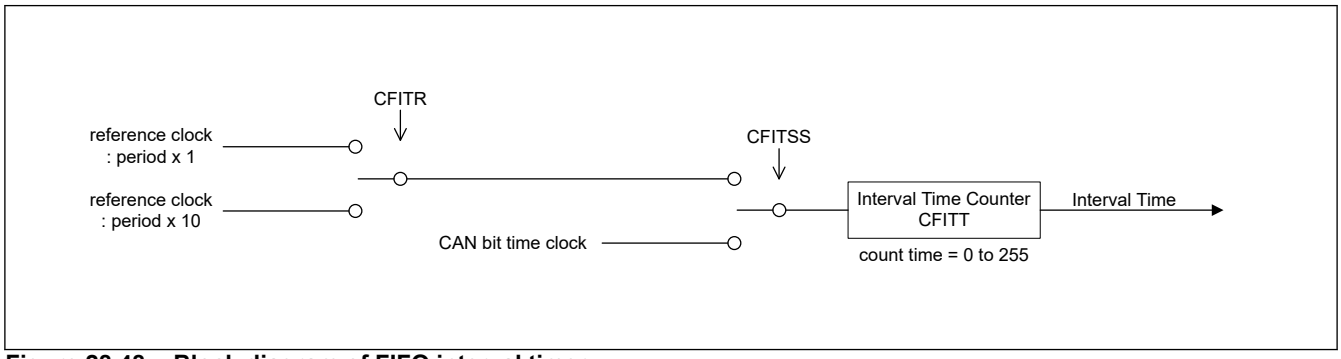


Figure 28.48 Block diagram of FIFO interval timer

28.8.2.4 TX Queue

Each enabled TX Queue for a specific channel consists of 3 to 4 TX message buffers, which are accessed through one access window.

- The first TX Queue can be configured with a depth of three up to four buffers and uses TX Message Buffer No. 0 as access window (referred to as TXQ)

All the TXQ messages enter the priority comparison for the transmission, which should be only ID Priority (CFDGCFG.TPRI = 0).

The registers for TXQ are:

- CFDTXQCC
- CFDTXQSTS
- CFDTXQPCTR

See related access registers TX Message Buffer ID Registers (TMID[m]), TX Message Buffer Pointer Registers (TMPTR[m]), TX Message Buffer Data Field 0 Registers, and TX Message Buffer Data Field 1 Registers (TMDF[0:1][m]) when access window TXQ0 is used.

The depth of each TXQ buffer can be configured by writing to the CFDTXQCC.TXQDC[1:0] bits of the TX Queue Configuration/Control Register. TXQ can be set from TXMB0 to TXMB3 as a queue buffer at the maximum.

The 4 available options for the depth configuration of TXQ buffer are:

- 0x00: TX Queue disabled
- 0x01: reserved
- 0x10: 3 Messages
- 0x11: 4 Messages

Do not access all the TX message buffers forming the TX Queue directly (except TX Message Buffer No. 0, which act as TX Queue access window).

When a system writes in TXQ, it writes in send data, after checking the state of TXQ.

Do not access or configure the related TX Message Buffer Control Registers.

The messages stored to the TX Queue access window are internally stored to a free buffer of the TX Queue.

When the buffer is full, no further access should be done to the queue, until it is no longer full. If access is a software write when the buffer of TXQ is full, send data is overwritten.

The TX Queue can be disabled by clearing the TXQE bit in the TX Queue Configuration/Control Register. If this bit is cleared, the TX Queue Empty flag is set as follows:

- Immediately if the message from the TX Queue is neither scheduled for the next transmission nor in transmission
- Following the transmission completion, the detection of an error on the CAN bus, loss of arbitration or transition to Channel or Global Halt mode if the transmission from the TX Queue is already scheduled for transmission or already in transmission.

Note: The TX Queue is disabled only when the Empty flag is set after clearing the TXQE bit for the corresponding TX Queue.

Other possible messages pending from the TX Queue are lost and their transmission must be requested again.

Before TXQE is set again, ensure that the CFDTXQSTS.TXQEMP bit is set and that there is no pending abort from the TX Queue.

When the TXQE bit is cleared, all messages in the TX Queue buffers are lost and no further message should be stored in the TX Queue.

When a message has been stored to the TX Queue, write 0xFF in the TX Queue Pointer Control Register. This sets the transmit request automatically and changes the internal message buffer pointer to the next free message buffer location of the TX Queue.

Note: If two messages with the same ID are stored in the TX Queue, the order of transmission of these messages can be different from the order in which they were stored in the TX Queue.

To avoid this condition, it is important to confirm that the previous message with the same ID was successfully transmitted before a new message with the same ID is stored in the TX Queue.

For the TX Queue, a dedicated interrupt can be enabled by setting the TXQIE bit of the TX Queue Configuration/Control Register.

The interrupt mode can be configured with the CFDTXQCC.TXQIM bit of the same register either to generate an interrupt for every transmitted message or for the last transmitted message.

The TX Queue transmission request procedure after configuration is shown in Figure 28.49.

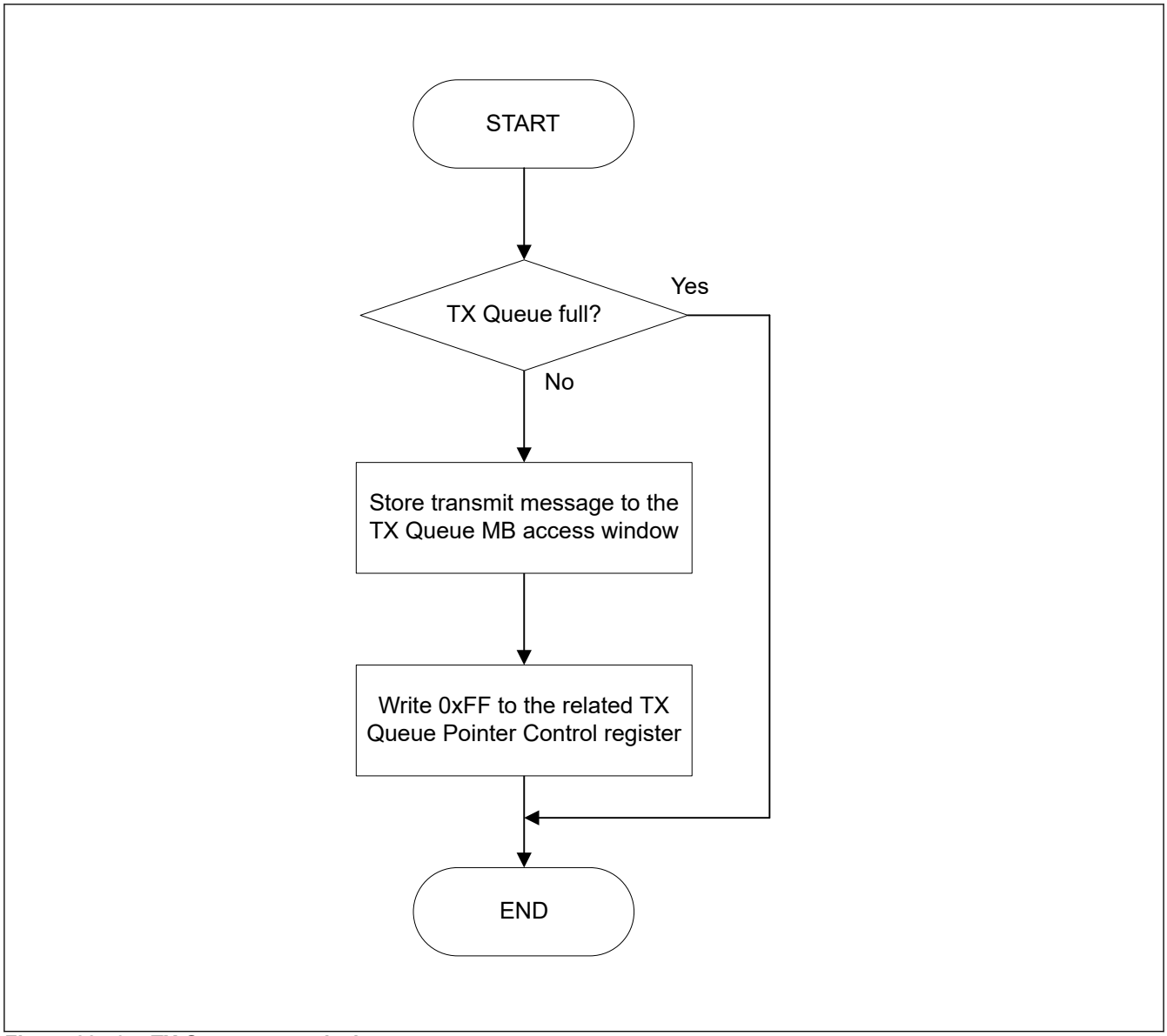


Figure 28.49 TX Queue transmission request

28.8.2.5 TX History List

The TX History List function records the information of the successfully transmitted message in the TX History List Buffers. Two TX History List buffers are provided and THL buffer can store up to 8 TX History List entries.

The CFDTHLCC.THLDTE bit of the TX History List Configuration/Control Register can be used to configure if only message information from TX FIFO or TX Queue is stored, or if all transmit message information from TX Queue, TX FIFO, or normal TX message buffers is stored in the TX History List.

Each transmit message can be individually configured for acceptance to the TX History List with the CFDCFID.THLEN bit in the Message Buffer Pointer Register.

The message information is stored to the TX History List Buffer of a CAN channel after the message is successfully transmitted.

Storing to the list is not synchronized with the status of CFDTMSTSj.TMTRF[1:0] bits in the TX Message Buffer Status Register.

Due to internal processing, the storage to the list can happen with a delay after the successful transmission indication.

Storing the TX History List data can be recognized by the condition that the THLIF is set to 1 when the THLIE bit is configured to 1 or when the TX History List counter CFDTHLSTS.THLMC[5:0] is increased.

In worst case when multi events like reception scan, internal message routing on happen.

- Maximum delay time from setting the CFDTMSTSj.TMTRF to store the TX History List data is 76 peripheral bus clock cycles.

The History list records the following information of a transmitted message:

- Buffer type:
  - 001: TX Message Buffer
  - 010: TX FIFO
  - 100: TX Queue
- Buffer number:
  TX message buffer, TX Queue message buffer or TX message buffer link for the Common FIFO buffer from which transmission occurred. The number depends on the buffer type. See Table 28.29.
- Transmission ID:
  Transmission pointer stored in the transmission message
- Transmit timestamp:
  Message timestamp captured at capture point as configured by CFDGFDCFG.TSCCFG.
- Transmission information label:
  Transmission information label stored in the transmission message.

Table 28.29 TX History List Buffer number entry

Buffer Number	BT[2:0] Buffer Type		
	001b TX Message Buffer	101b TX FIFO	100b TX Queue
00b	Message Buffer 0	Number shown corresponds to the common FIFO. TX Message Buffer Link CFTML of the related Common FIFO configuration	Number shown corresponds to the Message Buffer belonging to the TX Queue which the frame was transmitted
01b	Message Buffer 1		
10b	Message Buffer 2		
11b	Message Buffer 3		

The Transmission ID entry is used to identify which message of a TX FIFO or TX Queue has been successfully transmitted because the TX FIFO or TX Queue number alone is not sufficient.

Therefore, a unique number can be attached to each transmission message stored in a TX FIFO or TX Queue. This unique identification number should be written to the CFDCFFDCSTS.CFPTR[15:0] part of the Common FIFO Access Pointer Register for a TX FIFO or to the CFDTMFDCTRb.TMPTR[15:0] part of the TX Message Buffer Pointer Register of the TX Queue access window message buffer.

When the message is successfully transmitted, this identification number is stored together with the other message related information to the TX History List and can be read using the Transmission ID (TID) of the TX History List Access Register.

Also, for normal TX message buffers, the CFDTMFDCTRb.TMPTR[15:0] part of the TX Message Buffer Pointer Register is stored in the Transmission History List and the information label is the same.

Figure 28.50 shows a transmission preparation flow when TX History List is used.

Read access to the TX History List Access Register is done for every single entry.

After reading one entry, 0xFF must be written to the corresponding TX History List Pointer Control Register to be able to access the next entry until TX History List is empty.

Figure 28.51 shows an example flow for processing the TX History List information.

The TX History Lists have dedicated interrupts, which can be configured with the CFDTHLCC.THLIM bit of the corresponding TX History List Configuration/Control Register and enabled with the CFDTHLCC.THLIE bit of the same

registers, either to generate an interrupt when the History List reached a filling level of 75% or for every new TX History List entry.

An entry lost indication is flagged by the CFDTHLSTS.THLELT bit in the TX History List Status Register. The status of this bit is also shown by the THLES bit in the Global Error Flag Register.

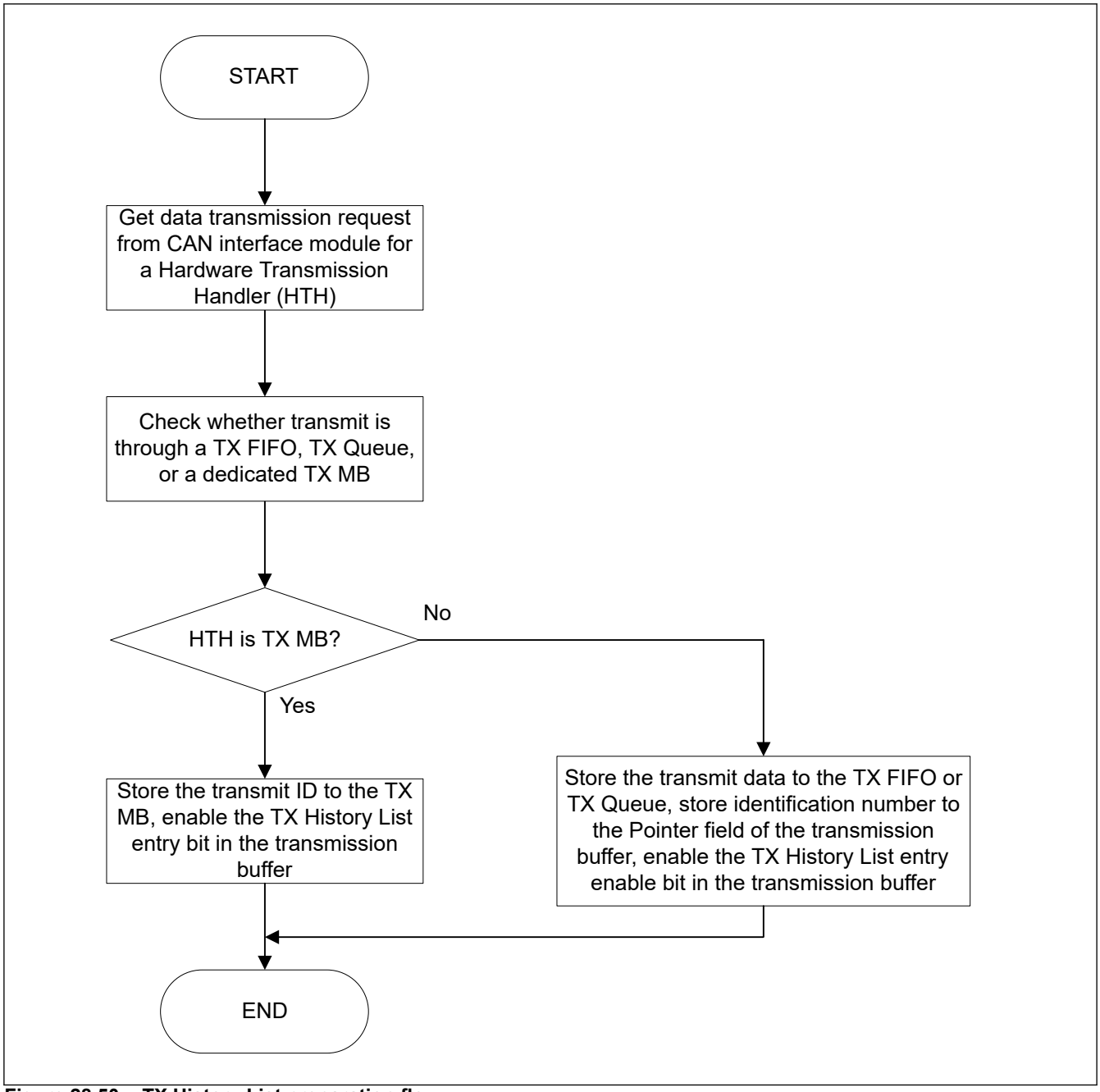


Figure 28.50 TX History List preparation flow

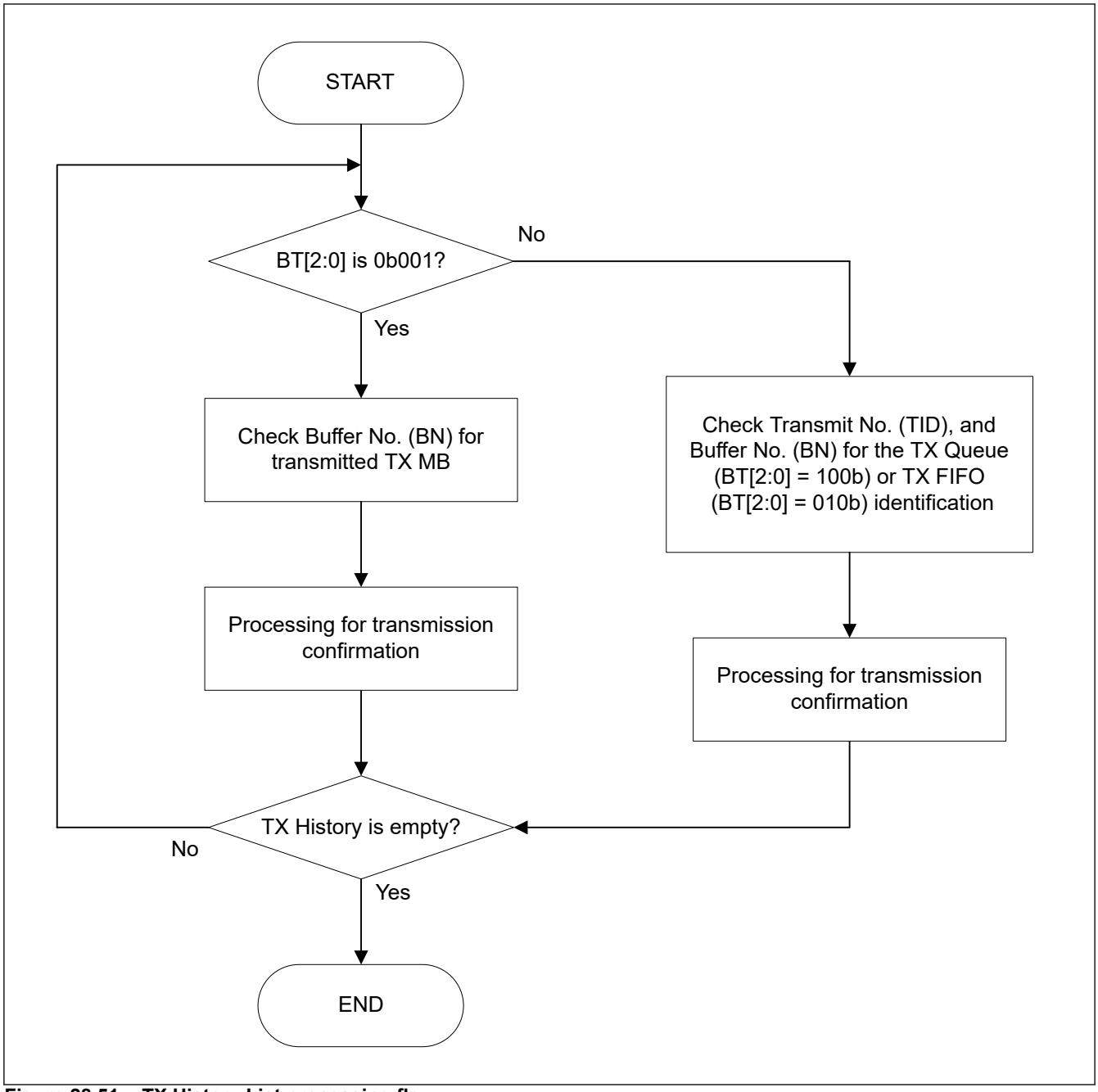


Figure 28.51 TX History List processing flow

28.8.2.6 TX Data Padding

This chapter is not valid for classical CAN.

If the data length code (DLC) of the transmitting message has a higher number of data bytes than the buffer size, the data bytes beyond the restricted range are replaced by bytes with the value of 0xCC.

This can happen for Common FIFO configured as (TX mode) when the transmit message DLC is higher than the CFDCFCC.CFPLS.

This can also happen in FD only mode, if a Classical frame is configured with a DLC bigger than 8.

28.9 Test Mode

The CANFD module can be configured into test modes to allow testing of certain features. These features are provided only for special purposes and care must be taken when configuring the CANFD module in test modes.

Note: All test modes are mutually exclusive unless it is explicitly stated that some functions can be enabled across other test modes.

Do not enable any combination of the various test modes specified in this section.

The test modes can be broadly split into 2 groups:

- Channel specific test modes
- Global test modes.

28.9.1 Channel Specific Test Modes

CAN channel can be configured into the following test modes:

- Basic test mode
- Listen-only mode
- Self-test mode 0 (External loop back mode)
- Self-test mode 1 (Internal loop back mode)
- Restricted operation mode.

28.9.1.1 Basic Test Mode

The basic test mode should be used when there is requirement for a particular test setting to be enabled other than when in Listen-only and Self-test modes.

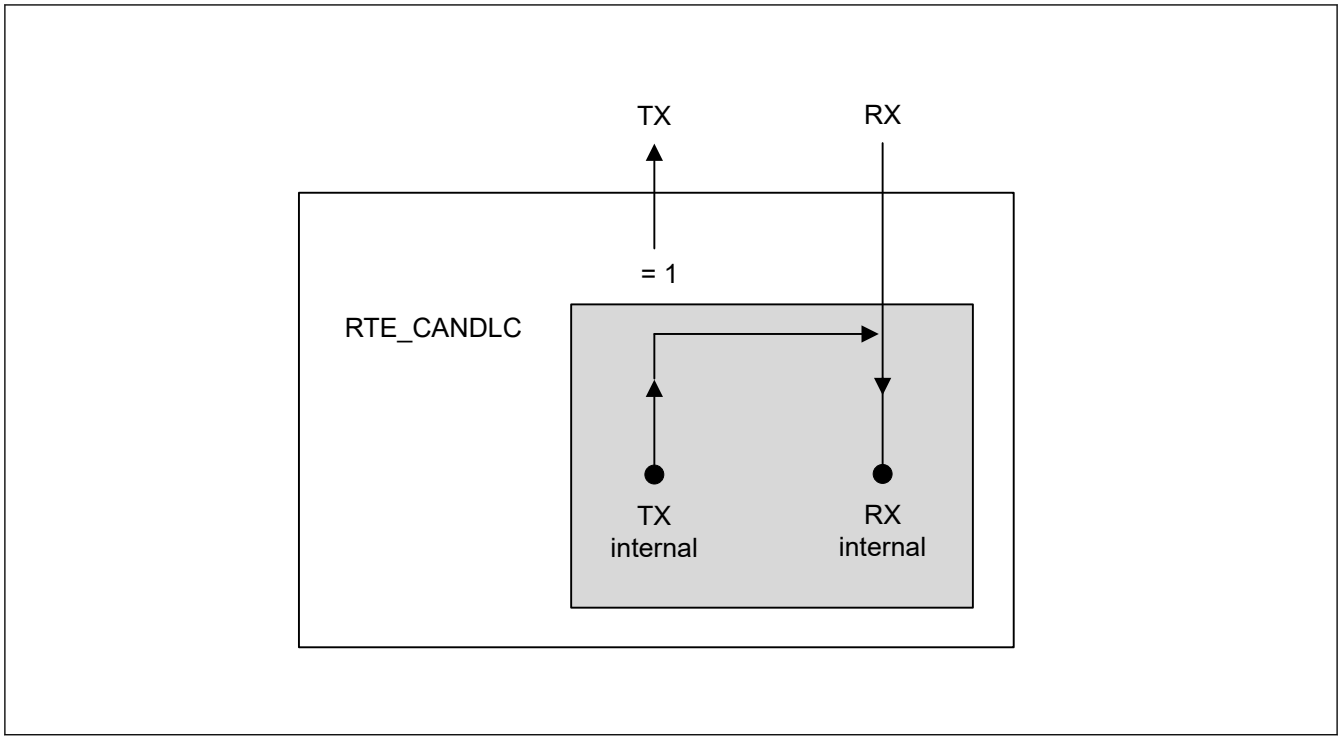
28.9.1.2 Listen-only Mode

The ISO 11898-1 recommends an optional bus-monitoring mode. In this mode, the CAN channel is able to receive valid data frames and valid remote frames. However, it sends only recessive bits on the CAN bus and is not allowed to transmit.

If the CAN engine is required to send a dominant bit (ACK bit, overload flag, active error flag), the bit is routed internally so that the CAN engine monitors this as dominant. The external TX pin remains in recessive state.

This mode can be used for baud rate detection. In this mode, an error interrupt is generated if a bus error occurs and the interrupt is enabled.

In this mode, it is not permitted to request transmission from any normal TX message buffer or TX FIFO.

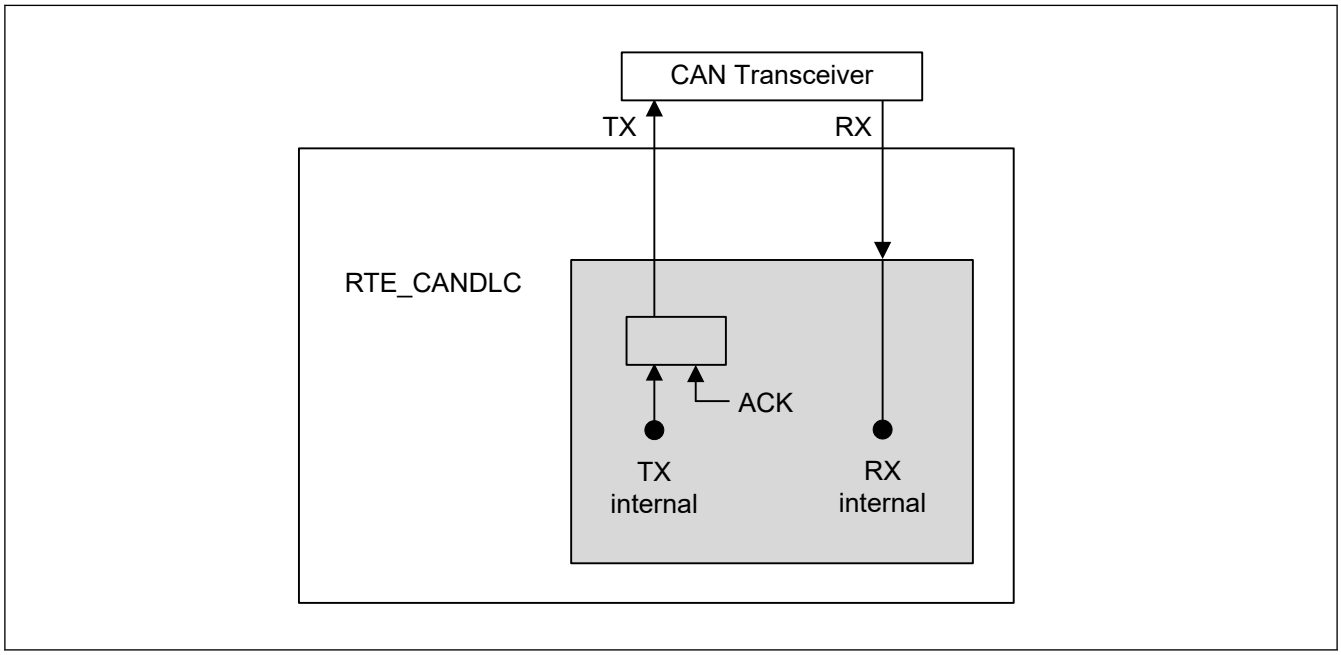


28.9.1.3 Self-test Mode 0 (External loopback mode)

In Self-test mode 0, the CAN engine treats its own transmitted messages as received messages through the CAN transceiver and stores them into its receive message buffers.

To be independent from external stimulation, the engine generates its own Acknowledge bit.

This test can be used for CAN transceiver tests and the RX/TX pins should be connected to the transceiver.

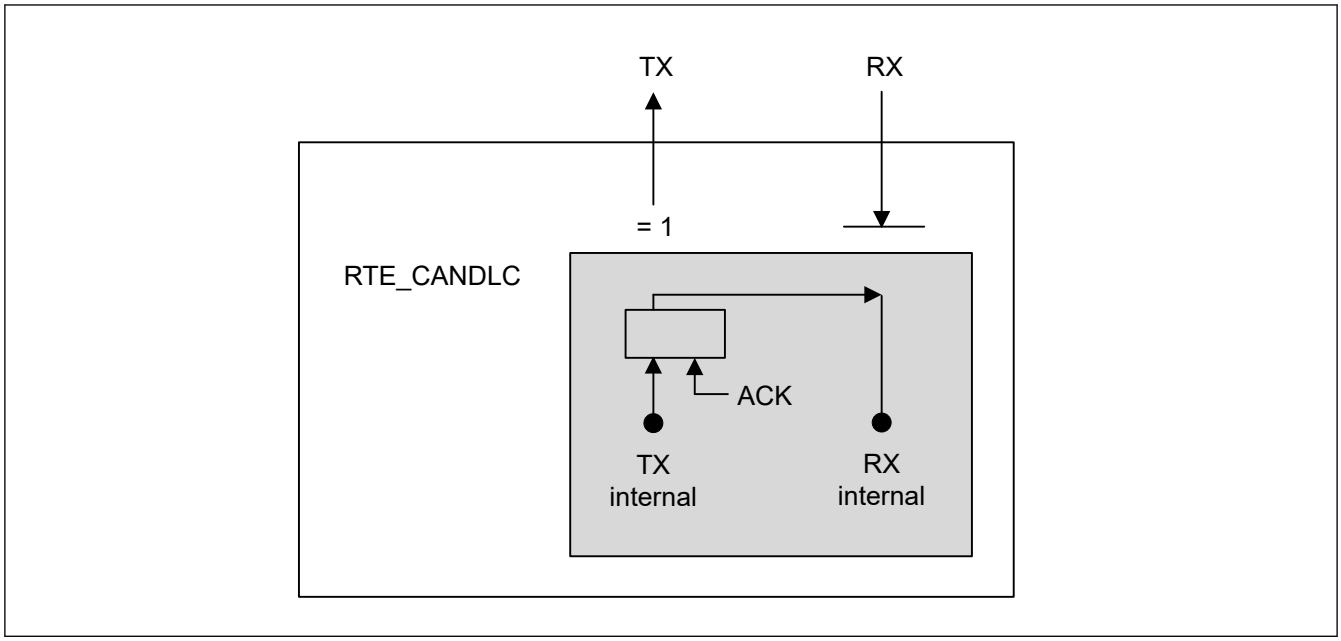


28.9.1.4 Self-test Mode 1 (Internal loopback mode)

In Self-test mode 1, the CAN engine treats its own transmitted messages as received messages and stores them into the receive buffer. This mode is provided for self-test functions. To be independent from external stimulation, the CAN engine generates its own Acknowledge bit. In this mode the CAN engine performs an internal feedback from TX internal to RX internal. The actual value of the external RX input is disregarded by the CAN engine.

The external TX pin outputs only recessive bits. The RX/TX pins do not need to be connected to the CAN bus or any external device.

Note: The channel pins are also disconnected from the internal CAN bus communication line.



28.9.1.5 Restricted Operation Mode

This chapter is not valid for classical CAN.

In Restricted operation mode, the CAN node is able to receive valid data and remote frames generating the Acknowledge bit.

Active error or overload frames cannot be transmitted, instead it waits for the occurrence of bus idle condition to resynchronize itself to the CAN communication after an error or overload condition occurs.

Additionally, the Receive and Transmit Error Counter (REC and TEC) are frozen independently from the occurrence of errors. The mode is specified in ISO 11898-1 and the setting of transmit request is permitted.

28.9.2 Global Test Modes

The CANFD module can be configured into the following test modes:

- RAM test mode
- Bit Flip Test

The test modes in the following table are protected by a special software procedure to enable the mode. This software procedure enables write access to the test mode by a specific unlock key as shown in the table.

Test mode	Unlock key 1	Unlock key 2
RAM test mode	0x7575	0x8A8A

If the software sequence of the two consecutive unlock key write accesses (half-word or word access) is interrupted by any other write access to the register or if incorrect data is written to the Global Unlock Key Register, the corresponding test mode cannot be set and the sequence must be restarted.

After the two unlock key write accesses, the next write access should be to set the corresponding test mode enable bit. If this is not followed, the unlock mechanism reset and the test mode enable bit cannot be set and the unlock sequence must be restarted.

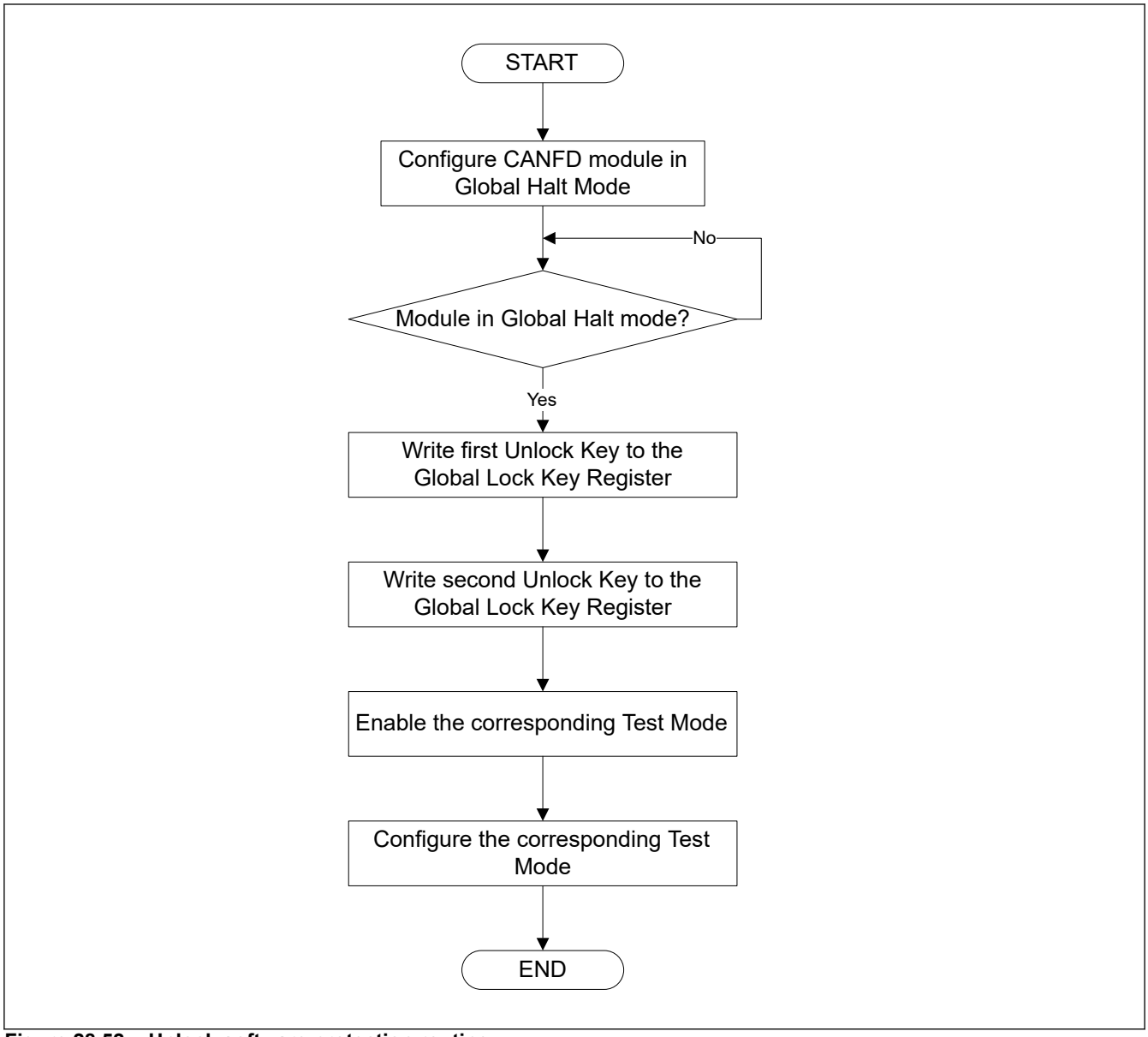


Figure 28.52 Unlock software protection routine

28.9.2.1 RAM Test Mode

The CANFD module can be configured in RAM test mode by setting the CFDGTSTCTR.RTME bit in the Global Test Control Register when the corresponding lock key is previously written. This is a special test mode, in which, the complete RAM area can be accessed.

Note: The actual RAM size is bigger than the RAM area initialized after a hardware reset. Therefore, ECC error flag (of the ECC macro) may be set if CPU reads data from this uninitialized RAM area while CANFD module is in RAM test mode.

In this mode, the RAM area is split into number of pages (pn) of 256 bytes, each which can be accessed with the CFDRPGACCk register.

The page should be selected for read/write access by writing to the CFDGTSTCFG.RTMPS[3:0] bits in the Global Test Control Register. Data can then be read from or written in to the RAM Test Page Access Registers.

Figure 28.53 shows the structure of the pages in the RAM when performing a RAM test mode.

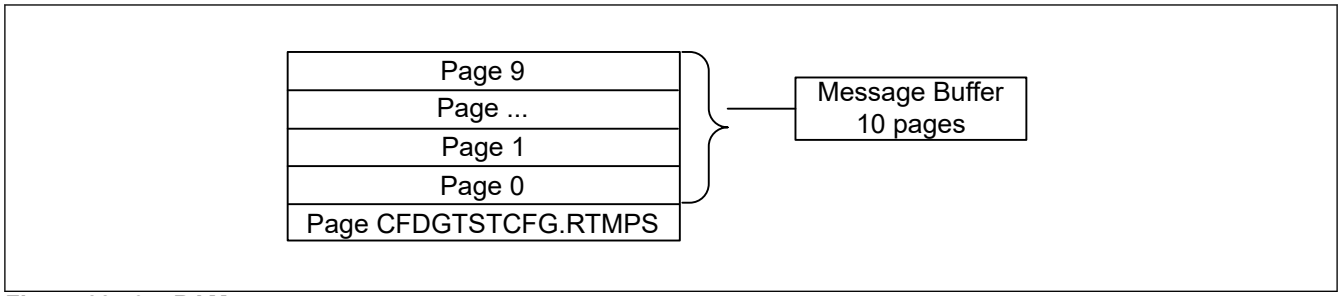


Figure 28.53 RAM page structure

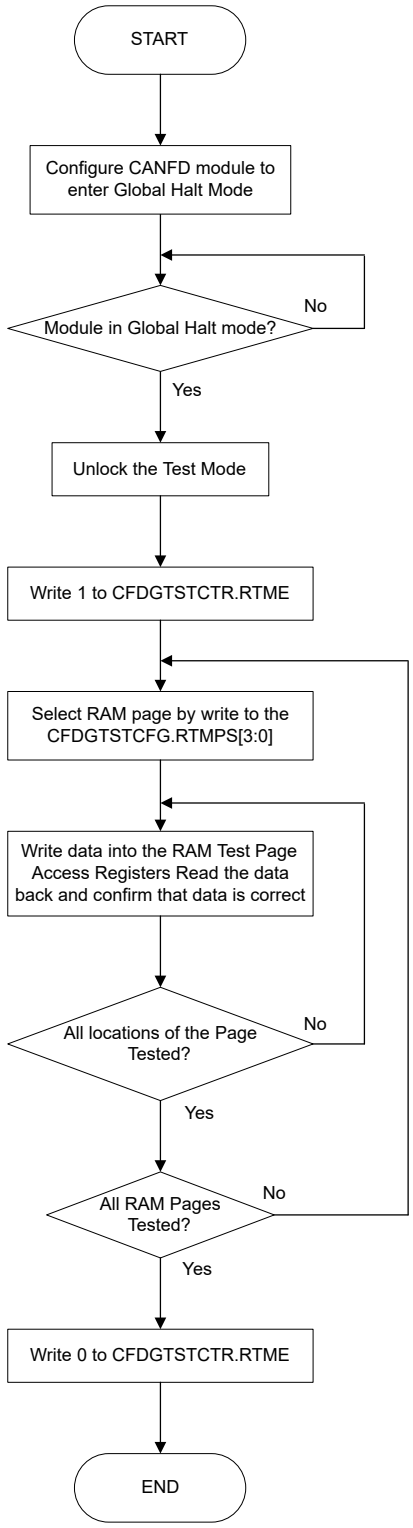
The total available RAM size is 2328 bytes for the Message Buffer RAM.

The pn and CFDGTSTCFG.RTMPS[3:0] values for the MB RAMs are calculated in the following way:

pn = ceil (total RAM size in bytes / number of bytes per page)

- MB RAM:
  pn = ceil (2328 / 256) = 10 pages
  CFDGTSTCFG.RTMPS[3:0] = 0 to 9 inclusive

Figure 28.54 shows the software flow for RAM test mode.



Note 1. Change into the following status before changing to RAMTEST.
- Cancel of a request for transmission.
- Disable of all the FIFO and TXQ.
- Clear of the receiving flag for a receiving buffer.

Figure 28.54 Software flow for RAM test mode

To exit this test mode, the CFDGTSTCTR.RTME bit must be cleared. The CFDGTSTCTR.RTME bit is cleared by writing 0 to it.

The CFDGTSTCTR.RTME bit is cleared automatically when the CANFD module enters Global Reset mode from the test mode.

28.9.2.2 Bit Flip Test

Bit Flip Test can invert the bit (the 1st bit of ID) of the beginning of the bit stream to receive.

If this function is used by a transmitting node, a bit error or an arbitration lost will occur.

If this function is used by a receiving node, a CRC error or a stuff error will occur.

Users should refer to the bit stuffing rule when using this feature, as there is the possibility of receiving a stuff error (due to the inversion) rather than a CRC error.

The following sequence should be used to perform CRC Error testing. In the sequence below CANFD module is the receiver.

1. Set the CFDC0CTR.BFT bit to 1’b1, in order to invert the first bit of the incoming bit stream from sending node
2. Wait for the can_cherr_int output signal to set to 1’b1
3. Read either the CFDC0ERFL.CRCREG or the CFDC0FDCRC.CRCREG (depending on the received frame type: Classical or FD). The value should be different from the received CRC value of the reference message from sending node.
4. Check that CFDC0ERFL.CERR is 1’b1

As the CRC generator logic is shared for RX and TX there is no need to create a separate TX CRC Error test.

29. CANFD ECC (CNECC)

29.1 Overview

MBRAM have ECC function of 2-bit ECC error detection and 1-bit ECC error detection and correction*1. The ECC module adds 7 bits ECC data to 32 bits RAM data.

Note 1. The ECC module cannot detect 3 or more bits error. In this case, the ECC module detects 1-bit or 2-bit error, does not detect errors, or corrects the erroneous bit to erroneous data by setting. When all RAM data are fixed to 0 or 1, it is detected as 2-bit ECC error.

29.2 Register Descriptions

29.2.1 EC710CTL : ECC Control Register

Base address: ECCMB = 0x4012_F200

Offset address: 0x00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ECDEDF0	ECSEDF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	EMCA[1:0]		—	—	ECOVFF	ECER2C	ECER1C	—	—	ECERVF	EC1ECP	EC2EDIC	EC1EDIC	ECER2F	ECER1F	ECEMF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit	Symbol	Function	R/W
0	ECEMF	ECC Error Message Flag 0: There is no bit error in present RAM output data 1: There is bit error in present RAM output data	R
1	ECER1F	ECC Error Detection and Correction Flag 0: After clearing this bit, 1-bit error correction has not occurred 1: 1-bit error has occurred	R
2	ECER2F	2-bit ECC Error Detection Flag 0: After clearing this bit, 2-bit error has not occurred 1: 2-bit error has occurred	R
3	EC1EDIC	ECC 1-bit Error Detection Interrupt Control 0: Disable 1-bit error detection interrupt request 1: Enable 1-bit error detection interrupt request	R/W
4	EC2EDIC	ECC 2-bit Error Detection Interrupt Control 0: Disable 2-bit error detection interrupt request 1: Enable 2-bit error detection interrupt request	R/W
5	EC1ECP	ECC 1-bit Error Correction Permission 0: At 1-bit error detection, the error correction is executed 1: At 1-bit error detection, the error correction is not executed	R/W
6	ECERVF	ECC Error Judgment Enable Flag 0: Error judgment disable 1: Error judgment enable	R/W
8:7	—	These bits are read as 0. The write value should be 0.	R/W
9	ECER1C	Accumulating ECC Error Detection and Correction Flag Clear 0: No effect 1: Clear accumulating ECC error detection and correction flag	R/W
10	ECER2C	2-bit ECC Error Detection Flag Clear 0: No effect 1: Clear 2-bit ECC error detection flag	R/W

Bit	Symbol	Function	R/W
11	ECOVFF	ECC Overflow Detection Flag 0: No effect 1: ECC overflow detection flag	R
13:12	—	These bits are read as 0. The write value should be 0.	R/W
15:14	EMCA[1:0]	Access Control to ECC Mode Select bit These bits enable or disable write access to ECERVF bit.	R/W
16	ECSEDF0	ECC Single bit Error Address Detection Flag 0: There is no bit error in EC710EAD0 after reset or clearing ECER1F bit 1: Address captured in EC710EAD0 shows that 1-bit error occurred and captured	R
17	ECDEDF0	ECC Dual Bit Error Address Detection Flag 0: There is no bit error in EC710EAD0 after reset or clearing ECER2F bit 1: Address captured in EC710EAD0 shows that 2-bit error occurred and captured	R
31:18	—	These bits are read as 0. The write value should be 0.	R/W

ECEMF bit (ECC Error Message Flag)

The ECEMF bit shows that there is error in present read data bus. This bit is updated by every RAM output data.

When RAM output data is undefined and the ECERVF bit is set to 1, the value of this bit is undefined.

[Setting condition]

There is bit error in present RAM output data under the condition that error judgement is enabled.

[Clearing condition]

- Under the condition that there is no 1-bit error in input data to decode circuit
- When ECC error judgement is disabled (ECERVF = 0).

ECER1F bit (ECC Error Detection and Correction Flag)

The ECER1F bit shows that the bit errors are detected in the one part of RAM read data [38:0] at RAM read access when the error judgment is enabled.

When the 1-bit error interrupt output is enabled, error interrupt is generated by setting this flag.

This bit is read-only, so writing 1 or 0 has no effect.

At clearing, write 1 to the ECER1C bit.

When 1-bit error is detected again under the condition that this bit is set, the interrupt is not generated.

[Setting condition]

When the error judgment is enabled and there is 1-bit error to RAM output data (when not setting ECER1C = 1).

[Clearing condition]

- Writing ECER1C = 1
- When ECC error judgement is disabled (ECERVF = 0).

ECER2F bit (2-bit ECC Error Detection Flag)

The ECER2F bit shows that the bit errors are detected in the two parts of RAM read data [38:0] at RAM read access when the error judgment is enabled.

When the 2-bit error interrupt output is enabled, error interrupt is generated by setting this flag.

This bit is read-only, so writing 1 or 0 has no effect.

At clearing, write 1 to the ECER2C bit.

When 2-bit error is detected again under the condition that this bit is set, the interrupt is not generated.

[Setting condition]

When the error judgment is enabled and there is 2-bit error to RAM output data (when not setting ECER2C = 1).

[Clearing condition]

- Writing ECER2C = 1
- When ECC error judgement is disabled (ECERVF = 0).

EC1EDIC bit (ECC 1-bit Error Detection Interrupt Control)

The EC1EDIC controls the interrupt output at detecting 1-bit error. By setting 1 to this bit, the 1-bit error interrupt is outputted when 1-bit error detected.

EC2EDIC bit (ECC 2-bit Error Detection Interrupt Control)

The EC2EDIC controls the interrupt output at detecting 2-bit error. By setting 1 to this bit, the 2-bit error interrupt is outputted when 2-bit error detected.

EC1ECP bit (ECC 1-bit Error Correction Permission)

The EC1ECP sets enable or disable to correct the 1-bit error when ECC error detection and correction is valid. By setting 1 to this bit, the non-corrected data is outputted if 1-bit error is detected.

ECERVF bit (ECC Error Judgment Enable Flag)

Setting the ECERVF bit to 1 enables the judgment of error. The correction of output data and the interrupt output depend on setting of the EC1ECP bit, EC2EDIC bit, and EC1EDIC bit.

The write access to this bit is valid when the write value of the EMCA[1:0] is 01b. So only the 16 bits or 32 bits operation command is valid in the case of the write access to this bit.

ECER1C bit (Accumulating ECC Error Detection and Correction Flag Clear)

The ECER1C bit clears the status flag of the ECER1F bit.

The read value is always 0. By writing 0, the internal condition is not changed. When the competition between writing 1 to this bit and setting the ECER1F bit, the former has priority.

The ECER1F bit is cleared by writing 1 to this bit while the ECER1F bit is set. Additionally, the Overflow Detection flag (ECOVFF), ECC Dual Bit Error flag (ECDEDF0) and ECC Single Bit Error flag (ECSEDF0) are also cleared.

ECER2C bit (2-bit ECC Error Detection Flag Clear)

The ECER2C bit clears the status flag of the ECER2F bit.

The read value is always 0. By writing 0, the internal condition is not changed. When the competition between writing 1 to this bit and setting the ECER2F bit, the former has priority.

The ECER2F bit is cleared by writing 1 to this bit while the ECER2F bit is set. Additionally, the Overflow Detection flag (ECOVFF), ECC Dual Bit Error flag (ECDEDF0), and ECC Single Bit Error flag (ECSEDF0) are also cleared.

ECOVFF bit (ECC Overflow Detection Flag)

The ECOVFF bit is set and the overflow interruption is outputted by detecting the new error address under the condition that error address is already captured in the EC710EAD0 register. The overflow interrupt is outputted again when this bit is set and new error is detected.

This bit is read-only, so writing 1 or 0 has no effect.

To clear this bit, write 1 to the ECER2C bit and the ECER1C bit.

[Setting condition]

When new error address is captured under the condition that error address is already captured in the EC710EAD0 register (when not setting ECER2C = 1 or ECER1C = 1).

[Clearing condition]

- Writing ECER2C = 1 or ECER1C = 1
- When ECC error judgement is disabled (ECERVF = 0).

EMCA[1:0] bit (Access Control to ECC Mode Select bit)

The EMCA[1:0] bits are the write trigger reserved bits to the ECERVF bit. The read value is always 0. When the value of these bits is 01b, it is possible to have write access to the ECERVF bit. If these bits are not 01b, write access to the ECERVF bit is ignored and the value is not written.

ECSEDF0 bit (ECC Single bit Error Address Detection Flag)

The ECSEDF0 bit shows that the error is captured in the error address register when error detection is valid. This bit is set by 1-bit error detection.

When 1-bit error is detected after the 2-bit error address is already captured in the EC710EAD0 register, this bit is not updated but the EC710EAD0 register is updated.

This bit is read-only, so writing 1 or 0 has no effect. To clear these bits, write 1 to the ECER1C bit.

[Setting condition]

When there is 1-bit error to RAM output data and error address is captured in EC710EAD0 under the condition that the error judgment is permitted (when not setting ECER1C = 1).

[Clearing condition]

- Writing ECER1C = 1
- When ECC error judgement is disabled (ECERVF = 0).

ECDEDF0 bit (ECC Dual Bit Error Address Detection Flag)

The ECDEDF0 bit shows that the error is captured in the error address register when error detection is valid. This bit is set by 2-bit error detection.

When 2-bit error is detected after the 1-bit error address is already captured in the EC710EAD0 register, this bit is not updated and the EC710EAD0 register is updated.

This bit is read-only, so writing 1 or 0 has no effect. To clear these bits, write 1 to the ECER2C bit.

[Setting condition]

When there is 2-bit error to RAM output data and error address is captured in EC710EAD0 under the condition that the error judgment is permitted (when not setting ECER2C = 1).

[Clearing condition]

- Writing ECER2C = 1
- When ECC error judgement is disabled (ECERVF = 0).

29.2.2 EC710TMC : ECC Test Mode Control Register

Base address: ECCMB = 0x4012_F200

Offset address: 0x04

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

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
1	ECDCS	ECC Decode Input Select 0: Input lower 32 bits of RAM output data to data area of decode circuit 1: Input ECEDB31-0 in EC710TED register to data area of decode circuit	R/W
6:2	—	These bits are read as 0. The write value should be 0.	R/W
7	ECTMCE	ECC Test Mode Control Enable 0: The access to test mode register and bit is disabled 1: The access to test mode register and bit is enabled	R/W
13:8	—	These bits are read as 0. The write value should be 0.	R/W
15:14	ETMA[1:0]	ECC Test Mode Bit Access Control These bits enable or disable write access to ECTMCE bit.	R/W

ECDCS bit (ECC Decode Input Select)

The ECDCS bit selects either the lower 32 bits data value from RAM or value from the internal test register (EDEDB[31:0] in EC710TED) as input signal to decoder.

The write access to this bit is valid under the condition of ECTMCE = 1 (it is possible to set them at the same time.)

This bit is cleared by setting ECTMCE = 0.

ECTMCE bit (ECC Test Mode Control Enable)

The ECTMCE bit selects the access enable or disable to test register and test control bit.

The write access to this bit is valid under the condition that the value of the ETMA[1:0] bits is 10b.

ETMA[1:0] bits (ECC Test Mode Bit Access Control)

The ETMA[1:0] bits are the write trigger reserved bits to the ECTMCE bit. The read value is always 0. When the value of these bits is 10b, it is possible to have write access to the ECTMCE bit. If these bits are not 10b, the write access to the ECTMCE bit is ignored and the value is not written.

29.2.3 EC710TED : ECC Test Substitute Data Register

Base address: ECCMB = 0x4012_F200

Offset address: 0x0C

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

Bit	Symbol	Function	R/W
31:0	ECEDB[31:0]	ECC Test Substitute Data Substitute data in ECC test mode.	R/W

This register is for the 32 bits data for ECC decode. It is possible to read and write using 32 bits operation command when ECTMCE = 1. When ECTMCE = 0, all bits are always 0.

ECEDB[31:0] bits (ECC Test Substitute Data)

When ECDCS in EC710TMC register is 1, the value of this register is bits [31:0] of the input data to the decode circuit.

29.2.4 EC710EAD0 : ECC Error Address Register

Base address: ECCMB = 0x4012_F200

Offset address: 0x10

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

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

Bit	Symbol	Function	R/W
10:0	ECEAD[10:0]	ECC Error Address	R
31:11	—	These bits are read as 0. The write value should be 0.	R

This is a read-only register to hold the ECC error address.

ECEAD[10:0] bits (ECC Error Address)

When ECC error is detected for permitting ECC error judgment, RAM address is captured by the detected signal as a trigger and is hold as the error occurring address. The error address is not captured when the error occurred again to the one held by the same factor.

If 2-bit error occurred under the condition that 1-bit error address is already captured, the 2-bit error address is over-written and the ECDEDF0 bit is set to 1.

If 1-bit error occurred under the condition that 2-bit error address is already captured, the 1-bit error address is not overwritten and the ECSEDF0 bit is not set to 1.

29.3 Operation

29.3.1 ECC Function Setting

Figure 29.1 shows a procedure for ECC function setting.

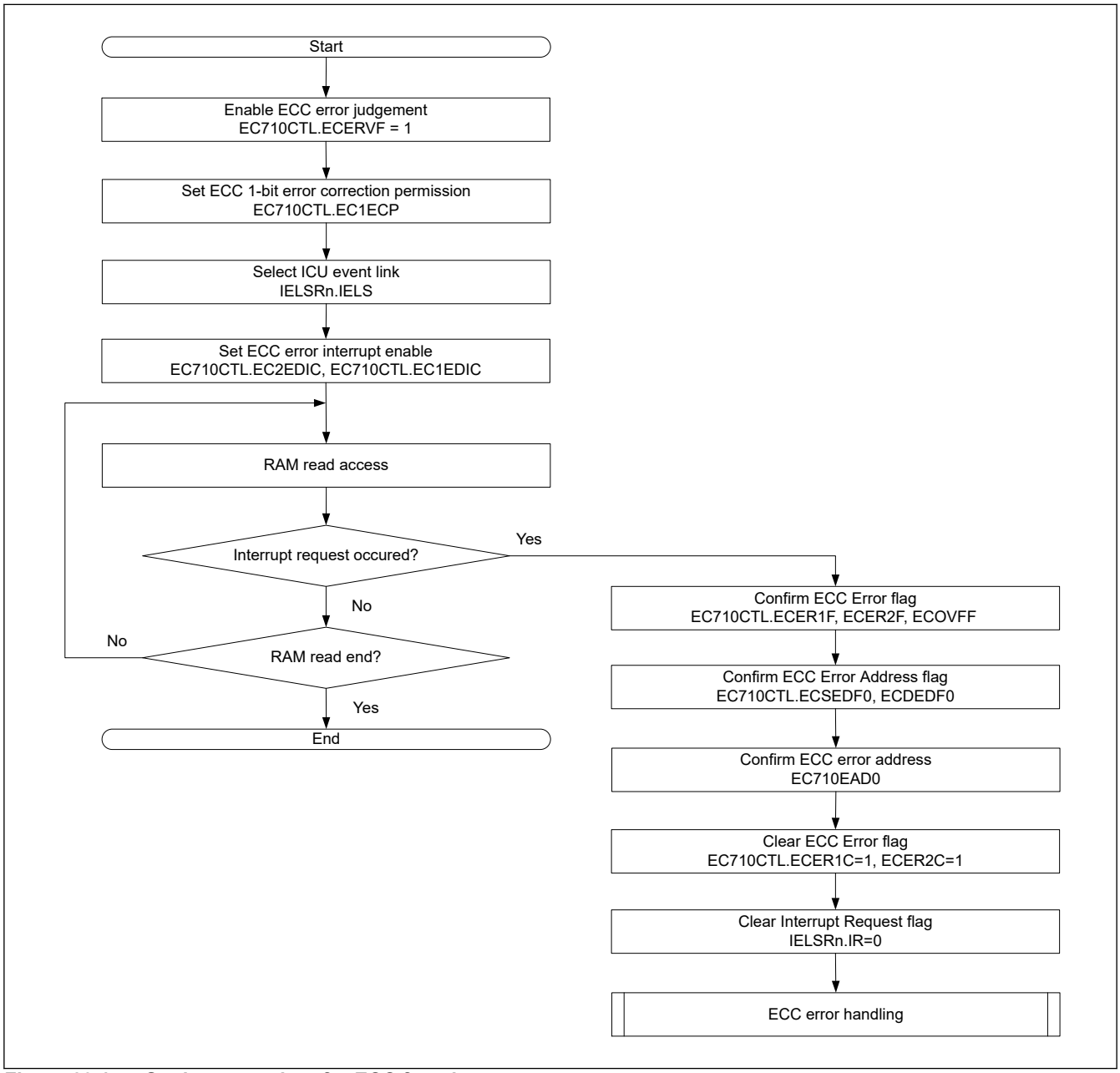


Figure 29.1 Setting procedure for ECC function

29.3.2 ECC Decoder Testing

ECC interrupts can be intentionally generated by ECC test mode. Figure 29.2 shows a procedure for ECC decoder testing.

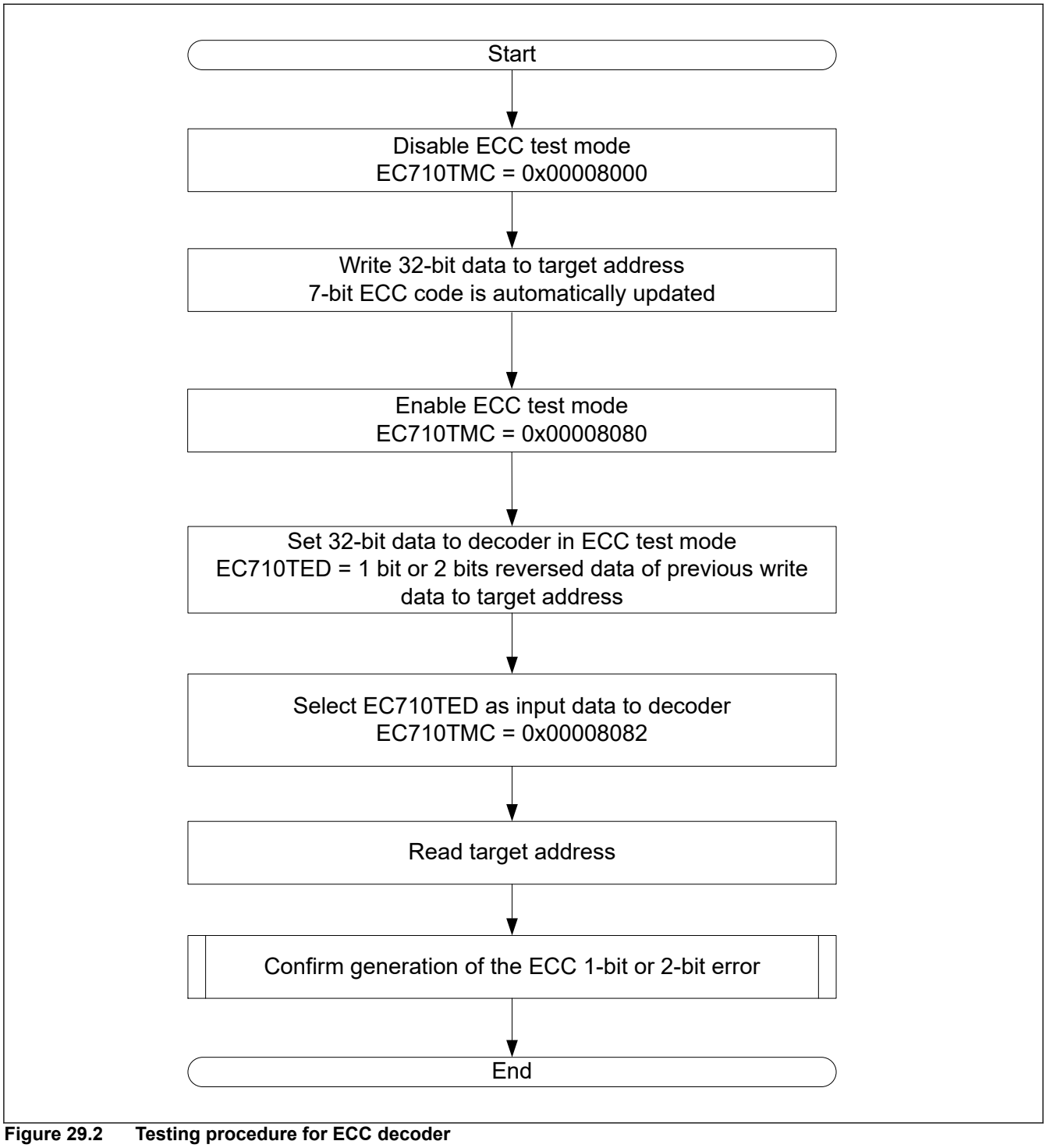


Figure 29.2 Testing procedure for ECC decoder

29.4 Interrupts

The ECC module issues three interrupt requests:

- CAN_MRAM_ERI.

Interrupt sources of each interrupt request include:

- 1-bit ECC error

- 2-bit ECC error
- ECC error overflow.

30. Serial Peripheral Interface (SPI)

This is the SPI_B version of the SPI peripheral module.

SPI_B is referred to as SPI in this chapter.

30.1 Overview

The Serial Peripheral Interface (SPI) has 2 channels. The SPI provides high-speed full-duplex synchronous serial communications with multiple processors and peripheral devices. Table 30.1 lists the SPI specifications, Figure 30.1 shows a block diagram, and Table 30.2 lists the I/O pins.

In this section, PCLK refers to PCLKA. TCLK refers to SPITCLK.

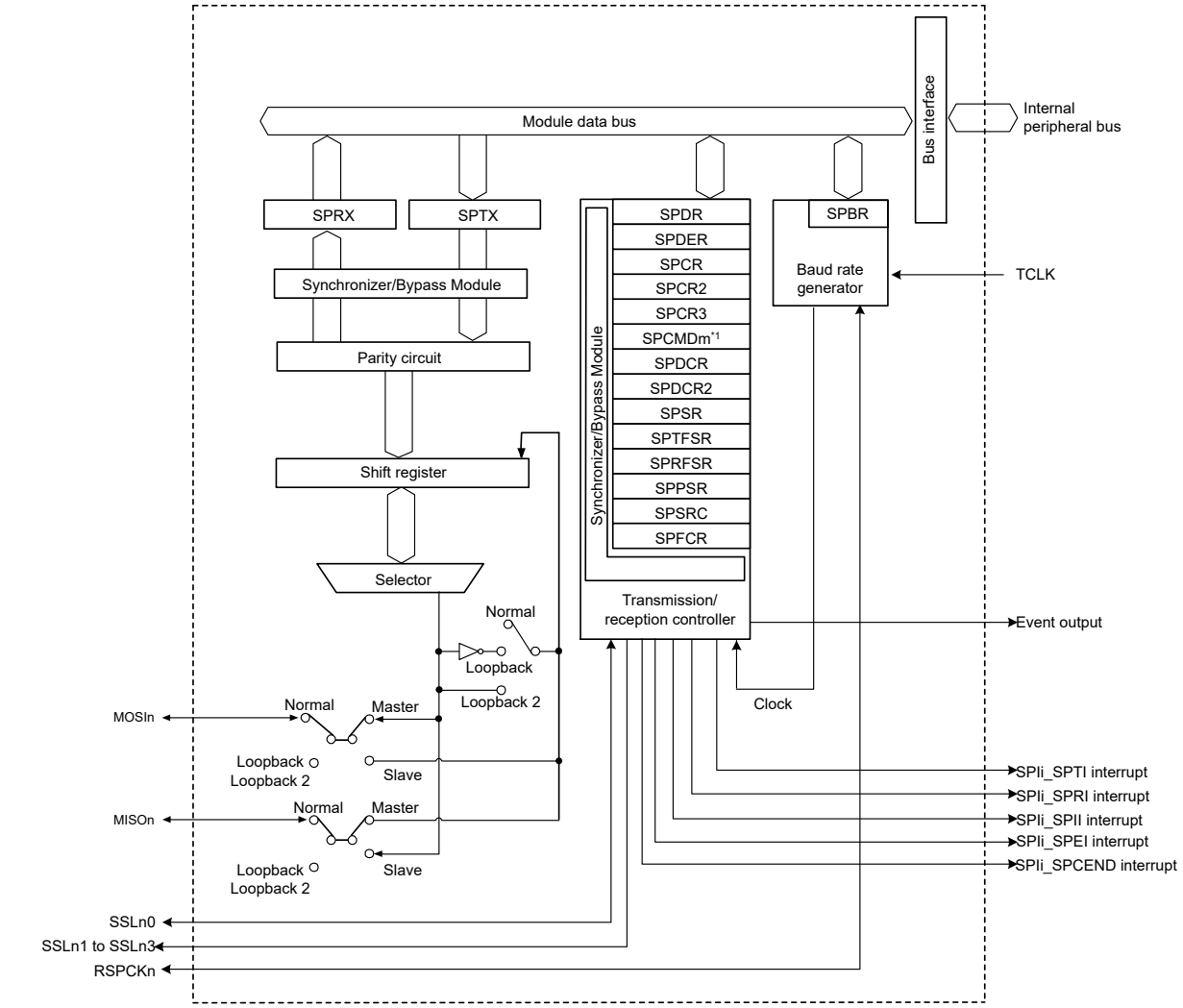
Table 30.1 SPI specifications (1 of 2)

Parameter	Specifications
Number of channels	Two channels
SPI transfer functions	• Use of MOSI (master out/slave in), MISO (master in/slave out), SSL (slave select), and RSPCK (SPI clock) signals allows serial communications through SPI operation (4-wire method) or clock synchronous operation (3-wire method) • Transmit-only operation available • Receive-only operation is available • Communication mode selectable to full-duplex, transmit-only or receive-only • RSPCK polarity switching • RSPCK phase switching
Data format	• MSB-first or LSB-first selectable • Transfer bit length selectable from 4 to 32 bits • 32 bit × 4 stages FIFO is available as transmit buffer or receive buffer • Byte swap operating function • Transmit/receive data can be inverted.
Bit rate	• In master mode, the on-chip baud rate generator generates RSPCK by frequency-dividing TCLK (the division ratio ranges from divided by 2 to divided by 4096) • In slave mode, the minimum TCLK clock divided by 2 can be input as RSPCK (TCLK divided by 2 is the maximum RSPCK frequency) Width at high level: 1 TCLK cycles; width at low level: 1 TCLK cycles
Buffer configuration	• Double buffer configuration for the transmit and receive buffers
Error detection	• Mode fault error detection • Underrun error detection • Overrun error detection*1 • Parity error detection • Receive data ready detection
SSL control function	[motorola SPI mode/TI SSP mode common] • Four SSL pins (SSLni: SSLn0 to SSLn3) (n = A, B) for each channel • In single-master mode, SSLn0 to SSLn3 pins are output • In multi-master mode, SSLn0 pin for input, and SSLn1 to SSLn3 pins either for output or unused • In slave mode, SSLn0 pin for input and SSLn1 to SSLn3 pins unused • Controllable delay from SSL output assertion to RSPCK operation (RSPCK delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) • Controllable delay from RSPCK stop to SSL output negation (SSL negation delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) • Controllable wait for next-access SSL output assertion (next-access delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) • Function for changing SSL polarity • Delay between frames in burst transfer is settable [only Motorola mode] • Controllable delay from SSL output assertion to RSPCK operation (RSPCK delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) [only TI SSP mode] • Controllable delay from OE output assertion to RSPCK operation (RSPCK delay) Range: 0 to 8 RSPCK cycles (set in RSPCK-cycle units)
Communication protocol	• Motorola SPI • TI SSP(Synchronous Serial Protocol)

Table 30.1 SPI specifications (2 of 2)

Parameter	Specifications
Control in master transfer	● Transfers of up to eight commands each can be executed sequentially in looped execution ● For each command, the following can be set: SSL signal value, bit rate, RSPCK polarity and phase, transfer data length, MSB- or LSB-first, burst, RSPCK delay, SSL negation delay, and next-access delay ● Transfers can be initiated by writing to the transmit buffer ● MOSI signal value specifiable in SSL negation ● RSPCK auto-stop function
Interrupt sources	Interrupt sources: ● Receive buffer full / Receive data ready interrupt ● Transmit buffer empty interrupt ● SPI error interrupt (mode fault error, under run error, over run error, parity error, receive data ready) ● SPI idle interrupt (SPI idle) ● Communication end interrupt
Event link function	The following events can be output to the Event Link Controller (ELC): ● Receive buffer full / receive data ready signal ● Transmit buffer empty signal ● Mode fault, underrun, overrun, parity error, or receive data ready signal ● SPI idle signal ● Communication end signal
Other functions	● Switching between CMOS output and open-drain output ● SPI initialization function ● Loopback mode
Module-stop function	Module-stop state can be set to reduce power consumption.
TrustZone Filter	Security attribution can be set

Note 1. In master reception and when the RSPCK auto-stop function is enabled, an overrun error does not occur because the transfer clock is stopped on overrun error detection.



Note 1. m = 0 to 7

Figure 30.1 SPI block diagram

The SPI automatically switches the I/O direction of the SSLn0 pin. SSLn0 is set as an output when the SPI is a single master, and as an input when the SPI is a multi-master or a slave. The RSPCKn, MOSIn, and MISOn pins are automatically set as inputs or outputs based on the master or slave setting and the level input on the SSLn0 pin. For details, see section 30.3.2. Controlling the SPI Pins.

Table 30.2 SPI I/O pins (1 of 2)

Channel	Pin name	I/O	Description
SPI0	RSPCKA	I/O	Clock input/output pin
	SSLA0	I/O	Slave selection input/output
	SSLA1 to SSLA3	Output	Slave selection output
	MOSIA	I/O	Master transmit data input/output
	MISOA	I/O	Slave transmit data input/output
SPI1	RSPCKB	I/O	Clock input/output pin
	MOSIB	I/O	Master transmit data input/output
	MISOB	I/O	Slave transmit data input/output
	SSLB0	I/O	Slave selection input/output

Table 30.2 SPI I/O pins (2 of 2)

Channel	Pin name	I/O	Description
	SSLB1 to SSLB3	Output	Slave selection output

Note: Pin names are indicated as “...A” or “...An” for SPI0, and “...B” or “...Bn” for SPI1 (n = 0, 1, 2, or 3).

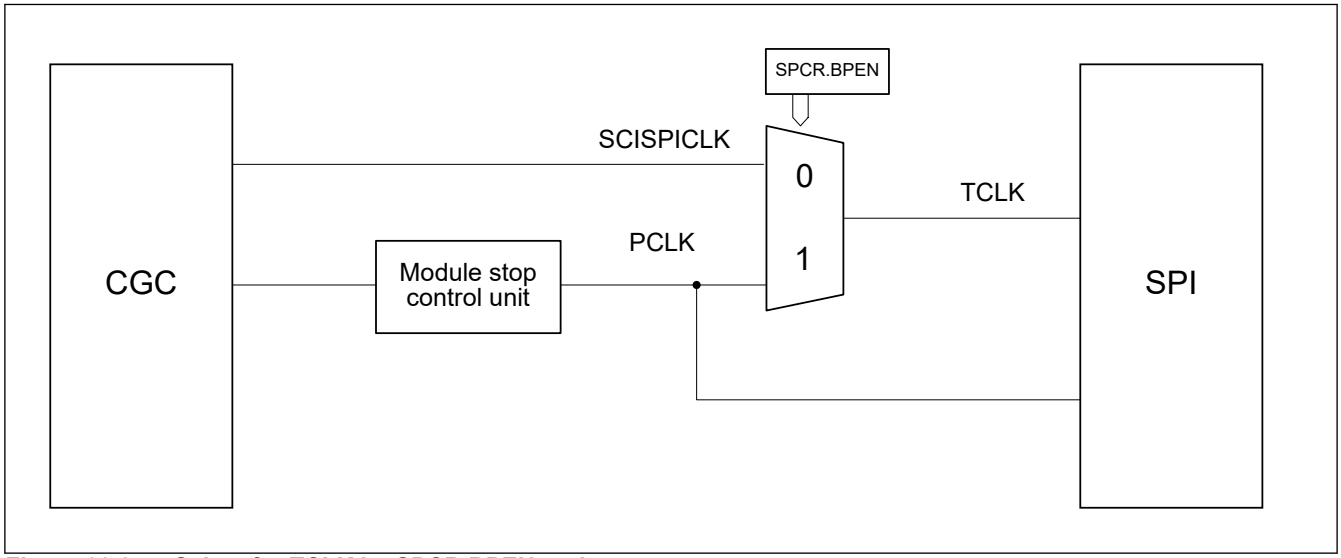


Figure 30.2 Select for TCLK by SPCR.BPEN register

30.2 Register Descriptions

30.2.1 SPDR : SPI Data Register

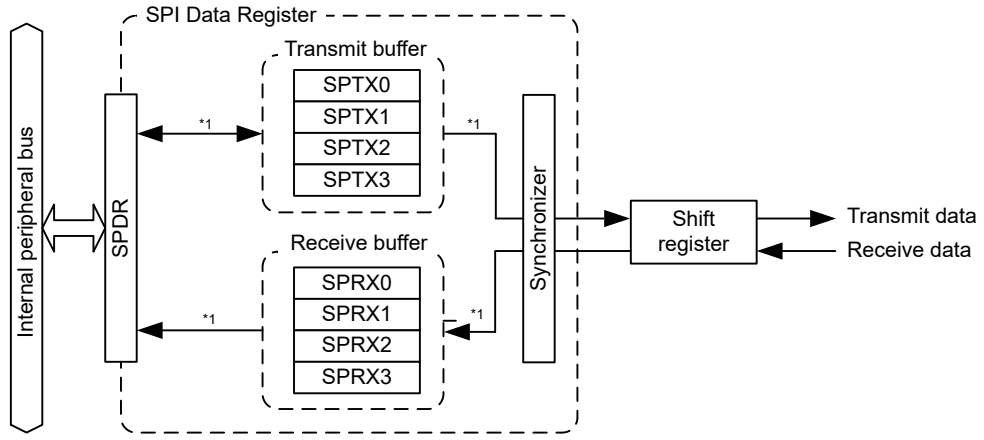
Base address: SPI_Bn = 0x4011_A000 + 0x0100 × n (n = 0, 1)

Offset address: 0x00

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

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

SPDR is the interface with the buffers that hold data for transmission and reception by the SPI. When accessing this register in words, access SPDR. The transmit buffer (SPTX) and receive buffer (SPRX) are independent but both are mapped to SPDR. Figure 30.3 shows the configuration of the SPDR register.



Note 1. The destination buffer and stage for access is automatically switched by hardware.

Figure 30.3 Structure of SPDR

32 bits × 4 stage transmit FIFO and 32 bits × 4 receive FIFO are provided. These 8 stage FIFO are mapped to one address in the SPDR. Transmit buffers (SPTXn, n = 0 to 3) can be written by writing data to SPDR to transmit written data.

Upon completion of receiving data, receive buffers store received data. When an overrun error occurs, data in the receive buffer is not updated.

(1) Bus Interface

The SPI data register has 32 bits × 4 stage transmit FIFO and 32 bits × 4 receive FIFO (32 bytes in total). These 32 bytes are mapped to the 4-byte space of SPDR. Write transmit data from the LSB. Received data is stored from the LSB.

SPDR register write operation and read operation are described below.

1. Write
   A transmit buffer write pointer is provided for transmit buffers. When data is written to SPDR, the pointer automatically switches to the next buffer. The following illustrates the structure of the transmit buffer bus interface (write).

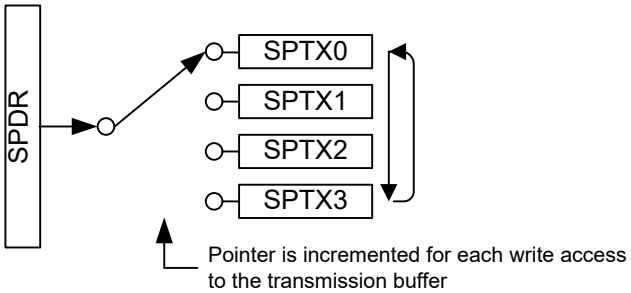


Figure 30.4 Structure of SPDR (Write)

   The transmit buffer (SPTX0 to SPTX3) switching order:
   SPTX0→SPTX1→SPTX2→SPTX3→SPTX0→SPTX1→…
   When writing transmit data to transmit buffers (SPTXn), write transmit data of frames +1 specified by the Transmission FIFO threshold setting bits of SPI data control register 2 (SPDCR2.TTRG[1:0]) while an SPI transmit buffer empty interrupt is present (SPSR.SPTEF flag = 1). Writing to the transmit buffer (SPTXn, n = 0 to 3) in the state where there is no empty stage in the transmit FIFO does not update the buffer value.

2. Read
   Values can be read from receive buffers (SPRXn, n = 0 to 3) or transmit buffers (SPTXn, n = 0 to 3) by reading the SPDR register. Reading a receive buffer or reading a transmit buffer can be selected by the SPI receive data or transmit data select bit (SPDCR.SPRDTD) in the SPI data control register.
   The SPDR register is read according to the independent receive buffer read pointer and the transmit buffer read pointer.

The following illustrates the structure of the receive buffer and transmit buffer bus interface (read).

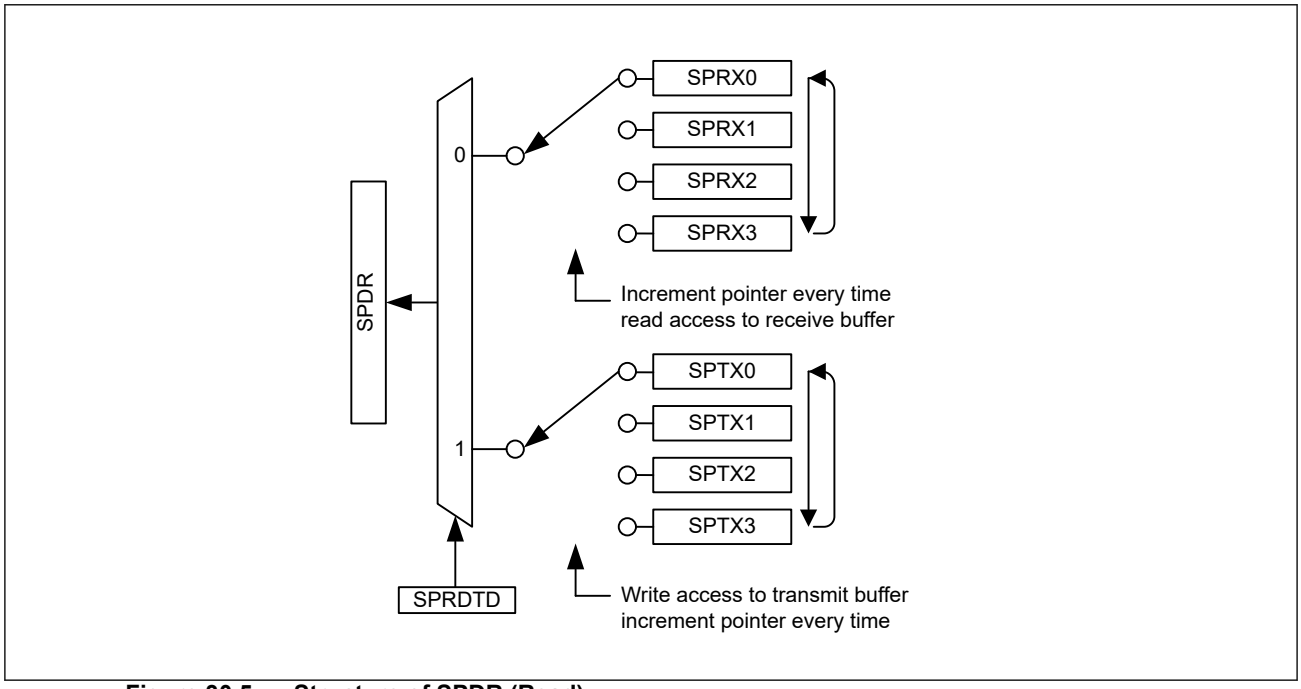


Figure 30.5 Structure of SPDR (Read)

When a receive buffer is read, the receive buffer read pointer automatically switches to the next buffer. The receive buffer read pointer switches in the same order as the transmit buffer write pointer.

The transmit buffer read pointer is updated during the SPDR write access, but it is not updated during the transmit buffer read access. When a transmit buffer is read, the value written to SPDR last can be read.

30.2.2 SPDECR : SPI Delay Control Register

Base address: SPI_Bn = 0x4011_A000 + 0x0100 × n (n = 0, 1)

Offset address: 0x04

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

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

Bit	Symbol	Function	R/W
2:0	SCKDL[2:0]	RSPCK Delay 0 0 0: 1RSPCK 0 0 1: 2RSPCK 0 1 0: 3RSPCK 0 1 1: 4RSPCK 1 0 0: 5RSPCK 1 0 1: 6RSPCK 1 1 0: 7RSPCK 1 1 1: 8RSPCK	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
10:8	SLNDL[2:0]	SSL Negation Delay [Master Mode] 0 0 0: 1RSPCK 0 0 1: 2RSPCK 0 1 0: 3RSPCK 0 1 1: 4RSPCK 1 0 0: 5RSPCK 1 0 1: 6RSPCK 1 1 0: 7RSPCK 1 1 1: 8RSPCK [TI-SSP case in Slave Mode] 0 0 0: 1 TCLK 0 0 1: 2 TCLK 0 1 0: 3 TCLK 0 1 1: 4 TCLK 1 0 0: 5 TCLK 1 0 1: 6 TCLK 1 1 0: 7 TCLK 1 1 1: 8 TCLK	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W
18:16	SPNDL[2:0]	SPI Next-Access Delay 0 0 0: 1RSPCK + 5TCLK 0 0 1: 2RSPCK + 5TCLK 0 1 0: 3RSPCK + 5TCLK 0 1 1: 4RSPCK + 5TCLK 1 0 0: 5RSPCK + 5TCLK 1 0 1: 6RSPCK + 5TCLK 1 1 0: 7RSPCK + 5TCLK 1 1 1: 8RSPCK + 5TCLK	R/W
31:19	—	These bits are read as 0. The write value should be 0.	R/W

SCKDL[2:0] bit (RSPCK Delay)

[In the Motorola-SPI case]

The RSPCK delay bits (SCKDL) are used to set the period (RSPCK delay) from SSL signal assertion start until RSPCK oscillates while the SCKDEN bit in the SPI command register (SPCMD) is 1. If SCKDL is modified while the MSTR bit and the SPE bit in the SPI control register (SPCR) are 1, subsequent operation is not guaranteed.

To use the SPI in slave mode, set SCKDL[2:0] bits to 000b.

[In the TI-SSP case]

The RSPCK delay bits (SCKDL) are used to set the period (RSPCK delay) from SSL signal assertion start until RSPCK oscillates while the SCKDEN bit in the SPI command register (SPCMD) is 1. Also that is used to set the period until the SSL signal is negated. If SCKDL is modified while the MSTR bit and the SPE bit in the SPI control register (SPCR) are 1, subsequent operation is not guaranteed.

To use the SPI in slave mode, set SCKDL[2:0] bits to 000b.

SLNDL[2:0] bit (SSL Negation Delay)

[In the Motorola-SPI case]

The SSL negation delay bits (SLNDL) are used to set the period (SSL negation delay) after the SPI in master mode sends the final RSPCK edge during serial transfer until it negates the SSL signal while the SLNDEN bit in the SPI command register (SPCMD) is 1. If SLNDL is modified while the MSTR bit and the SPE bit in the SPI control register (SPCR) are 1, subsequent operation is not guaranteed.

To use the SPI in slave mode except TI-SSP, set SLNDL[2:0] bits to 000b.

[In the TI-SSP case]

The SSL negation delay bits (SLNDL) are used to set the period (OE negation delay) after the SPI in master mode sends the final RSPCK edge during serial transfer until it negates the OE signal while the SLNDEN bit in the SPI command register (SPCMD) is 1. Also, that is used to set the period from when the SPI in slave mode detects the last RSPCK edge of serial

transfer to when the OE signal is negated. If SLNDL is modified while the SPE bit in the SPI control register (SPCR) are 1, subsequent operation is not guaranteed.

SPNDL[2:0] bit (SPI Next-Access Delay)

The SPI next-access delay register (SPDECR.SPNDL) is used to set the SSL signal inactive period (next-access delay) after completion of serial transfer while the SPNDEN bit in the SPI command register (SPCMD) is 1. If SPNDL is modified while the MSTR bit and the SPE bit in the SPI control register (SPCR) are 1, subsequent operation is not guaranteed.

These bits are used to set the next-access delay value when the SPNDEN bit in SPCMD is 1. To use the SPI in slave mode, set SPNDL[2:0] bits to 000b.

30.2.3 SPCR : SPI Control Register

Base address: SPI_Bn = 0x4011_A000 + 0x0100 × n (n = 0, 1)

Offset address: 0x08

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	BPEN	MSTR	TXMD[1:0]		—	—	SPFRF	SPMS	—	—	CENDIE	SPTIE	SPDRES	SPIIE	SPRIE	SPEIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	MODFEN	BFDS	SCKASE	PTE	—	SPOE	SPPE	—	—	—	—	—	—	—	SPE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SPE	SPI Function Enable 0: SPI function is disabled. 1: SPI function is enabled.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R
8	SPPE	Parity Enable 0: A parity bit is not added to transmit data. Received-data parity check is not performed. 1: A parity bit is added to transmit data. Received-data parity check is performed.	R/W
9	SPOE	Parity Mode 0: Even parity is used for transmission and reception. 1: Odd parity is used for transmission and reception.	R/W
10	—	This bit is read as 0. The write value should be 0.	R
11	PTE	Parity Self-Diagnosis Enable 0: Parity circuit self-diagnosis function is disabled. 1: Parity circuit self-diagnosis function is enabled.	R/W
12	SCKASE	RSPCK Auto-Stop Function Enable 0: RSPCK auto-stop function is disabled. 1: RSPCK auto-stop function is enabled.	R/W
13	BFDS	Between Burst Transfer Frames Delay Select 0: Delay (RSPCK delay, SSL negation delay and next-access delay) between frames is inserted in burst transfer 1: Delay between frames is not inserted in burst transfer.	R/W
14	MODFEN	Mode Fault Error Detection Enable 0: Mode fault error detection is disabled. 1: Mode fault error detection is enabled.	R/W
15	—	This bit is read as 0. The write value should be 0.	R
16	SPEIE	SPI Error Interrupt Enable 0: SPI error interrupt request is disabled. 1: SPI error interrupt request is enabled.	R/W

Bit	Symbol	Function	R/W
17	SPRIE	SPI Receive Buffer Full Interrupt Enable 0: SPI receive buffer full interrupt request is disabled. 1: SPI receive buffer full interrupt request is enabled.	R/W
18	SPIIE	SPI Idle Interrupt Enable 0: Idle interrupt request is disabled. 1: Idle interrupt request is enabled.	R/W
19	SPDRES	SPI receive data ready error select Select the interrupt request to be generated when the reception data ready is detected 0: Receive data full interrupt 1: Error interrupt	R/W
20	SPTIE	SPI Transmit Buffer Empty Interrupt Enable 0: SPI transmit buffer empty interrupt request is disabled. 1: SPI transmit buffer empty interrupt request is enabled.	R/W
21	CENDIE	SPI Communication End Interrupt Enable 0: Communication end interrupt request is disabled. 1: Communication end interrupt request is enabled.	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R
24	SPMS	SPI Mode Select 0: SPI operation (4-wire) 1: Clock synchronous operation (3-wire)	R/W
25	SPFRF	SPI Frame Format Select 0: Motorola-SPI 1: TI-SSP Note: When SPMS = 1 (clock synchronous operation (3-wire)), this bit setting is invalid.	R/W
27:26	—	These bits are read as 0. The write value should be 0.	R
29:28	TXMD[1:0]	Communication Mode Select 0 0: Transmit-Receive 0 1: Transmit only Others: Receive only	R/W
30	MSTR	SPI Master/Slave Mode Select 0: Slave mode 1: Master mode	R/W
31	BPEN	Synchronization Circuit Bypass Enable 0: Non-Bypass 1: Bypass	R/W

The SPI control register (SPCR) is used to set operating mode of the SPI. If the set BPEN, MSTR, TXMD[1:0], SPFRF, SPMS, MODFEN, BFDS, SCKASE, PTE, SPOE, SPPE bit value is modified while the SPE bit = 1, subsequent operation is not guaranteed.

SPE bit (SPI Function Enable)

This bit is used to enable or disable SPI functions. Setting this bit to 1 enables SPI functions. When the MODF flag in the SPI status register (SPSR) is 1, the SPE bit is cleared to 0 and the SPE bit cannot be set to 1 until the MODF flag is cleared to 0. (See section 30.3.10. Error Detection) Setting the SPE bit to 0 disables SPI functions and initializes a part of module functions. (See section 30.3.11. Initializing the SPI)

SPPE bit (Parity Enable)

This bit is used to enable or disable the parity function.

SPOE bit (Parity Mode)

This bit is used to specify even parity or odd parity.

In even parity mode, the parity bit is determined so that the sum of 1 (parity bit + transmit characters or receive characters) becomes an even number. In the same way, in odd parity mode, a parity bit is determined so that the sum of 1 (parity bit + transmit characters or receive characters) becomes an odd number. The SPOE bit is valid only when the SPPE bit in SPCR is set to 1.

PTE bit (Parity Self-Diagnosis Enable)

This bit is used to enable or disable self-diagnosis of the parity circuit to confirm that the parity function is normal.

SCKASE bit (RSPCK Auto-Stop Function Enable)

This bit is used to enable or disable the RSPCK auto-stop function. When this function is enabled, the RSPCK clock stops immediately before an overrun error occurs during data reception in master mode. For details, see section 30.3.10.1. Overrun errors.

BFDS bit (Between Burst Transfer Frames Delay Select)

This bit controls whether insert the delay time between the burst transfer frames.

Valid in the master mode (SPCR.MSTR = 1) for frames with the SPCMDn.SSLKP bit set to 1.

This bit should be set to 0 in slave mode. The usage of SSL delay control between transfer frames is shown as below. For details, see section 30.3.12.1. Master mode operation.

1. Non-burst transmits
2. Burst transmit with delay between frames
   2-1. From the 1st frame to the last previous frame
   2-2. The last frame
3. Burst transmit with no delay between frames
   3-1. From the 1st frame to the last previous frame
   3-2. The last frame

Table 30.3 Usage of SSL delay control between transfer frames (Master mode)

	SPCMDn. SSLKP bit	SPCR.BFDS bit	SSL delay control register*1 (RSPCK clock delay, SSL negation delay, next access delay)
1	0	0	Any given value. You can control each delay value according to setting for RSPCK clock delay, SSL negation delay and next access delay
2-1	1	0	
2-1	0	0	
3-1	1	1	Any given value. But delay is inserted only below. • RSPCK clock delay of the 1st frame • SSL negation delay and next access delay of the last frame
3-2	0	1	

Note 1. Whether the setting value of following bits are valid or not depends on the setting value of the SPCMD.SPNDEN bit. (See section 30.2.6. SPCMDm : SPI Command Register (m = 0 to 7).)
The SPDECR.SCKDL[2:0] bits: RSPCK delay
The SPDECR.SLNDL[2:0] bits: SSL negate delay
The SPDECR.SPNDL[2:0] bits: Next access delay

< Setting / operation example > (Motorola SPI, BFDS = 1 Case)

SPCMD0.SSLKP = 1 → Burst transfer / no interframe delay between 0 and 1 (SSL keep active)

SPCMD1.SSLKP = 1 → Burst transfer / no interframe delay between 1 and 2 (SSL keep active)

SPCMD2.SSLKP = 1 → Burst transfer / no interframe delay between 2 and 3 (SSL keep active))

SPCMD3.SSLKP = 1 → Burst transfer / no interframe delay between 3 and 4 (SSL keep active)

SPCMD4.SSLKP = 0 → do not Burst Transfer, and inactive SSL. (BFDS setting is invalid because it does not Burst Transfer.)

SPCMD5.SSLKP = 1 → Burst transfer / no interframe delay between 5 and 6 (SSL keep active)

SPCMD6.SSLKP = 1 → Burst transfer / no interframe delay between 6 and 7 (SSL keep active)

SPCMD7.SSLKP = 0 → do not Burst Transfer, and inactive SSL. (BFDS setting is invalid because it does not Burst Transfer.)

MODFEN bit (Mode Fault Error Detection Enable)

This bit is used to enable or disable detection of a mode fault error. (See section 30.3.10. Error Detection.) The SPI determines SSL0 pin input or output direction according to the combination of the MODFEN and MSTR bits. (See section 30.3.2. Controlling the SPI Pins.)

SPEIE bit (SPI Error Interrupt Enable)

This bit is used to enable or disable an SPI error interrupt request when the SPI detects a mode fault error or an underrun error and sets the MODF flag in the SPI status register (SPSR) to 1, when the SPI detects an overrun error and sets the OVRF flag in SPSR to 1, or when the SPI detects a parity error and sets the PERF flag in SPSR to 1. (See section 30.3.10. Error Detection)

SPRIE bit (SPI Receive Buffer Full Interrupt Enable)

This bit is used to enable or disable a receive buffer full interrupt request of the SPI.

SPIIE bit (SPI Idle Interrupt Enable)

This bit is used to enable or disable an idle interrupt request of the SPI after the SPI detects the idle state and sets the IDLNF flag in the SPI status register (SPSR) to 0.

SPDRES bit (SPI receive data ready error select)

When a receive data ready is detected (SPSR.SPDRF = 1), select whether to use SPIi_SPRI interrupt request or SPIi_SPEI interrupt request.

SPTIE bit (SPI Transmit Buffer Empty Interrupt Enable)

This bit is used to enable or disable a transmit buffer empty interrupt request of the SPI.

A transmit buffer empty interrupt request at the beginning of transmission is generated by setting the SPE bit to 1 simultaneously when or after the SPTIE bit is set to 1. Note that a transmit buffer empty interrupt is generated while the SPTIE bit is 1 even though SPI functions are disabled (SPE bit = 0).

CENDIE bit (SPI Communication End Interrupt Enable)

This bit controls generation of a communication end interrupt request.

SPMS bit (SPI Mode Select)

This bit is used to select SPI operation (4-wire) or clock synchronous operation (3-wire).

For clock synchronous operation, the SSL pin is not used but three pins RSPCK, MOSI, and MISO are used for communication. When SPMS = 1 (clock synchronous operation (3-wire)), the setting of the SPFRF bit is invalid.

To perform clock synchronous operation in master mode (SPCR.MSTR = 1), set the CPHA bit in the SPI command register (SPCMD) to 0 or 1. To perform clock synchronous operation in slave mode (SPCR.MSTR = 0), set the CPHA bit to 1. If this bit is set to 0 for clock synchronous operation in slave mode (SPCR.MSTR = 0), subsequent operation is not guaranteed.

The communication status according to the settings of the MSTR bit, TXMD[1:0] bits, SPFRF bit, and SPMS bit of the SPI control register (SPCR) as follows.

Table 30.4 SPI Communication Status (1 of 2)

SPCR.MSTR	SPCR.TXMD[1]	SPCR.TXMD[0]	SPCR.SPFRF	SPCR.SPMS	Communication Status	Communication Status No
1	0	0	0	0	Transmit-Receive Master / Motorola SPI / SPI operation (4-wire)	1-(1)
1	0	0	1	0	Transmit-Receive Master / TI-SSP / SPI operation (4-wire)	1-(2)
1	0	0	—	1	Transmit-Receive Master /Clock synchronous operation (3-wire)	1-(3)
1	0	1	0	0	Transmit only Master / Motorola SPI / SPI operation (4-wire)	1-(4)
1	0	1	1	0	Transmit only Master / TI-SSP / SPI operation (4-wire)	1-(5)
1	0	1	—	1	Transmit only Master /Clock synchronous operation (3-wire)	1-(6)
1	1	—	0	0	Receive only Master / Motorola SPI / SPI operation (4-wire)	1-(7)

Table 30.4 SPI Communication Status (2 of 2)

SPCR. MSTR	SPCR.TXMD[1]	SPCR.TXMD[0]	SPCR. SPFRF	SPCR. SPMS	Communication Status	Communication Status No
1	1	—	1	0	Receive only Master / TI-SSP / SPI operation (4-wire)	1-(8)
1	1	—	—	1	Receive only Master /Clock synchronous operation (3-wire)	1-(9)
0	0	0	0	0	Transmit-Receive Slave / Motorola SPI / SPI operation (4-wire) (default)	0-(1)
0	0	0	1	0	Transmit-Receive Slave / TI-SSP / SPI operation (4-wire)	0-(2)
0	0	0	—	1	Transmit-Receive Slave /Clock synchronous operation (3-wire)	0-(3)
0	0	1	0	0	Transmit only Slave / Motorola SPI / SPI operation (4-wire)	0-(4)
0	0	1	1	0	Transmit only Slave / TI-SSP / SPI operation (4-wire)	0-(5)
0	0	1	—	1	Transmit only Slave /Clock synchronous operation (3-wire)	0-(6)
0	1	—	0	0	Receive only Slave / Motorola SPI / SPI operation (4-wire)	0-(7)
0	1	—	1	0	Receive only Slave / TI-SSP / SPI operation (4-wire)	0-(8)
0	1	—	—	1	Receive only Slave /Clock synchronous operation (3-wire)	0-(9)

SPFRF bit (SPI Frame Format Select)

This bit selects the communication protocol.

The format of the SPI terminal (RSPCK, SSL0 to 7) can be set according to the set communication protocol.

When SPMS = 1 (clock synchronous operation (3-wire)), this bit is invalid because SSL is not used.

TXMD[1:0] bit (Communication Mode Select)

This bit is used to select the transmit-receive, transmit-only, and receive-only serial communication.

When TXMD[1:0] is set to 01 for communication, transmit-only is performed without reception.

When TXMD[1] is set to 1 for communication, receive-only is performed without transmission.

When TXMD[1:0] is set to 01 for communication, a receive buffer full interrupt request cannot be used.

When TXMD[1] is set to 1 for communication, a transmit buffer empty interrupt request cannot be used.

(See section 30.3.6. Communication Operating Mode.)

MSTR bit (SPI Master/Slave Mode Select)

This bit is used to select master mode or slave mode of the SPI. The SPI determines input/output directions of pins RSPCK, MOSI, MISO, and SSL1 to SSL3 according to the MSTR bit setting.

BPEN bit (Synchronization Circuit Bypass Enable)

This bit selects whether to enable or disable the synchronization bypass function. This bit can be used to bypass the synchronization circuit only when the same clock is input to the bus clock (PCLK) and operation clock (TCLK).

30.2.4 SPCR2 : SPI Control Register 2

Base address: SPI_Bn = 0x4011_A000 + 0x0100 × n (n = 0, 1)

Offset address: 0x0C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	MOIFE	MOIFV	—	—	SPLP2	SPLP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
4:0	RMFM[4:0]	Frame processing count setting in Master Receive only The number of received frames can be adjusted in master receive only. 0x00: This function is not used*1 0x01: Automatically stop communication after processing 1 received frame ⋮ 0x1F: Automatically stop communication after processing 31 received frames	R/W
5	—	This bit is read as 0. The write value should be 0.	R
6	RMEDTG	End Trigger in Master Receive only 1: Receive End (Writable only when Master Receive only) Reading value is always 0	W
7	RMSTTG	Start Trigger in Master Receive only 1: Receive Start (Writable only when Master Receive only) Reading value is always 0	W
15:8	SPDRC[7:0]	SPI received data ready detect adjustment 0x0: Disable receive data ready detection function 0x1: Performs reception data ready judgment after 1 TCLK ⋮ 0xFF Performs reception data ready judgment after 255 TCLK	R/W
16	SPLP	SPI Loopback 0: Normal mode 1: Loopback mode (inverted transmit data = receive data)	R/W
17	SPLP2	SPI Loopback 2 0: Normal mode 1: Loopback mode (transmit data = receive data)	R/W
19:18	—	These bits are read as 0. The write value should be 0.	R
20	MOIFV	MOSI Idle Fixed Value 0: The fixed value of MOSI idle = 0. 1: The fixed value of MOSI idle = 1.	R/W
21	MOIFE	MOSI Idle Fixed Value Enable 0: The MOSI output value is the last data of previous transfer. 1: The MOSI output value is the set MOIFV bit value.	R/W
31:22	—	These bits are read as 0. The write value should be 0.	R

Note 1. See SW flow in the Figure 30.66.

RMFM[4:0] bit (Frame processing count setting in Master Receive only)

The number of received frames can be adjusted when operating in master receive only. Valid only when the master mode (SPCR.MSTR = 1) and the communication operation mode select bits (SPCR.TXMD [1:0]) are 10b.

Only the start bit in master mode reception automatically stops communication after starts frame processing according to the value set in this bit after reception starts.

If the RMFM [4:0] bits are rewritten while the SPE bit of the SPI control register (SPCR) is 1, subsequent operations are not guaranteed.

RMEDTG bit (End Trigger in Master Receive only)

This bit is used to end reception when master receive only. Valid only when the master mode (SPCR.MSTR = 1) and the communication mode select bits (SPCR.TXMD [1:0]) are 10b.

RMSTTG bit (Start Trigger in Master Receive only)

This bit is used to start reception when master receive only. Valid only when the master mode (SPCR.MSTR = 1) and the communication mode select bits (SPCR.TXMD [1:0]) are 10b.

Writing 1 to this bit during reception is not accepted. Write again after reception is completed.

SPDRC[7:0] bit (SPI received data ready detect adjustment)

The receive data ready detection function can be disabled or, if used, the period until detection can be set from 1 to 255 TCLK.

The value set in the SPDRC [7:0] bits is used to 1 set the SPDRF flag. For details, see the description of SPDRF in section 30.2.9. SPSR : SPI Status Register.

If the set value is changed while the SPE bit is 1, subsequent operations are not guaranteed.

SPLP bit (SPI Loopback)

When the SPLP bit is set to 1, the SPI shuts down the route between the MISO pin and the shift register (when the MSTR bit in the SPI control register is 1) or shuts down the route between the MOSI pin and the shift register, inverts the input route value in the shift register, and then connects the route to the output route (when the MSTR bit in the SPI control register is 0) (loopback mode).

SPLP2 bit (SPI Loopback 2)

When the SPLP2 bit is set to 1, the SPI shuts down the route between the MISO pin and the shift register (when the MSTR bit in the SPI control register is 1) or shuts down the route between MOSI pin and the shift register and then connects the route to the output route without inverting the input route value in the shift register (when the MSTR bit in the SPI control register is 0) (loopback mode). If this bit is set to 1 together with the SPLP bit, setting this bit takes precedence.

MOIFV bit (MOSI Idle Fixed Value)

This bit is used to select the MOSI pin output value during the SSL negation period (including SSL retention period in burst transfer) when the MOIFE bit is 1 in master mode.

If this bit is modified with the SPE bit in the SPI control register (SPCR) set to 1, subsequent operation is not guaranteed.

MOIFE bit (MOSI Idle Fixed Value Enable)

This bit is used for the SPI in master mode to fix the MOSI output value during the SSL negation period (including SSL retention period in burst transfer). When MOIFE bit = 0, the SPI outputs the last data of the previous serial transfer to MOSI during the SSL negation period. When MOIFE bit = 1, the SPI outputs the fixed MOIFV bit value to MOSI.

If this bit is modified with the SPE bit in the SPI control register (SPCR) set to 1, subsequent operation is not guaranteed.

30.2.5 SPCR3 : SPI Control Register 3

Base address: SPI_Bn = 0x4011_A000 + 0x0100 × n (n = 0, 1)

Offset address: 0x10

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SPBR[7:0]								—	—	—	—	SSL3P	SSL2P	SSL1P	SSL0P
Value after reset:	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SSL0P	SSL0 Signal Polarity [In the Motorola-SPI case] 0: The SSL0 signal is active low (0). 1: The SSL0 signal is active high (1). [In the TI-SSP case] 0: The SSL0 signal is active high (1). 1: The SSL0 signal is active low (0).	R/W

Bit	Symbol	Function	R/W
1	SSL1P	SSL1 Signal Polarity [In the Motorola-SPI case] 0: The SSL1 signal is active low (0). 1: The SSL1 signal is active high (1). [In the TI-SSP case] 0: The SSL1 signal is active high (1). 1: The SSL1 signal is active low (0).	R/W
2	SSL2P	SSL2 Signal Polarity [In the Motorola-SPI case] 0: The SSL2 signal is active low (0). 1: The SSL2 signal is active high (1). [In the TI-SSP case] 0: The SSL2 signal is active high (1). 1: The SSL2 signal is active low (0).	R/W
3	SSL3P	SSL3 Signal Polarity [In the Motorola-SPI case] 0: The SSL3 signal is active low (0). 1: The SSL3 signal is active high (1). [In the TI-SSP case] 0: The SSL3 signal is active high (1). 1: The SSL3 signal is active low (0).	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R
15:8	SPBR[7:0]	SPI Bit Rate	R/W
23:16	—	These bits are read as 0. The write value should be 0.	R
26:24	SPSLN[2:0]	SPI Sequence Length Registers SPCMD0 to SPCMD7 to be referenced and reference sequence are changed according to the set sequence length. Relationship among this bit value, sequence length, and SPCMD0 to SPCMD7 to be referenced by the SPI are shown below. The SPI in slave mode always references SPCMD0. 0 0 0: Sequence Length is 1 (Referenced SPCMDn, n = 0→0→…) 0 0 1: Sequence Length is 2 (Referenced SPCMDn, n = 0→1→0→…) 0 1 0: Sequence Length is 3 (Referenced SPCMDn, n = 0→1→2→0→…) 0 1 1: Sequence Length is 4 (Referenced SPCMDn, n = 0→1→2→3→0→…) 1 0 0: Sequence Length is 5 (Referenced SPCMDn, n = 0→1→2→3→4→0→…) 1 0 1: Sequence Length is 6 (Referenced SPCMDn, n = 0→1→2→3→4→5→0→…) 1 1 0: Sequence Length is 7 (Referenced SPCMDn, n = 0→1→2→3→4→5→6→0→…) 1 1 1: Sequence Length is 8 (Referenced SPCMDn, n = 0→1→2→3→4→5→6→7→0→…)	R/W
31:27	—	These bits are read as 0. The write value should be 0.	R

SSLiP bits (SSL Signal Polarity Bits)

These bits are used to specify the polarity of SSL signals. The set SSLiP bit (i = 3 to 0) values indicate the active polarity of SSLi signals.

If any of these SSLiP bits is modified with the SPE bit in the SPI control register (SPCR) set to 1, subsequent operation is not guaranteed.

Note: SSL0 is different from SSL1-SSL3. When slave or multi-master, it functions as an input.

For details, see section 30.3.3.2. Single-master/single-slave with the MCU as a slave, and section 30.3.3.5. Multi-master/multi-slave with the MCU as a master.

SPBR[7:0] bit (SPI Bit Rate)

The SPI bit rate bits (SPBR) is used to set the bit rate in master mode. If SPBR is modified while the MSTR bit in the SPI control register (SPCR) is 1, subsequent operation is not guaranteed.

When the SPI is used in slave mode, the bit rate depends on the input clock bit rate regardless of the SPCMD.BRDV setting. (Specify a bit rate that meets electrical characteristics.)

The bit rate is determined by a combination of the set SPBR value and the set BRDV[1:0] bits value in the SPI command register (SPCMD0 to SPCMD7).

The bit rate is calculated by the following expression, where n is the set SPBR value (0 to 255) and N is the set BRDV[1:0] bits value (0 to 3).

$$\text{Bit rate} = \frac{f(\text{TCLK})}{2 \times (n+1) \times 2^N}$$

The following table shows an example of correspondence between bit rates and set values of SPBR and BRDV[1:0].

Table 30.5 Corresponding Between Bit Rates and Set Values (Example)

SPBR Value (n)	BRDV Value (N)	Division Ratio	Bit Rate				
			TCLK = 32 MHz	TCLK = 36 MHz	TCLK = 40 MHz	TCLK = 50 MHz	TCLK = 120MHz
0	0	2	16.0 Mbps	18.0 Mbps	20.0 Mbps	25.0 Mbps	60.0Mbps
1	0	4	8.00 Mbps	9.00 Mbps	10.0 Mbps	12.5 Mbps	30.0Mbps
2	0	6	5.33 Mbps	6.00 Mbps	6.67 Mbps	8.33 Mbps	20.0Mbps
3	0	8	4.00 Mbps	4.50 Mbps	5.00 Mbps	6.25 Mbps	15.0Mbps
4	0	10	3.20 Mbps	3.60 Mbps	4.00 Mbps	5.00 Mbps	12.0Mbps
5	0	12	2.67 Mbps	3.00 Mbps	3.33 Mbps	4.16 Mbps	10.0Mbps
5	1	24	1.33 Mbps	1.50 Mbps	1.67 Mbps	2.08 Mbps	5.0Mbps
5	2	48	677 kbps	750 kbps	833 kbps	1.04 Mbps	2.5Mbps
5	3	96	333 kbps	375 kbps	417 kbps	521 kbps	1.25Mbps
255	3	4096	7.81 kbps	8.80 kbps	9.78 kbps	12.2 kbps	29.3kbps

SPSLN[2:0] bit (SPI Sequence Length)

These bits are used to set the sequence length for the SPI in master mode to perform sequence operation. According to the sequence length specified by SPSLN[2:0] bits, the SPI in master mode changes SPI command registers 0 to 7 (SPCMD0 to SPCMD7) to be referenced and the reference sequence. For details, see section 30.3.13.1. Master mode operation.

The SPI in slave mode always references SPCMD0.

30.2.6 SPCMDm : SPI Command Register (m = 0 to 7)

Base address: SPI_Bn = 0x4011_A000 + 0x0100 × n (n = 0, 1)

Offset address: 0x14 + 0x04 × m

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	SSLA[2:0]			—	—	—	SPB[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SCKDEN	SLNDEN	SPNDEN	LSBF	—	—	—	—	SSLKP	—	—	—	BRDV[1:0]		CPOL	CPHA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CPHA	RSPCK Phase 0: Data is sampled at an odd edge and changes at an even edge. 1: Data changes at an odd edge and is sampled at an even edge.	R/W
1	CPOL	RSPCK Polarity 0: RSPCK in idle state is 0. 1: RSPCK in idle state is 1.	R/W

Bit	Symbol	Function	R/W
3:2	BRDV[1:0]	Bit Rate Division 0 0: The base bit rate is selected. 0 1: Two-divided base bit rate is selected. 1 0: Four-divided base bit rate is selected. 1 1: Eight-divided base bit rate is selected.	R/W
6:4	—	These bits are read as 0. The write value should be 0.	R
7	SSLKP	SSL Signal Level Hold 0: All SSL signals are negated at the end of transfer. 1: SSL signal level is held after the transfer ends until the next access starts.	R/W
11:8	—	These bits are read as 0. The write value should be 0.	R
12	LSBF	SPI LSB First 0: MSB first 1: LSB first	R/W
13	SPNDEN	SPI Next-Access Delay Enable 0: Next-access delay is 1RSPCK + 5TCLK 1: Next-access delay is the set value of the SPI next-access delay register (SPDECR.SPNDL).	R/W
14	SLNDEN	SSL Negation Delay Setting Enable 0: [Master] SSL negation delay is 1RSPCK. [Slave in the TI-SSP] SSL negation delay is 1TCLK 1: SSL negation delay is the set value of the slave select negation delay register (SPDECR.SLNDL).	R/W
15	SCKDEN	RSPCK Delay Setting Enable [In the Motorola-SPI case] 0: RSPCK delay is 1 RSPCK. 1: RSPCK delay is the set value of the RSPCK delay register (SPDECR.SCKDL). [In the TI-SSP case] 0: RSPCK delay is 0 RSPCK. 1: RSPCK delay is the set value of the RSPCK delay register (SPDECR.SCKDL).	R/W
20:16	SPB[4:0]	SPI Data Length 0x00 to 0x02: Setting prohibited 0x03: 4bit 0x04: 5bit 0x05: 6bit ⋮ 0x1E: 31bit 0x1F: 32bit	R/W
23:21	—	These bits are read as 0. The write value should be 0.	R
26:24	SSLA[2:0]	SSL Signal Assertion 0 0 0: SSL0 0 0 1: SSL1 0 1 0: SSL2 0 1 1: SSL3 Others: Setting prohibited	R/W
31:27	—	These bits are read as 0. The write value should be 0.	R

The SPI has eight SPI command registers (SPCMD0 to SPCMD7) that are used to set the transfer format of the SPI in master mode. Furthermore, some bits in SPCMD0 are used to set the transfer format of the SPI in slave mode. The SPI in master mode sequentially references SPCMD0 to SPCMD7 according to the setting of the SPSLN[2:0] bits in the SPI Control register 3 (SPCR3), and then performs serial transfer specified in the referenced SPCMD.

SPI set the SPCMD register before setting data to be transmitted by referencing the SPCMD when the transmit buffer is empty (while the next-transfer data has not been set).

The SPCMD referenced by the SPI in master mode is indicated by SPCP[2:0] in the SPI status register (SPSR). If SPCMD0 is modified while the SPI in slave mode is enabled (SPCR.SPE = 1), subsequent operation is not guaranteed.

CPHA bit (RSPCK Phase)

This bit is used to set the RSPCK phase of the SPI in master mode or slave mode. To perform data communication between SPI modules, the same RSPCK phase must be set for both modules.

When SPCR.SPMS = 0 and SPCR.SPFRF = 1 (in TI SSP mode), setting CPHA = 0 is invalid.

CPOL bit (RSPCK Polarity)

This bit is used to set the RSPCK polarity of the SPI in master mode or slave mode. To perform data communication between SPI modules, the same RSPCK polarity must be set for both modules.

BRDV[1:0] bit (Bit Rate Division)

This register is used to determine the bit rate with a combination of the set values of the BRDV[1:0] bits and the SPI bit rate register (SPCR3.SPBR). The set SPBR value determines the base bit rate. The set BRDV[1:0] bits value is used to select undivided, 2-divided, 4-divided, or 8-divided base bit rate. SPCMD0 to SPCMD7 enable setting of different BRDV[1:0] values. This makes it possible to perform serial transfer with a different bit rate for each command.

SSLKP bit (SSL Signal Level Hold)

This bit is used to set whether to hold or negate the SSL signal level of the current command during a period from the SSL negation timing for the current command to the SSL assertion timing for the next command when the SPI in master mode performs serial transfer. Setting this bit to 1 enables burst transfer in SPI operation master mode. For details, see section 30.3.12.1. Master mode operation.

To use the SPI in slave mode, set SSLKP bit to 0.

LSBF bit (SPI LSB First)

This bit is used to set the data format of the SPI in master mode or slave mode to MSB first or LSB first.

SPNDEN bit (SPI Next-Access Delay Enable)

This bit is used to set the period (next-access delay) after the SPI in master mode inactivates the SSL signal at the end of serial transfer until it enables SSL signal assertion of the next access. When SPNDEN bit = 0, the SPI sets the next-access delay to 1 RSPCK + 5TCLK. When SPNDEN bit = 1, the SPI inserts the next-access delay in accordance with the SPI next-access delay register (SPDECR.SPNDL) setting.

To use the SPI in slave mode, set SPNDEN bit to 0.

SLNDEN bit (SSL Negation Delay Setting Enable)

[In the Motorola-SPI case]

This bit is used to set the period (SSL negation delay) after the SPI in master mode stops RSPCK oscillation until it inactivates the SSL signal. When SLNDEN bit = 0, the SPI sets the SSL negation delay to 1 RSPCK. When SLNDEN bit = 1, the SPI negates the SSL signal with the RSPCK delay in accordance with the slave select negation delay register (SPDECR.SLNDL) setting.

To use the SPI in slave mode, set SLNDEN bit to 0.

[In the TI-SSP case]

This bit is used to set the period from when the master mode SPI stops RSPCK oscillation to when the OE signal is inactivated, or when the slave mode SPI detects the last edge of RSPCK and then negates the OE signal. When the SLNDEN bit is 0, the SSL negate delay is 1 RSPCK in master mode and 1 TCLK in slave mode. When SLNDEN bit = 1, the SPI negates the SSL signal with the RSPCK delay in accordance with the slave select negation delay register (SPDECR.SLNDL) setting.

When using SPI in slave mode except TI SSP setting, set the SLNDEN bit to 0.

SCKDEN bit (RSPCK Delay Setting Enable)

[In the Motorola-SPI case]

This bit is used to set the period (RSPCK delay) after the SPI in master mode activates the SSL signal until it oscillates RSPCK. When SCKDEN bit = 0, the SPI sets the RSPCK delay to 1 RSPCK. When SCKDEN bit = 1, the SPI starts RSPCK oscillation with the RSPCK delay in accordance with the RSPCK delay register (SPDECR.SCKDL) setting.

To use the SPI in slave mode, set SCKDEN bit to 0.

[In the TI-SSP case]

This bit is used to set the period from the start of assertion of the SSL signal to the RSPCK oscillation (RSPCK delay) and the period of the SSL signal to negation by the SPI in master mode. When SCKDEN bit = 0, the SPI does not sets the RSPCK delay. When SCKDEN bit = 1, the SPI starts RSPCK oscillation with the RSPCK delay in accordance with the RSPCK delay register (SPDECR.SPCKDL) setting.

To use the SPI in slave mode, set SCKDEN bit to 0.

SPB[4:0] bit (SPI Data Length)

These bits are used to set the transfer data length of the SPI in master mode or slave mode.

SSLA[2:0] bit (SSL Signal Assertion)

These bits are used to control SSL signal assertion for the SPI in master mode to perform serial transfer. The set SSLA[2:0] bits value controls assertion of the SSL3 to SSL0 signals. The signal polarity when the SSL signal is asserted depends on the set value of the SPI slave select polarity register (SPCR3.SSLiP). When SSLA[2:0] bits are set to 000b in multi-master mode, serial transfer is performed with all SSL signals negated (because SSL0 is input).

To use the SPI in slave mode, set SSLA[2:0] bits to 000b.

30.2.7 SPDCR : SPI Data Control Register

Base address: SPI_Bn = 0x4011_A000 + 0x0100 × n (n = 0, 1)

Offset address: 0x40

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

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

Bit	Symbol	Function	R/W
0	BYSW	Byte Swap Operating Mode Select 0: Byte Swap OFF 1: Byte Swap ON	R/W
2:1	—	These bits are read as 0. The write value should be 0.	R/W
3	SPRDTD	SPI Receive Data or Transmit Data Select 0: The SPDR reads the receive buffer. 1: The SPDR reads the transmit buffer	R/W
4	SINV	Serial data invert bit 0: Not invert serial data 1: Invert serial data.	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R
9:8	SPFC[1:0]	Frame Count 0 0: 1 frame 0 1: 2 frame 1 0: 3 frame 1 1: 4 frame	R/W
31:10	—	These bits are read as 0. The write value should be 0.	R

The SPI data control register (SPDCR) controls the data format.

If the value set in this register is changed while the SPE bit is 1, subsequent operations are not guaranteed.

BYSW bit (Byte Swap Operating Mode Select)

It is a setting bit, that is to swap a transmit/receive data in byte units. A data after byte swap is different by a data length (setting of SPCMD.SPB[4:0]).

When byte swap, A data length (setting of SPB[4:0]) must be set to 32bit or 16bit. Other case of data length (i.e. 4 to 15, 17 to 31-bit length), byte swap is not guaranteed. For the arrangement of data before and after swapping data lengths of 32 bits and 16 bits, see section 30.3.4.3. Byte Swap Transmission and section 30.3.4.4. Byte Swap Reception.

When the parity function set to valid, the behavior is not guaranteed.

SPRDTD bit (SPI Receive Data or Transmit Data Select)

This bit is used to select receive buffer or transmit buffer from which the SPI data register (SPDR) value is read.

When the transmit buffer is read, the value that was written to SPDR immediately before is read.

SINV bit (Serial data invert bit)

This bit is used to invert transmit data and receive data.

When the SINV bit is set to 1, transmit buffer (SPTX) data is inverted to invert transmit data and receive data, and then the inverted data is stored in the receive buffer (SPRX). The parity bit is the value corresponding to the inverted transmission/ reception data.

SPFC[1:0] bit (Frame Count)

Used for the condition to set the CENDF flag in slave receive only mode.

For details on the CENDF flag setting conditions, see section 30.2.9. SPSR : SPI Status Register.

Note that this bit is invalid except in the slave receive only mode.

30.2.8 SPDCR2 : SPI Data Control Register 2

Base address: SPI_Bn = 0x4011_A000 + 0x0100 × n (n = 0, 1)

Offset address: 0x44

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

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

Bit	Symbol	Function	R/W
1:0	RTRG[1:0]	Receive FIFO threshold setting 0 0: threshold 0 0 1: threshold 1 1 0: threshold 2 1 1: threshold 3	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R
9:8	TTRG[1:0]	Transmission FIFO threshold setting 0 0: threshold 0 0 1: threshold 1 1 0: threshold 2 1 1: threshold 3	R/W
31:10	—	These bits are read as 0. The write value should be 0.	R/W

SPI data control register 2 (SPDCR2) controls the FIFO threshold. If the value set in this register is changed while the SPE bit is 1, subsequent operations are not guaranteed.

RTRG[1:0] bit (Receive FIFO threshold setting)

Set the receive FIFO threshold.

When the number of data stored in the receive FIFO > the number of frames set by RTRG[1:0], the receive buffer full flag is set.

TTRG[1:0] bit (Transmission FIFO threshold setting)

Set the transmit FIFO threshold.

When the number of empty stages in the transmit FIFO > the number of frames set in TTRG[1:0], the transmit buffer empty flag is set.

30.2.9 SPSR : SPI Status Register

Base address: SPI_Bn = 0x4011_A000 + 0x0100 × n (n = 0, 1)

Offset address: 0x50

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	SPRF	CENDF	SPTEF	UDRF	PERF	MODF	IDLNF	OVRF	SPDRF	—	—	—	—	—	—	—
Value after reset:	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0. The write value should be 0.	R
10:8	SPCP[2:0]	SPI Command Pointer 0 0 0: SPCMD0 0 0 1: SPCMD1 0 1 0: SPCMD2 0 1 1: SPCMD3 1 0 0: SPCMD4 1 0 1: SPCMD5 1 1 0: SPCMD6 1 1 1: SPCMD7	R
11	—	This bit is read as 0. The write value should be 0.	R
14:12	SPECM[2:0]	SPI Error Command 0 0 0: SPCMD0 0 0 1: SPCMD1 0 1 0: SPCMD2 0 1 1: SPCMD3 1 0 0: SPCMD4 1 0 1: SPCMD5 1 1 0: SPCMD6 1 1 1: SPCMD7	R
22:15	—	These bits are read as 0. The write value should be 0.	R
23	SPDRF	SPI Receive Data Ready Flag 0: Receive data ready not detected 1: Receive data ready detected	R
24	OVRF	Overrun Error Flag 0: No overrun error is present. 1: An overrun error is present.	R
25	IDLNF	SPI Idle Flag 0: The SPI is in the idle state. 1: The SPI is in the transfer state.	R

Bit	Symbol	Function	R/W
26	MODF	Mode Fault Error Flag 0: Neither mode fault error nor underrun error is present. 1: A mode fault error or underrun error is present.	R
27	PERF	Parity Error Flag 0: No parity error is present. 1: A parity error is present.	R
28	UDRF	Underrun Error Flag This bit indicates error status in combination with the MODF flag. 0: When MODF=0, neither mode fault error nor underrun error is present. When MODF=1, a mode fault error is present. 1: When MODF=0, neither mode fault error nor underrun error is present. When MODF=1, an underrun error is present.	R
29	SPTEF	SPI Transmit Buffer Empty Flag 0: The number of empty stages in the transmit FIFO ≤ the value set in SPDCR2.TTRG 1: The number of empty stages in the transmit FIFO > the value set in SPDCR2.TTRG	R
30	CENDF	Communication End Flag 0: The SPI is not communicating or communicating. 1: The SPI communication completed.	R
31	SPRF	SPI Receive Buffer Full Flag 0: The number of data stored in the receive FIFO ≤ number of frames set by the SPDCR2.RTRG bit. 1: The number of data stored in the receive FIFO > number of frames set by the SPDCR2.RTRG bit.	R

The SPI status register (SPSR) stores flags that indicate SPI's operating status.

SPCP[2:0] bit (SPI Command Pointer)

These bits indicate SPI command registers 0 to 7 (SPCMD0 to SPCMD7) indicated by the current pointer in the SPI sequence control. For details about the SPI sequence control, see section 30.3.13.1. Master mode operation.

SPECM[2:0] bit (SPI Error Command)

These bits indicate SPI command registers 0 to 7 (SPCMD0 to SPCMD7) indicated by the command pointer (SPCP[2:0] bits) when an error was detected in the SPI sequence control. The SPI updates the SPECM[2:0] bits value only when an error is detected. When no error is present (OVRF, MODF, and PERF flags in SPSR are 0), the SPECM[2:0] bits value has no meaning. For the SPI's error detection function, see section 30.3.10. Error Detection. For the SPI's sequence control, see section 30.3.13.1. Master mode operation.

SPDRF bit (SPI Receive Data Ready Flag)

During communication (SPCR.SPE = 1), a certain period of time has elapsed while the number of data stored in the reception FIFO ≤ the reception FIFO threshold.

This bit is set to 0 when the reception operation is not performed (SPCR.TXMD[1:0] = 01b).

[Setting (to 1) condition]

All the following two conditions are met.

- SPCR2.SPDRC[7:0] ≠ 0x00.
- After the receive FIFO has been written, when the number of data stored in the receive FIFO ≤ the receive FIFO threshold and the value set by SPDRC[7:0] has elapsed

[Clearing (to 0) condition]

- When 1 is written to the SPSRC.SPDRFC bit.

OVRF bit (Overrun Error Flag)

This flag indicates whether an overrun error is present. When the RSPCK clock auto-stop function is enabled (SPCR.SCKASE = 1) in master mode (SPCR.MSTR = 1), no overrun error occurs and, therefore, this flag is not set to 1. For details, see section 30.3.10.1. Overrun errors.

[Setting (to 1) condition]

When serial transfer is completed in one of the following two conditions with data stored in the receive FIFO for the number of FIFO stages.

- SPCR.TXMD[1:0] = 00b. (transmit-receive mode)
- SPCR.TXMD[1:0] = 10b. (receive only)

[Clearing (to 0) condition]

- When 1 is written to the SPSRC.OVRFC bit.

IDLNF bit (SPI Idle Flag)

This flag indicates transfer status of the SPI.

[Setting (to 1) condition]

[Transmit-Receive, Transmit-only Master mode]

- None of the clearing (to 0) conditions (Transmit-Receive / Transmit-only in Master mode) below is met.

[Receive-only Master mode]

- When 1 is written to RMSTTG of SPCR2.

[Slave mode]

- The SPE bit in SPCR is 1 (SPI function enabled).

[Clearing (to 0) conditions]

Communication status: 1-(1) to (6) * For details of communication status, see Table 30.4.

[Transmit-Receive, Transmit-only Master mode]

Any of the following two conditions is met.

- The SPE bit in SPCR is 0 (SPI initialization).
- All the following three conditions are met.
  - The next transfer data is not set in the transmission buffer (SPTXn, n = 0 to 3)
  - The SPCP bits in SPSR are 000b (at the beginning of sequence control).
  - The operation completed by the next access delay (the master main state machine has transitioned to the idle state)

[Receive-only Master mode]

Communication status: 1-(7) to (9)

Any of the following two conditions is met.

- The SPE bit in SPCR is 0 (SPI initialization).
- Any of the following 3 conditions is met.
  When RMFM[4:0] = 0x0, after writing 1 to RMEDTG, the operation completed by the next access delay (the master main state machine has transitioned to the idle state)
  When RMFM[4:0] ≠ 0x0, after writing 1 to RMEDTG, the operation completed by the next access delay (the master main state machine has transitioned to the idle state)
  When RMFM[4:0] ≠ 0x0, the operation completed by the next access delay after processing is completed for the number of received frames set in RMFM[4:0] (the master main state machine has transitioned to the idle state)

[Slave mode]

Communication status: 0-(1) to (9)

- The SPE bit in SPCR is 0 (SPI initialization).

MODF bit (Mode Fault Error Flag)

This flag indicates whether a mode fault error or an underrun error is present. The UDRF flag allows you to see which error (mode fault error or underrun error) has occurred.

[Setting (to 1) condition]

[Multi-master mode]

- The SSL0 pin input level becomes active level while the SPCR.MSTR bit = 1 (master mode) and the SPCR.MODFEN bit = 1 (mode fault error detection enabled), and then the SPI has detected a mode fault error.

[Slave, Motorola-SPI mode]

Any of the following two conditions is met.

- The SSL0 pin is negated before the RSPCK cycles necessary for data transfer end while the SPCR.MSTR bit = 0 (slave mode), SPCR.SPFRF bit = 0 (Motorola-SPI) and the SPCR.MODFEN bit = 1 (mode fault error detection enabled), and then the SPI has detected a mode fault error.
- Serial transfer is started before transmit data output becomes ready while the SPCR.SPE bit = 1 (SPI function enabled), and then the SPI has detected an underrun error.

[Slave, TI-SSP mode]

Any of the following two conditions is met.

- The SSL0 pin is asserted before the RSPCK cycles necessary for data transfer end while the SPCR.MSTR bit = 0 (slave mode), SPCR.SPFRF bit = 1 (TI-SSP) and the SPCR.MODFEN bit = 1 (mode fault error detection enabled), and then the SPI has detected a mode fault error.
- Serial transfer is started before transmit data output becomes ready while the SPCR.SPE bit = 1 (SPI function enabled), and then the SPI has detected an underrun error.

The SSL0 signal active level depends on the SPCR3.SSLiP bits (SSL signal polarity bits).

[Clearing (to 0) condition]

- When 1 is written to the SPSRC.MODFC bit.

PERF bit (Parity Error Flag)

This flag indicates whether a parity error is present.

[Setting (to 1) condition]

When the serial transfer ends and a parity error is detected with the SPPE bit of SPCR set to 1 under any of the following 2 conditions.

- SPCR.TXMD[1:0] = 00b. (transmit-receive master mode or transmit-receive slave mode)
- SPCR.TXMD[1:0] = 10b. (receive-only master mode or receive-only slave mode)

[Clearing (to 0) condition]

- When 1 is written to the SPSRC.PERFC bit.

UDRF bit (Underrun Error Flag)

This flag indicates that a mode fault error or an underrun error is present.

[Setting (to 1) condition]

- Serial transfer is started before transmit data output becomes ready while the SPCR.MSTR bit = 0 and the SPCR.TXMD[1:0] bit = 00b or 01b (transmit-receive slave mode or transmit-only slave mode) and the SPCR.SPE bit = 1 (SPI function enabled), and then the SPI has detected an underrun error.

[Clearing (to 0) condition]

- When 1 is written to the SPSRC.UDRFC bit.

SPTEF bit (SPI Transmit Buffer Empty Flag)

This flag indicates the transmit buffer (SPTX) status in the SPI data register (SPDR).

[Setting (to 1) condition]

Any of the following 3 conditions is met.

- The SPE bit is set to 0 (SPI initialization).
- When the number of empty transmission FIFO stages > the value set in SPDCR2.TTRG[1:0].
- When 1 is written to SPFCR.SPFRST.

[Clearing (to 0) condition]

Any of the following two conditions is met.

- At the time of final access when transmission data is written to SPDR (SPTXn, n = 0 to 3) in one processing routine using DTC / DMAC.
- When 1 is written to the SPSRC.SPTEFC bit.

Writing a value to the SPDR register is enabled only while the SPTEF flag = 1. If a value is written to the SPDR register while the SPTEF flag = 0, transmit buffer data is not updated.

CENDF bit (Communication End Flag)

This flag indicates communication end status of SPI. It turns 1 at communication end and turns 0 at starting next communication.

[Setting (to 1) condition]

Transmit-Receive / Transmit-only Master mode

Communication status: 1-(1) to (6) * For details of communication status, see Table 30.4.

The following 3 conditions are met.

- The next transfer data is not set in the transmission buffer (SPTXn, n = 0 to 3)
- The SPSR.SPCP[2:0] are 000b. (It means the head of the sequential control.)
- The operation completed by the next access delay (the master main state machine has transitioned to the idle state)

Receive-only Master mode

Communication status: 1-(7) to (9)

Any of the following 3 conditions is met.

- When RMFM[4:0] = 0x0, after writing 1 to RMEDTG, the operation completed by the next access delay (the master main state machine has transitioned to the idle state)
- When RMFM[4:0] ≠ 0x0, after writing 1 to RMEDTG, the operation completed by the next access delay (the master main state machine has transitioned to the idle state)
- When RMFM[4:0] ≠ 0x0, the operation completed by the next access delay after processing is completed for the number of received frames set in RMFM[4:0] (the master main state machine has transitioned to the idle state)

Transmit-receive / transmit-only slave, Motorola-SPI mode @ SPI serial communication (4-wire: the SPCR.SPMS bit is 0)

Communication status: 0-(1), (4)

The following 3 conditions are met.

- The next transfer data is not set in the transmission buffer
- The transmission shift register is empty. (It means SPI does not do serial transfer.)
- SSL0 was negated.

Transmit-receive / transmit-only slave, TI-SSP mode @ SPI serial communication (4-wire: the SPCR.SPMS bit is 0)

Communication status: 0-(2), (5)

The following 3 conditions are met.

- The next transfer data is not set in the transmission buffer
- The transmission shift register is empty. (It means SPI does not do serial transfer.)
- When the SSL negate delay is completed.

Transmit-receive / transmit only slave mode @ clock synchronous (3-wire: the SPCR.SPMS bit is 1)

Communication status: 0-(3), (6)

The following 3 conditions are met.

- The next transfer data is not set in the transmission buffer
- The transmission shift register is empty. (It means SPI does not do serial transfer.)
- The last even edge of RSPCK of the frame was detected. (When the SPCMD.CPHA bit is 1.)

Receive only slave, Motorola-SPI mode @ SPI serial communication (4-wire: the SPCR.SPMS bit is 0)

Communication status: 0-(7)

The following condition is met.

- SSL0 input was negated after getting frames for SPDCR.SPFC set value in the receive buffer.

Receive only slave, TI-SSP mode @ SPI serial communication (4-wire: the SPCR.SPMS bit is 0)

Communication status: 0-(8)

The following condition is met.

- SSL0 negate delay is completed after getting frames for SPDCR.SPFC set value in the receive buffer.

Receive only slave mode @ clock synchronous (3-wire: the SPCR.SPMS bit is 1)

Communication status: 0-(9)

The following condition is met.

- The last even edge of RSPCK of the Last frame received for SPFC sets value. (When the SPCMD.CPHA bit is 1.)

[Clearing (to 0) condition]

Transmit-Receive / Transmit-only Master mode

Communication status: 1-(1) to (6)

Any of the following 2 conditions is met.

- The next transmit data was written to the transmit buffer (SPTX).
- When 1 is written to the SPSRC.CENDFC bit.

Receive -only Master mode

Communication status: 1-(7) to (9)

Any of the following 2 conditions is met.

- When 1 is written to the SPCR2.RMSTTG bit with SPE = 1.
- When 1 is written to the SPSRC.CENDFC bit.

Transmit-receive / transmit only slave mode

Communication status: 0-(1) to (6)

Satisfy one of following.

- The next transmit data was written to the transmit buffer (SPTX).
- When 1 is written to the SPSRC.CENDFC bit.

Receive only slave mode @ SPI serial communication (4-wire: the SPCR.SPMS bit is 0)

Communication status: 0-(7) to (8)

Satisfy one of following.

- SSL0 assertion of next data was detected.
- When 1 is written to the SPSRC.CENDFC bit.

Receive only slave mode @ clock synchronous (3-wire: the SPCR.SPMS bit is 1)

Communication status: 0-(9)

Satisfy one of following.

- The first edge of RSPCK of the next data was detected.
- When 1 is written to the SPSRC.CENDFC bit.

SPRF bit (SPI Receive Buffer Full Flag)

This flag indicates the receive buffer (SPRX) status in the SPI data register (SPDR).

[Setting (to 1) condition]

When the number of data stored in the receive FIFO > the number of frames set in the SPDCR2.RTRG[1:0] bits in Transmit-Receive, receive-only mode. However, the SPRF flag does not change from 0 to 1 while the OVRF flag = 1. (See section 30.3.10. Error Detection.)

[Clearing (to 0) condition]

Any of the following 3 conditions is met.

- At the last access when read data is read from SPDR (SPRXn, n = 0 to 3) in one processing routine using DTC / DMAC
- When 1 is written to the SPSRC.SPRFC bit
- When 1 is written to the SPFCR.SPFRST bit

30.2.10 SPTFSR : SPI Transfer FIFO Status Register

Base address: SPI_Bn = 0x4011_A000 + 0x0100 × n (n = 0, 1)

Offset address: 0x58

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

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

Bit	Symbol	Function	R/W
2:0	TFDN[2:0]	Transmit FIFO data empty stage number 0 0 0: Number of empty stages 0 ⋮ 1 0 0: Number of empty stages 4	R
31:3	—	These bits are read as 0. The write value should be 0.	R

TFDN[2:0] bit (Transmit FIFO data empty stage number)

Displays the number of empty transmission FIFO stages. By clearing the SPCR.SPE bit, TFDN[2:0] will be the initial value after reset (= all empty).

30.2.11 SPRFSR : SPI Receive FIFO Status Register

Base address: SPI_Bn = 0x4011_A000 + 0x0100 × n (n = 0, 1)

Offset address: 0x5C

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

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

Bit	Symbol	Function	R/W
2:0	RFDN[2:0]	Receive FIFO data store stage number 0 0 0: Number of store stages 0 ⋮ 1 0 0: Number of store stages 4	R
31:3	—	These bits are read as 0. The write value should be 0.	R

RFDN[2:0] bit (Receive FIFO data store stage number)

Displays the number of stores receive FIFO stages. RFDN [2:0] is cleared by clearing the SPCR.SPE bit.

30.2.12 SPPSR : SPI Polling Register

Base address: SPI_Bn = 0x4011_A000 + 0x0100 × n (n = 0, 1)

Offset address: 0x60

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

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

Bit	Symbol	Function	R/W
0	SPEPS	SPI Polling Status 0: SPCR.SPE is 0 1: SPCR.SPE is 1	R
31:1	—	These bits are read as 0. The write value should be 0.	R

SPEPS bit (SPI Polling Status)

This bit selects whether to enable or disable the synchronization bypass function. This bit can be used to bypass the synchronization circuit only when the same clock is input to the bus clock (PCLK) and operation clock (TCLK).

30.2.13 SPSRC : SPI Status Clear Register

Base address: SPI_Bn = 0x4011_A000 + 0x0100 × n (n = 0, 1)

Offset address: 0x68

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	SPRFC	CENDFC	SPTEFC	UDRFC	PERFC	MODFC	—	OVRFC	SPDRFC	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
22:0	—	These bits are read as 0. The write value should be 0.	R
23	SPDRFC	SPI Receive Data Ready Flag Clear The SPI receive data ready flag can be cleared by writing 1. When read, 0 is read.	W
24	OVRFC	Overrun Error Flag Clear By writing 1, the Overrun Error Flag can be cleared. Reading value is always 0.	W
25	—	This bit is read as 0. The write value should be 0.	R
26	MODFC	Mode Fault Error Flag Clear By writing 1, the Mode Fault Error Flag can be cleared. Reading value is always 0.	W*1
27	PERFC	Parity Error Flag Clear By writing 1, the Parity Error Flag can be cleared. Reading value is always 0.	W
28	UDRFC	Underrun Error Flag Clear By writing 1, the Underrun Error Flag can be cleared. Reading value is always 0.	W*2
29	SPTEFC	SPI Transmit Buffer Empty Flag Clear By writing 1, the SPI Transmit Buffer Empty Flag can be cleared. Reading value is always 0.	W
30	CENDFC	Communication End Flag Clear By writing 1, the Communication End Flag can be cleared. Reading value is always 0.	W
31	SPRFC	SPI Receive Buffer Full Flag Clear By writing 1, the SPI Receive Buffer Full Flag can be cleared. Reading value is always 0.	W

Note 1. Before setting MODFC and UDRFC, make sure that SPSR.MODF and UDRF are set to 1.
Note 2. When clearing the UDRF flag, clear the MODF flag at the same time (MODFC = 1).

The SPI status clear register (SPSRC) is a register that clears the status flag (SPSR) that indicates the operating status of SPI.

30.2.14 SPFCR : SPI FIFO Clear Register

Base address: SPI_Bn = 0x4011_A000 + 0x0100 × n (n = 0, 1)

Offset address: 0x6C

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

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

Bit	Symbol	Function	R/W
0	SPFRST	SPI FIFO clear By writing 1, the pointer in the FIFO and the stored data are initialized. Reading value is always 0.	W
31:1	—	These bits are read as 0. The write value should be 0.	R

The FIFO clear register (SPFCR) is used to clear the FIFO.

If SPFCR is rewritten while the SPE bit of the SPI control register (SPCR) is 1, subsequent operations are not guaranteed.

SPFRST bit (SPI FIFO clear)

Initializing the pointer and stored data in the transmit / receive FIFO by writing 1

30.3 Operation

In this section, the serial transfer period refers to the period from the beginning of driving valid data to the fetching of the final valid data.

30.3.1 Overview of SPI Operation

The SPI is capable of synchronous serial transfers in the following modes:

- Slave mode (SPI operation)
- Single master mode (SPI operation)
- Multi-master mode (SPI operation)
- Slave mode (clock synchronous operation)
- Master mode (clock synchronous operation)

The SPI mode can be selected by using the MSTR, MODFEN, SPMS, and SPFRF bits in SPCR. Table 30.6 lists the relationship between SPI modes and SPCR settings, and a description of each mode.

Table 30.6 Relationship between SPCR settings and SPI modes (1 of 2)

Mode	Slave (SPI operation)	Single-master (SPI operation)	Multi-master (SPI operation)	Slave (clock synchronous operation)	Master (clock synchronous operation)
MSTR bit setting	0	1	1	0	1
MODFEN bit setting	0 or 1	0	1	0	0
SPMS bit setting	0	0	0	1	1
SPFRF bit setting	valid	valid	valid	Invalid	Invalid
RSPCKn pins	Input	Output	Output/Hi-Z	Input	Output
MOSIn pin	Input	Output	Output/Hi-Z	Input	Output
MISOn pin	Output/Hi-Z	Input	Input	Output	Input
SSLn0 pins	Input	Output	Input	Hi-Z*1	Hi-Z*1
SSLn1 to SSLn3 pins	Hi-Z*1	Output	Output/Hi-Z	Hi-Z*1	Hi-Z*1
SSL polarity change function	Supported	Supported	Supported	—	—
Max transfer rate	TCLK/2	TCLK/2	TCLK/2	TCLK/2	TCLK/2
Clock source	RSPCK input	On-chip baud rate generator	On-chip baud rate generator	RSPCK input	On-chip baud rate generator
Clock polarity	Two				
Clock phase	Two*6	Two*6	Two*6	One (CPHA = 1)	Two
Transfer data length	4 to 32 bits				
Burst transfer	Possible (CPHA = 1)	Possible (CPHA = 0, 1)	Possible (CPHA = 0, 1)	—	—

Table 30.6 Relationship between SPCR settings and SPI modes (2 of 2)

Mode	Slave (SPI operation)	Single-master (SPI operation)	Multi-master (SPI operation)	Slave (clock synchronous operation)	Master (clock synchronous operation)
RSPCK delay control	Not supported	Supported	Supported	Not supported	Supported
SSL negation delay control	Not supported*7	Supported	Supported	Not supported	Supported
Next-access delay control	Not supported	Supported	Supported	Not supported	Supported
Transfer trigger	SSL input active or RSPCK oscillation	Write to transmit buffer on generation of transmit buffer empty interrupt request (SPTEF = 1)	Write to transmit buffer on generation of transmit buffer empty interrupt request (SPTEF = 1)	RSPCK oscillation	Write to transmit buffer on generation of transmit buffer empty interrupt request (SPTEF = 1)
Sequence control	Not supported	Supported	Supported	Not supported	Supported
Transmit buffer empty detection	Supported*5				
Receive buffer full detection	Supported*2				
Overrun error detection	Supported*2	Supported*2 *4	Supported*2 *4	Supported*2	Supported*2
Parity error detection	Supported*3 *2				
Mode fault error detection	Supported (MODFEN = 1)	Not supported	Supported	Not supported	Not supported
Underrun error detection	Supported*5	Not supported	Not supported	Supported*5	Not supported

Note 1. This function is not supported in this mode.
Note 2. When SPI is transmit-master mode or transmit-slave mode (see Table 30.4), detection of receive buffer full, overrun error, and parity error are not performed.
Note 3. When the SPCR.SPPE bit is 0, parity error detection is not performed.
Note 4. When the SPCR.SCKASE bit is 1, overrun error detection does not proceed.
Note 5. When SPI is receive only slave mode, none of transmit buffer empty and underrun error is detected.
Note 6. CPHA = 0 is invalid in TI SSP mode. (If it is set, the operation is the same as when CPHA = 1.)
Note 7. Supported only in TI SSP mode.

30.3.2 Controlling the SPI Pins

Based on the settings of the MSTR, MODFEN, and SPMS bits in SPCR and the PmnPFS.NCODR bit for I/O Ports, the SPI can switch pin states. Table 30.7 lists the relationship between pin states and bit settings. Setting the PmnPFS.NCODR bit for an I/O port to 0 selects the CMOS output. Setting it to 1 selects the open-drain output. The I/O port settings must follow this relationship.

Table 30.7 Relationship between pin states and bit settings (1 of 2)

Mode	Pin	Pin state*2	
		PmnPFS.NCODR bit for I/O ports = 0	PmnPFS.NCODR bit for I/O ports = 1
Single-master mode (SPI operation) (MSTR = 1, MODFEN = 0, SPMS = 0)	RSPCKn	CMOS output	Open-drain output
	SSLn0 to SSLn3	CMOS output	Open-drain output
	MOSIn	CMOS output	Open-drain output
	MISOn	Input	Input
Multi-master mode (SPI operation) (MSTR = 1, MODFEN = 1, SPMS = 0)	RSPCKn*3	CMOS output/Hi-Z	Open-drain output/Hi-Z
	SSLn0	Input	Input
	SSLn1 to SSLn3*3	CMOS output/Hi-Z	Open-drain output/Hi-Z
	MOSIn*3	CMOS output/Hi-Z	Open-drain output/Hi-Z
	MISOn	Input	Input

Table 30.7 Relationship between pin states and bit settings (2 of 2)

Mode	Pin	Pin state*2	
		PmnPFS.NCODR bit for I/O ports = 0	PmnPFS.NCODR bit for I/O ports = 1
Slave mode (SPI operation) (MSTR = 0, SPMS = 0)	RSPCKn	Input	Input
	SSLn0	Input	Input
	SSLn1 to SSLn3*5	Hi-Z*1	Hi-Z*1
	MOSIn	Input	Input
	MISOn*4	CMOS output/Hi-Z	Open-drain output/Hi-Z
Master mode (clock synchronous operation) (MSTR = 1, MODFEN = 0, SPMS = 1)	RSPCKn	CMOS output	Open-drain output
	SSLn0 to SSLn3*5	Hi-Z*1	Hi-Z*1
	MOSIn	CMOS output	Open-drain output
	MISOn	Input	Input
Slave mode (clock synchronous operation) (MSTR = 0, SPMS = 1)	RSPCKn	Input	Input
	SSLn0 to SSLn3*5	Hi-Z*1	Hi-Z*1
	MOSIn	Input	Input
	MISOn	CMOS output	Open-drain output

Note 1. This function is not supported in this mode.
Note 2. SPI settings are not reflected in multiplexed pins for which the SPI function is not selected.
Note 3. Motorola-SPI: When SSLn0 is at the active level, the pin state is Hi-Z. Whether or not the input signal is at the active level determines the setting of the SPCR3.SSL0P bit.
TI-SSP: From when SSL0 is at the active level until communication is completed, the pin state is Hi-Z under the condition SPCR.SPE = 1.
Note 4. Motorola-SPI: When SSLn0 is at the non-active level or the SPCR.SPE bit is 0, the pin state is Hi-Z. Whether or not the input signal is at the active level determines the setting of the SPCR3.SSL0P bit.
TI-SSP: When SSL0 is except the communication period or when the SPE bit of SPCR is 0 (assertion after SPE = 1 and communication is completed), the pin status changes to Hi-Z.
Note 5. These pins are available for use as I/O port pins.

The SPI in single-master mode (SPI operation) or multi-master mode (SPI operation) determines the MOSI signal values during the SSL negation period (including the SSL retention period during a burst transfer) based on the MOIFE and MOIFV bit settings in SPCR2, as listed in Table 30.8.

Table 30.8 MOSI signal value determination during SSL negation

MOIFE bit	MOIFV bit	MOSIn signal value during SSL negation
0	0, 1	Final data from previous transfer
1	0	Low
1	1	High

30.3.3 SPI System Configuration Examples

This configuration example describes that 0 level of SSL0n signals is active level.

When connecting and using in a multi-slave or multi-master mode, the transfer format of the connected device should be unified to either Motorola-SPI or TI-SSP.

30.3.3.1 Single-master/single-slave with the MCU as a master

Figure 30.6 shows a single-master/single-slave SPI system configuration example where the MCU is used as a master. In the single-master/single-slave configuration, the SSLni outputs of the MCU (master) are not used. The SSL input of the SPI slave is fixed to the low level, and the SPI slave is maintained in the selected state.*1

Note 1. In the transfer format configured when the SPCMDm.CPHA bit is 0, the SSL signal for some slave devices cannot be fixed to an active level. In this case, always connect the SSLni output of the MCU to the SSL input of the slave device.

The MCU (master) drives the RSPCKn and MOSIn signals. The SPI slave drives the MISO signal.

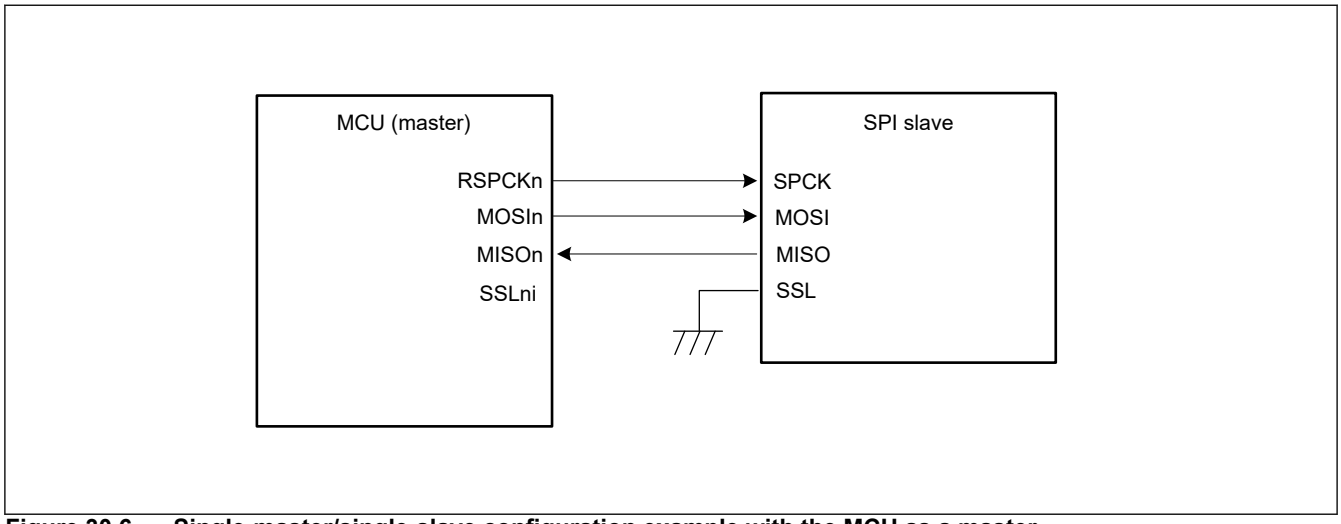


Figure 30.6 Single-master/single-slave configuration example with the MCU as a master

30.3.3.2 Single-master/single-slave with the MCU as a slave

Figure 30.7 shows a single-master/single-slave SPI system configuration example where the MCU is used as a slave. When the MCU operates as a slave, the SSLn0 pin is used as SSL input. The SPI master drives the RSPCK and MOSI signals. The MCU (slave) drives the MISOn signal.*1

Note 1. When SSLn0 is at a non-active level, the pin state is Hi-Z.

In the single-slave configuration when the SPCMDm.CPHA bit is set to 1, the SPCR.SPFRF bit is set to 0, and SPCR.SPMS is set to 0, the SSLn0 input of the MCU (slave) is fixed to the low level and the MCU (slave) is maintained in the selected state. This enables serial transfer execution (Figure 30.8). However, the communication end interrupt does not output when SSL0 input is fixed as Figure 30.8.

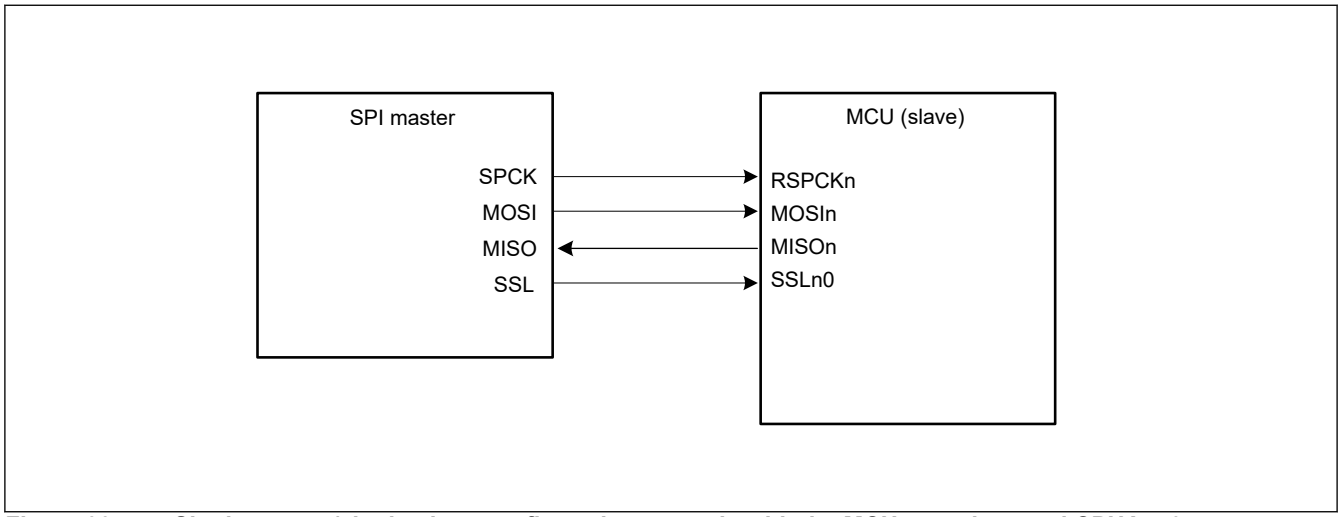


Figure 30.7 Single-master/single-slave configuration example with the MCU as a slave and CPHA = 0

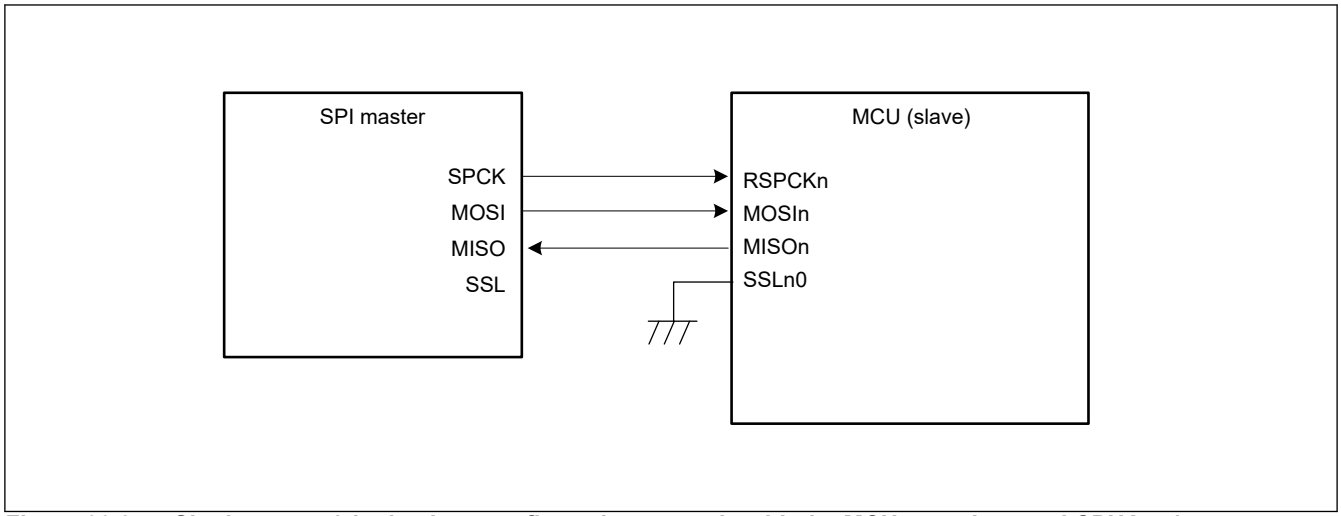


Figure 30.8 Single-master/single-slave configuration example with the MCU as a slave and CPHA = 1

30.3.3.3 Single-master/multi-slave with the MCU as a master

Figure 30.9 shows a single-master/multi-slave SPI system configuration example where the MCU is used as a master. In this example, the SPI system includes the MCU (master) and four slaves (SPI slave 0 to SPI slave 3).

The RSPCKn and MOSIn outputs of the MCU (master) are connected to the RSPCK and MOSI inputs of SPI slaves 0 to 3. The MISO outputs of SPI slaves 0 to 3 are all connected to the MISOn input of the MCU (master). The SSLn0 to SSLn3 outputs of the MCU (master) are connected to the SSL inputs of SPI slaves 0 to 3, respectively.

The MCU (master) drives the RSPCKn, MOSIn, and SSLn0 to SSLn3 signals. Out of the SPI slaves 0 to 3, the slave that receives low-level input into the SSL input drives the MISO signal.

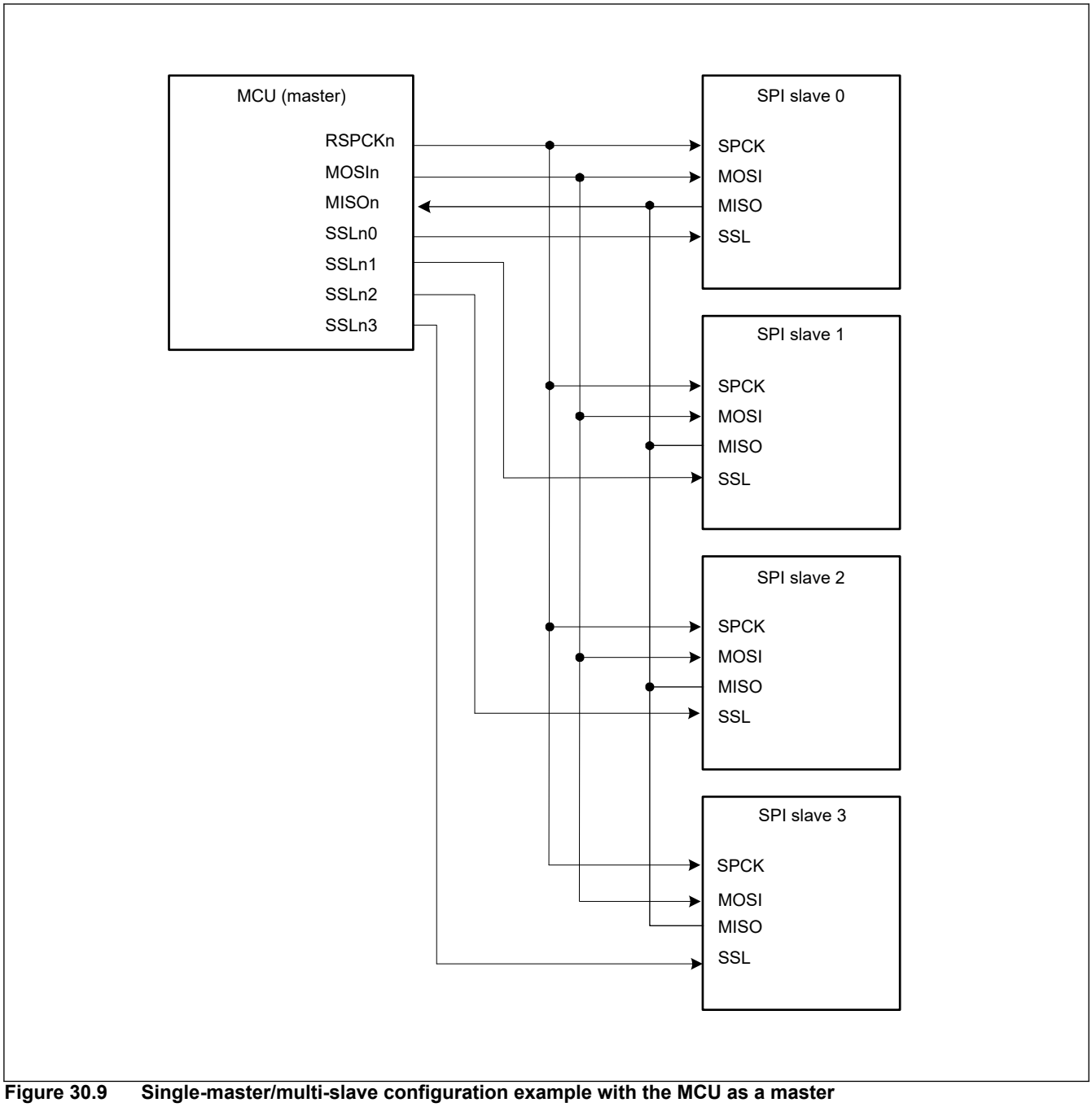


Figure 30.9 Single-master/multi-slave configuration example with the MCU as a master

30.3.3.4 Single-master/multi-slave with the MCU as a slave

Figure 30.10 shows a single-master/multi-slave SPI system configuration example where the MCU is used as a slave. In this example, the SPI system includes an SPI master and two MCUs (slaves X and Y).

The SPCK and MOSI outputs of the SPI master are connected to the RSPCKn and MOSIn inputs of the MCUs (slaves X and Y). The MISOn outputs of the MCUs (slaves X and Y) are all connected to the MISO input of the SPI master. The SSLX and SSLY outputs of the SPI master are connected to the SSLn0 inputs of the MCUs (slaves X and Y, respectively).

The SPI master drives the SPCK, MOSI, SSLX, and SSLY signals. Of the MCUs (slaves X and Y), the slave that receives low-level input into the SSLn0 input drives the MISOn signal.

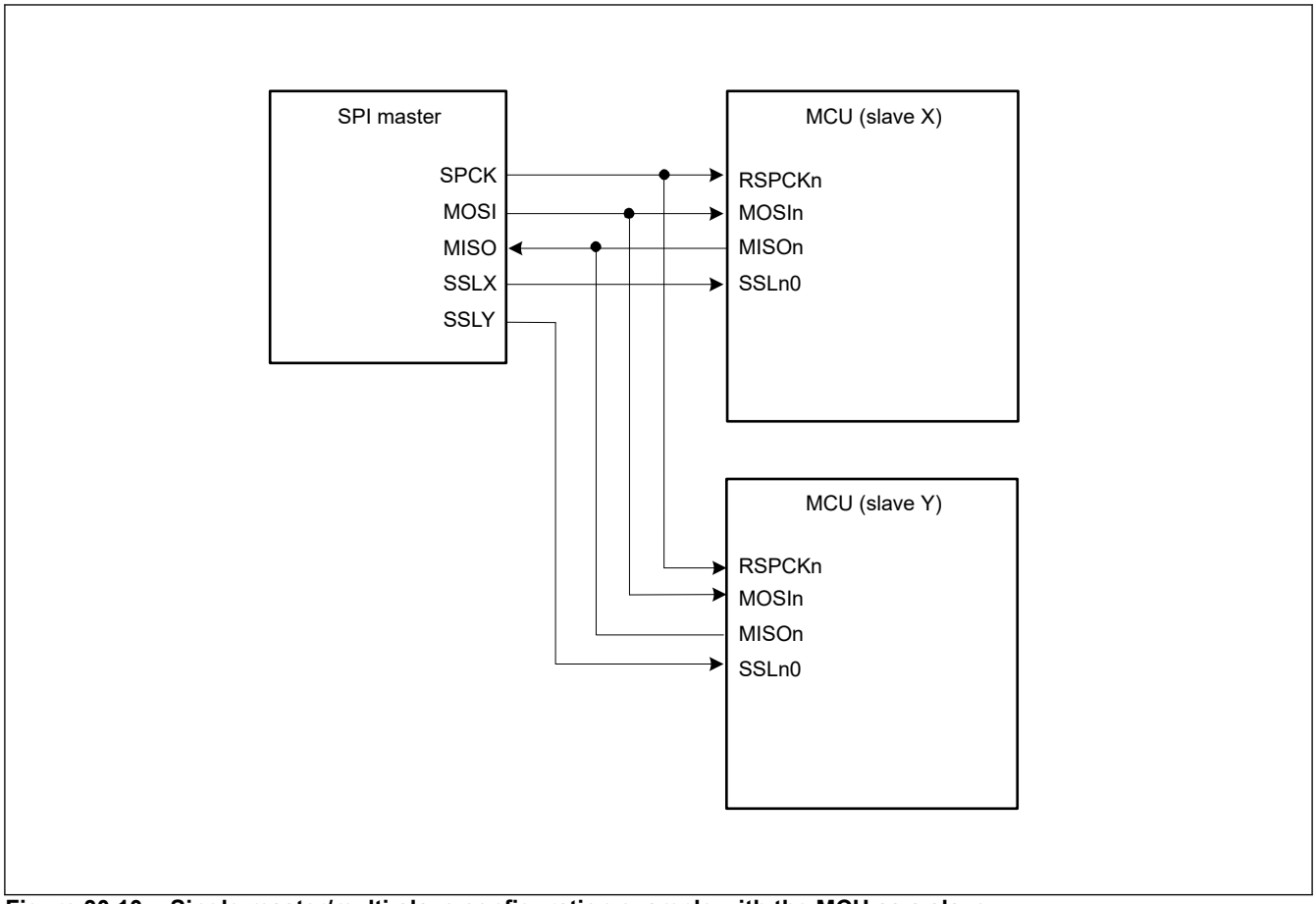


Figure 30.10 Single-master/multi-slave configuration example with the MCU as a slave

30.3.3.5 Multi-master/multi-slave with the MCU as a master

Figure 30.11 shows a multi-master/multi-slave SPI system configuration example where the MCU is used as a master. In this example, the SPI system includes two MCUs (masters X and Y) and two SPI slaves (SPI slaves 1 and 2).

The RSPCKn and MOSIn outputs of the MCUs (masters X and Y) are connected to the RSPCK and MOSI inputs of SPI slaves 1 and 2. The MISO outputs of SPI slaves 1 and 2 are connected to the MISOn inputs of the MCUs (masters X and Y). Any generic port Y output from the MCU (master X) is connected to the SSLn0 input of the MCU (master Y). Any generic port X output of the MCU (master Y) is connected to the SSLn0 input of the MCU (master X). The SSLn1 and SSLn2 outputs of the MCUs (masters X and Y) are connected to the SSL inputs of the SPI slaves 1 and 2. In this configuration example, because the system can be comprised solely of SSLn0 input, and SSLn1 and SSLn2 outputs for slave connections, the SSLn3 output of the MCU is not required.

The MCU drives the RSPCKn, MOSIn, SSLn1, and SSLn2 signals when the SSLn0 input level is high. When the SSLn0 input level is low, the MCU detects a mode fault error, sets RSPCKn, MOSIn, SSLn1, and SSLn2 to Hi-Z, and releases the SPI bus directly to the other master. Of the SPI slaves 1 and 2, the slave that receives low-level input into the SSL input drives the MISO signal.

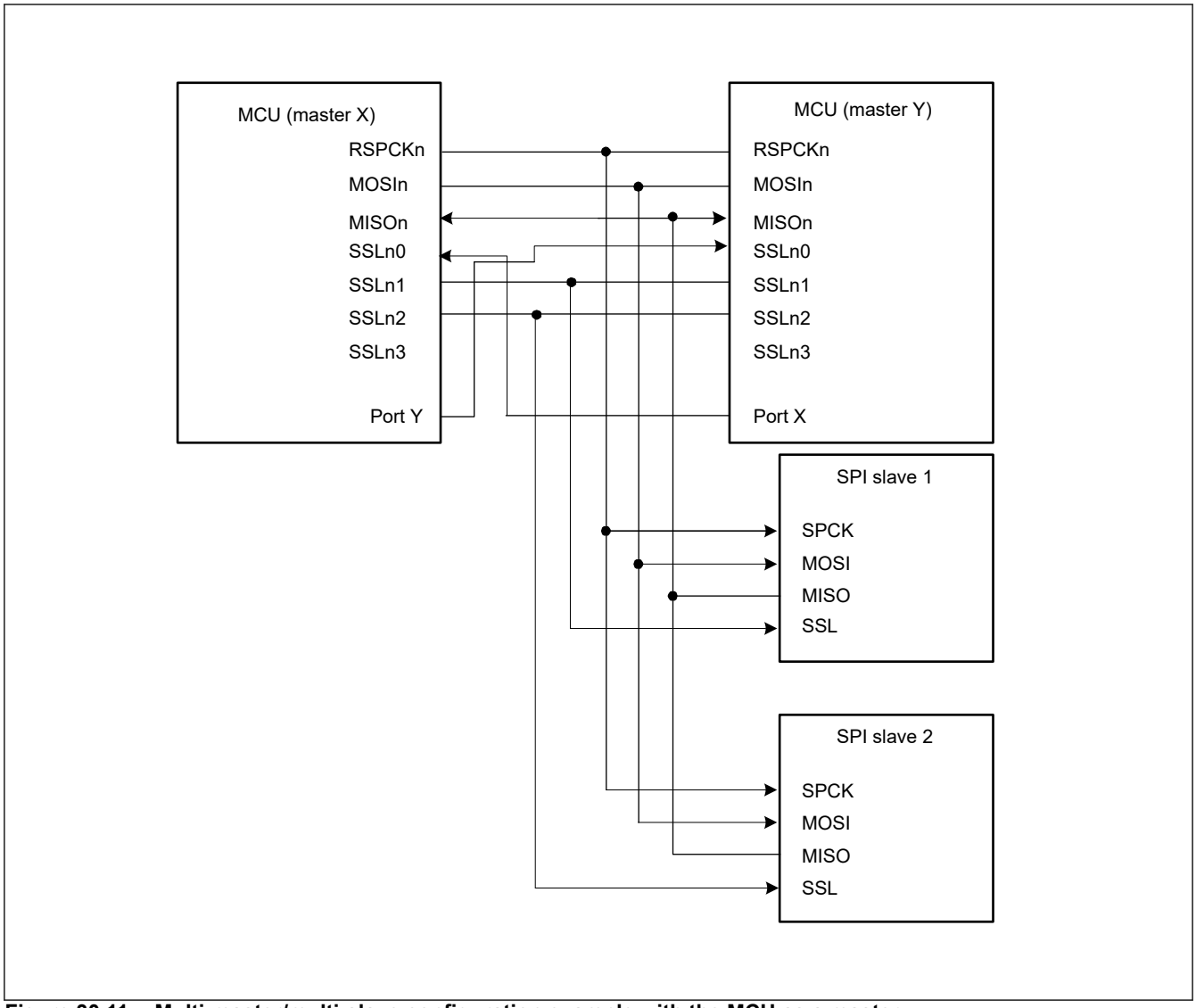


Figure 30.11 Multi-master/multi-slave configuration example with the MCU as a master

When setting TI SSP, enter the following levels for port X and port Y.

- Start of communication: the value of SPCR3.SSL0P of the other master.
- End of communication: the inverted value of SPCR3.SSL0P of the other master.

30.3.3.6 Master and slave in clock synchronous mode with the MCU configured as a master

Figure 30.12 shows a master and slave in clock synchronous mode configuration example where the MCU is used as a master. In this configuration, SSLni of the MCU (master) are not used.

The MCU (master) drives the RSPCKn and MOSIn signals. The SPI slave drives the MISO signal.

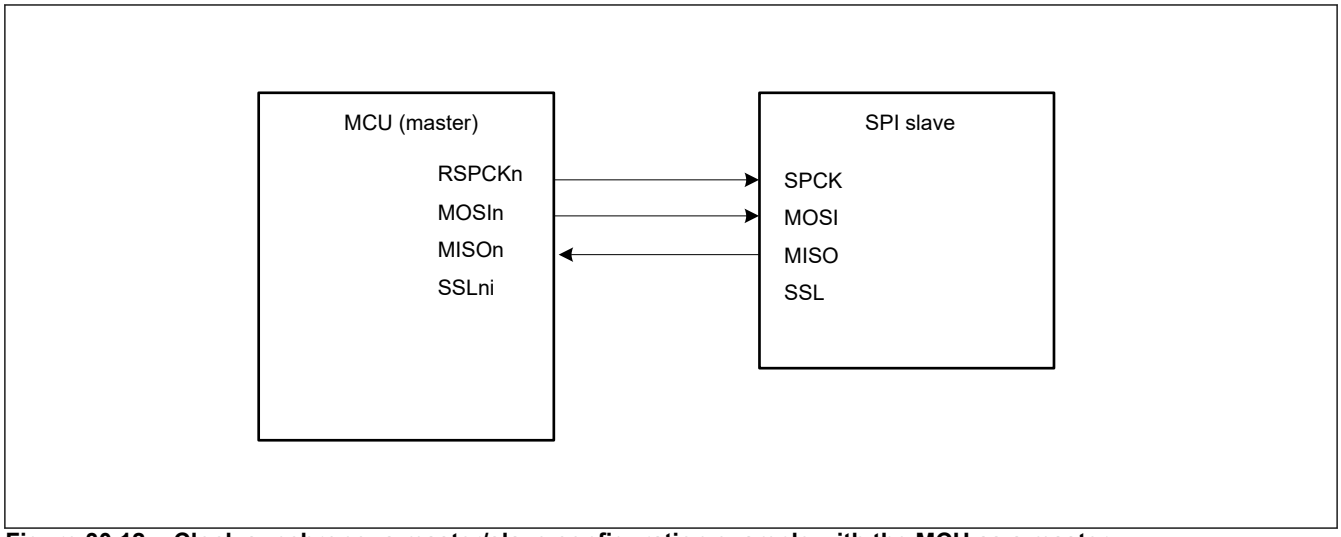


Figure 30.12 Clock synchronous master/slave configuration example with the MCU as a master

30.3.3.7 Master and slave in clock synchronous mode with the MCU as a slave

Figure 30.13 shows a master and slave in clock synchronous mode configuration example where the MCU is used as a slave. When the MCU operates as a slave (clock synchronous operation), the MCU (slave) drives the MISOn signal and the SPI master drives the SPCK and MOSI signals. In addition, SSLn0 to SSLn3 of the MCU (slave) are not used.

The MCU (slave) can only execute serial transfers in the single-slave configuration when the SPCMDm.CPHA bit is set to 1.

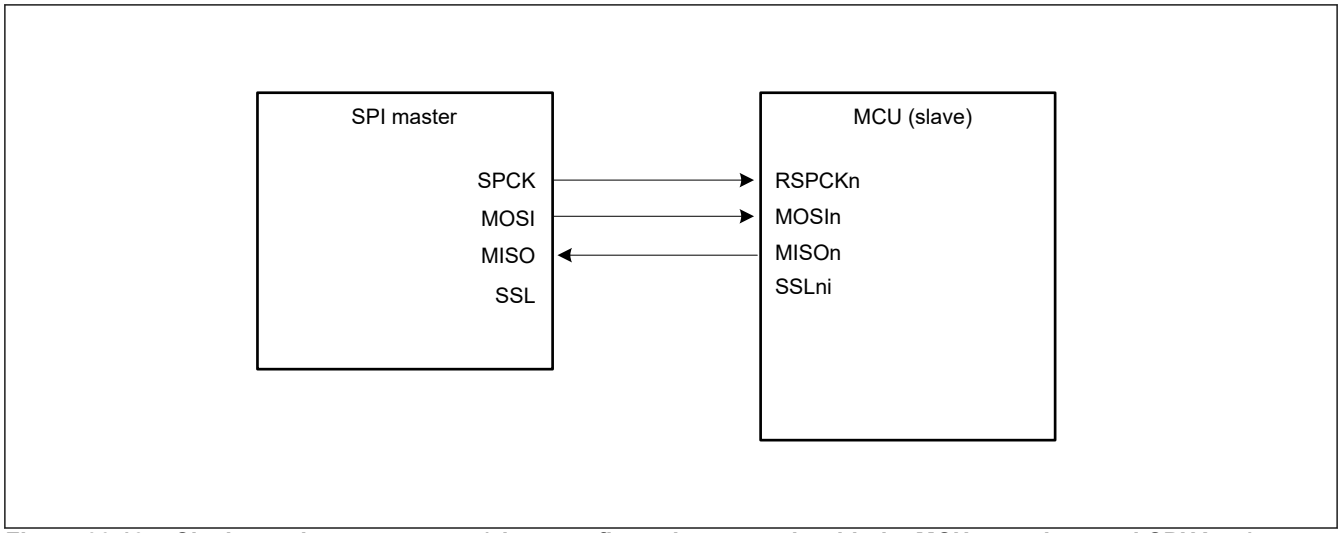


Figure 30.13 Clock synchronous master/slave configuration example with the MCU as a slave and CPHA = 1

30.3.4 Data Formats

The data format of the SPI depends on the settings in SPI Command Register m (SPCMDm) and the parity enable bit in SPI Control Register (SPCR.SPPE). Regardless of whether the MSB or LSB is first, the SPI treats the range from the LSB bit in the SPI Data Register (SPDR) to the bit associated with the selected data length, as transfer data.

This section shows the format of one frame of data before or after transfer.

Data format with parity disabled

When parity is disabled, transmission or reception of data proceeds with the length in bits selected in the SPI data length setting in SPI Command Register m (SPCMDm.SPB[4:0]).

Data format with parity enabled

When parity is enabled, transmission or reception of data proceeds with the length in bits selected in the SPI data length setting in SPI Command Register m (SPCMDm.SPB[4:0]). In this case, however, the last bit is a parity bit.

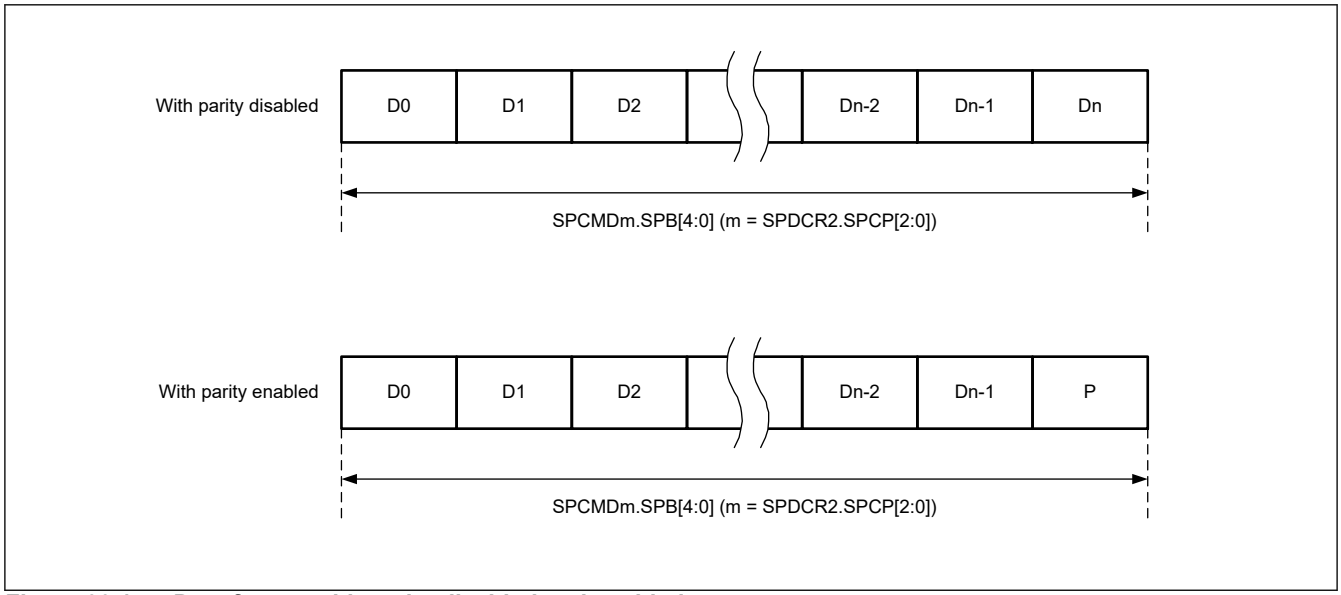


Figure 30.14 Data format with parity disabled and enabled

30.3.4.1 Operation when parity is disabled (SPCR.SPPE = 0)

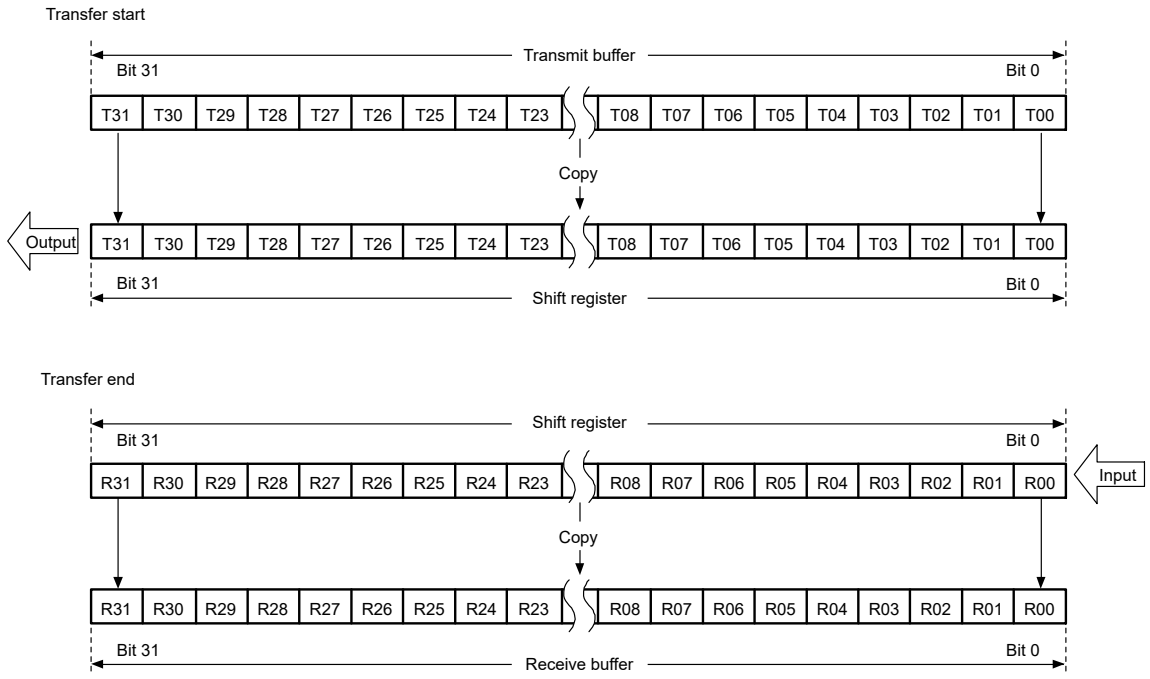
When parity is disabled, data for transmission is copied to the shift register with no pre-processing. This section describes the connection between the SPI Data Register (SPDR) and the shift register in terms of the combination of MSB- or LSB-first order and data length.

(1) MSB-first transfer with 32-bit data

Figure 30.15 shows the operation of the SPI Data Register (SPDR) and the shift register in a transfer with parity disabled, a SPI data length of 32 bits, and MSB-first selected.

In transmission, bits T31 to T00 from the current stage of the transmit buffer are copied to the shift register. Data for transmission is shifted out from the shift register from T31 to T30, and continuing to T00.

In reception, received data is shifted in bit-by-bit through bit[0] of the shift register. When the R31 to R00 bits are collected after input of the required number of RSPCK cycles, the value in the shift register is copied to the receive buffer.



Note: Output = MOSI (master)/MISO (slave), input = MISO (master)/MOSI (slave)

Figure 30.15 MSB-first transfer with 32-bit data and parity disabled

(2) MSB-first transfer with 24-bit data

Figure 30.16 shows the operation of the SPI Data Register (SPDR) and the shift register in a transfer with parity disabled, an SPI data length of 24 bits for an example that is not 32 bits, and MSB-first selected.

In transmission, the lower 24 bits (T23 to T00) from the current stage of the transmit buffer are copied to the shift register. Data for transmission is shifted out from the shift register from T23 to T22, and continuing to T00.

In reception, received data is shifted in bit-by-bit through bit[0] of the shift register. When the R23 to R00 bits are collected after input of the required number of RSPCK cycles, the value in the shift register is copied to the receive buffer.

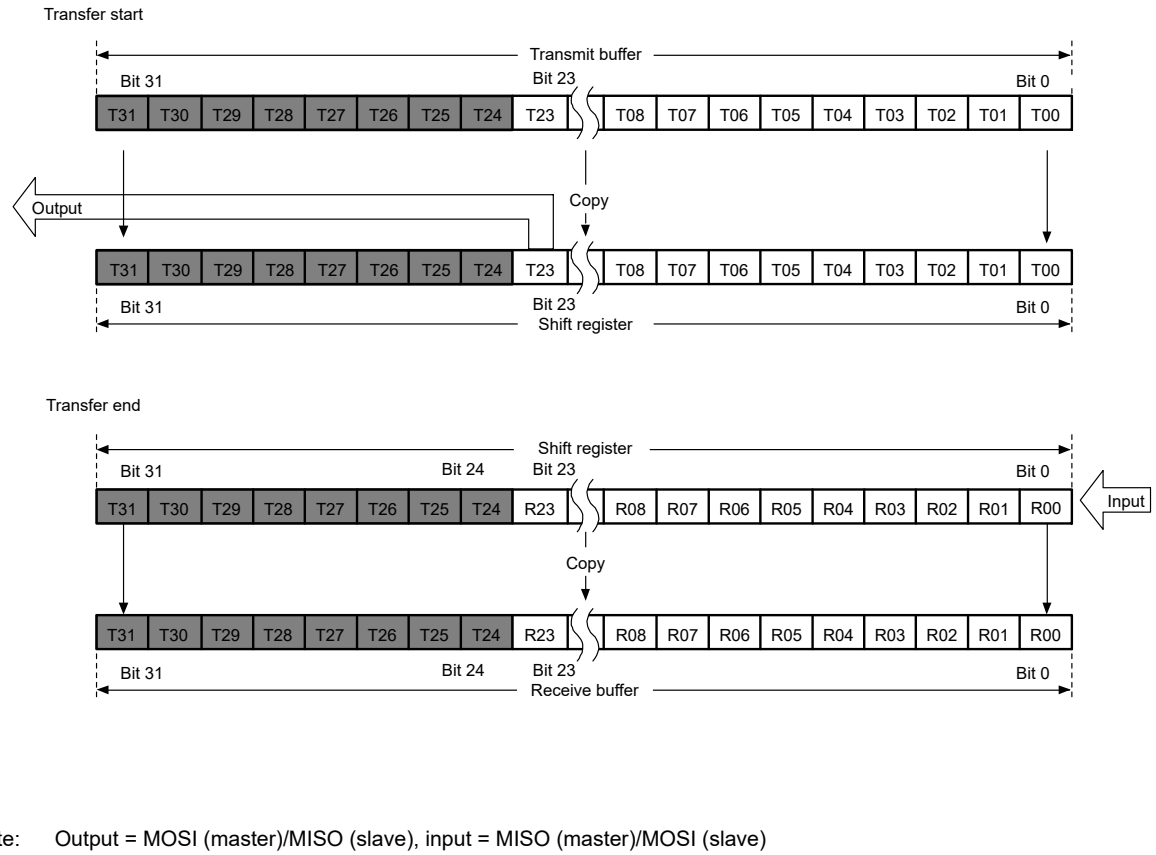


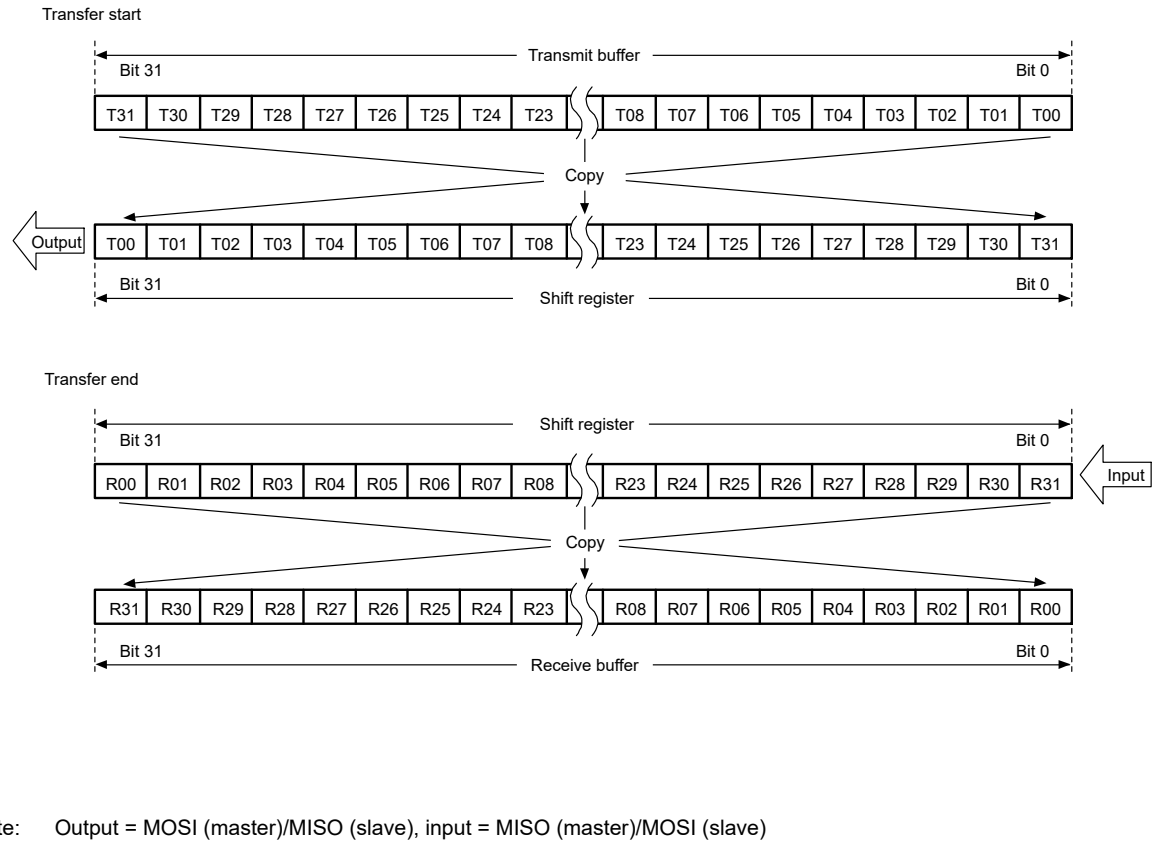
Figure 30.16 MSB-first transfer with 24-bit data and parity disabled

(3) LSB-first transfer with 32-bit data

Figure 30.17 shows the operation of the SPI Data Register (SPDR) and the shift register in a transfer with parity disabled, an SPI data length of 32 bits, and LSB-first selected.

In transmission, bits T31 to T00 from the current stage of the transmit buffer are reordered bit-by-bit to obtain the order T00 to T31 for copying to the shift register. Data for transmission is shifted out from the shift register in order from T00 to T01, and continuing to T31.

In reception, received data is shifted in bit-by-bit through bit[0] of the shift register. When the R00 to R31 bits are collected after input of the required number of RSPCK cycles, the value in the shift register is copied to the receive buffer.



Note: Output = MOSI (master)/MISO (slave), input = MISO (master)/MOSI (slave)

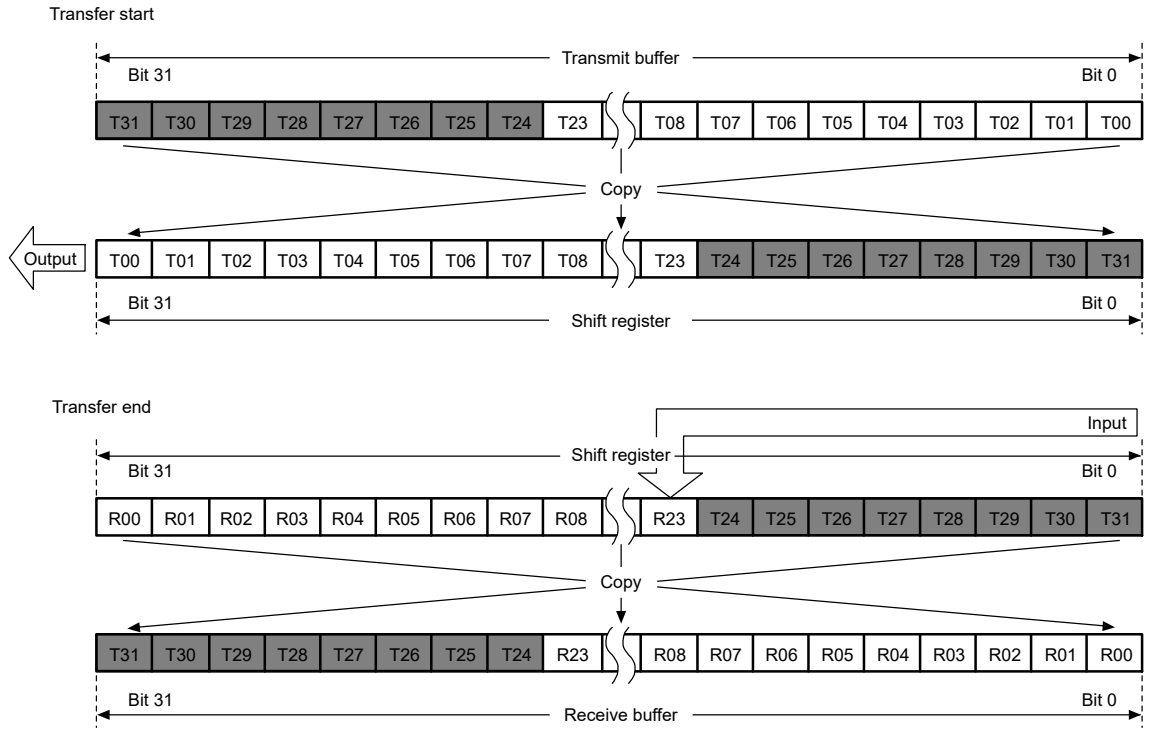
Figure 30.17 LSB-first transfer with 32-bit data and parity disabled

(4) LSB-first transfer with 24-bit data

Figure 30.18 shows the operation of the SPI Data Register (SPDR) and the shift register in transfers with parity disabled, an SPI data length of 24 bits for an example that is not 32, and LSB-first selected.

In transmission, the lower 24 bits (T23 to T00) from the current stage of the transmit buffer are reordered bit-by-bit to obtain the order T00 to T23 for copying to the shift register. Data for transmission is shifted out from the shift register from T00 to T01, and continuing to T23.

In reception, received data is shifted in bit-by-bit through bit[8] of the shift register. When the R00 to R23 bits are collected after input of the required number of RSPCK cycles, the value in the shift register is copied to the receive buffer.



Note: Output = MOSI (master)/MISO (slave), input = MISO (master)/MOSI (slave)

Figure 30.18 LSB-first transfer with 24-bit data and parity disabled

30.3.4.2 Operation when parity is enabled (SPCR.SPPE = 1)

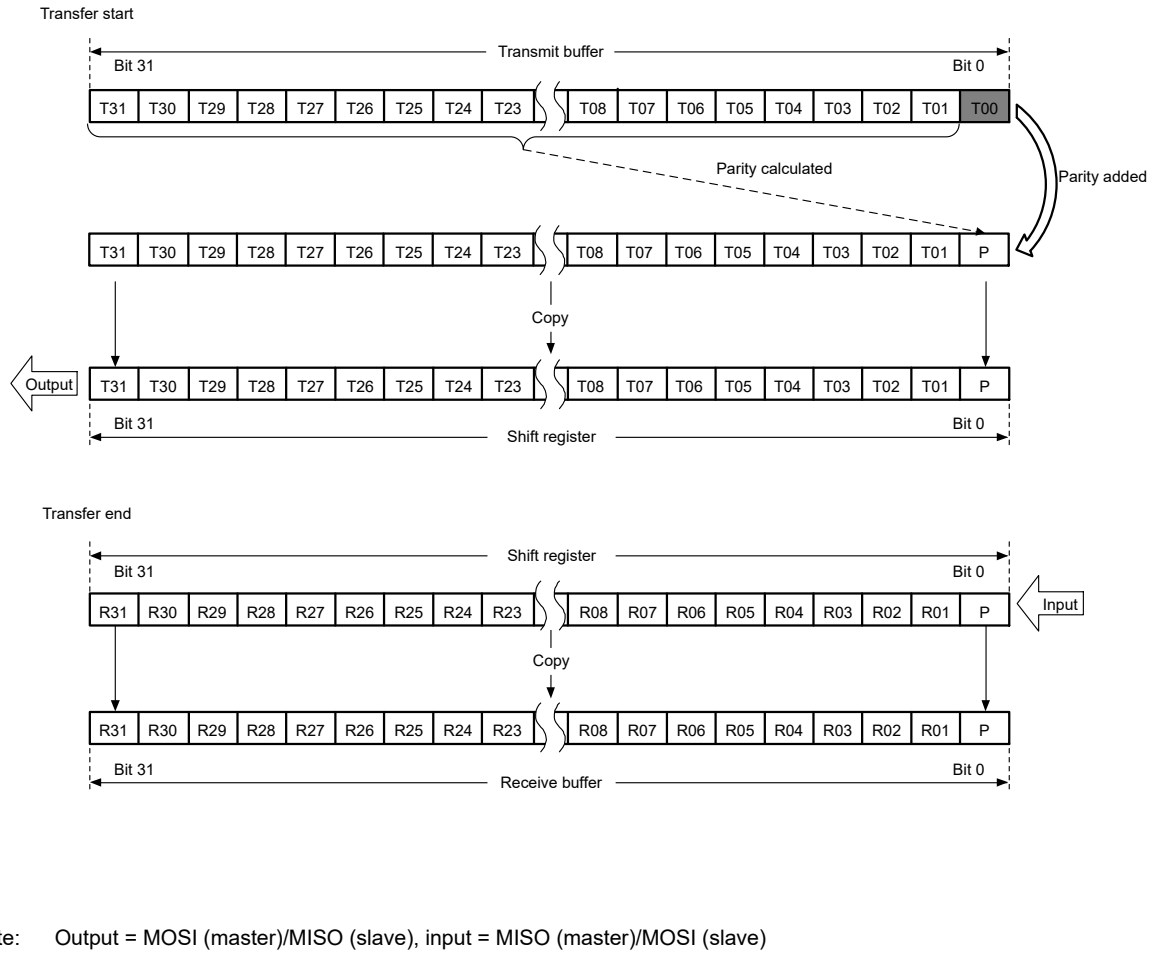
When parity is enabled, the lowest-order bit of the data for transmission becomes a parity bit. Hardware calculates the value of the parity bit.

(1) MSB-first transfer with 32-bit data

Figure 30.19 shows the operation of the SPI Data Register (SPDR) and the shift register in a transfer with parity enabled, an SPI data length of 32 bits, and MSB-first selected.

In transmission, the value of the parity bit (P) is calculated from bits T31 to T01. This replaces the final bit, T00, and the whole value is copied to the shift register. Data is transmitted in the order T31, T30, …, T01, and P.

In reception, received data is shifted in bit-by-bit through bit[0] of the shift register. When the R31 to P bits are collected after input of the required number of RSPCK cycles, the value in the shift register is copied to the receive buffer. On copying of data to the shift register, the data from R31 to P is checked for parity.



Note: Output = MOSI (master)/MISO (slave), input = MISO (master)/MOSI (slave)

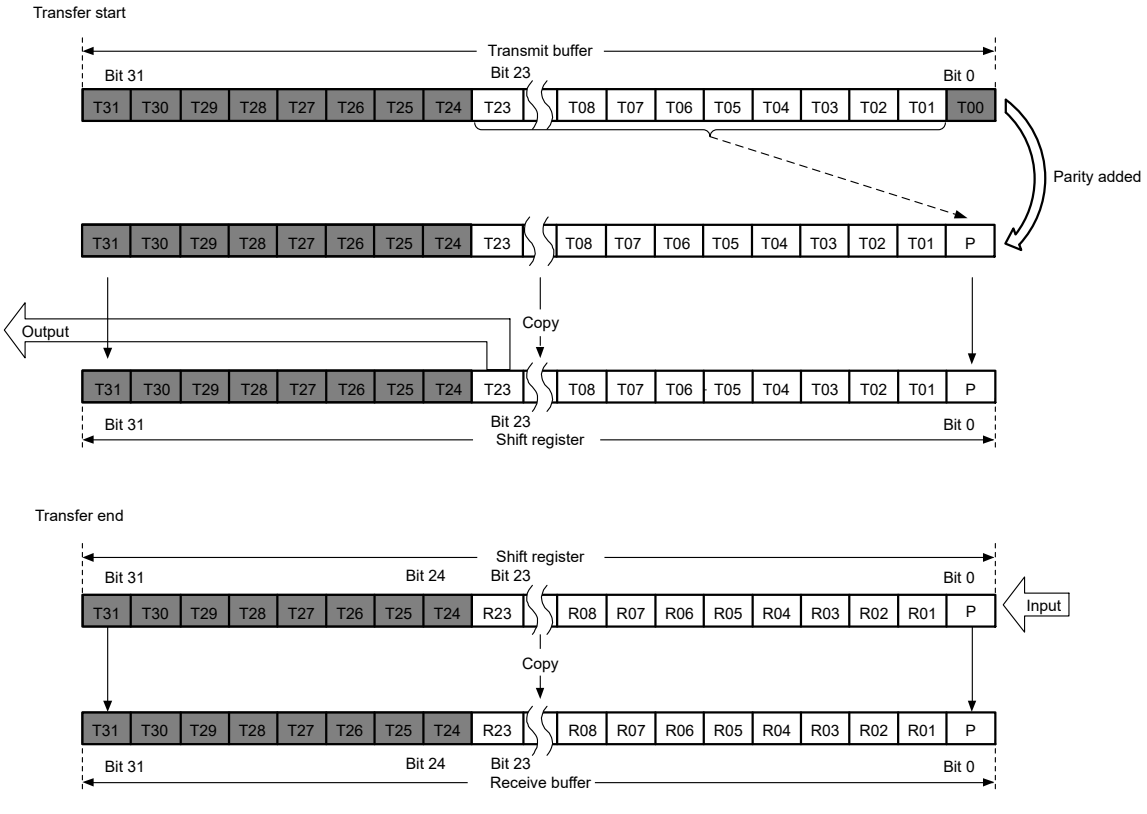
Figure 30.19 MSB-first transfer with 32-bit data and parity enabled

(2) MSB-first transfer with 24-bit data

Figure 30.20 shows the operation of the SPI Data Register (SPDR) and the shift register in a transfer with parity enabled, a SPI data length of 24 bits, and MSB-first selected.

In transmission, the value of the parity bit (P) is calculated from bits T23 to T01. This replaces the final bit, T00, and the whole value is copied to the shift register. Data is transmitted in the order T23, T22, …, T01, and P.

In reception, received data is shifted in bit-by-bit through bit[0] of the shift register. When the R23 to P bits are collected after input of the required number of RSPCK cycles, the value in the shift register is copied to the receive buffer. On copying of data to the shift register, the data from R23 to P is checked for parity.



Note: Output = MOSI (master)/MISO (slave), input = MISO (master)/MOSI (slave)

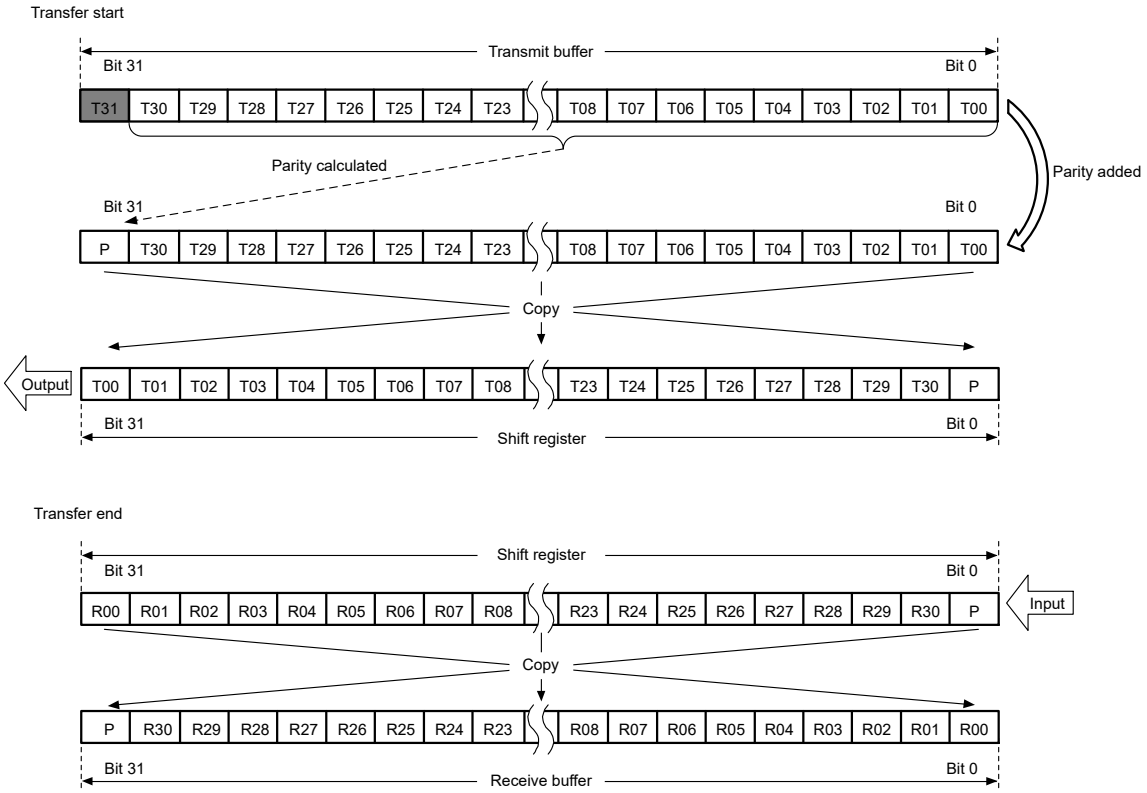
Figure 30.20 MSB-first transfer with 24-bit data and parity enabled

(3) LSB-first transfer with 32-bit data

Figure 30.21 shows the operation of the SPI Data Register (SPDR) and the shift register in a transfer with parity enabled, an SPI data length of 32 bits, and LSB-first selected.

In transmission, the value of the parity bit (P) is calculated from bits T30 to T00. This replaces the final bit, T31, and the whole value is copied to the shift register. Data is transmitted in the order T00, T01, …, T30, and P.

In reception, received data is shifted in bit-by-bit through bit[0] of the shift register. When the R00 to P bits are collected after input of the required number of RSPCK cycles, the value in the shift register is copied to the receive buffer. On copying of data to the shift register, the data from R00 to P is checked for parity.



Note: Output = MOSI (master)/MISO (slave), input = MISO (master)/MOSI (slave)

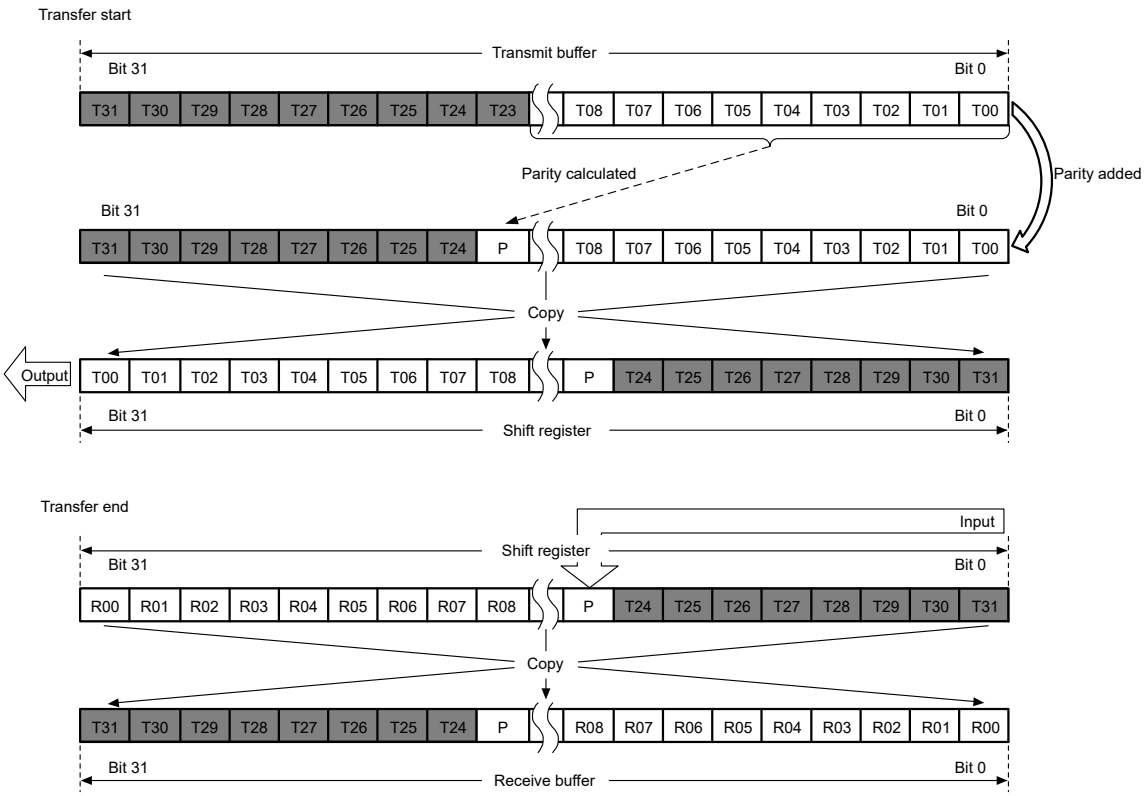
Figure 30.21 LSB-first transfer with 32-bit data and parity enabled

(4) LSB-first transfer with 24-bit data

Figure 30.22 shows the operation of the SPI Data Register (SPDR) and the shift register in a transfer with parity enabled, a SPI data length of 24 bits, and LSB-first selected.

In transmission, the value of the parity bit (P) is calculated from bits T22 to T00. This replaces the final bit, T23, and the whole value is copied to the shift register. Data is transmitted in the order T00, T01, …, T22, and P.

In reception, received data is shifted in bit-by-bit through bit[8] of the shift register. When the R00 to P bits are collected after input of the required number of RSPCK cycles, the value in the shift register is copied to the receive buffer. On copying of data to the shift register, the data from R00 to P is checked for parity.



Note: Output = MOSI (master)/MISO (slave), input = MISO (master)/MOSI (slave)

Figure 30.22 LSB-first transfer with 24-bit data and parity enabled

30.3.4.3 Byte Swap Transmission

When byte swapping is enabled, the data in the transmission buffer, swapped in 8-bit units, is copied to the shift register. Figure 30.23 shows the relationship between the SPDR (transmit buffer) and the shift register when transferring data with a 32-bit data length, using a combination of MSB / LSB first and with / without byte swap.

(1) MSB-first transfer. (When the byte swap is disabled.)

Data (Byte3 [T31 to T24] to Byte0 [T07 to T00]) in the transmit buffer are copied to the shift register.

Bit values in the shift register are shifted and transmitted in the order of T31 → T30 → … → T00 as transmit data.

(2) MSB-first transfer. (When the byte swap is enabled.)

Byte values of the transmit buffer (Byte3 [T31 to T24] to Byte0 [T07 to T00]) are reversed in byte units and are copied to the shift register in the order of Byte0 [T07 to T00] to Byte3 [T31 to T24].

Bit values in the shift register are shifted and transmitted in the order of T07 → T06 → … → T00 → T15 → T14 → … → T08 → T23 → T22 → … → T16 → T31 → T30 → … → T24 as transmit data.

(3) LSB-first transfer. (When the byte swap is disabled.)

Bit values of the transmit buffer (Byte3 [T31 to T24] to Byte0 [T07 to T00]) are reversed in bit units and are copied to the shift register in the order of Byte0 [T00 to T07] to Byte3 [T24 to T31].

Bit values in the shift register are shifted and transmitted in the order of T00 → T01 → … → T31 as transmit data.

(4) LSB-first transfer. (When the byte swap is enabled.)

Bit values of each byte of the transmit buffer (Byte3 [T31 to T24] to Byte0 [T07 to T00]) are reversed in bit units and are copied to the shift register in the order of Byte3 [T24 to T31] to Byte0 [T00 to T07].

Bit values in the shift register are shifted and transmitted in the order of T24 → T25 → … → T31 → T16 → T17 → … → T23 → T08 → T09 → … → T15 → T00 → T01 → … → T07 as transmit data.

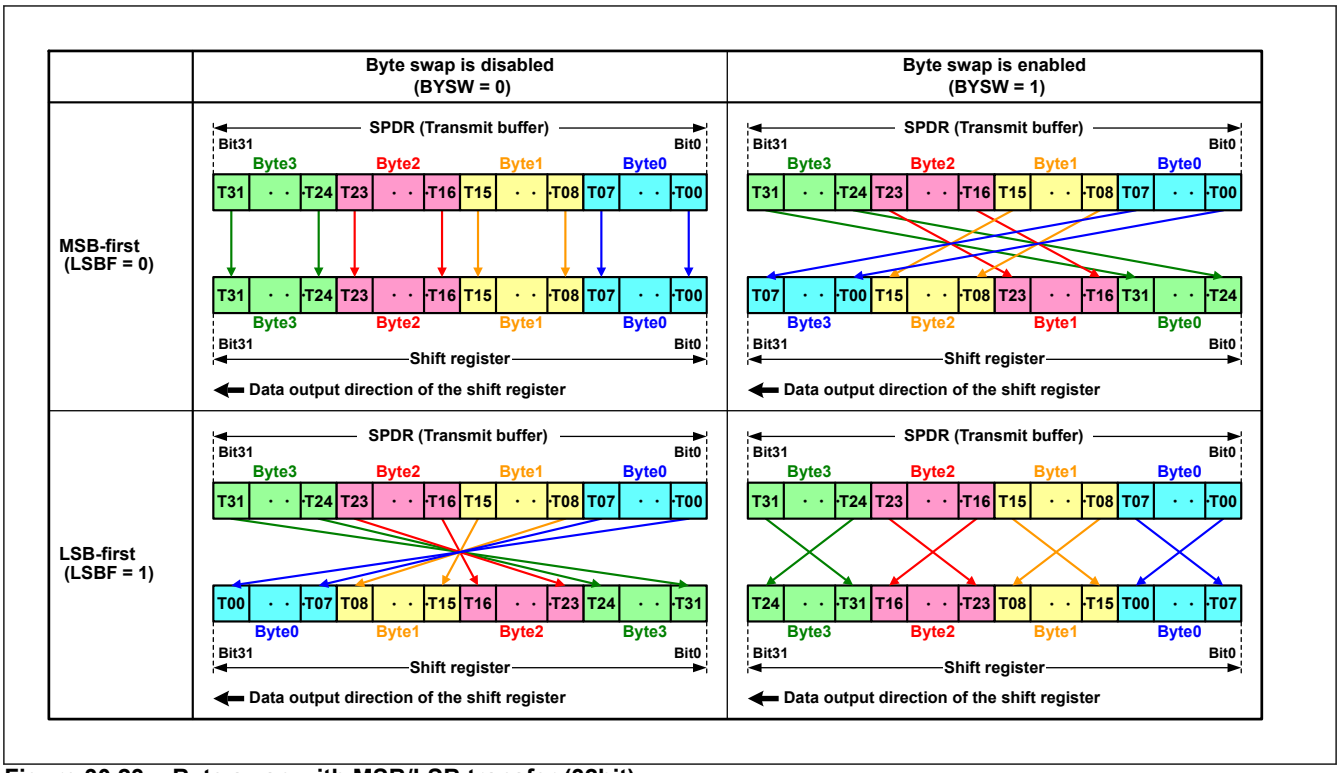


Figure 30.23 Byte swap with MSB/LSB transfer (32bit)

Figure 30.24 shows the relationship between the SPDR (transmit buffer) and the shift register when transferring data with a 16-bit data length, using a combination of MSB / LSB first and with / without byte swap.

1. MSB-first transfer. (When the byte swap is disabled.)
   Data (Byte1 [T15 to T08] to Byte0 [T07 to T00]) in the transmit buffer are copied to the shift register in the order of Byte1 [T15 to T08] to Byte0 [T07 to T00], Byte1 [T15 to T08] to Byte0 [T07 to T00]. Bit values in the shift register are shifted and transmitted in the order of T15 → T14 → … T00 as transmit data.
2. MSB-first transfer. (When the byte swap is enabled.)
   Byte values of the transmit buffer (Byte1 [T15 to T08] to Byte0 [T07 to T00]) are reversed in byte units and are copied to the shift register in the order of Byte0 [T07to T00] to Byte1 [T15 to T08], Byte0 [T07 to T00] to Byte1 [T15 to T08]. Bit values in the shift register are shifted and transmitted in the order of T07 → T06 → … T00 → T15 → T14 → … T08 as transmit data.
3. LSB-first transfer. (When the byte swap is disabled.)
   Bit values of the transmit buffer (Byte1 [T15 to T08] to Byte0 [T07 to T00]) are reversed in bit units and are copied to the shift register in the order of Byte0 [T00 to T07] to Byte1 [T08 to T15], Byte0 [T00 to T07] to Byte1 [T08 to T15]. Bit values in the shift register are shifted and transmitted in the order of T00 → T01 → … T15 as transmit data.
4. LSB-first transfer. (When the byte swap is enabled.)
   Bit values of each byte of the transmit buffer (Byte1 [T15 to T08] to Byte0 [T07 to T00]) are reversed in bit units and are copied to the shift register in the order of Byte1 [T08 to T15] to Byte0 [T00 to T07], Byte1 [T08 to T15] to Byte0 [T00 to T07]. Bit values in the shift register are shifted and transmitted in the order of T08 → T09 → … T15 → T00 → T01 → … T07 as transmit data.

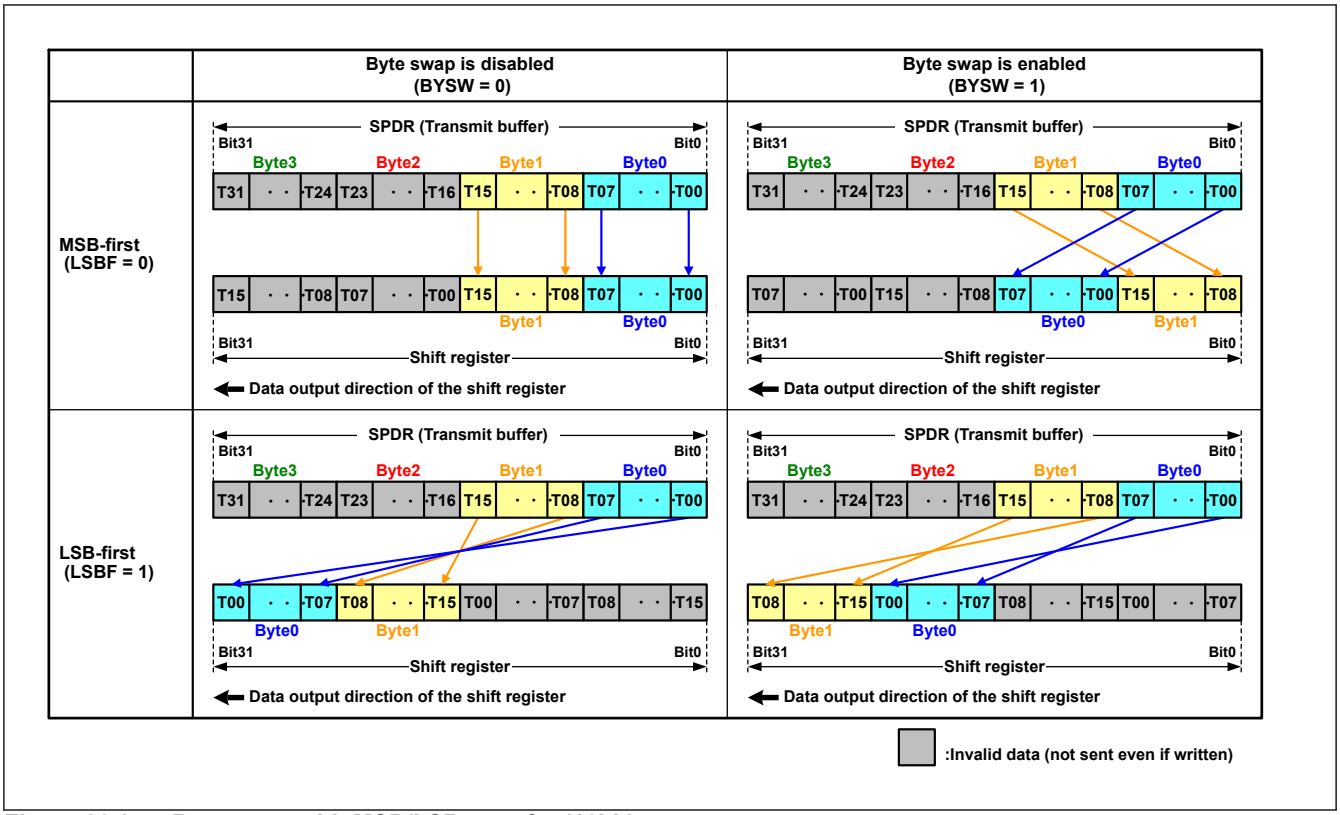


Figure 30.24 Byte swap with MSB/LSB transfer (16bit)

Note: 1. When using the byte swap, set 16 bits or 32 bits to the data length (SPCMDm.SPB[4:0] setting). If setting the other length, the behavior is not guaranteed.
2. When the byte swap is valid, set the parity function as invalid (SPCR.SPPE bit = 0). If setting the parity function as valid (SPPE bit = 1), the behavior is not guaranteed.
3. Set SPDCR.BYSW bit, when SPCR.SPE bit is 0. If rewriting BYSW bit, when SPE bit is 1, the behavior after it is not guaranteed.

30.3.4.4 Byte Swap Reception

When byte swap is enabled, the data in the shift register, swapped in 8-bit units, is copied to the receive buffer. Figure 30.25 shows the relationship between the shift register and SPDR (reception buffer) when transferring data with a 32-bit data length, using a combination of MSB / LSB first and with / without byte swap.

(1) MSB-first transfer. (When the byte swap is disabled.)

The first received data (R31) is stored in bit 0 of the shift register, and received data is shifted in the order of R31 → R30 → … → R00.

When necessary RSPCK cycles are input and data is stored from Byte3 [R31 to R24] to Byte0 [R07 to R00], the shift register value is copied to the receive buffer.

(2) MSB-first transfer. (When the byte swap is enabled.)

The first received data (R07) is stored in bit 0 of the shift register, and received data is shifted in the order of R07 → R06 → … → R00 → R15 → R14 → … → R08 → R23 → R22 → … → R16 → R31 → R30 → … → R24.

When necessary RSPCK cycles are input and data is stored from Byte0 [R07 to R00] to Byte3 [R31 to R24], byte values in the shift register are reversed in byte units and are copied to the receive buffer in the order of Byte3 [R31 to R24] to Byte0 [R07 to R00].

(3) LSB-first transfer. (When the byte swap is disabled.)

The first received data (R00) is stored in bit 0 of the shift register, and received data is shifted in the order of R00 → R01 → … → R31.

When necessary RSPCK cycles are input and data is stored from Byte0 [R00 to R07] to Byte3 [R24 to R31], bit values in the shift register are reversed in bit units and are copied to the receive buffer in the order of Byte3 [R31 to R24] to Byte0 [R07 to R00].

(4) LSB-first transfer. (When the byte swap is enabled.)

The first received data (R24) is stored in bit 0 of the shift register, and received data is shifted in the order of R24 → R25 → … → R31 → R16 → R17 → … → R23 → R08 → R09 → … → R15 → R00 → R01 → … → R07.

When necessary RSPCK cycles are input and data is stored from Byte3 [R24 to R31] to Byte0 [R00 to R07], bit values of each byte in the shift register are reversed in bit units and are copied to the receive buffer in the order of Byte3 [R31 to R24] to Byte0 [R07 to R00].

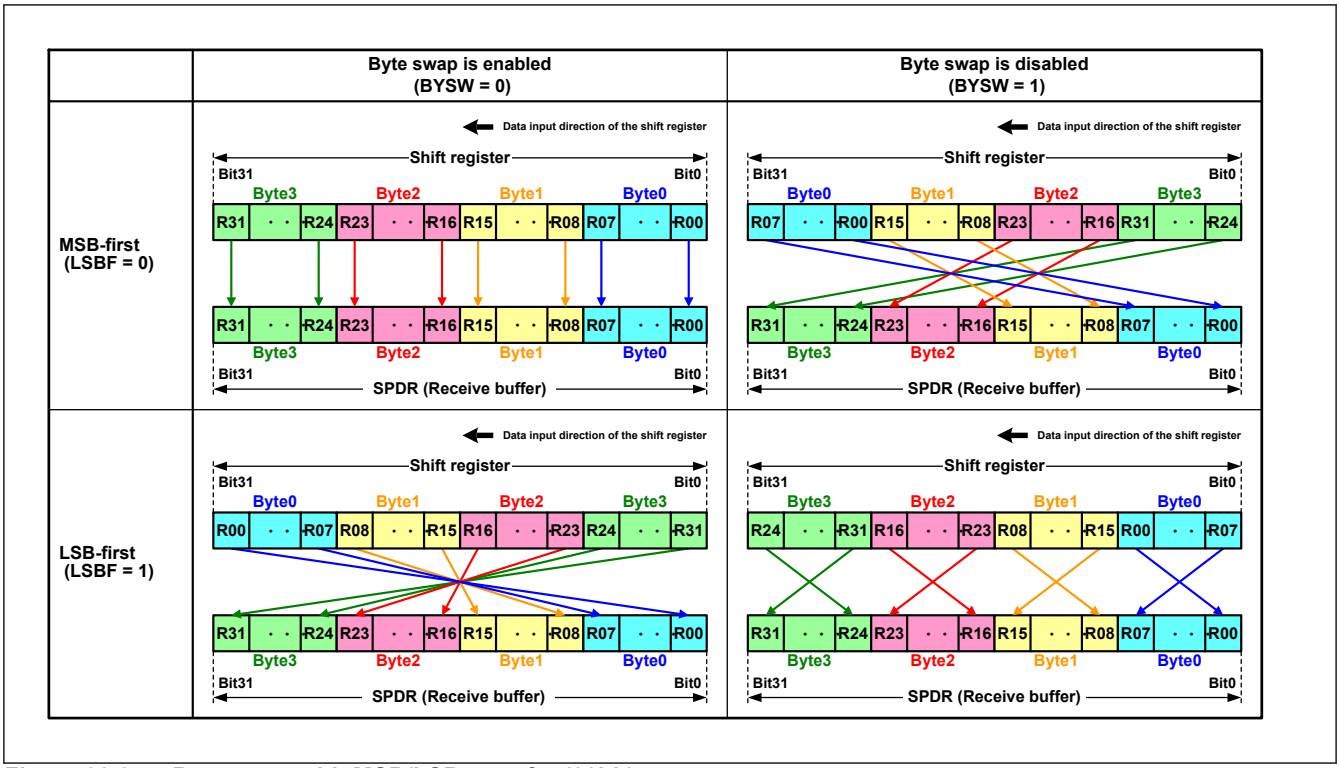


Figure 30.25 Byte swap with MSB/LSB transfer (32bit)

Figure 30.26 shows the relationship between the shift register and SPDR (reception buffer) when transferring data with a 16-bit data length, using a combination of MSB / LSB first and with / without byte swap.

1. MSB-first transfer. (When the byte swap is disabled.)
   The first received data (R15) is stored in bit 0 of the shift register, and received data is shifted in the order of R15 → R14 → … R00. When necessary RSPCK cycles are input and data is stored from Byte3 [R31 to R24] to Byte0 [R07 to R00], the shift register value is copied to the receive buffer.
2. MSB-first transfer. (When the byte swap is enabled.)
   The first received data (R07) is stored in bit 0 of the shift register, and received data is shifted in the order of R07 → R06 → … R00 → R15 → R14 → … R08. When necessary RSPCK cycles are input and data is stored from Byte0 [R07 to R00] to Byte1 [R15 to R08], byte values in the shift register are reversed in byte units and are copied to the receive buffer in the order of Byte3 [R31 to R24] to Byte0 [R07 to R00].
3. LSB-first transfer. (When the byte swap is disabled.)
   The first received data (R00) is stored in bit 15 of the shift register, and received data is shifted in the order of R00 → R01 → … R07 → R08 → R09 → … R15. When necessary RSPCK cycles are input and data is stored from Byte0 [R00 to R07] to Byte1 [R08 to R15], bit values in the shift register are reversed in bit units and are copied to the receive buffer in the order of Byte3 [R31 to R24] to Byte0 [R07-R00].
4. LSB-first transfer. (When the byte swap is enabled.)
   The first received data (R08) is stored in bit 15 of the shift register, and received data is shifted in the order of R08 → R09 → … R15 → R00 → R01 → … R07. When necessary RSPCK cycles are input and data is stored from Byte1 [R08

to R15] to Byte0 [R00 to R07], bit values of each byte in the shift register are reversed in bit units and are copied to the receive buffer in the order of Byte3 [R31 to R24] to Byte0 [R07 to R00].

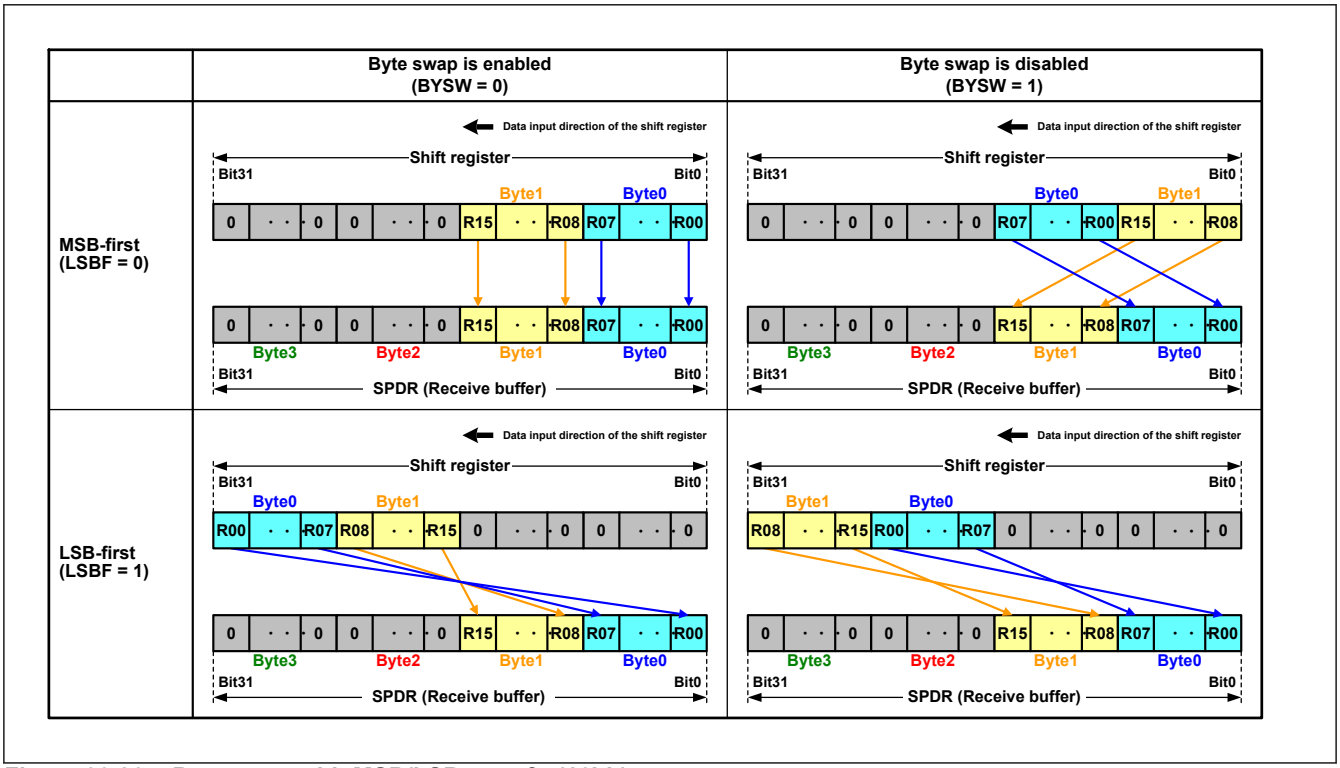


Figure 30.26 Byte swap with MSB/LSB transfer(16bit)

Note: 1. When using the byte swap, set 16 bits or 32 bits to the data length (SPCMDm.SPB[4:0] setting). If setting the other length, the behavior is not guaranteed.
2. When the byte swap is valid, set the parity function as invalid (SPCR.SPPE bit = 0). If setting the parity function as valid (SPPE bit = 1), the behavior is not guaranteed.
3. Set SPDCR.BYSW bit, when SPCR.SPE bit is 0. If rewriting BYSW bit, when SPE bit is 1, the behavior after it is not guaranteed.

30.3.5 Transfer Formats

30.3.5.1 When CPHA = 0

Figure 30.27 shows an example transfer format for the serial transfer of 8-bit data when the SPCMDm.CPHA bit is 0. Do not perform clock synchronous operation (SPCR.SPMS = 1) when the SPI operates in slave mode (SPCR.MSTR = 0) and the CPHA bit is 0. In Figure 30.27, RSPCKn (CPOL = 0) indicates the RSPCKn signal waveform when the SPCMDm.CPOL bit is 0, and RSPCKn (CPOL = 1) indicates the RSPCKn signal waveform when the CPOL bit is 1. The sampling timing represents the timing at which the SPI fetches serial transfer data into the shift register. The I/O directions of the signals depend on the SPI settings. For details, see section 30.3.2. Controlling the SPI Pins.

When the SPCMDm.CPHA bit is 0, the driving of valid data to the MOSIn and MISOn signals begins at an SSLni signal assertion. The first RSPCKn signal change that occurs after the SSLni signal assertion becomes the first transfer data fetch. After this, data is sampled every 1 RSPCKn cycle. The change timing for MOSIn and MISOn signals is 1/2 RSPCK cycles after the transfer data fetch timing. The CPOL bit setting does not affect the RSPCK signal operation timing as it only affects the signal polarity.

t1 denotes the RSPCK delay, the period from an SSLni signal assertion to RSPCKn oscillation. t2 denotes the SSL negation delay, the period from the termination of RSPCKn oscillation to an SSLni signal negation. t3 denotes the next-access delay, the period in which SSLni signal assertion is suppressed for the next transfer after the end of serial transfer. t1, t2, and t3 are controlled by a master device running on the SPI system. For a description of t1, t2, and t3 when the SPI is in master mode, see section 30.3.12.1. Master mode operation.

[In the Motorola-SPI case]

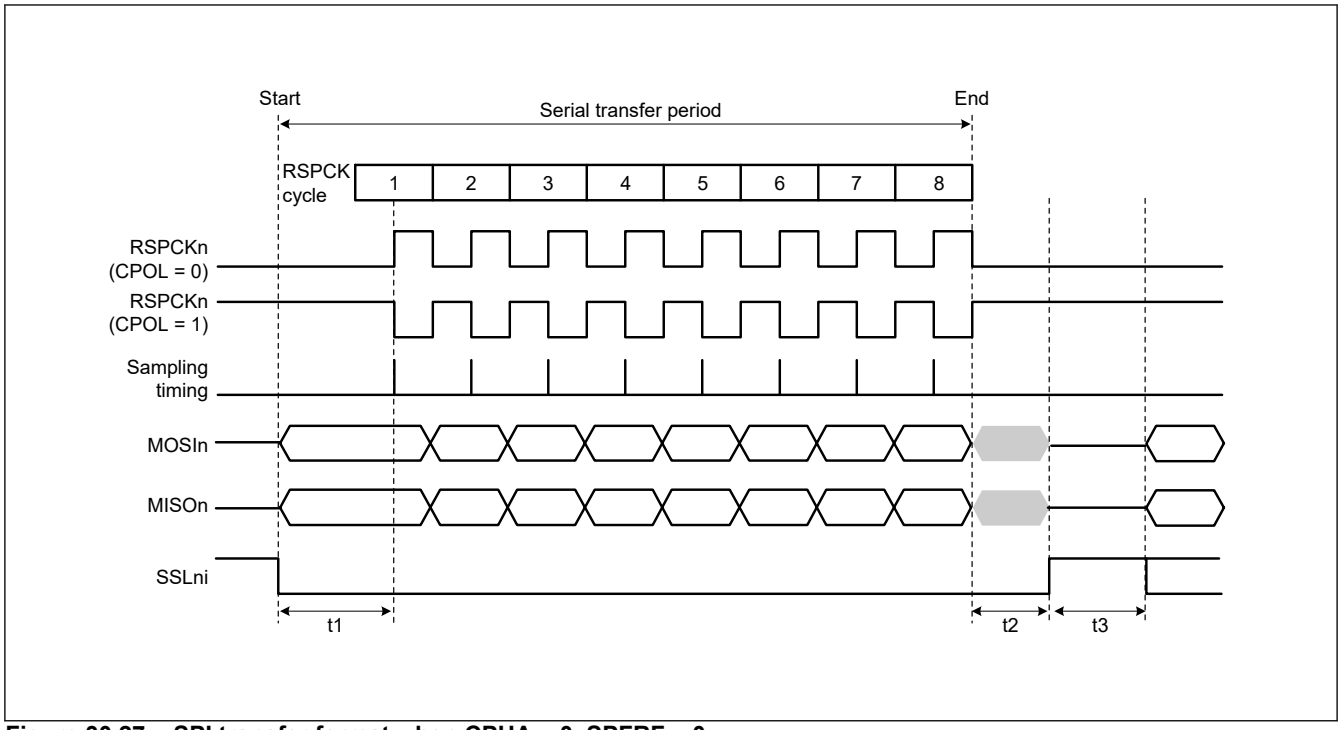


Figure 30.27 SPI transfer format when CPHA = 0, SPFRF = 0

[In TI-SSP case]

Not supported in CPHA = 0

30.3.5.2 When CPHA = 1

Figure 30.28 shows an example transfer format for the serial transfer of 8-bit data when the SPCMDm.CPHA bit is 1. However, when the SPCR.SPMS bit is 1, the SSLni signals are not used, and only the three signals RSPCKn, MOSIn, and MISOn handle communications. In Figure 30.28, RSPCK (CPOL = 0) indicates the RSPCKn signal waveform when the SPCMDm.CPOL bit is 0 and RSPCK (CPOL = 1) indicates the RSPCKn signal waveform when the CPOL bit is 1. The sampling timing represents the timing at which the SPI fetches serial transfer data into the shift register. The I/O directions of the signals depend on the SPI mode (master or slave mode). For details, see section 30.3.2. Controlling the SPI Pins.

When the SPCMDm.CPHA bit is 1, the driving of invalid data to the MISOn signal begins at an SSLni signal assertion. The output of valid data to the MOSIn and MISOn signals begins at the first RSPCKn signal change that occurs after the SSLni signal assertion. After this, data is updated every 1 RSPCK cycle. The transfer data fetch timing is 1/2 RSPCK cycles after the data update timing. The SPCMDm.CPOL bit setting does not affect the RSPCKn signal operation timing. It only affects the signal polarity.

t1, t2, and t3 are the same as those when CPHA = 0. For a description of t1, t2, and t3 when the SPI of the MCU is in master mode, see section 30.3.12.1. Master mode operation.

[In the Motorola-SPI case]

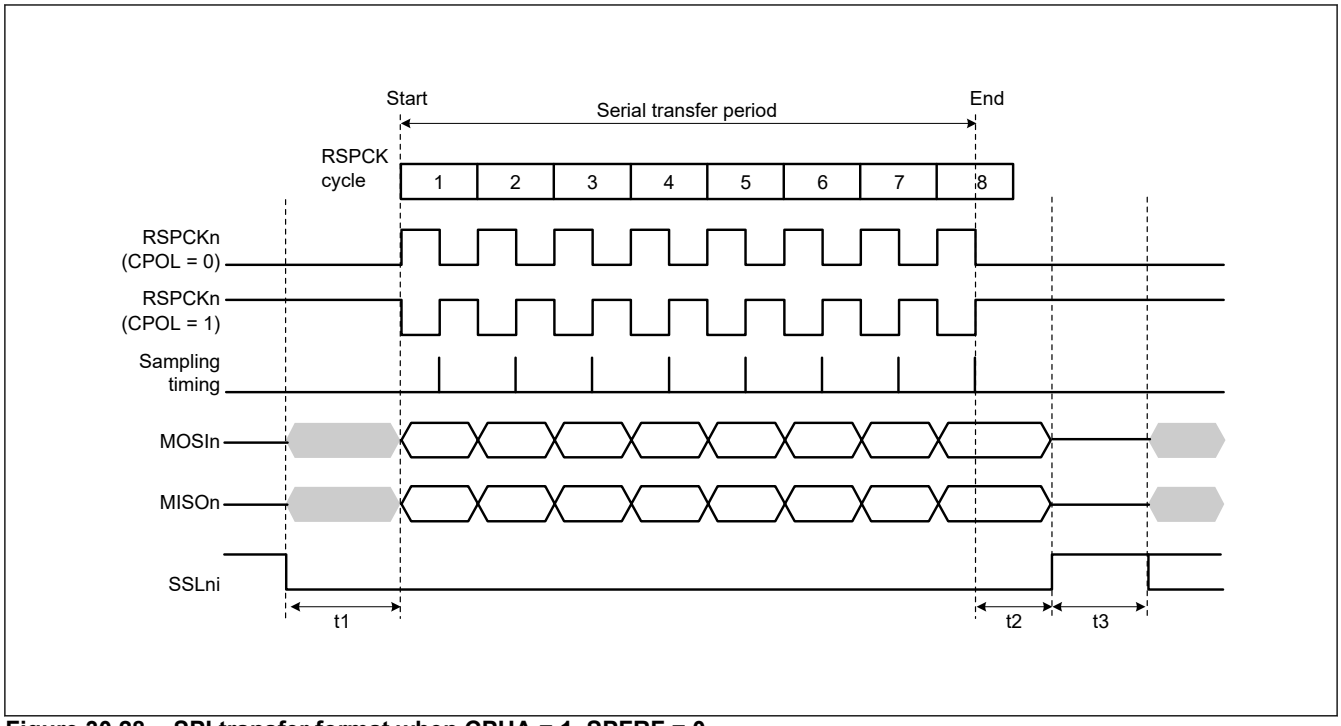


Figure 30.28 SPI transfer format when CPHA = 1, SPFRF = 0

[In the TI-SSP case]

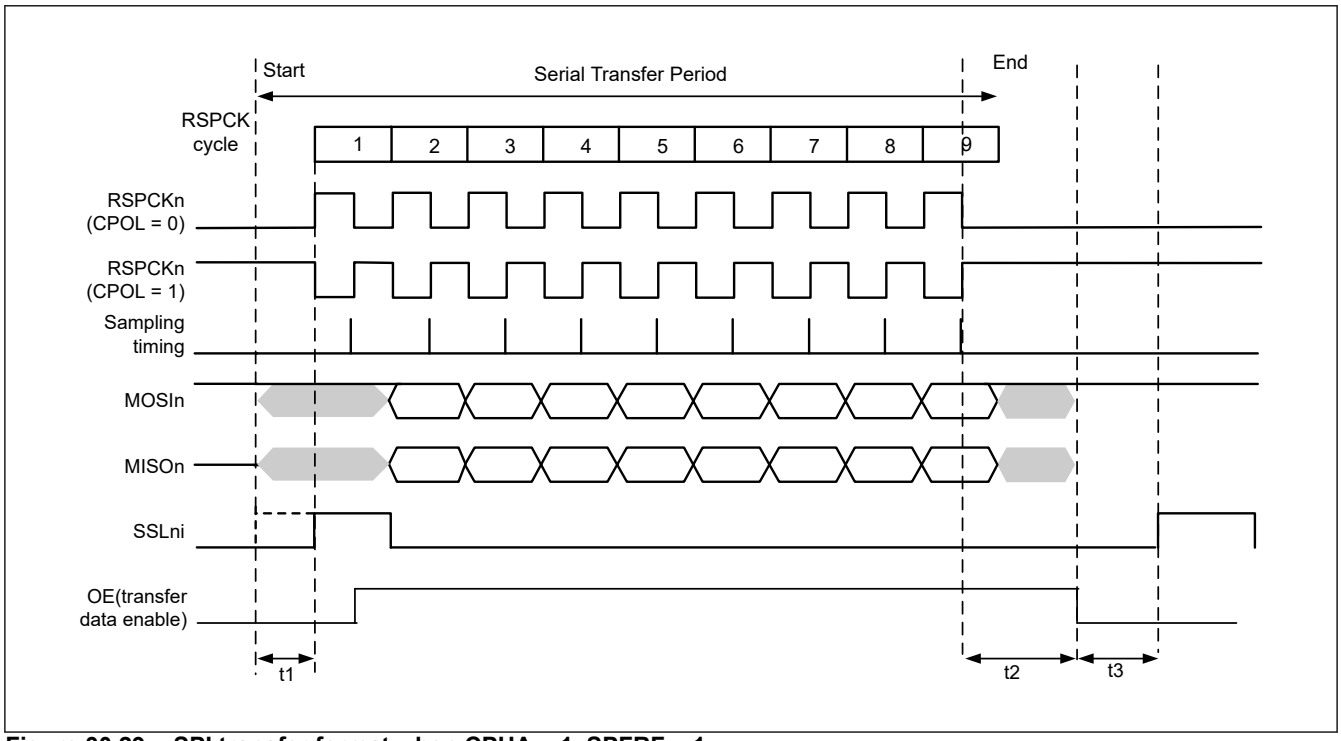


Figure 30.29 SPI transfer format when CPHA = 1, SPFRF = 1

30.3.6 Communication Operating Mode

Transmit-Receive serial communication, transmit-only operation, and Receive-only operation are selected by setting the Communication Mode Select bits (TXMD [1:0]) of the SPI control register (SPCR).

SPDR access described in Figure 30.30, Figure 30.31, Figure 30.32 shows an access to the SPI data register (SPDR). W shows a write cycle.

30.3.6.1 Transmit-Receive Serial Communication (TXMD[1:0] = 00b)

Figure 30.30 shows an example of operation when the communication mode select bit (TXMD[1:0]) in the SPI control register (SPCR) is set to 00b. In the example in Figure 30.30, the SPI performs 8-bit data serial transfer with the settings of RTRG = FIFO stage - 1, TTRG = 0 in the SPI data control register 2 (SPDCR2), CPHA in the SPI command register (SPCMDm) = 1, and CPOL in SPCMDm = 0. Numbers under the RSPCK waveform show the number of RSPCK cycles (number of transfer bits).

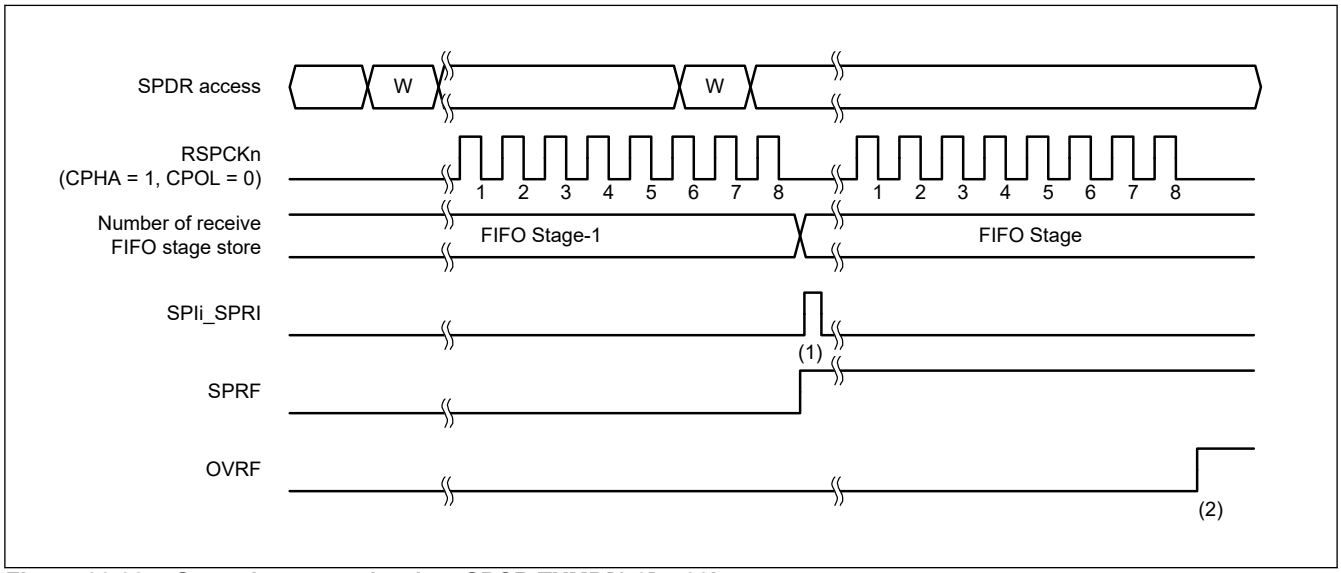


Figure 30.30 Operation example when SPCR.TXMD[1:0] = 00b

The operation of the flags at timings (1) and (2) in Figure 30.30 is as follows:

1. When serial transfer ends while the number of SPDR receive buffer store matches the number of frames set in SPDCR2.RTRG, the SPI generates a receive buffer full interrupt request (SPIi_SPRI), the SPI sets the SPSR.SPRF flag to 1, and copies the received data in the shift register to the receive buffer.
2. When serial transfer ends with data for the number of FIFO stages stored in the SPDR receive buffer, the SPI sets the SPSR.OVRF flag to 1, and discards the received data in the shift register. For details about the operation of the SPSR.OVRF flag, see section 30.3.10.1. Overrun errors.

In Transmit-Receive serial communication (TXMD[1:0] = 00b), transmit data is transmitted and receive data is received. Therefore, the SPRF flag and the OVRF flag are set to 1 at timings (1) and (2) respectively.

30.3.6.2 Transmit-Only Serial Communications (TXMD[1:0] = 01b)

Figure 30.31 shows an example of operation when the communication mode select bit (TXMD[1:0]) in the SPI control register (SPCR) is set to 01b. In this example, the SPI performs an 8-bit serial transfer when the SPDCR2.TTRG is 0, the SPDCR2.RTRG is 0, the SPCMDm.CPHA bit is 1, and the SPCMDm.CPOL bit is 0. The numbers given for RSPCKn in the waveform represent the number of RSPCK cycles, such as the number of transferred bits.

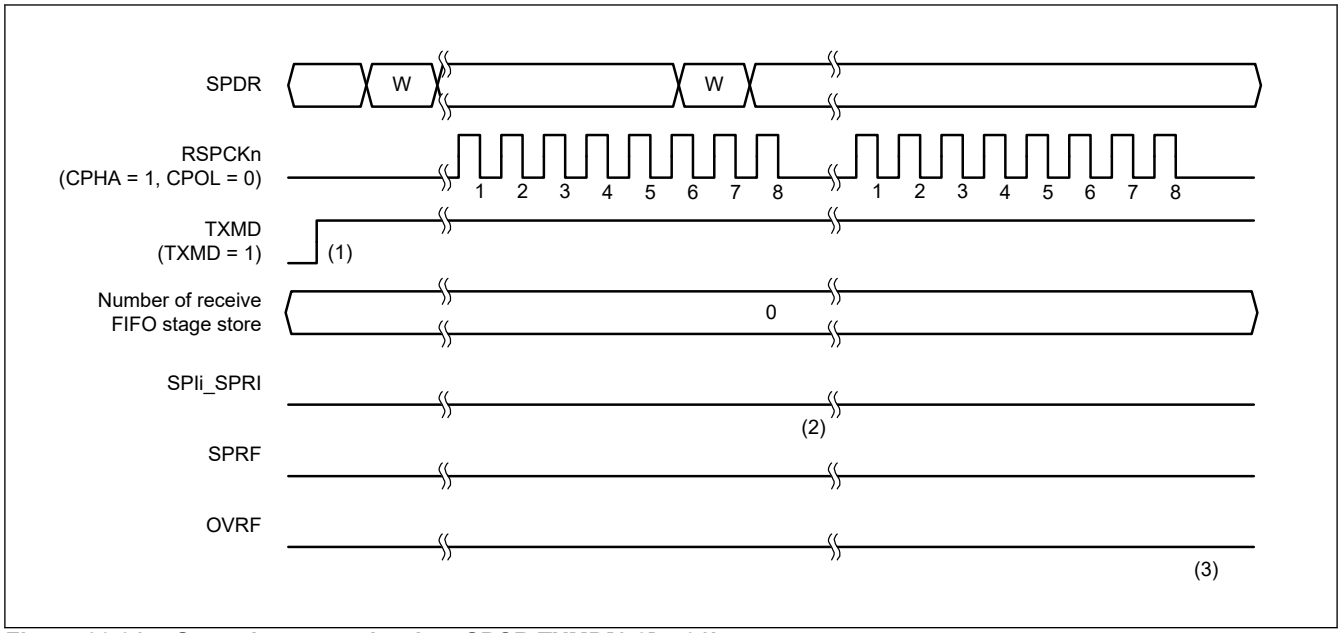


Figure 30.31 Operation example when SPCR.TXMD[1:0] = 01b

The operation of the flags at timings (1) to (3) in Figure 30.31 is as follows:

1. Make sure there is no data left in the receive buffer (the SPSR.SPRF flag is 0) and the SPSR.OVRF flag is 0 before entering transmit-only mode (SPCR.TXMD[1:0] = 01b).
2. When a serial transfer ends without receiving data in the receiving FIFO of SPDR, if the transmit-only mode is selected (SPCR.TXMD[1:0] = 01b), the SPSR.SPRF flag retains the value of 0, and the SPI does not copy the data in the shift register to the receive buffer.
3. Because the receive buffer of SPDR does not hold data that was received in the previous serial transfer, even when a serial transfer ends, the SPSR.OVRF flag retains the value of 0, and the data in the shift register is not copied to the receive buffer.

In transmit-only mode (SPCR.TXMD[1:0] = 01b), the SPI transmits data but does not receive data. Therefore, the SPSR.SPRF and SPSR.OVRF flags remain 0 at timings (1) to (3).

30.3.6.3 Receive-Only Serial Communication (TXMD[1:0] = 10b)

Figure 30.32 shows an example of operation when the communication mode select bit (TXMD[1]) in the SPI control register (SPCR) is set to 1. In the example in Figure 30.32, the SPI performs 8-bit data serial transfer with the settings of TTRG = FIFO stage - 1, RTRG = 0 in the SPI data control register 2 (SPDCR2), CPHA in the SPI command register (SPCMDm) = 1, and CPOL in SPCMDm = 0. Numbers under the RSPCK waveform show the number of RSPCK cycles (number of transfer bits).

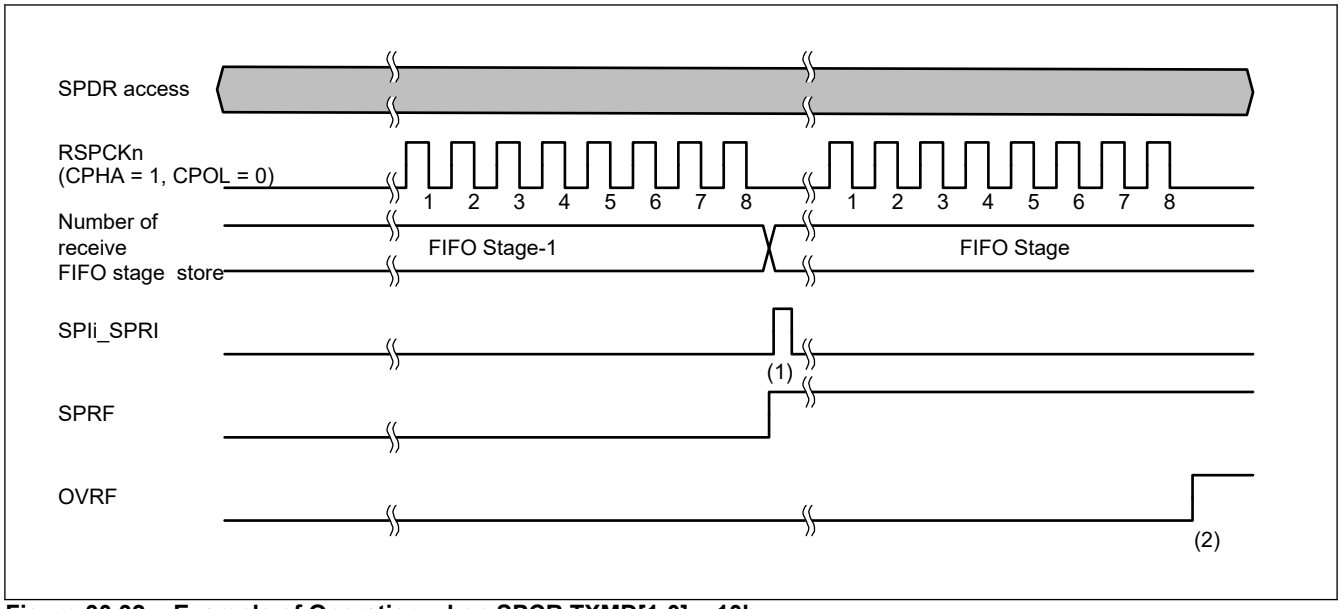


Figure 30.32 Example of Operation when SPCR.TXMD[1:0] = 10b

The following describes operation of flags at timings (1) and (2) in the figure above.

1. When serial transfer ends while the number of SPDR receive buffer store matches the number of frames set in SPDCR2.RTRG, the SPI generates a receive buffer full interrupt request (SPIi_SPRI), the SPI sets the SPSR.SPRF flag to 1, and copies the received data in the shift register to the receive buffer
2. When serial transfer ends with data for the number of FIFO stages stored in the SPDR receive buffer, the SPI sets the SPSR.OVRF flag to 1 and discards the received data in the shift register.

30.3.7 Transmit Buffer Empty and Receive Buffer Full Interrupts

Figure 30.33 show examples of operation of the transmit buffer empty interrupt (SPIi_SPTI) and the receive buffer full interrupt (SPIi_SPRI). The SPDR register accesses shown in these figures indicate the conditions of access to the register, where W denotes a write cycle and R a read cycle. In Figure 30.33, the SPI performs an 8-bit serial transfer when SPCR.TXMD[1:0] bits are 00b, the SPDCR2.TTRG bit is 0, the SPDCR2.RTRG bit is 0, the SPCMDm.CPHA bit is 0, and the SPCMDm.CPOL bit is 0. The numbers given for RSPCKn in the waveform represent the number of RSPCK cycles, such as the number of transferred bits.

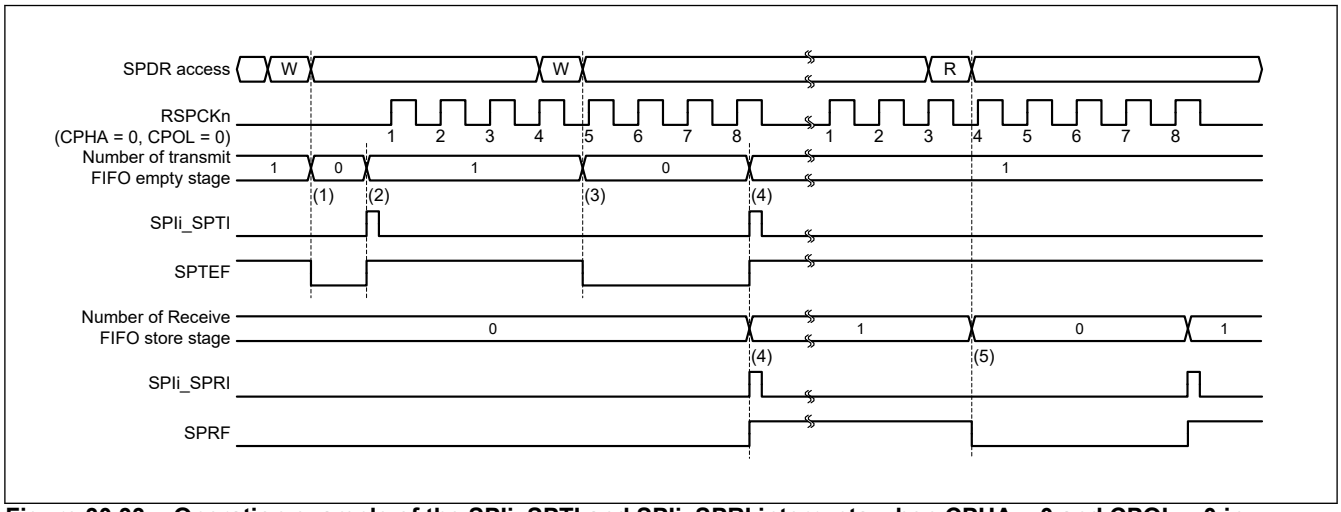


Figure 30.33 Operation example of the SPIi_SPTI and SPIi_SPRI interrupts when CPHA = 0 and CPOL = 0 in master mode

The operation of the SPI at timings (1) to (5) in Figure 30.33 is as follows:

1. When transmit data is written to SPDR with the transmit buffer of SPDR is before the next transfer data is set, the SPI writes data to the transmit buffer. When transmit data is written to SPDR in one processing routine using DTC / DMAC, the SPSR.SPTEF flag is cleared to 0 at the last access.
2. If the shift register is empty, the SPI copies the data in the transmit buffer to the shift register. At this time, if transmit FIFO empty stage number > TTRG value, then the SPI generates a transmit buffer empty interrupt request (SPIi_SPTI), and sets the SPSR.SPTEF flag to 1. How a serial transfer is started depends on the SPI mode. For details, see section 30.3. Operation, and section 30.3.13. Clock Synchronous Operation.
3. When transmit data is written to SPDR either by the transmit buffer empty interrupt routine, or by the processing of the transmit buffer empty using the SPTEF flag, the SPI writes data to the transmit buffer. When the transmit data is written to SPDR in one processing routine using DTC / DMAC, the SPTEF flag is cleared to 0 at the last access. Because the data being transferred serially is stored in the shift register, the SPI does not copy the data in the transmit buffer to the shift register.
4. When the serial transfer ends with the receive buffer of SPDR > FIFO stage number, the SPI copies the receive data in the shift register to the receive buffer, generates a receive buffer full interrupt request (SPIi_SPRI), and sets the SPRF flag to 1. Because the shift register becomes empty on completion of the serial transfer, if the next transfer data is set in the transmit FIFO before the serial transfer ended, the SPI sets the SPTEF flag to 1 and copies data in the transmit buffer to the shift register. Even when received data is not copied from the shift register to the receive buffer in an overrun error status, on completion of the serial transfer, the SPI determines that the shift register is empty, so data transfer from the transmit buffer to the shift register is enabled.
5. When SPDR is read either by the receive buffer full interrupt routine or processing of the receive buffer full interrupt using the SPRF flag, the receive data can be read. If the received data is read from SPDR in one processing routine using DTC / DMAC, the SPRF flag is cleared to 0 at the last access.

When transmit data is written to the SPDR register while no empty stages in the transmit FIFO, the SPI does not update data in the transmit buffer. When writing to SPDR, always use either a transmit buffer empty interrupt request or check the empty or processing of the transmit buffer empty interrupt using the SPTEF flag. To use a transmit buffer empty interrupt, set the SPTIE bit in SPCR to 1. If the SPI function is disabled (the SPCR.SPE bit is 0), set the SPTIE bit to 0.

When serial transfer ends while data is stored in the receive FIFO for the number of FIFO stages, the SPI does not copy data from the shift register to the receive buffer, and it detects an overrun error (see section 30.3.10. Error Detection). To prevent a receive data overrun error, read the received data using a receive buffer full interrupt request before the next serial transfer ends. To use an SPI receive buffer full interrupt, set the SPCR.SPRIE bit to 1.

Transmission and reception interrupts or the associated IELSRn.IR flags (where n is the interrupt vector number) in the ICU can be used to confirm the states of the transmit and receive buffers.

Similarly, the SPTEF and SPRF flags can be used to confirm the states of the transmit and receive buffers. See section 12, Interrupt Controller Unit (ICU) for the interrupt vector numbers.

30.3.8 Idle Interrupt

When the SPCP[2:0] of the SPI status register (SPSR) becomes 000b (start of sequence control), the IDLNF flag in the SPI status register (SPSR) is set to 1 and an idle interrupt request is made during master mode operation. An interrupt request is also made by clearing the SPCR.SPE bit to 0.

[In the Motorola-SPI case]

Figure 30.34 shows an example of idle interrupt operation during normal operation.

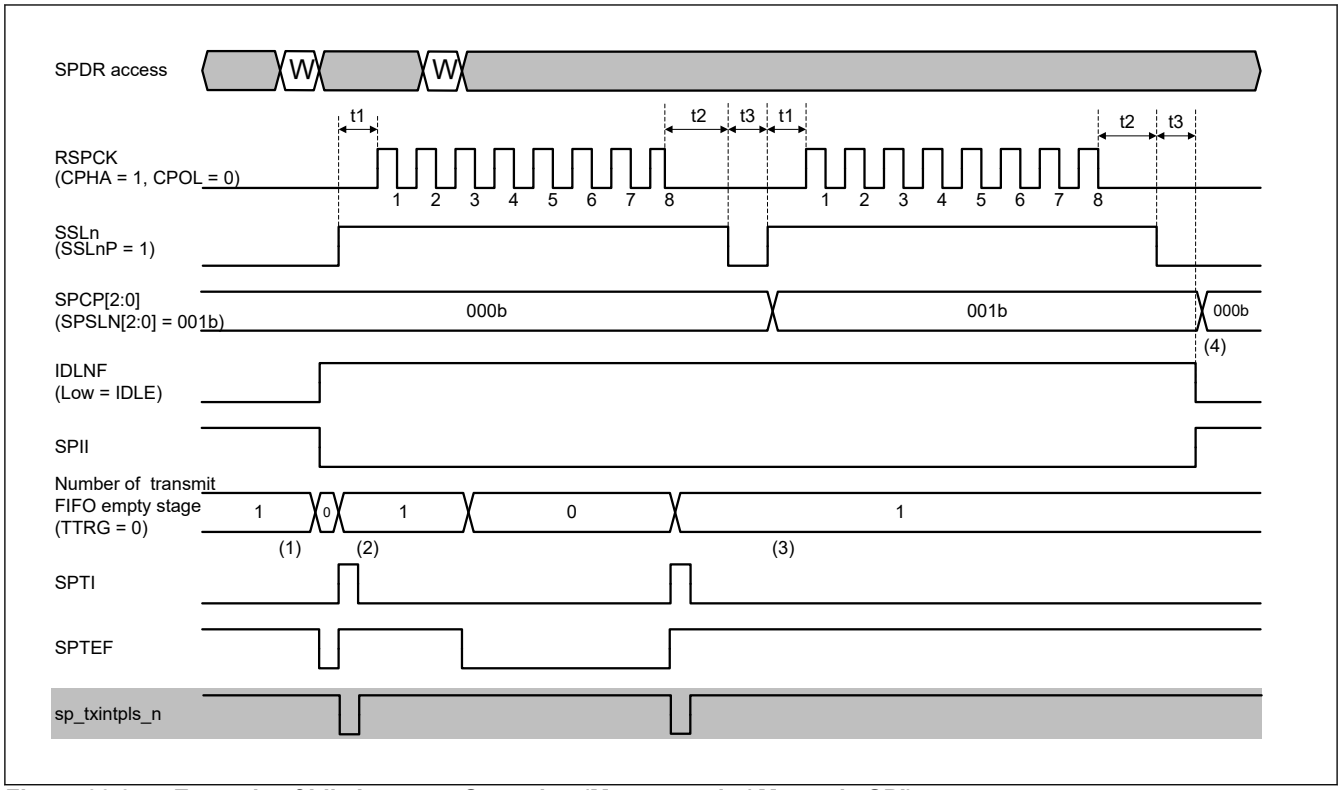


Figure 30.34 Example of Idle Interrupt Operation (Master mode / Motorola-SPI)

1. At the start of transmission, if the next transfer data is not set in the transmission buffer, the IDLNF flag is 0 (IDLE). Writing transmit data sets the IDLNF flag to 1 (BUSY). When the SPIIE bit in the SPI control register (SPCR) is set to 1 before transmit data is written, interrupt processing is required before transmission start. For this reason, set the SPIIE bit to 0 before starting transmission.
2. After transmission has started, the IDLNF flag remains 1 (BUSY) regardless of the transmit buffer state.
3. The SPCP[2:0] bits change the command to the next command at the end of t3 cycle. When the next command is not 000b, the IDLNF flag remains unchanged even when the next transmit data has not been written.
4. The IDLNF flag is cleared to 0 (IDLE) at the end of t3 cycle because the next command is 000b and the next transmit data is not present. When the SPIIE bit is 1 currently, an SPIi_SPII interrupt is output.

[In the TI-SSP case]

Figure 30.35 shows an example of idle interrupt operation during normal operation.

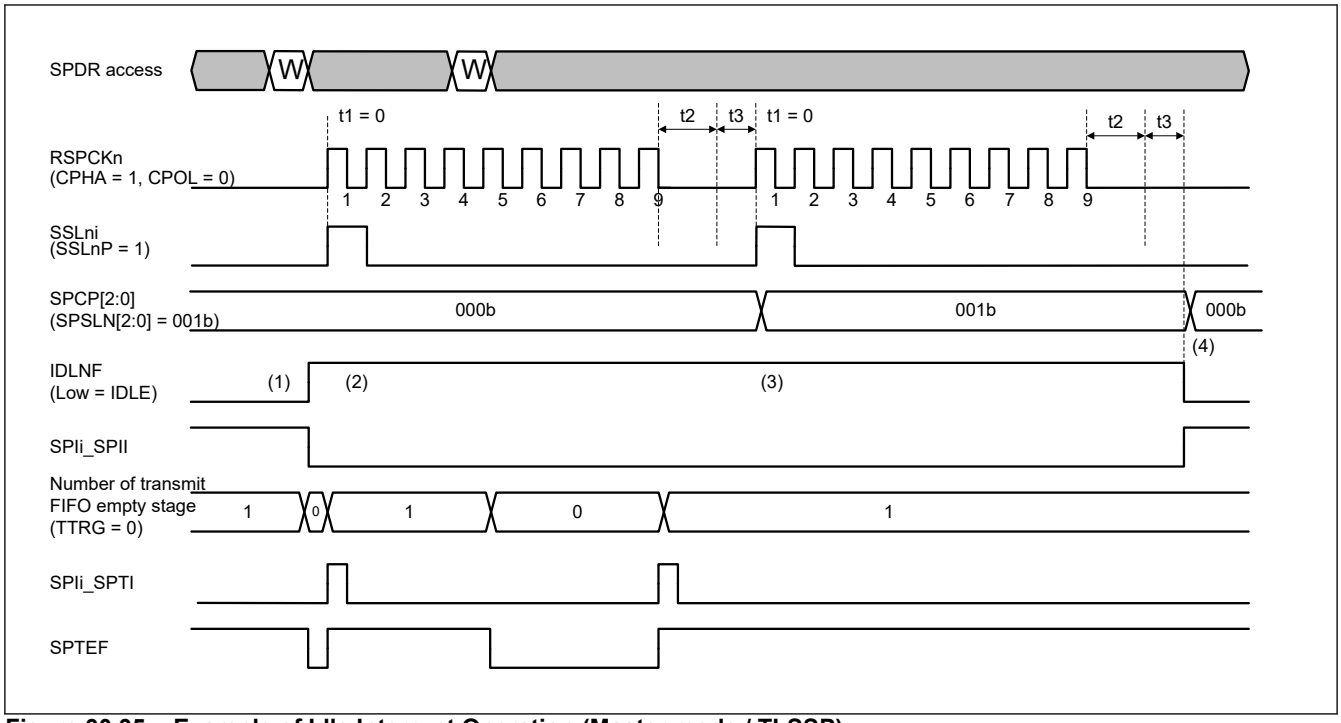


Figure 30.35 Example of Idle Interrupt Operation (Master mode / TI-SSP)

1. At the start of transmission, if the next transfer data is not set in the transmission buffer, the IDLNF flag is 0 (IDLE). Writing transmit data makes sets the IDLNF flag to 1 (BUSY). When the SPIIE bit in the SPI control register (SPCR) is set to 1 before transmit data is written, interrupt processing is required before transmission start. For this reason, set the SPIIE bit to 0 before starting transmission.
2. After transmission has started, the IDLNF flag remains 1 (BUSY) regardless of the transmit buffer state.
3. The SPCP[2:0] bits change the command to the next command at the end of t3 cycle. When the next command is not 000b, the IDLNF flag remains unchanged even when the next transmit data has not been written.
4. The IDLNF flag is cleared to 0 (IDLE) at the end of t3 cycle because the next command is 000b and the next transmit data is not present. When the SPIIE bit is 1 currently, an SPIi_SPII interrupt is output.

30.3.9 Communication End Interrupt

30.3.9.1 Transmit-Receive/Transmit in Master Mode

See the description of the CENDF bit in section 30.2.9. SPSR : SPI Status Register for the setting / clearing conditions of the communication completion flag during Transmit-Receive/Transmit-only in Master Mode.

[In the Motorola-SPI case]

Figure 30.36 shows an example of communication end interrupt operation during transmit-receive/transmit master mode.

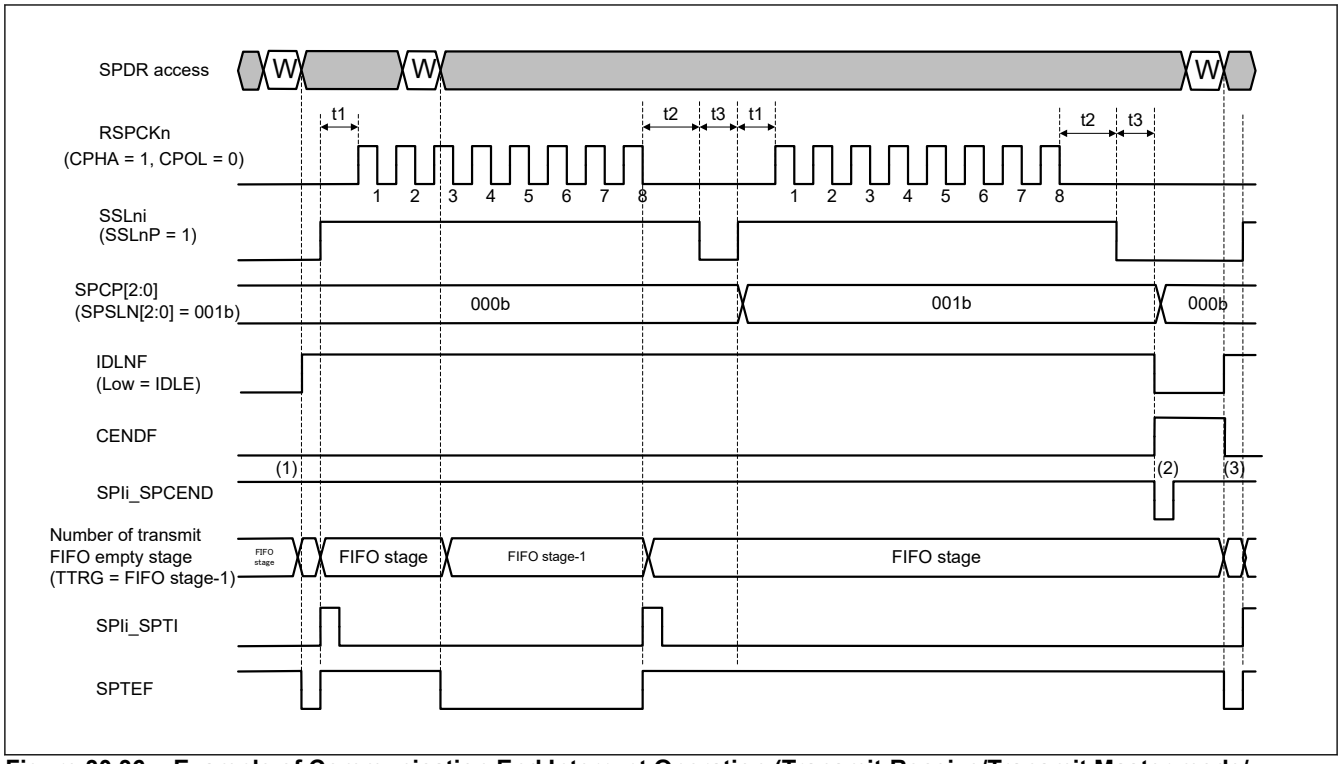


Figure 30.36 Example of Communication End Interrupt Operation (Transmit-Receive/Transmit Master mode/ Motorola-SPI)

1. The CENDF flag is 0 and the level of SPIi_SPCEND is 1 before communication start, and these have kept during communication.
2. The CENDF flag will be 1 (Communication End) at the end of t3 cycle because the next command is 000b and there is no next transmit data, and then the SPIi_SPCEND interrupt output when the CENDIE bit is 1.
3. The CENDF flag is cleared when the next transmission data is written to the transmit buffer (SPTX). Or when 1 is written to the SPSRC.CENDFC bit, then the CENDF flag is 0.

[In the TI-SSP case]

Figure 30.37 shows an example of communication end interrupt operation during transmit-receive/transmit-only master mode.

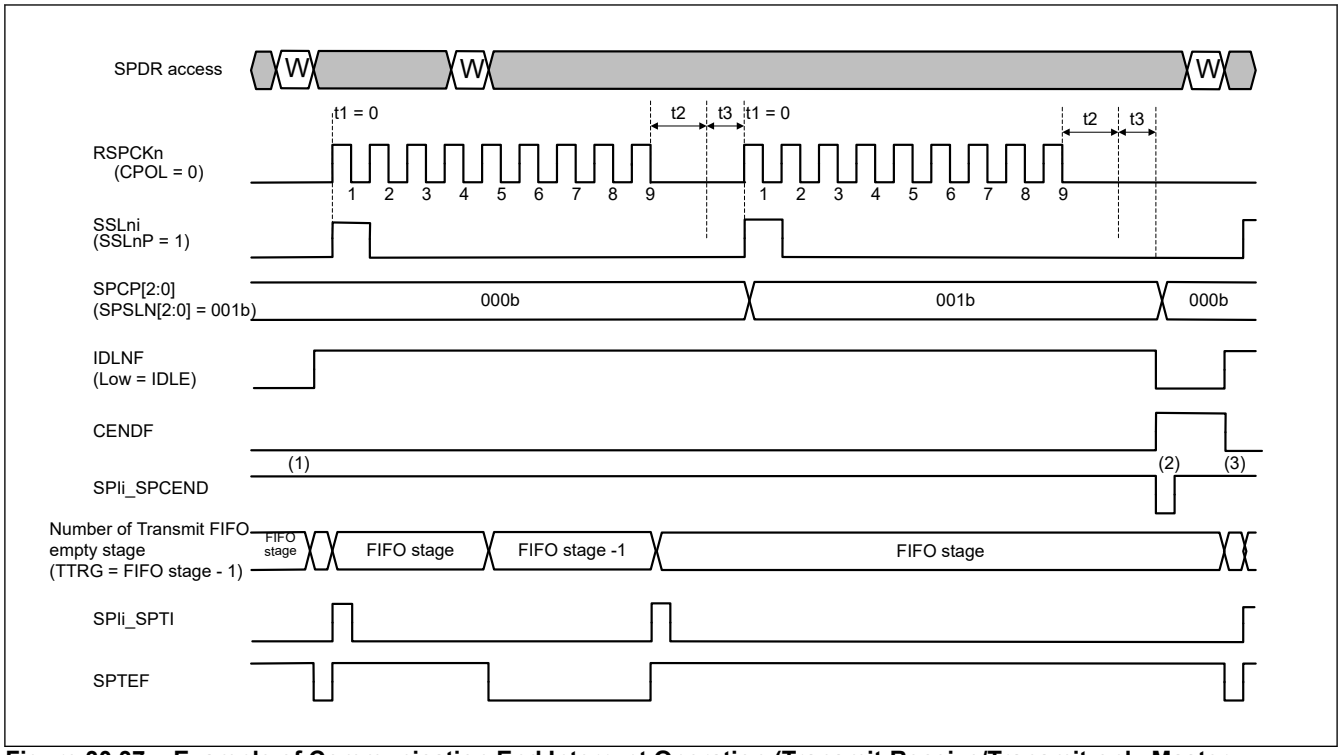


Figure 30.37 Example of Communication End Interrupt Operation (Transmit-Receive/Transmit-only Master mode/TI-SSP)

1. The CENDF flag is 0 and the level of SPIi_SPCEND is 1 before communication start, and these have kept during communication.
2. The CENDF flag will be 1 (Communication End) at the end of t3 cycle because the next command is 000b and there is no next transmit data, and then the SPIi_SPCEND interrupt outputs with PCLK 1 cycle width if the CENDIE bit is 1.
3. The CENDF flag is cleared when the next transmission data is written to the transmit buffer (SPTX). Or when 1 is written to the SPSRC.CENDFC bit, then the CENDF flag is 0.

In slave mode operation, the output timing of the communication end interrupt is deferent due to the value of the SPCR.SPMS bit (SPI mode select bit), and the clear timing of the communication end interrupt is deferent due to the communication mode (transmit-receive or transmit-only or receive-only).

30.3.9.2 Receive-only in Master Mode

See the description of the CENDF bit in section 30.2.9. SPSR : SPI Status Register for the setting / clearing conditions of the communication completion flag during Receive-only in Master Mode.

Figure 30.38 shows an example of communication end interrupt operation during receive-only master mode at RMFM[4:0] = 0.

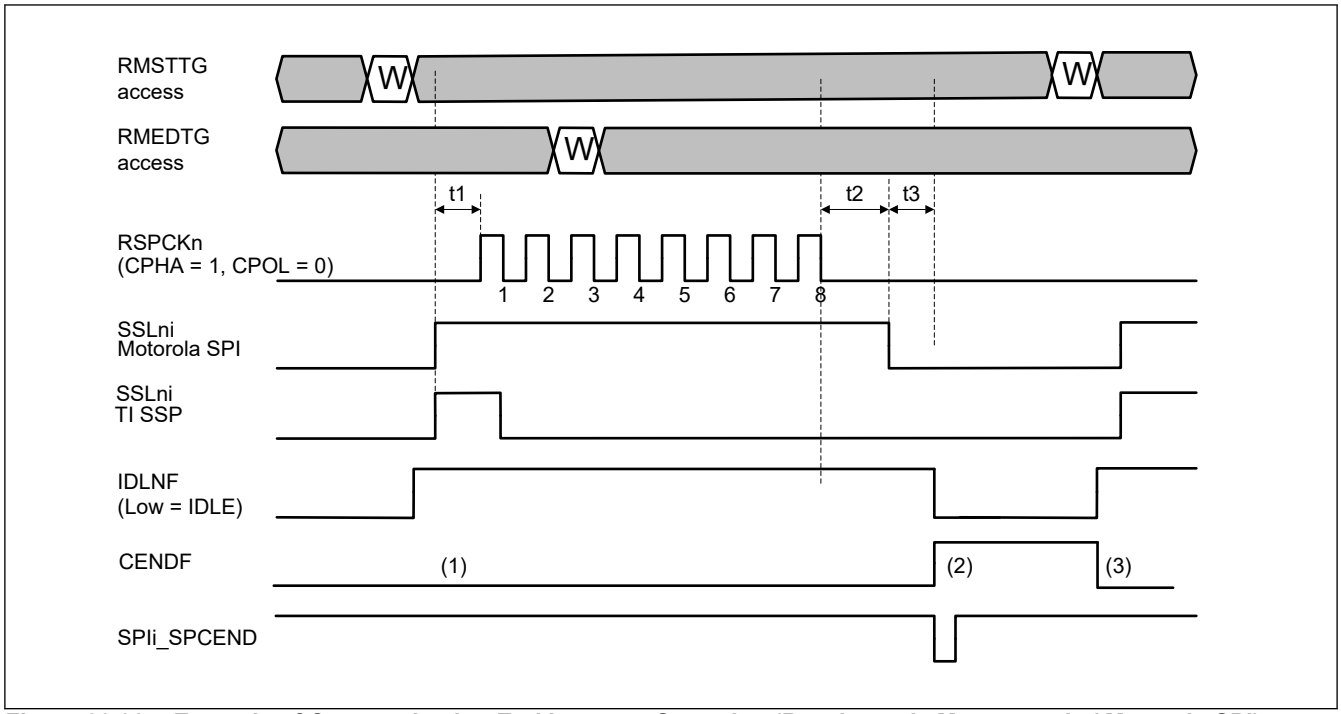


Figure 30.38 Example of Communication End Interrupt Operation (Receive-only Master mode / Motorola-SPI) at RMFM [4:0] = 0

1. The CENDF flag is 0 and the level of SPIi_SPCEND is 1 before communication start. These have kept during communication.
2. The CENDF flag will be 1 (Communication End) at the end of t3 cycle by writing 1 to RMEDTG during the communication frame. Then the SPIi_SPCEND interrupt outputs with PCLK 1 cycle width if the CENDIE bit is 1.
3. The CENDF flag is cleared when writing 1 to RMSTTG. Also when 1 is written to the SPSRC.CENDFC bit, then the CENDF flag is 0.

Figure 30.39 shows an example of communication end interrupt operation during receive-only master mode at RMFM[4:0] ≠ 0.

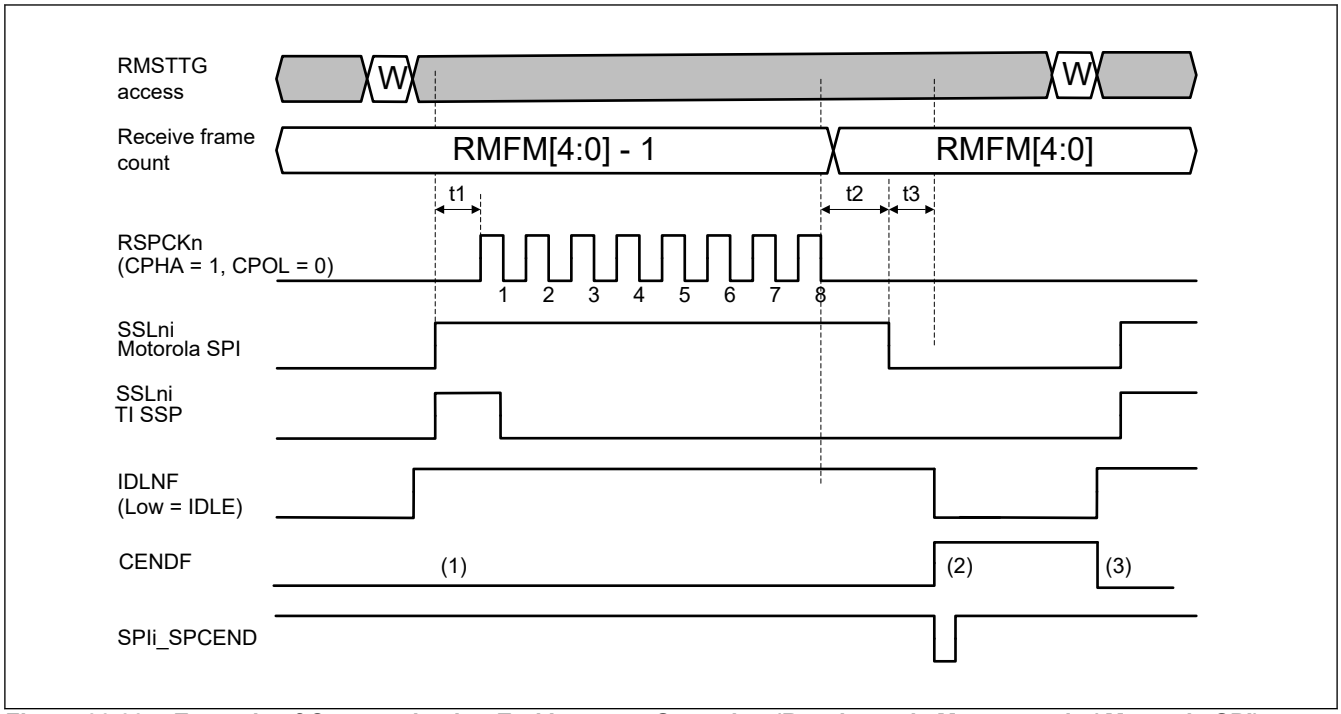


Figure 30.39 Example of Communication End Interrupt Operation (Receive-only Master mode / Motorola-SPI) at RMFM [4:0] ≠ 0

1. The CENDF flag is 0 and the level of SPIi_SPCEND is 1 before communication start. These have kept during communication.
2. The CENDF flag will be 1 (Communication End) at the end of t3 cycle after receiving the number of frames set by RMFM[4:0]. Then the SPIi_SPCEND interrupt outputs with PCLK 1 cycle width if the CENDIE bit is 1.
3. The CENDF flag is cleared when writing 1 to RMSTTG. Also when 1 is written to the SPSRC.CENDFC bit, then the CENDF flag is 0.

30.3.9.3 Transmit-Receive/Transmit in Slave Mode on SPI Operation (4-wire)

See the description of the CENDF bit in section 30.2.9. SPSR : SPI Status Register for the setting / clearing conditions of the communication completion flag during Transmit-Receive/Transmit-only in Slave Mode (4-wire).

[In the Motorola-SPI case]

Figure 30.40 shows an example of communication end interrupt operation during transmit-receive/transmit slave mode on SPI operation.

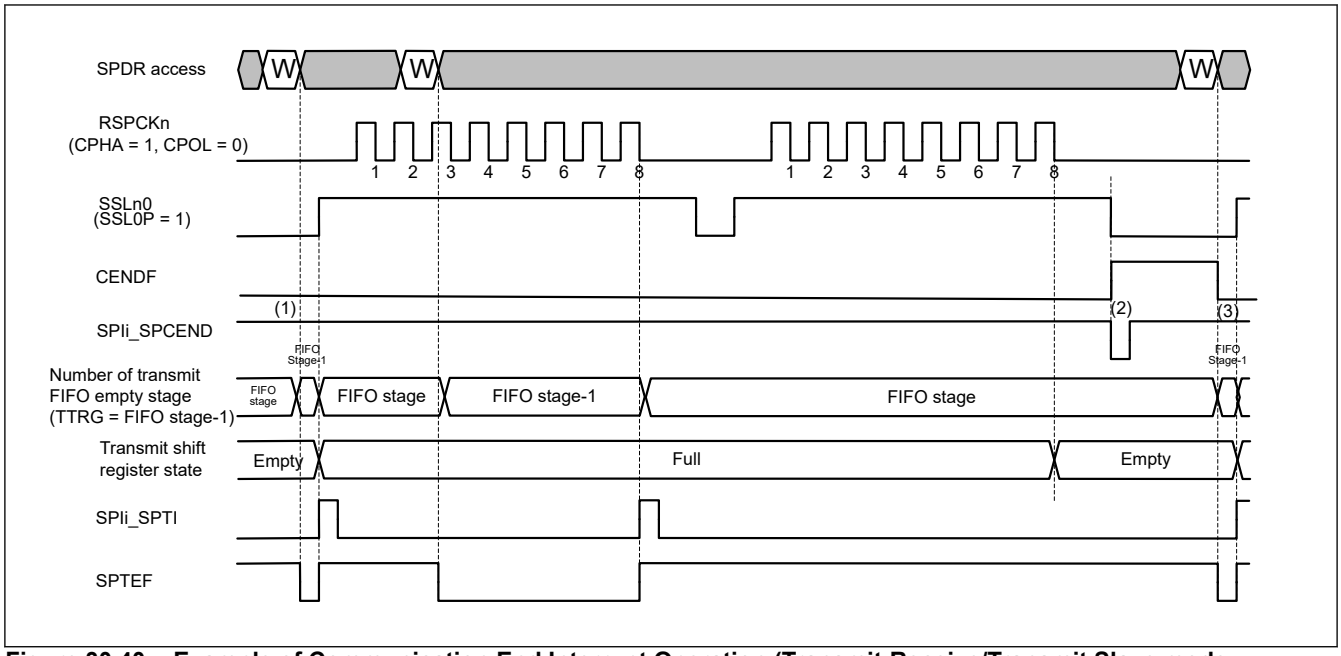


Figure 30.40 Example of Communication End Interrupt Operation (Transmit-Receive/Transmit Slave mode on SPI Operation/Motorola-SPI)

1. The CENDF flag is 0 and the level of SPIi_SPCEND is 1 before communication start, and these have kept during communication.
2. The CENDF flag will be 1 (Communication End) at the timing of SSLn0 negate when the next transfer data is not set in the transmit FIFO and the transmit shift register is empty. Then the SPIi_SPCEND interrupt outputs with PCLK 1 cycle width if the CENDIE bit is 1.
3. The CENDF flag is cleared when the next transmission data is written to the transmit buffer (SPTX). Or when 1 is written to the SPSRC.CENDFC bit, then the CENDF flag is 0.

[In the TI-SSP case]

Figure 30.41 shows an example of communication end interrupt operation during transmit-receive/transmit-only slave mode on SPI operation.

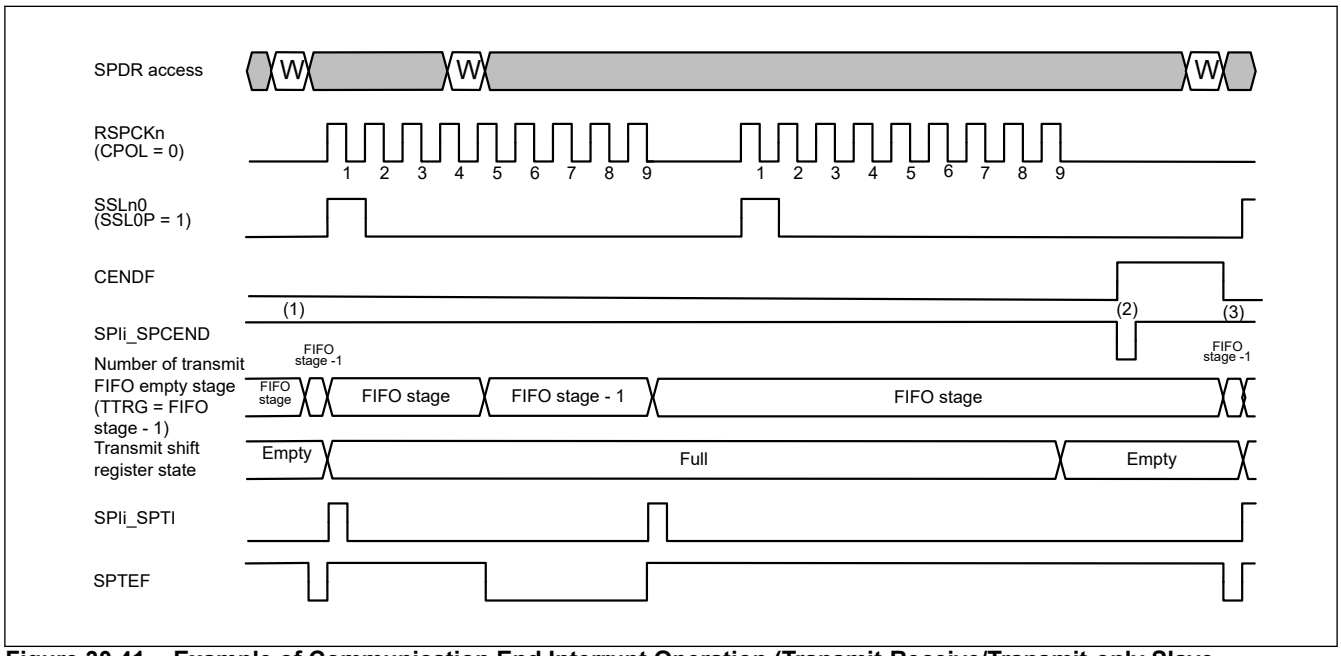


Figure 30.41 Example of Communication End Interrupt Operation (Transmit-Receive/Transmit-only Slave mode on SPI Operation / TI-SSP)

1. The CENDF flag is 0 and the level of SPIi_SPCEND is 1 before communication start, and these have kept during communication.
2. The CENDF flag will be 1 (Communication End) at the RSPCKn last data bit sampling when the next transfer data is not set in the transmit FIFO and the transmit shift register is empty. Then the SPIi_SPCEND interrupt outputs with PCLK 1 cycle width if the CENDIE bit is 1.
3. The CENDF flag is cleared when the next transmission data is written to the transmit buffer (SPTX). Or when 1 is written to the SPSRC.CENDFC bit, then the CENDF flag is 0.

30.3.9.4 Receive Only in Slave Mode on SPI Operation (4-wire)

See the description of the CENDF bit in section 30.2.9. SPSR : SPI Status Register for the setting / clearing conditions of the communication completion flag during Receive-only in Slave Mode (4-wire).

[In the Motorola-SPI case]

Figure 30.42 shows an example of communication end interrupt operation during receive only slave mode on SPI operation (4-wire).

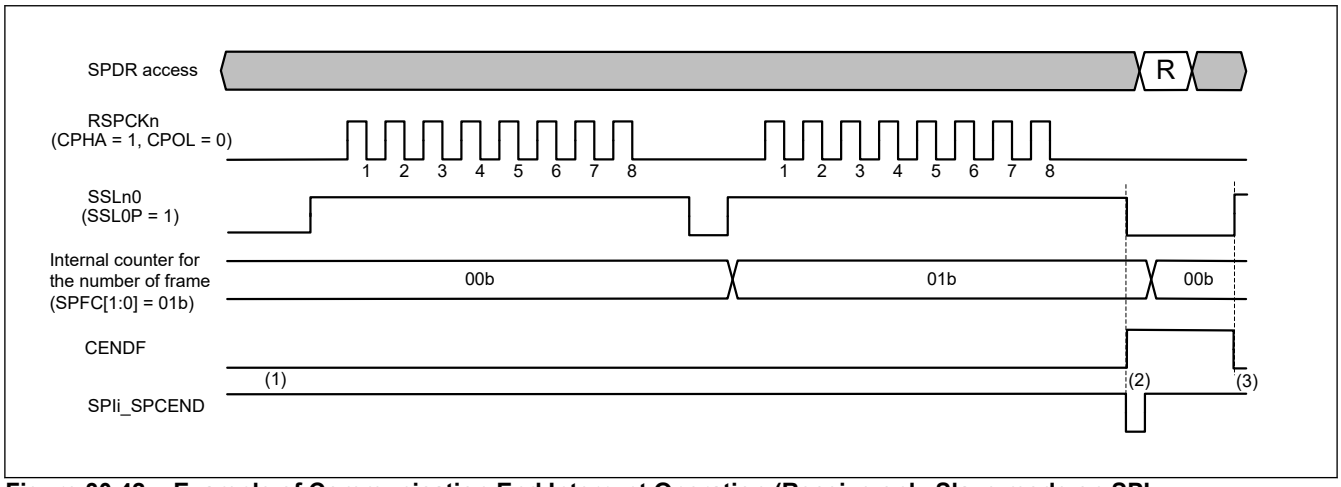


Figure 30.42 Example of Communication End Interrupt Operation (Receive only Slave mode on SPI Operation / Motorola-SPI)

1. The CENDF flag is 0 and the level of SPIi_SPCEND is 1 before communication start, and these have kept during communication.
2. After the frames for SPFC set value in the SPI data control register (SPDCR) are stored in the receive buffer, the CENDF flag becomes 1 (communication completed) at the timing of SSLn0 negation. Then the SPIi_SPCEND interrupt outputs with PCLK 1 cycle width if the CENDIE bit is 1.
3. The CENDF flag is cleared at the SSLn0 assert when the next transmission start. Or when 1 is written to the SPSRC.CENDFC bit, then the CENDF flag is 0.

[In the TI-SSP case]

Figure 30.43 shows an example of communication end interrupt operation during receive only slave mode on SPI operation (4-wire).

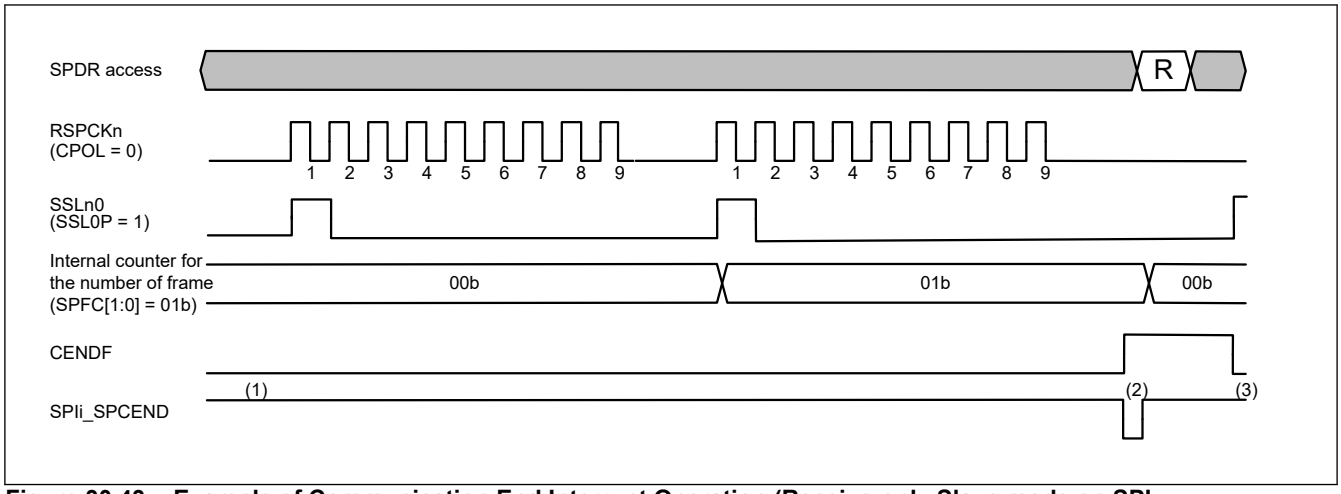


Figure 30.43 Example of Communication End Interrupt Operation (Receive-only Slave mode on SPI Operation / TI-SSP)

1. The CENDF flag is 0 and the level of SPIi_SPCEND is 1 before communication start, and these have kept during communication.
2. The CENDF flag will be 1 (Communication End) at the RSPCK last data bit sampling when the last frame transmission ends. Then the SPIi_SPCEND interrupt outputs with PCLK 1 cycle width if the CENDIE bit is 1.
3. The CENDF flag is cleared at the SSLn0 assert when the next transmission start. Or when 1 is written to the SPSRC.CENDFC bit, then the CENDF flag is 0.

30.3.9.5 Transmit-Receive/Transmit in Slave Mode on Clock Synchronous Operation (3-wire)

See the description of the CENDF bit in section 30.2.9. SPSR : SPI Status Register for the setting / clearing conditions of the communication completion flag during Transmit-Receive/Transmit-only in Slave Mode on Clock Synchronous (3-wire).

Figure 30.44 shows an example of communication end interrupt operation during transmit-receive/transmit slave mode on clock synchronous operation (3-wire).

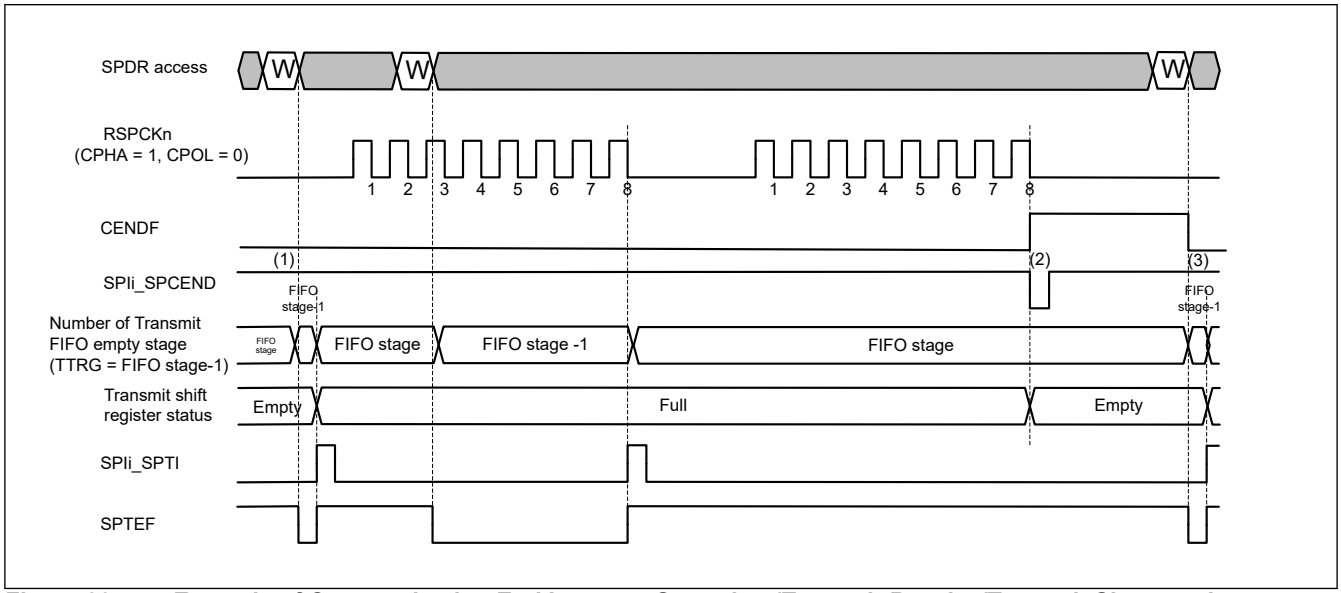


Figure 30.44 Example of Communication End Interrupt Operation (Transmit-Receive/Transmit Slave mode on Clock Synchronous Operation)

1. The CENDF flag is 0 and the level of SPIi_SPCEND is 1 before communication start, and these have kept during communication.

2. When the next transfer data is not set in the transmit FIFO and the transmit shift register is empty, then the SPIi_SPCEND interrupt outputs with PCLK 1 cycle width if the CENDIE bit is 1.
3. The CENDF flag is cleared when the next transmission data is written to the transmit buffer (SPTX). Or when 1 is written to the SPSRC.CENDFC bit, then the CENDF flag is 0.

30.3.9.6 Receive Only in Slave Mode on Clock Synchronous Operation (3-wire)

See the description of the CENDF bit in section 30.2.9. SPSR : SPI Status Register for the setting / clearing conditions of the communication completion flag during Receive -only in Slave Mode on Clock Synchronous (3-wire).

Figure 30.45 shows an example of communication end interrupt operation during receive only slave mode on clock synchronous operation.

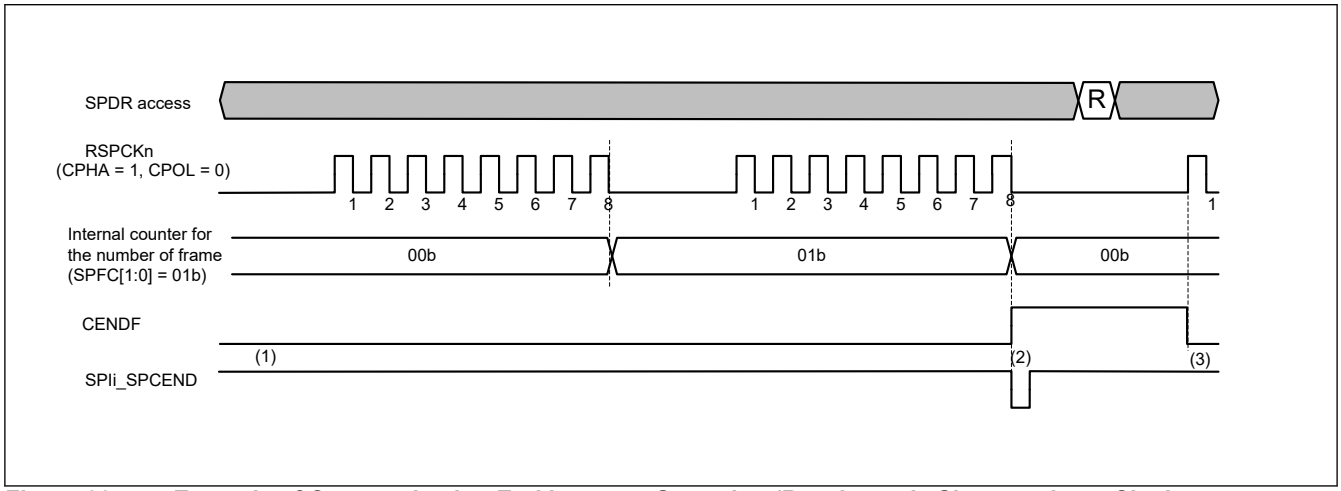


Figure 30.45 Example of Communication End Interrupt Operation (Receive-only Slave mode on Clock Synchronous Operation)

1. The CENDF flag is 0 and the level of SPIi_SPCEND is 1 before communication start, and these have kept during communication.
2. The CENDF flag is set to 1 (communication completed) at the timing of the last data bit sampling of RSPCKn in the last frame communication when the last frame of the SPI data control register (SPDCR) SPFC set value is received. Then the SPIi_SPCEND interrupt outputs with PCLK 1 cycle width if the CENDIE bit is 1.
3. The CENDF flag is cleared at the first edge of RSPCKn for the next transmission. Or when 1 is written to the SPSRC.CENDFC bit, then the CENDF flag is 0.

30.3.9.7 Common Operation

In this chapter, the operation common to each mode / area option communication in section 30.3.9.1. Transmit-Receive/ Transmit in Master Mode to section 30.3.9.6. Receive Only in Slave Mode on Clock Synchronous Operation (3-wire) is explained. When the enable of SPI communication end interrupt (CENDIE) is 0, at the time of communication completion, a flag of communication end (CENDF) is set and an event of communication end (sp_elccend) is output, but no interrupt is output. However, if the enable of communication end interrupt (CENDIE) is set to 1 before clearing the flag of communication end (CENDF) while the enable of SPI function (SPE) is 1, the communication end interrupt is output.

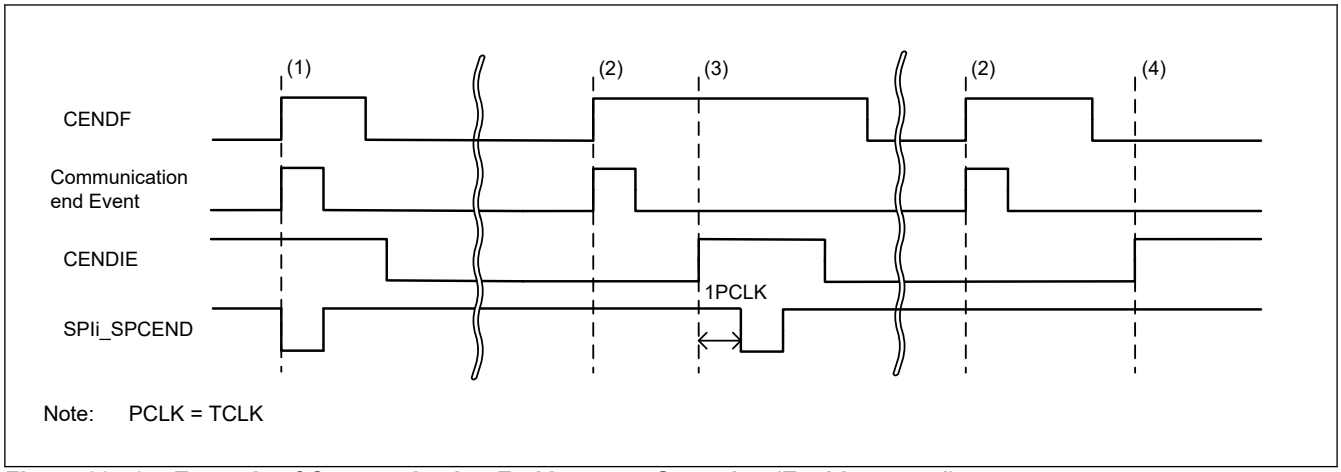


Figure 30.46 Example of Communication End Interrupt Operation (Enable control)

1. When the enable of SPI communication end interrupt (CENDIE) is 1, at the time of communication completion, the following three are the same timing.
   - A flag of communication end (CENDF)
   - An event of communication end (sp_elccend)
   - The communication end interrupt
2. When the enable of SPI communication end interrupt (CENDIE) is 0, at the time of communication completion, the following two are the same timing, but no interrupt.
   - A flag of communication end (CENDF)
   - An event of communication end (sp_elccend)
3. After (2), if the enable of communication end interrupt (CENDIE) is set when the enable of SPI function (SPE) and the flag of communication end (CENDF) are 1, the communication end interrupt is output after 1 TCLK.
4. After (2), even if the enable of communication end interrupt (CENDIE) is set when the enable of SPI function (SPE) or the flag of communication end (CENDF) is 0, the communication end interrupt is not output.

30.3.10 Error Detection

In normal SPI serial transfers, data written to the transmit buffer of SPDR is transmitted, and received data can be read from the receive buffer of SPDR. If access is made to SPDR, an abnormal transfer might occur, depending on the status of the transmit or receive buffer or the status of the SPI at the beginning or end of serial transfer.

If an abnormal transfer occurs, the SPI detects the event as an underrun error, overrun error, parity error, or mode fault error. Table 30.9 lists the relationship between non-normal transfer operations and the SPI error detection function.

Table 30.9 Relationship between non-normal transfer operations and SPI error detection (1 of 2)

Operation	Occurrence condition	SPI operation	Error detection
1	SPDR is written while no empty stages in the transmit FIFO.	• The contents of the transmit buffer are kept • Write data is missing	None
2	SPDR is read while no data stored in receive FIFO.	The contents of the receive buffer and previously received data are output.	None
3	Serial transfer is started in slave mode when the SPI is not able to transmit data.	• Serial transfer is suspended • Transmit or receive data is missing • Driving of the MISOn output signal is stopped • SPI function is disabled	Underrun error
4	Serial transfer ends when data is stored in the receive FIFO for the number of FIFO stages.	• Keeps the contents of the receive FIFO • Missing receive data	Overrun error

Table 30.9 Relationship between non-normal transfer operations and SPI error detection (2 of 2)

Operation	Occurrence condition	SPI operation	Error detection
5	An incorrect parity bit is received during full-duplex synchronous serial communication with the parity function enabled in following mode: • Transmit-receive master mode • Receive-only master mode • Transmit-receive slave mode • Receive-only slave mode	The parity error flag is asserted	Parity error
6	The SSLn0 input signal is asserted when the serial transfer is idle in multi-master mode.	• Driving of the RSPCKn, MOSIn, SSLn1 to SSLn3 output signals is stopped • SPI function is disabled	Mode fault error
7	The SSLn0 input signal is asserted during serial transfer in multi-master mode.	• Serial transfer is suspended • Transmit or receive data is missing • Driving of the RSPCKn, MOSIn, SSLn1 to SSLn3 output signals is stopped • SPI function is disabled	Mode fault error
8	[In the Motorola-SPI case] The SSLn0 input signal is negated during serial transfer in slave mode.	• Serial transfer is suspended • Transmit or receive data is missing • Driving of the MISOn output signal is stopped • SPI function is disabled	Mode fault error
9	[In the TI-SSP case] The SSL0 input signal is asserted during serial transfer in slave mode.	• Serial transfer is suspended • Transmit or receive data is missing • Driving of the MISOn output signal is stopped • SPI function is disabled	Mode fault error
10	After data is stored in the receive FIFO with SPDRES = 1, the number of stored data is less than the threshold value and no receive data is written for the set value of SPDRC [7:0]	Assert the receive data ready flag	Receive data ready

In operation 1 described in Table 30.9, the SPI does not detect an error. To prevent data omission during writes to SPDR, the writes to SPDR must be executed using a transmit buffer empty interrupt request (when the SPSR.SPTEF flag is 1).

Similarly, the SPI does not detect an error in operation 2. To prevent extraneous data from being read, SPDR read must be executed with an SPI receive buffer full interrupt request (when the SPSR.SPRF flag is 1).

For information on the other errors, see the following sections:

- Underrun error, indicated in operation 3, see section 30.3.10.4. Underrun errors
- Overrun error, indicated in operation 4, see section 30.3.10.1. Overrun errors
- Parity error, indicated in operation 5, see section 30.3.10.2. Parity errors
- Mode fault error, indicated in operations 6 to 9, see section 30.3.10.3. Mode fault errors
- For the transmit and receive interrupts, see section 30.3.7. Transmit Buffer Empty and Receive Buffer Full Interrupts.
- For the reception data ready in operations 10, see section 30.3.10.5. Received data ready.

30.3.10.1 Overrun errors

If a serial transfer ends when the receive buffer of SPDR is full, the SPI detects an overrun error and sets the SPSR.OVRF flag to 1. When the OVRF flag is 1, the SPI does not copy data from the shift register to the receive buffer, so the data prior to the error occurrence is retained in the receive buffer. To set the OVRF flag to 0, issue a system reset or 1 is written to the SPSRC.OVRFC bit.

Figure 30.47 shows an example of operation of the OVRF and SPRF flags. The SPSRC and SPDR accesses shown in Figure 30.47 indicate the condition of accesses to the SPSRC and SPDR register, where W denotes a write cycle, and R a read cycle. In this example, the SPI performs an 8-bit serial transfer when SPCMDm.CPHA bit is 1 and the SPCMDm.CPOL bit is 0. The numbers given for RSPCKn in the waveform represent the number of RSPCK cycles, such as the number of transferred bits.

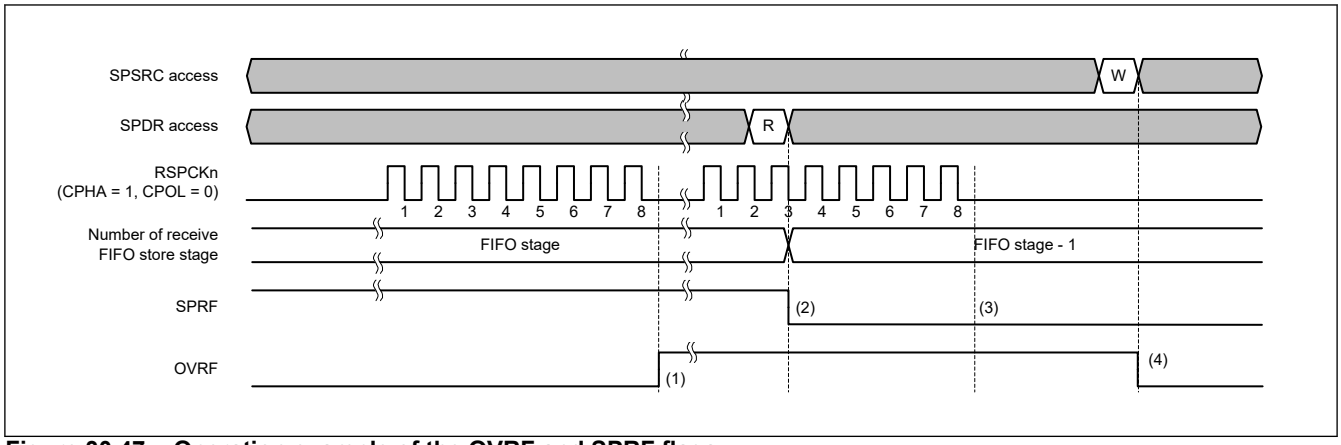


Figure 30.47 Operation example of the OVRF and SPRF flags

The operation of the flags at timings (1) to (4) in Figure 30.47 is as follows:

1. When serial transfer ends while data is stored for the number of FIFO stages, the SPI detects an overrun error and sets the OVRF flag to 1. The SPI does not copy shift register data to the receive buffer. The SPI does not detect a parity error even when SPPE = 1. In master mode, the SPI copies the value of pointer to the SPI command register (SPCMDm) to the SPECM[2:0] bits in the SPI status register (SPSR).
2. When SPDR is read, the SPI outputs the data in the receive buffer. At this time, the SPRF flag is cleared to 0 at the last access when the received data is read from SPDR in one processing routine using DTC / DMAC.
3. If the serial transfer ends with the OVRF flag set to 1 (overrun error occurred), the SPI does not copy data in the shift register to the receive buffer (the SPRF flag does not set to 1). A receive buffer full interrupt is not generated. Even when the SPPE bit is 1, parity errors are not detected. In an overrun error state when the SPI does not copy the received data from the shift register to the receive buffer, on termination of the serial transfer, the SPI determines that the shift register is empty. This enables data transfer from the transmit buffer to the shift register.
4. When 1 is written to the SPSRC.OVRFC bit, the SPSR.OVRF flag is cleared.

The occurrence of an overrun can be checked either by reading SPSR or by using an SPI error interrupt and reading SPSR. When executing a serial transfer, you must ensure that overrun errors are detected early, for example by reading SPSR immediately after SPDR is read.

If an overrun error occurs and the OVRF flag sets to 1, normal reception operations cannot be performed until the OVRF flag is set to 0.

When the RSPCK auto-stop function is enabled (SPCR.SCKASE = 1) in master mode, an overrun error does not occur. Figure 30.48 and Figure 30.49 show the clock stop waveform when a serial transfer continues while the receive buffer is full in master mode.

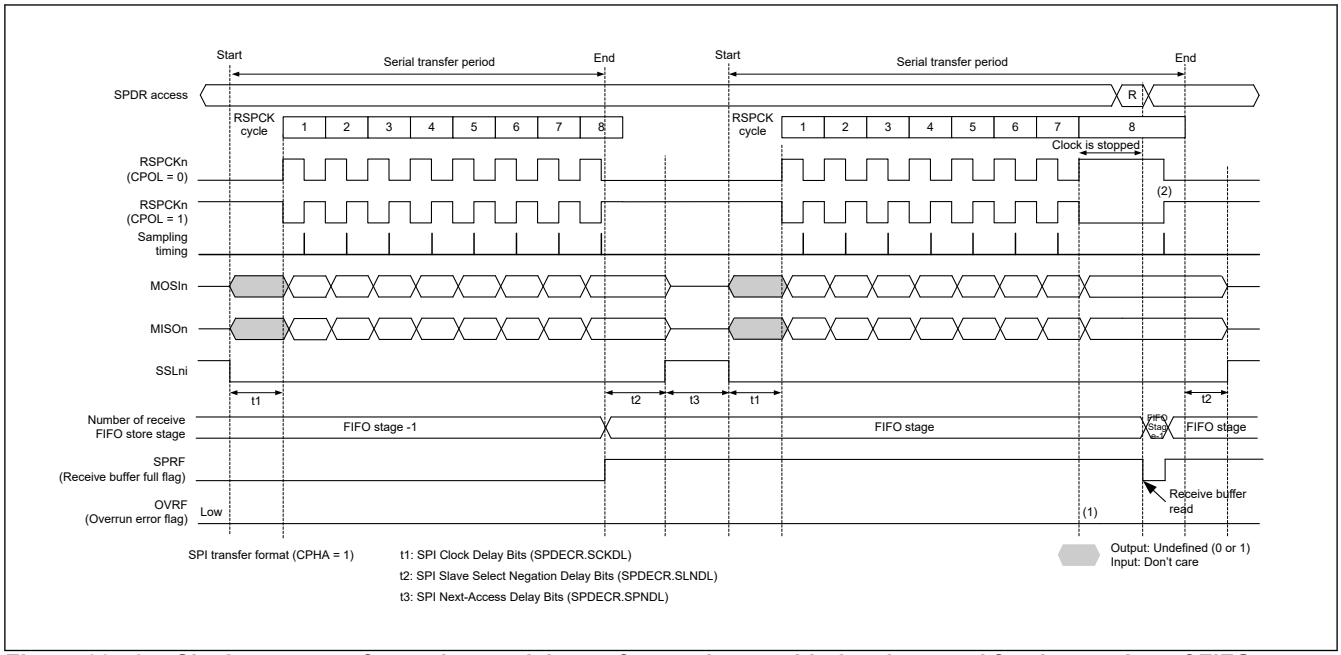


Figure 30.48 Clock stop waveform when serial transfer continues with data is stored for the number of FIFO stages in master mode (CPHA = 1)

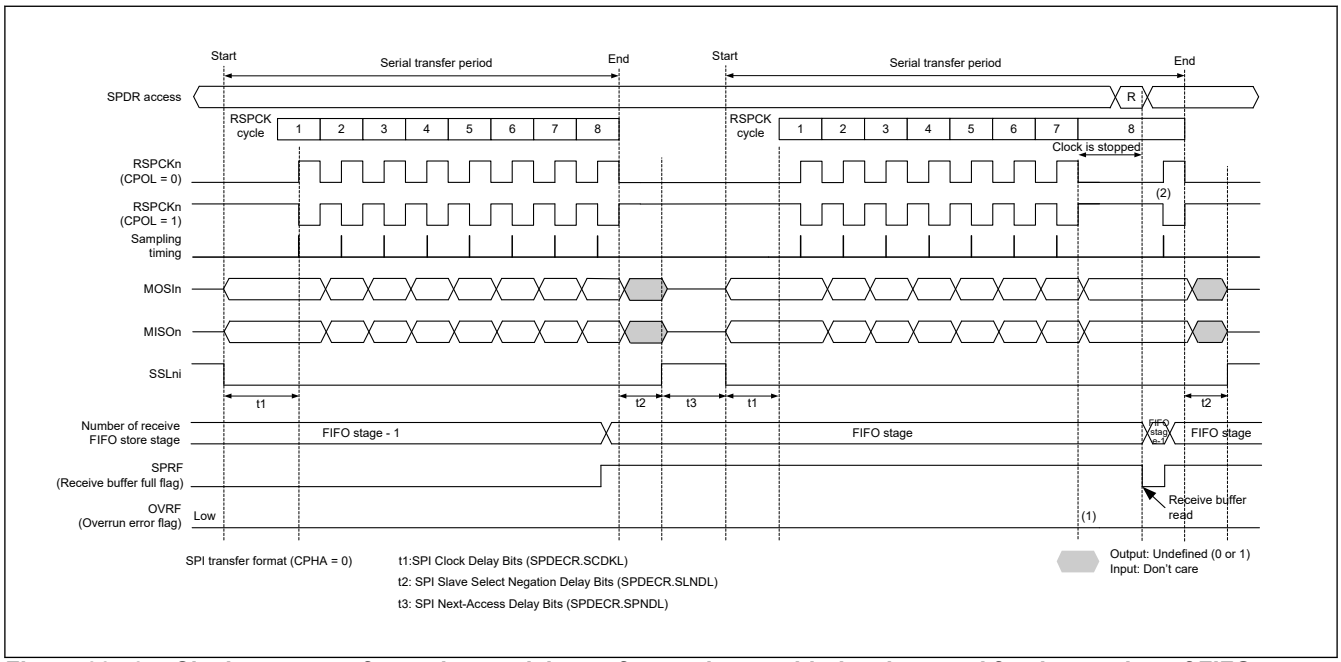


Figure 30.49 Clock stop waveform when serial transfer continues with data is stored for the number of FIFO stages in master mode (CPHA = 0)

The operation of the flags at timings (1) and (2) in Figure 30.48 and Figure 30.49 is as follows:

1. While data is stored in the receive FIFO for the number of FIFO stages, the RSPCK clock is deactivated and no overrun error occurs.
2. If SPDR is read while the clock is stopped, data in the receive buffer can be read. The RSPCK clock restarts.

Overrun error does not occur when RSPCK automatic stop function is enabled for transfer with no delay of between frames during burst transfer in master mode. Figure 30.50 and Figure 30.51 show the clock stop waveform, when there is no delay between frames at burst transfer and the serial transfer continues in the data is stored in the receive FIFO for the number of FIFO stages.

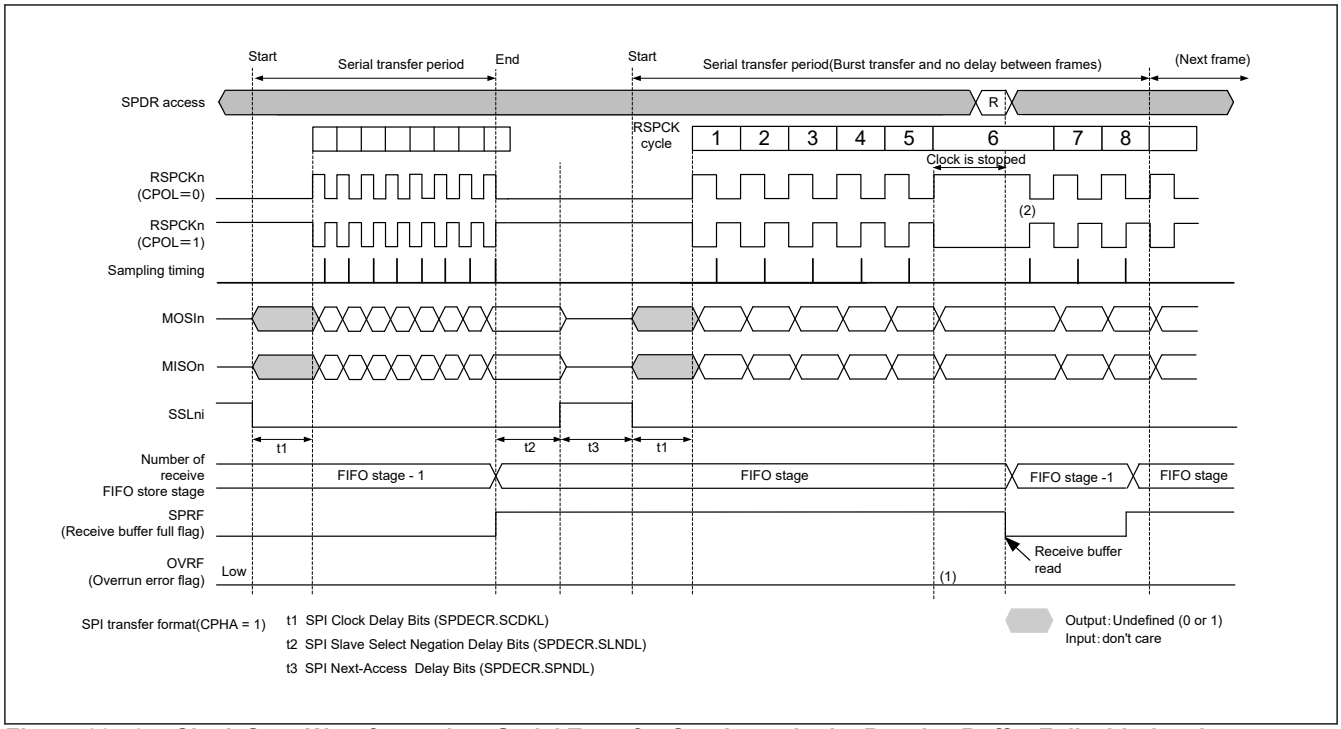


Figure 30.50 Clock Stop Waveform when Serial Transfer Continues in the Receive Buffer Full with data is stored for the number of FIFO stages (at burst transfer and no delay between frames CPHA = 1)

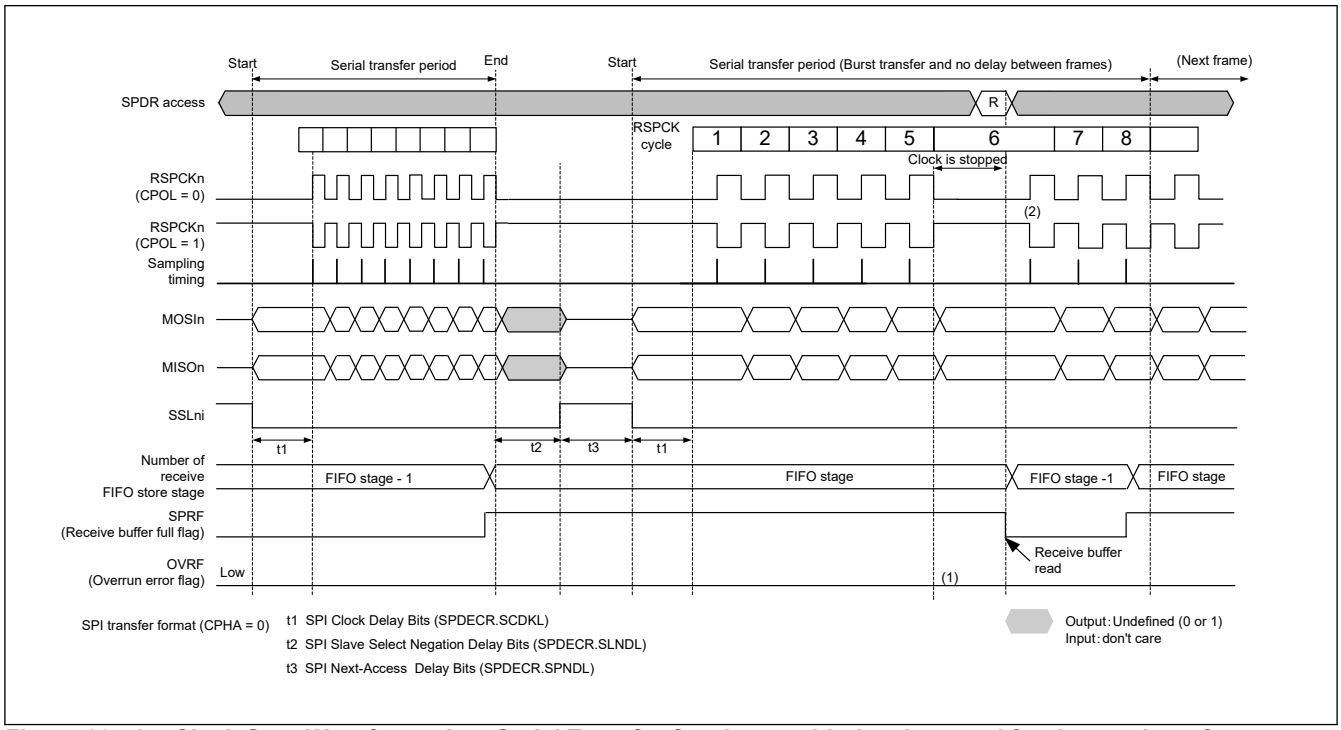


Figure 30.51 Clock Stop Waveform when Serial Transfer Continues with data is stored for the number of FIFO stages in Master Mode (at burst transfer and no delay between frames CPHA = 0)

The following describes operation of flags at timings (1) and (2) in the figure above.

1. While the data is stored for the number of FIFO stages, the RSPCK clock is deactivated and no overrun error occurs.
2. Receive buffer data can be read by reading SPDR during clock stop. After the receive buffer data has been read the RSPCK clock restarts.

30.3.10.2 Parity errors

After transfer in transmit-receive or receive-only master mode, transmit-receive slave mode or receive only slave mode while the SPPE bit in the SPI control register (SPCR) is 1, the SPI checks occurrence of a parity error. When the SPI detects a parity error in received data, the PERF flag in the SPI status register (SPSR) is set to 1. While the OVRF flag is 1, the SPI does not copy shift register data to the receive buffer. Therefore, parity error in received data is not detected. To clear the PERF flag in SPSR to 0, issue a system reset or 1 is written to the SPSRC.PERFC bit.

Figure 30.52 shows an example of operation of the OVRF and PERF flags. The SPSR access shown in Figure 30.52 indicates the condition of access to the register, where W denotes a write cycle, and R a read cycle. In this example, full-duplex serial communication is performed while the SPCR.SPPE bit is 1. The SPI performs an 8-bit serial transfer when SPCMDm.CPHA bit is 1 and the SPCMDm.CPOL bit is 0. The numbers given for RSPCKn in the waveform represent the number of RSPCK cycles, such as the number of transferred bits.

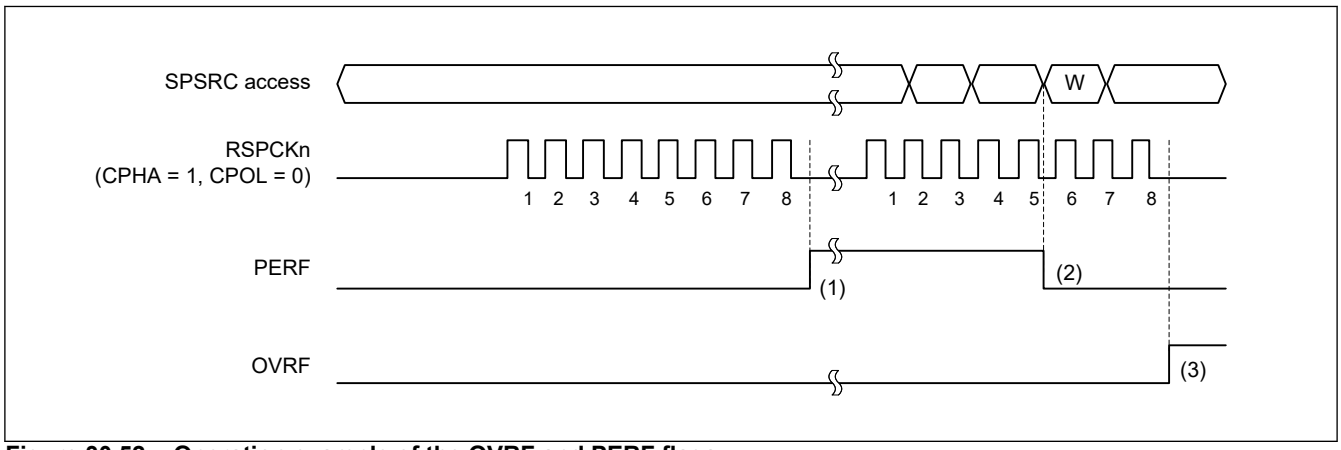


Figure 30.52 Operation example of the OVRF and PERF flags

The operation of the flags at timings (1) to (3) in Figure 30.52 is as follows:

1. When the SPI does not detect an overrun error and terminates the serial transfer, the SPI copies shift register data to the receive buffer. When the SPI checks the received data and detects a parity error at this time, the PERF flag is set to 1. In master mode, the SPI copies the value of pointer to the SPI command register (SPCMDm) to the SPECM[2:0] bits in the SPI Data control register 2 (SPDCR2).
2. When 1 is written to the SPSRC.PERFC bit, then clear the PERF flag.
3. When the SPI detects an overrun error and serial transfer is terminated, the data in the shift register is not copied to the receive buffer. The SPI does not perform parity error detection at this time.

Parity errors can be checked for by either reading the SPSR register or using an SPI error interrupt and reading the SPSR register. When executing a serial transfer, such checks are required to ensure early detection of parity errors. When the SPI is used in master mode, the pointer value to the SPCMDm register at the occurrence of the error can be checked by reading the SPDCR2.SPECM[2:0] bits.

30.3.10.3 Mode fault errors

The SPI operates in multi-master mode when the SPCR.MSTR bit is 1, the SPCR.SPMS bit is 0, and the SPCR.MODFEN bit is 1.

If the active level is input for the SSLn0 input signal of the SPI in multi-master mode, the SPI detects a mode fault error regardless of the status of the serial transfer, and sets the SPSR.MODF flag to 1.

On detecting the mode fault error, the SPI copies the value of the pointer to SPCMD to the SPECM[2:0] bits.

The active level of the SSLn0 signal is determined by the SPCR3.SSL0P bit.

When the MSTR bit is 0, the SPI operates in slave mode.

When the SPCR.MODFEN bit = 1 and the SPMS bit = 0 in slave mode, if the SSLn0 input signal is negated during the serial transfer period (from valid data drive start to final valid data latch), the SPI detects a mode fault error, while any of the following 2 conditions is met.

[In the Motorola-SPI case]

When the SSLn0 input signal is negated while serial data transfer.

[In the TI-SSP case]

When the SSLn0 input signal is asserted while serial data transfer. However, during burst transfer, no error is detected even if the SSLn0 input signal is asserted during the last bit of frame.

When the SPI detects a mode fault error, it stops driving output signals and clears the SPE bit in the SPCR register. When the SPE bit is cleared, the SPI function is disabled (as described in section 30.3.12. SPI Operation). In a multi-master configuration, the mastership can be released by stopping driving output signals and disabling the SPI function by using a mode fault error.

Whether a mode fault error is present can be checked by reading SPSR or by reading an SPI error interrupt and SPSR. To detect a mode fault error without using an SPI error interrupt, poll SPSR. When the SPI is used in master mode, the pointer value to SPCMD when an error is present can be checked by reading the SPECM[2:0] bits in SPSR.

While the MODF flag = 1, the SPI ignores writing 1 to the SPE bit. To enable the SPI function after a mode fault error is detected, clear the MODF flag to 0 without fail.

30.3.10.4 Underrun errors

While the SPI is operating in slave mode (SPCR.MSTR bit = 0) and the communication mode select bit (TXMD[1:0]) in the SPI control register (SPCR) is set to 00b or 01b, if serial transfer is started before transmit data output is ready with the SPCR.SPE bit set to 1 (SPI function enabled), the SPI detects an underrun error and sets the SPSR.MODF and SPSR.UDRF flags to 1.

On detecting an underrun error, the SPI stops the driving of output signals and clears the SPCR.SPE bit to 0 (see section 30.3.11. Initializing the SPI).

The occurrence of underrun errors can be checked either by reading SPSR or by using an SPI error interrupt and reading SPSR. Detecting underrun errors without using the SPI error interrupt requires polling of SPSR.

When the MODF flag is 1, writing 1 to the SPE bit is ignored by the SPI. To enable the SPI function after the detection of an underrun error, the MODF flag must be set to 0.

30.3.10.5 Received data ready

When SPCR.TXMD[1:0] = 00b, 01b, or 11b, and SPCR2.SPDRC[7:0] ≠ 0x00, after receiving data in the receive FIFO during communication (SPE = 1), SPSR.SPDRF flag is set to 1 when the received data is not stored even after the number of received FIFOs is equal to or less than ≤ the threshold value and the value set in SPDRC[7:0] has elapsed.

When the receive data ready is detected, the interrupt and event link output can be selected as SPIi_SPRI or SPIi_SPEI with the SPCR.SPDRES bit.

Figure 30.53 shows an example of reception data ready detection operation.

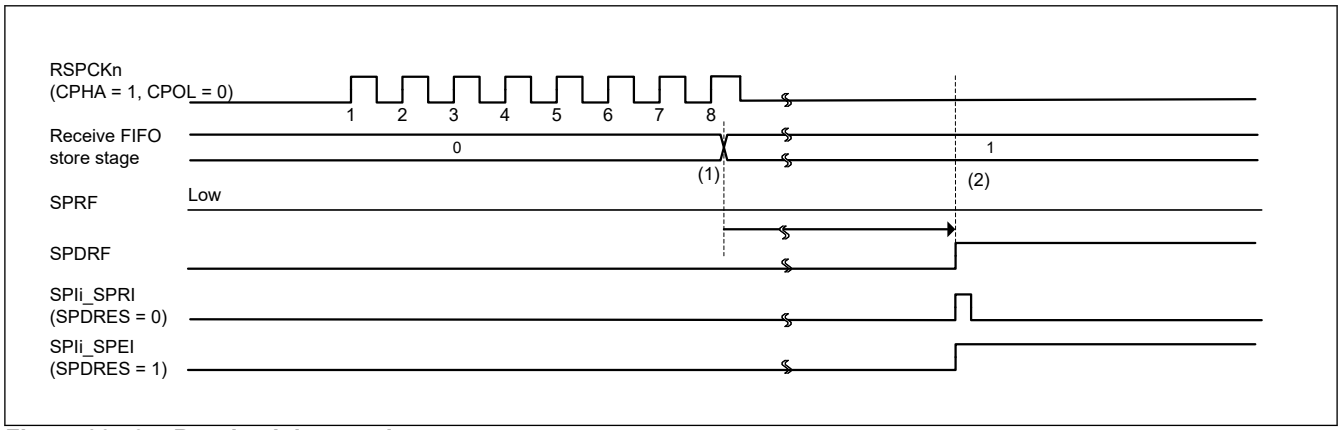


Figure 30.53 Received data ready

The following describes the operation at the timings indicated by (1) and (2) in the figure.

(1) Store the received data in the receive FIFO. SPRF is 0, because receive FIFO store stage ≤ number of frames set by SPDCR2.RTRG[1:0] .

(2) Set SPDRF and assert SPIi_SPRI or SPIi_SPEI because there is no writing to the receive FIFO for the amount of SPDRFC [7:0] set from above (1).

30.3.11 Initializing the SPI

If 0 is written to the SPCR.SPE bit or if the SPI sets the SPE bit to 0 because it detected a mode fault error or an underrun error, the SPI disables the SPI function and initializes some of the module functions. When a system reset is generated, the SPI initializes all of the module functions. This section describes initialization by clearing of the SPCR.SPE bit and by a system reset.

30.3.11.1 Initialization by clearing of the SPCR.SPE bit

When the SPCR.SPE bit is set to 0, the SPI initializes by:

- Suspending any serial transfer that is being executed
- Stopping the driving of output signals (Hi-Z) in slave mode
- Initializing the internal state of the SPI
- Initializing the transmit buffer of the SPI (the SPSR.STEF flag sets to 1)

Initialization by clearing of the SPE bit does not initialize the control bits of the SPI. For this reason, the SPI can be started in the same transfer mode in use prior to initialization when the SPE bit is set to 1 again.

The SPSR.CENDF, SPSR.SPRF, SPSR.OVRF, SPSR.MODF, SPSR.PERF, and SPSR.UDRF flags are not initialized, and the value of the SPDCR2.SPECM[2:0] and SPDCR2.SPCP[2:0] bits are not initialized. Therefore, even after the SPI is initialized, data from the receive buffer can be read to check the communication completion status and the error status during an SPI transfer.

The transmit buffer is initialized to an empty state (the SPSR.SPTEF flag sets to 1). Therefore, if the SPCR.SPTIE bit is set to 1 after SPI initialization, a transmit buffer empty interrupt is generated. To disable any transmit buffer empty interrupts when the SPI is initialized, write 0 to the SPTIE bit simultaneously while writing 0 to the SPE bit.

30.3.11.2 Initialization by system reset

A system reset completely initializes the SPI by initializing all SPI control bits, status bits, and data registers, in addition to meeting the requirements described in section 30.3.11.1. Initialization by clearing of the SPCR.SPE bit.

30.3.12 SPI Operation

30.3.12.1 Master mode operation

The only difference between single- and multi-master mode operation is the use of mode fault error detection (see section 30.3.10. Error Detection). In single-master mode, the SPI does not detect mode fault errors whereas in multi-master mode, it does. This section explains operations that are common to both modes.

(1) Starting a serial transfer

When data is written to the SPI data register (SPDR) while the next transfer data is not set in the transmit FIFO, the SPI updates the transmit buffer (SPTXn,n = 0 to 3) data in SPDR. When the shift register is empty after the number of frames set in the SPDCR.SPFC[1:0] bits are written to the SPDR, the SPI copies data from the transmit buffer to the shift register and starts serial transfer. On copying transmit data to the shift register, the SPI changes the status of the shift register to full. On termination of the serial transfer, it changes the status of the shift register to empty. The status of the shift register cannot be referenced.

The polarity of the SSLni output pins depends on the SPCR3.SSLnP (n = 0 to 3) bits settings. For details on the SPI transfer format, see section 30.3.5. Transfer Formats.

(2) Terminating a serial transfer

[Except Receive-only in Master Mode]

After the SPI detects the RSPCKn edge corresponding to the final sampling timing regardless of the CPHA bit value in the SPI command register (SPCMD), the SPI terminates serial transfer. When the number of data stored in the receive FIFO is

less than the number of FIFO stages, data is copied from the shift register to the receive buffer in the SPI data register (SPDR) after serial transfer.

The final sampling timing varies depending on the bit length of transfer data. In master mode, the SPI data length depends on the SPCMDm.SPB[4:0] bit settings. The polarity of the SSLni output pin depends on the SPCR3.SSLnP (n = 0 to 3) bits settings. For details on the SPI transfer format, see section 30.3.5. Transfer Formats.

[Receive-only in Master Mode]

When any of the following 2 conditions is met, then SPI terminating the serial transfer.

- After the SPI detects the RSPCKn edge corresponding to the final sampling timing regardless of the CPHA bit value in the SPI command register (SPCMD), the SPI terminates serial transfer.
- When writing SPCR2.RMEDTG = 1 during the serial transfer period, SPI terminating the serial transfer.

When the number of data stored in the receive FIFO is less than the number of FIFO stages, data is copied from the shift register to the receive buffer in the SPI data register (SPDR) after serial transfer.

The final sampling timing varies depending on the transfer data bit length. The data length of the SPI in master mode depends on the set value of the SPB[4:0] bits in the SPI command register (SPCMD). The SSLni output signal polarity depends on the set SPI SSLi signal polarity bit (SPCR3.SSLiP) (i = 0 to 3) value. For details about the SPI transfer format, see section 30.3.5. Transfer Formats.

(3) Sequence control

The transfer format in master mode is determined as follows.

The transfer format used in master mode is determined by the SPCR3, SPCMDm, and SPDECR registers.

The SPCR3.SPSLN[2:0] bits determine the sequence configuration for serial transfers that are executed by the SPI in master mode. The following items are set in the SPCMDm register:

- SSLni pin output signal value
- MSB- or LSB-first
- Data length
- Some of the bit rate settings
- RSPCKn polarity and phase
- Whether SPDECR.SCKDL is to be referenced
- Whether SPDECR.SLNDL is to be referenced
- Whether SPDECR.SPNDL is to be referenced

SPCR3.SPBR holds some of the bit rate settings, including SPDECR.SCKDL (SPI clock delay), SPDECR.SLNDL (SSL negation delay), and SPDECR.SPNDL (next-access delay).

Based on the sequence length assigned in SPCR3.SPSLN, the SPI makes up a sequence comprised of a part or all of the SPCMDm register. The SPI contains a pointer to the SPCMDm register that makes up the sequence. The value of this pointer can be checked by reading the SPDCR2.SPCP[2:0] bits. When the SPCR.SPE bit is set to 1 and the SPI function is enabled, the SPI loads the pointer to the commands in SPCMD0, and incorporates the SPCMD0 settings into the transfer format at the beginning of serial transfer. The SPI increments the pointer each time the next-access delay period for a data transfer ends. On completion of the serial transfer that corresponds to the final command in the sequence, the SPI sets the pointer to SPCMD0, and in this way the sequence is executed repeatedly.

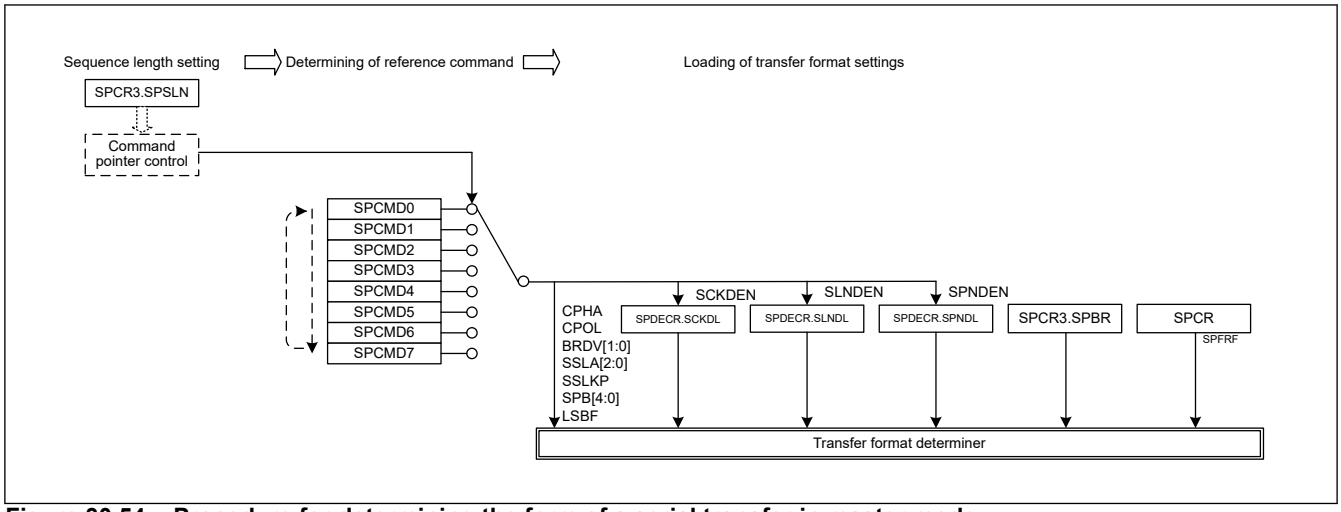


Figure 30.54 Procedure for determining the form of a serial transfer in master mode

In this section, a frame is the combination of the data in SPDR and the settings in SPCMDm.

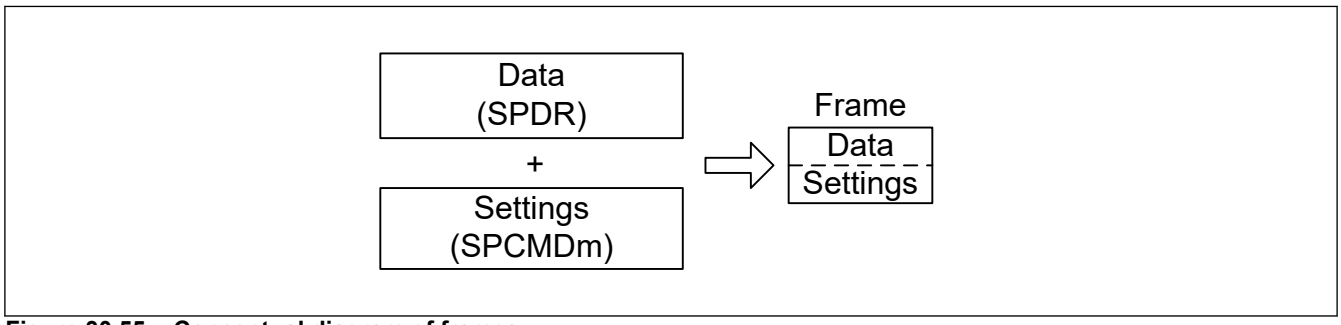


Figure 30.55 Conceptual diagram of frames

Figure 30.56 shows the correspondence between the commands and the transmit and receive buffers in the sequence of operations specified by the settings.

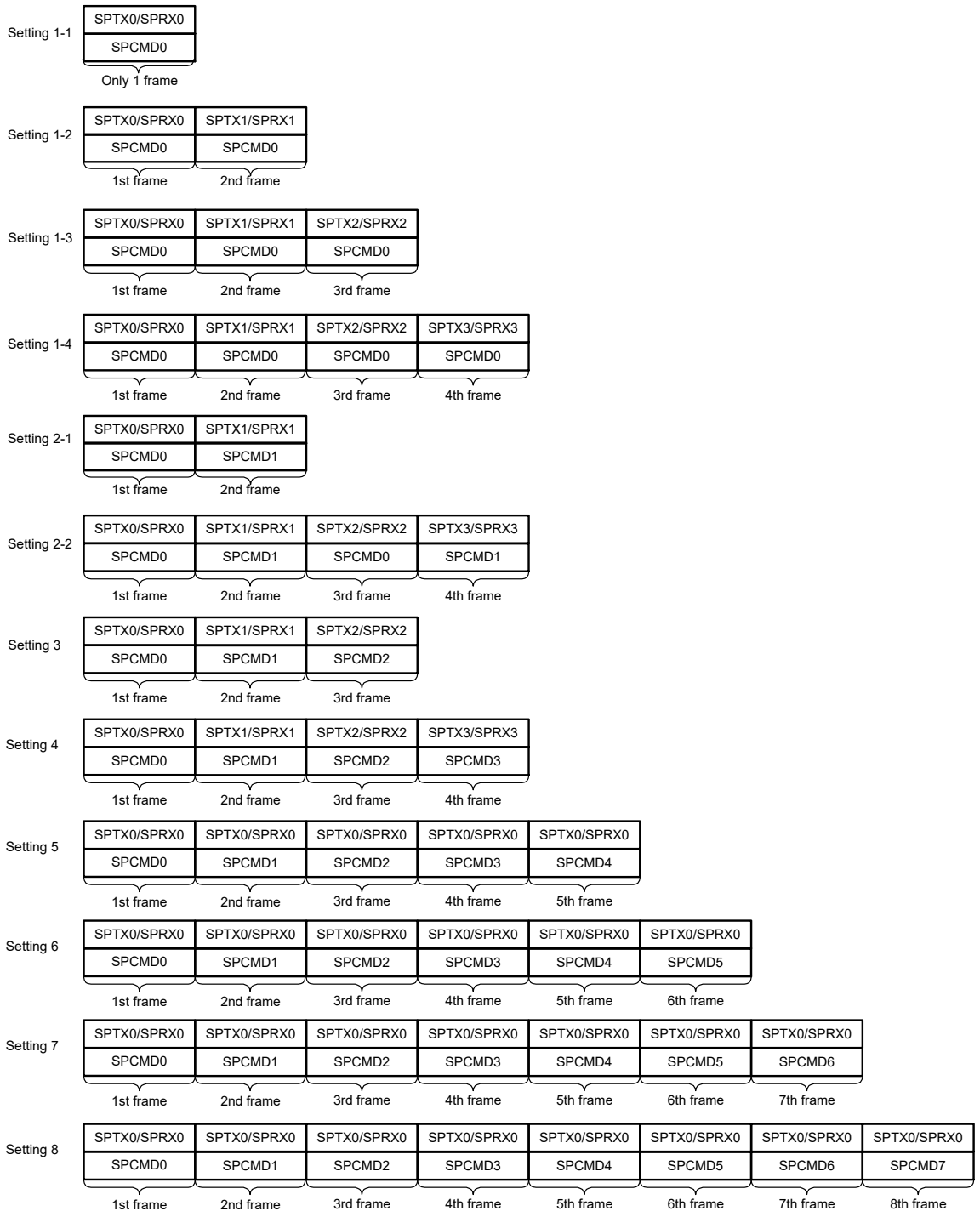


Figure 30.56 Correspondence between SPI Command Register and transmit and receive buffers in sequence operations

(4) Burst transfers

This section describes burst transfer during transmit-receive / transmit-only operation.

[In the Motorola-SPI case]

If the SPCMDm.SSLKP bit that the SPI references during the current serial transfer is 1, the SPI maintains the SSLni signal level during the serial transfer until the beginning of the SSLni signal assertion for the next serial transfer. If the SSLni signal level for the next serial transfer is the same as the SSLni signal level for the current serial transfer, the SPI can execute continuous serial transfers while keeping the SSLni signal assertion status (burst transfer).

- When Between Burst Transfer Frames Delay Select bit (BFDS) of SPI control register (SPCR) is 0.

Figure 30.57 shows an example of an SSLni signal operation for a burst transfer that is implemented using the SPCMD0 and SPCMD1 register settings. This section describes SPI operations (1) to (8) shown in Figure 30.57.

Note: The polarity of the SSLni output signal depends on the SPCR3.SSLnP (n = 0 to 3) bits settings.

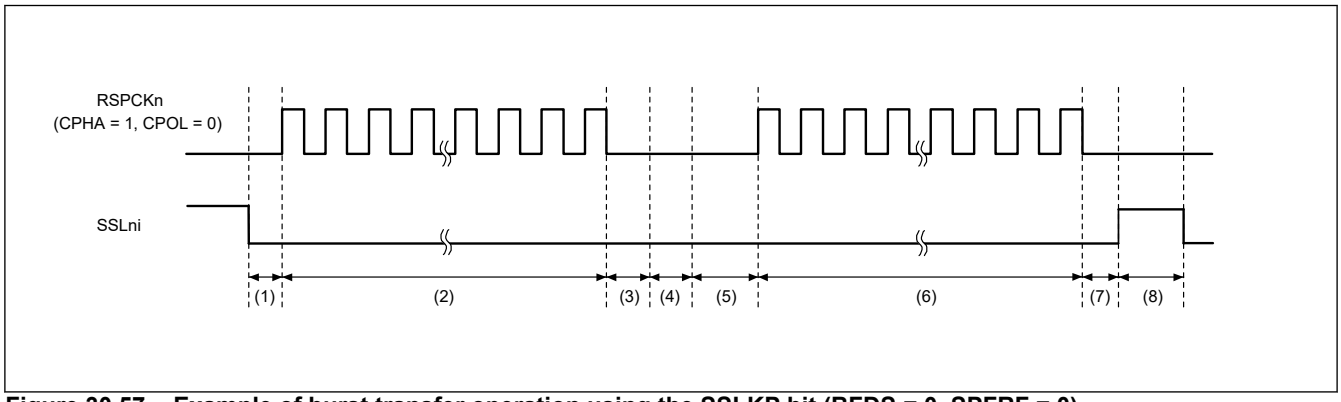


Figure 30.57 Example of burst transfer operation using the SSLKP bit (BFDS = 0, SPFRF = 0)

The SPI operation at times (1) to (8) in the figure is as follows:

1. Based on the SPCMD0 settings, the SPI asserts the SSLni signal and inserts RSPCK delays.
2. The SPI executes serial transfers in accordance with the SPCMD0 settings.
3. The SPI inserts an SSL negation delay.
4. Because the SPCMD0.SSLKP bit is 1, the SPI keeps the SSLni signal value specified in SPCMD0. This period additionally continues for 5 TCLK cycles (at minimum) that is the same as the next-access delay time of SPCMD0. If the shift register is empty after the passage of the minimum period, this period is sustained until the transmit data is stored in the shift register for the next transfer.
5. Based on the SPCMD1 settings, the SPI asserts the SSLni signal and inserts RSPCK delays.
6. The SPI executes serial transfers in accordance with the SPCMD1 settings.
7. Insert SSL negate delay.
8. Because the SPCMD1.SSLKP bit is 0, the SPI negates the SSLni signal. In addition, a next-access delay is inserted in accordance with SPCMD1.

If the SSLni signal output settings in the SPCMDm register where 1 is assigned to the SSLKP bit are different from the SSLni signal output settings in the SPCMDm register to be used in the next transfer, the SPI switches the SSLni signal status to SSLni signal assertion as shown in (5) in Figure 30.57. This corresponds to the command for the next transfer.

Note: If such an SSLni signal switching occurs, the slaves that drive the MISOn signal compete, and collision of signal levels might occur.

The SPI in master mode references the SSLni signal operation within the module when the SSLKP bit is not used. When the SPCMDm.CPHA bit is 0, the SPI can accurately start serial transfers by using the SSLni signal assertion for the next transfer that is detected internally.

- When Between Burst Transfer Frames Delay Select bit (BFDS) of SPI control register (SPCR) is 1.

Figure 30.58 shows an example of SSLni signal operation when burst transfer is achieved by using the settings of SPCMD0 and SPCMD1. The following describes SPI operations of (1) to (6) shown in Figure 30.58. The SSLni output signal polarity depends on the set SPCR3.SSLnP (n = 0 to 3) value.

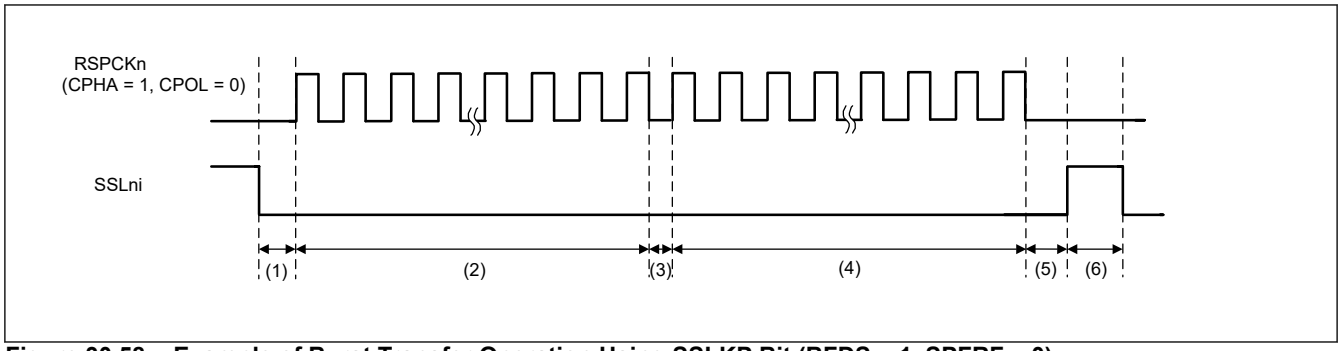


Figure 30.58 Example of Burst Transfer Operation Using SSLKP Bit (BFDS = 1, SPFRF = 0)

1. Assert the SSLni signal and insert an RSPCK delay according to SPCMD0. The RSPCK delay is inserted only the first frame of burst transmission.
2. Perform serial transfer according to SPCMD0. Wait last clock until the next transmit data is stored in the shift register, if the shift register is empty during RSPCK negate period between frames.
3. The value of SSLni signal according to SPCMD0 was hold, because the SPCMD0.SSLKP bit is 1. RSPCK negate period between frames is 0.5RSPCK, if the shift register is not empty.
4. Perform serial transfer according to SPCMD1.
5. Insert SSLni negate delay for the last frame.
6. The SSLni signal is negated because the SSLKP bit in SPCMD1 is 0. Furthermore, the next-access delay is inserted according to SPCMD1.

[In the TI-SSP case]

SPI asserts the SSLni signal for one cycle at the start of serial transfer.

Serial transfer can be executed continuously by asserting the SSLni signal for one cycle at the start of the next serial transfer (burst transfer).

- When the SSLni signal level holding bit (SSLKP) of the SPI command register (SPCMD) is 1 and the burst transfer frame delay selection bit (BFDS) of the SPI control register (SPCR) is 1, SPCMD0 to SPCMD1 are shown in Figure 30.59. The following shows an example of SSLni signal operation and serial data MISOn / MOSIn when burst transfer is realized using the settings. The SSLni output signal polarity depends on the set SPI SSLi signal polarity bit (SPCR3.SSLiP) (i = 0 to 3) value.

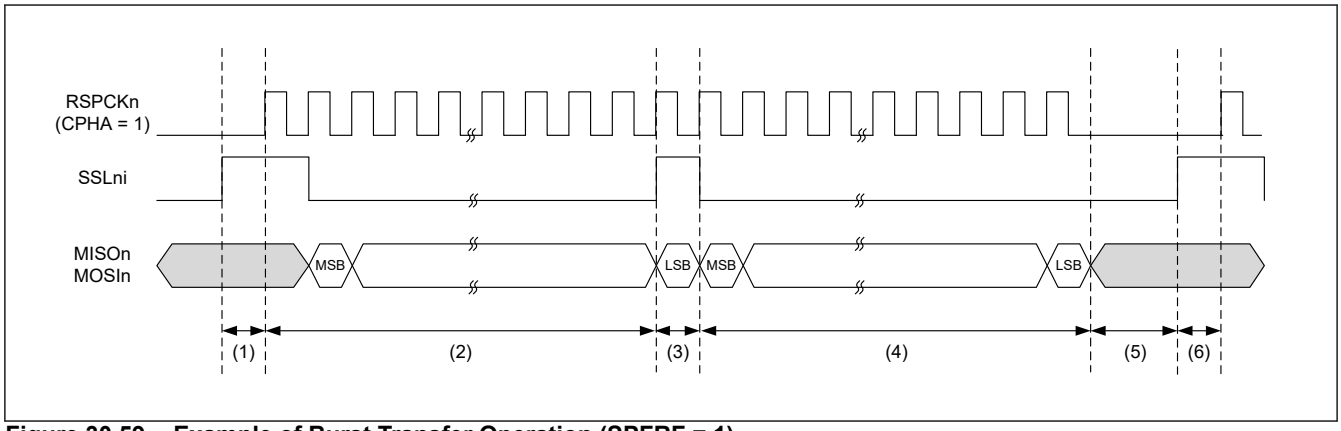


Figure 30.59 Example of Burst Transfer Operation (SPFRF = 1)

1. Assert the SSLni signal and insert an RSPCK delay according to SPCMD0.
   The RSPCK delay is inserted only the first frame of burst transmission.
2. Perform serial transfer according to SPCMD0.
3. Final data transfer and SSLni assertion are performed simultaneously. If the shift register is empty during the RSPCK negation period between frames, wait for the output of the last clock until the transmission data for the next transfer is stored in the shift register.

4. Perform serial transfer according to SPCMD1.
5. Insert OE negate delay for the last frame.
6. Insert the next access delay according to SPCMD1.

If the SSLni signal output setting in SPCMD with the SSLKP bit set to 1 differs from the SSLni signal output setting in SPCMD to be used for the next transfer, the SPI changes the SSLni signal state when the SSLni signal corresponding to the next-transfer command is asserted ((5)). Note that, if an SSLni signal change like this takes place, slaves that drive the MISOn signal may conflict with each other, which may cause collision of signal level.

This section describes burst transfer during receive-only operation.

[In the Motorola-SPI case]

When the SSLKP bit in the SPI command register (SPCMD), which the SPI references in the current serial transfer, is 1, the SPI retains the SSLni signal level during serial transfer until the SSLni signal assertion of the next serial transfer starts. When the SSLni signal level in the next serial transfer is the same as the SSLni signal level in the current serial transfer, the SPI can continuously perform serial transfer while holding the SSLni signal assertion status (burst transfer).

- When Between Burst Transfer Frames Delay Select bit (BFDS) of SPI control register (SPCR) is 0.

Figure 30.60 shows an example of SSLni signal operation when burst transfer is achieved by using the settings of SPCMD0 and SPCMD1. The following describes SPI operations of (1) to (8) shown in Figure 30.60. The SSLni output signal polarity depends on the set SPI SSLni signal polarity bit (SPCR3.SSLiP) (i = 0 to 3) value.

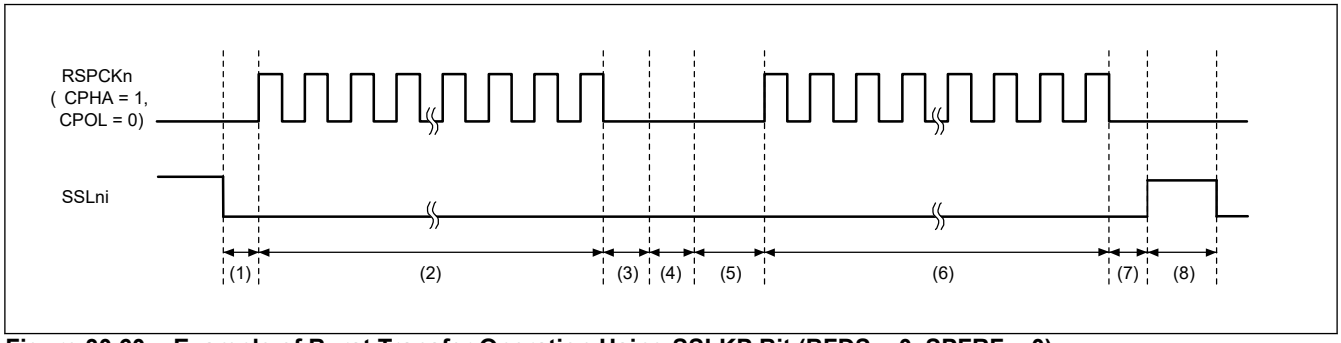


Figure 30.60 Example of Burst Transfer Operation Using SSLKP Bit (BFDS = 0, SPFRF = 0)

1. Assert the SSLni signal and insert an RSPCK delay according to SPCMD0.
2. Perform serial transfer according to SPCMD0.
3. Insert an SSLni negation delay.
4. The SSLni signal value in SPCMD0 is retained because the SSLKP bit in SPCMD0 is 1. This period additionally continues for 5 TCLK cycles (at minimum) that is the same as the next-access delay time of SPCMD0.
5. Assert the SSLni signal and insert an RSPCK delay according to SPCMD1.
6. Perform serial transfer according to SPCMD1.
7. Insert SSLni negate delay.
8. The SSLni signal is negated because the SSLKP bit in SPCMD1 is 0. Furthermore, the next-access delay is inserted according to SPCMD1.

If the SSLni signal output setting and the SSLni signal output setting between SPCMDs used for burst transfer are different, SPI switches the SSLni signal state when the SSLni signal corresponding to the next transfer command is asserted (5). Note that, if an SSLni signal change like this takes place, slaves that drive the MISOn signal may conflict with each other, which may cause collision of signal level.

The SPI in master mode references the SSLni signal operation in the module when SSLKP is not used.

Even when the CPHA bit in SPCMD is 0, the SPI can accurately start serial transfer by using the next transfer SSLni signal assertion detected internally. For this reason, burst transfer in master mode is enabled regardless of the set CPHA bit value. (See section 30.3.11. Initializing the SPI.)

- When Between Burst Transfer Frames Delay Select bit (BFDS) of SPI control register (SPCR) is 1.

Figure 30.61 shows an example of SSLni signal operation when burst transfer is achieved by using the settings of SPCMD0 and SPCMD1. The following describes SPI operations of (1) to (6) shown in Figure 30.61. The SSLni output signal polarity depends on the set SPI SSLi signal polarity bit (SPCR3.SSLiP) (i = 0 to 3) value.

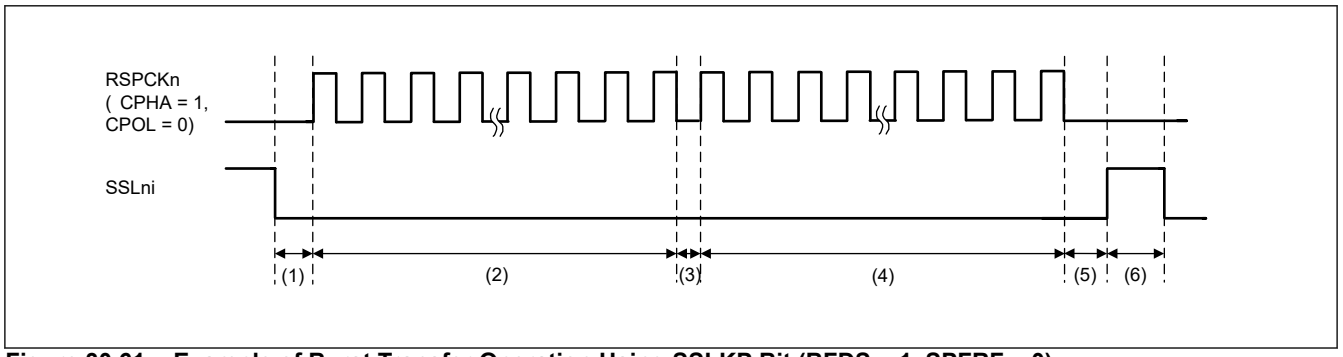


Figure 30.61 Example of Burst Transfer Operation Using SSLKP Bit (BFDS = 1, SPFRF = 0)

1. Assert the SSLni signal and insert an RSPCK delay according to SPCMD0.
   The RSPCK delay is inserted only the first frame of burst transmission.
2. Perform serial transfer according to SPCMD0.
3. Since it is not the last frame, the SSLni signal value at SPCMD0 is retained. RSPCKn negation between frames is 0.5 RSPCKn for the next frame.
4. Perform serial transfer according to SPCMD1.
5. Insert SSLni negate delay for the last frame.
6. The SSLni signal is negated. Furthermore, the next-access delay is inserted according to SPCMD1.

Note: Last frame: Frame set by RMFM [4: 0] bits when SPCR2.RMFM [4: 0] ≠ 00h
Or, a frame in which SPCR2.RMEDTG = 1 has been accepted.

[In the TI-SSP case]

SPI asserts the SSLni signal for one cycle at the start of serial transfer.

Serial transfer can be executed continuously by asserting the SSLni signal for one cycle at the start of the next serial transfer (burst transfer).

- When the SSLni signal level holding bit (SSLKP) of the SPI command register (SPCMD) is 1 and the burst transfer frame delay selection bit (BFDS) of the SPI control register (SPCR) is 1, SPCMD0 to SPCMD1 are shown in Figure 30.62. The following shows an example of SSLni signal operation and serial data MISOn / MOSIn when burst transfer is realized using the settings. The SSLni output signal polarity depends on the set SPI SSLi signal polarity bit (SPCR3.SSLiP) (i = 0 to 3) value.

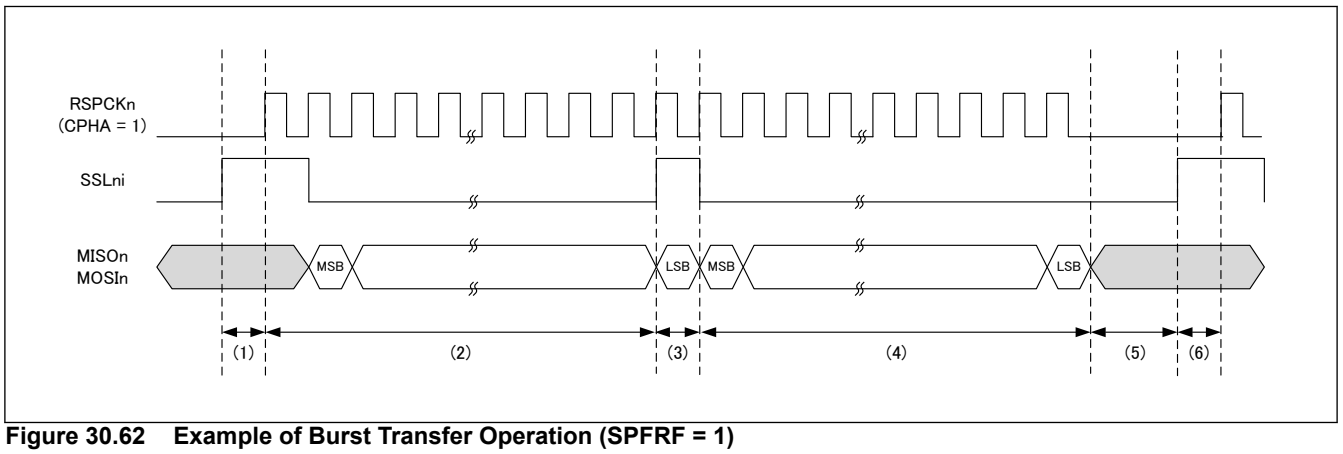


Figure 30.62 Example of Burst Transfer Operation (SPFRF = 1)

1. Assert the SSLni signal and insert an RSPCK delay according to SPCMD0.
   The RSPCK delay is inserted only the first frame of burst transmission.

2. Perform serial transfer according to SPCMD0.
3. Final data transfer and SSLni assertion are performed simultaneously.
4. Perform serial transfer according to SPCMD1.
5. Insert OE negate delay for the last frame.
6. Insert the next access delay according to SPCMD1.

Note: Last frame: Frame set by RMFM [4:0] bits when SPCR2.RMFM [4:0] ≠ 00h
Or, a frame in which SPCR2.RMEDTG = 1 has been accepted.

If the SSLni signal output setting between the SPCMDs used for burst transfer differs from the SSLni signal output setting, SPI switches the SSLni signal state when the SSLni signal corresponding to the next transfer command is asserted (5). Note that, if an SSLni signal change like this takes place, slaves that drive the MISOn signal may conflict with each other, which may cause collision of signal level.

(5) RSPCK delay (t1)

The RSPCK delay value of the SPI in master mode depends on the SPCMDm.SCKDEN bit setting and the SPDECR.SCKDL[2:0] bits setting. The SPI determines the SPCMDm register to be referenced during a serial transfer by pointer control, and determines an RSPCK delay using the SPCMDm.SCKDEN bit and SPDECR.SCKDL[2:0] bits, as listed in Table 30.10. For a definition of RSPCK delay, see section 30.3.5. Transfer Formats.

RSPCK delay insert to only the first frame of burst transmission, when transmit without “Between Burst Transfer Frames Delay”. (The SPCMD.SSLKP bit is 1 and the SPCR.BFDS bit is 1.)

Table 30.10 Relationship between the SPCMDm.SCKDEN bit, SPDECR.SCKDL[2:0] bits, and RSPCK delay

SPCMDm.SCKDEN bit	SPDECR.SCKDL[2:0] bits	RSPCK delay	
		Motorola-SPI	TI-SSP
0	000b to 111b	1 RSPCK	0 RSPCK
1	000b	1 RSPCK	1 RSPCK
	001b	2 RSPCK	2 RSPCK
	010b	3 RSPCK	3 RSPCK
	011b	4 RSPCK	4 RSPCK
	100b	5 RSPCK	5 RSPCK
	101b	6 RSPCK	6 RSPCK
	110b	7 RSPCK	7 RSPCK
	111b	8 RSPCK	8 RSPCK

(6) SSL negation delay (t2)

The SSL negation delay value of the SPI in master mode depends on the SPCMDm.SLNDEN bit setting and the SPDECR.SLNDL[2:0] bits setting. The SPI determines the SPCMDm register to be referenced by pointer control during a serial transfer, and determines an SSL negation delay using the SPCMDm.SLNDEN bit and SPDECR.SLNDL[2:0] bits, as listed in Table 30.11. For a definition of SSL negation delay, see section 30.3.5. Transfer Formats.

An SSL negation delay is inserted to only the last frame of the burst transmission, that is, transmit without between burst transfer frames delay. (SPCMD.SSLKP bit is 1 and SPCR.BFDS bit is 1).

Table 30.11 Relationship between the SPCMDm.SLNDEN bit, SPDECR.SLNDL[2:0] bits, and SSL negation delay (1 of 2)

SPCMDm.SLNDEN bit	SPDECR.SLNDL[2:0] bits	SSL negation delay
0	000b to 111b	1 RSPCK

Table 30.11 Relationship between the SPCMDm.SLNDEN bit, SPDECR.SLNDL[2:0] bits, and SSL negation delay (2 of 2)

SPCMDm.SLNDEN bit	SPDECR.SLNDL[2:0] bits	SSL negation delay
1	000b	1 RSPCK
	001b	2 RSPCK
	010b	3 RSPCK
	011b	4 RSPCK
	100b	5 RSPCK
	101b	6 RSPCK
	110b	7 RSPCK
	111b	8 RSPCK

(7) Next-access delay (t3)

The next-access delay value of the SPI in master mode depends on the SPCMDm.SPNDEN bit setting and the SPDECR.SPNDL[2:0] bits setting. The SPI determines the SPCMDm register to be referenced during serial transfer by pointer control, and then determines a next-access delay during serial transfer using the SPCMDm.SPNDEN bit and SPDECR.SPNDL[2:0] bits, as listed in Table 30.12. For a definition of next-access delay, see section 30.3.5. Transfer Formats.

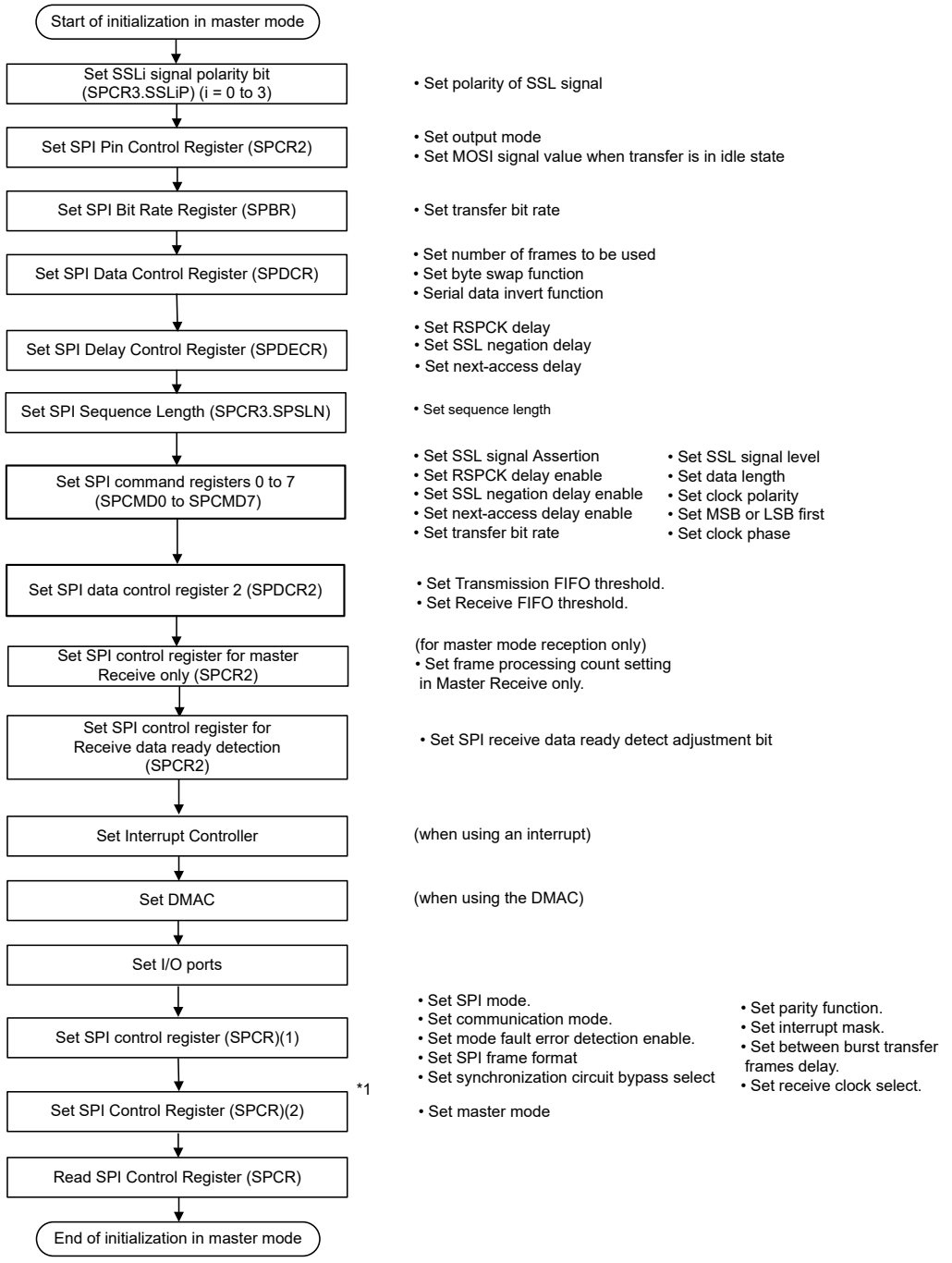
A next-Access delay is inserted to only the last frame of the burst transmission, that is, transmit without between burst transfer frames delay. (SPCMD.SSLKP bit is 1 and SPCR.BFDS bit is 1).

Table 30.12 Relationship between the SPCMDm.SPNDEN bit, SPDECR.SPNDL[2:0] bits, and next-access delay

SPCMDm.SPNDEN bit	SPDECR.SPNDL[2:0] bits	Next-access delay
0	000b to 111b	1 RSPCK + 5 TCLK
1	000b	1 RSPCK + 5 TCLK
	001b	2 RSPCK + 5 TCLK
	010b	3 RSPCK + 5 TCLK
	011b	4 RSPCK + 5 TCLK
	100b	5 RSPCK + 5 TCLK
	101b	6 RSPCK + 5 TCLK
	110b	7 RSPCK + 5 TCLK
	111b	8 RSPCK + 5 TCLK

(8) Initialization flow

Figure 30.63 shows an example of SPI initialization flow when the SPI is in master mode. For information on how to set up the Interrupt Controller Unit (ICU), DMAC and I/O ports, see the descriptions given in the individual blocks.



Note 1. Regarding the setting of the SPI control register(SPCR), leave an interval of 1TCLK between setting(1) (initial setting other than MSTR bit) and setting(2) (MSTR setting).

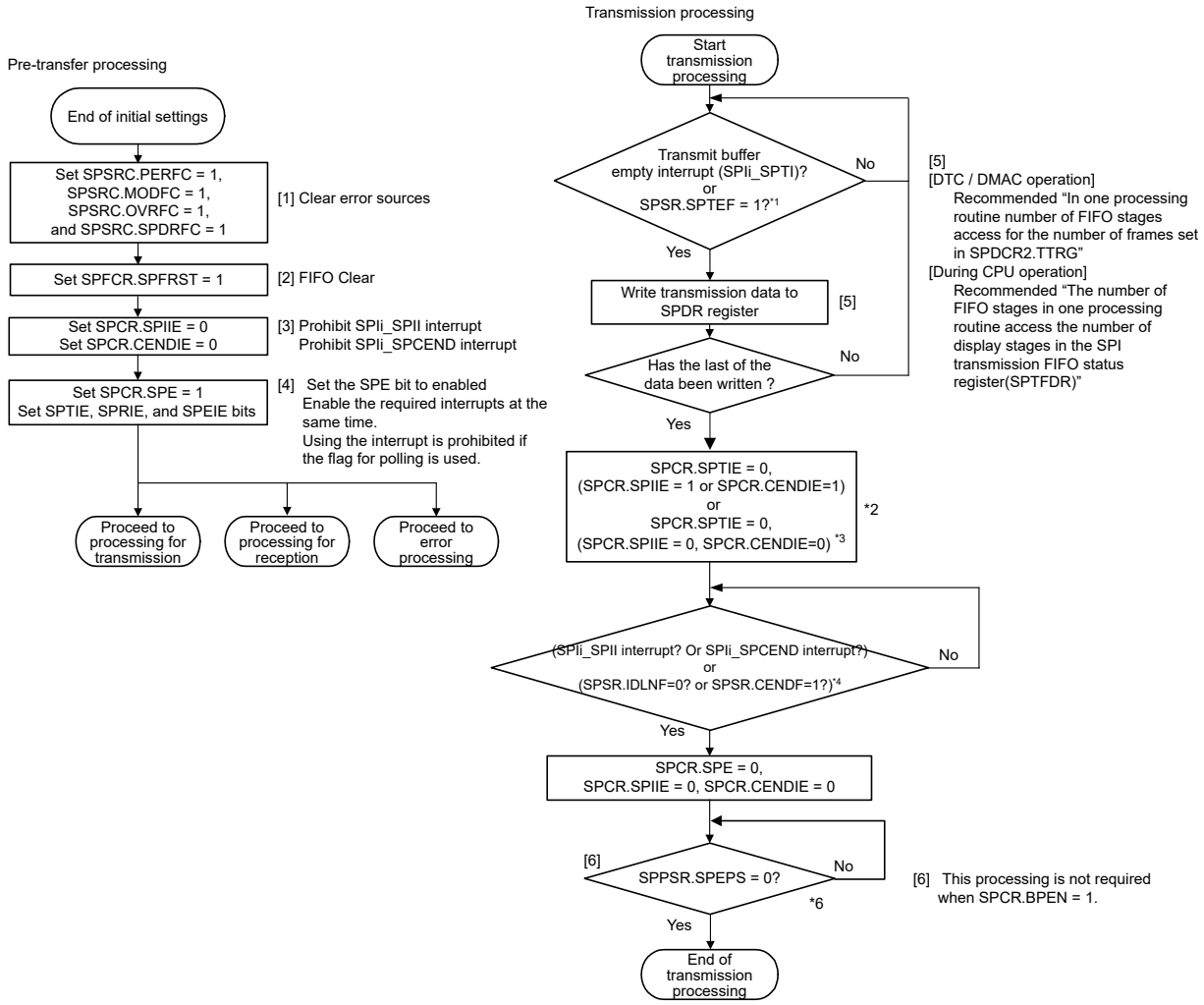
Figure 30.63 Example of initialization flow in master mode for SPI operation

(9) Software processing flow

Figure 30.64 to Figure 30.67 show examples of the software processing flow.

Transmit processing flow

When transmitting data, with the SPIi_SPII or SPIi_SPCEND interrupt enabled, the CPU is notified of the completion of data transmission after the last data write for transmission.

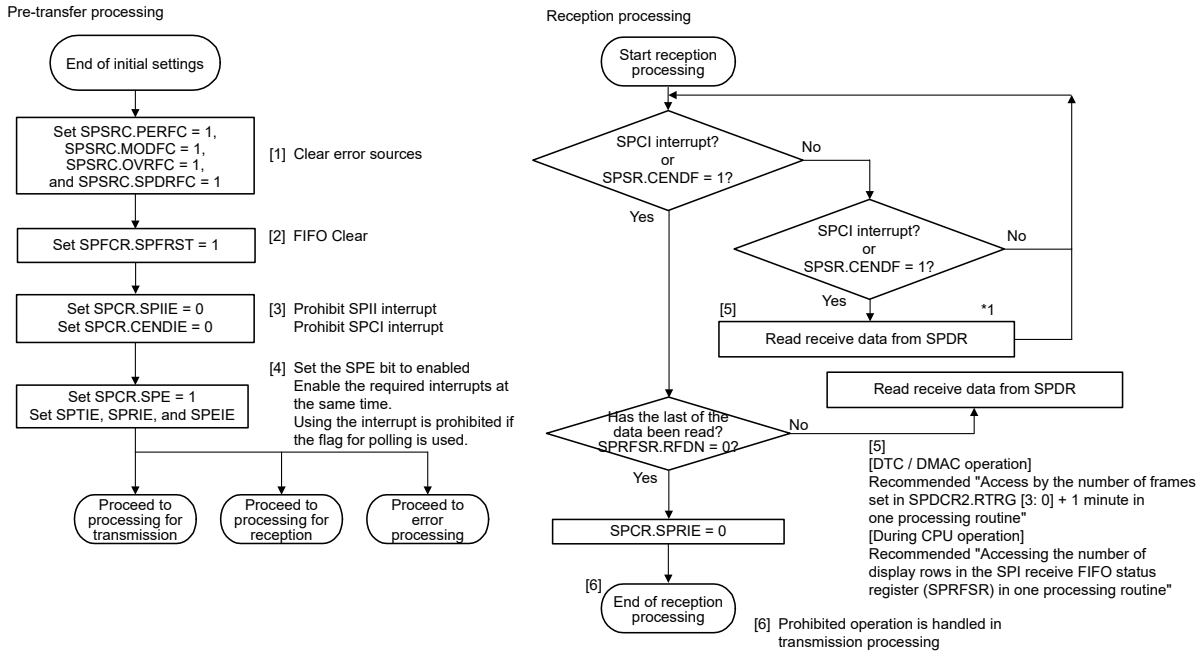


Note 1. Proceed to process of writing data for transmission after reading SPSR.SPTEF flag is 1, if the user uses the flag for polling.
Note 2. The user can know the communication end by SPIi_SPII interrupt (IDLNF flag) or SPIi_SPCEND interrupt (CENDF flag). Use either SPIi_SPII interrupt (IDLNF flag) or SPIi_SPCEND interrupt (CENDF flag).
Note 3. Set SPIi_SPII interrupt prohibited (SPCR.SPIIE = 0) and SPIi_SPCEND interrupt prohibited (SPCR.CENDIE = 0), if the user uses the flag for polling.
Note 4. Wait more than 1 PCLK, after writing data for transmission to SPDR, and before starting polling of SPSR.IDLNF or SPSR.CENDF, if the user uses the flag for polling.
Note 5. During CPU operation, Write 1 to SPI transmit buffer empty flag clear bit (SPSRC.SPTEFC) after one processing routine.
Note 6. When setting to 0 in slave mode and SPCR.BPEN = 0, check that SPPSR.SPEPS has become 0.(Not required when BPEN = 1.)

Figure 30.64 Transmission flow in master mode

Receive processing flow

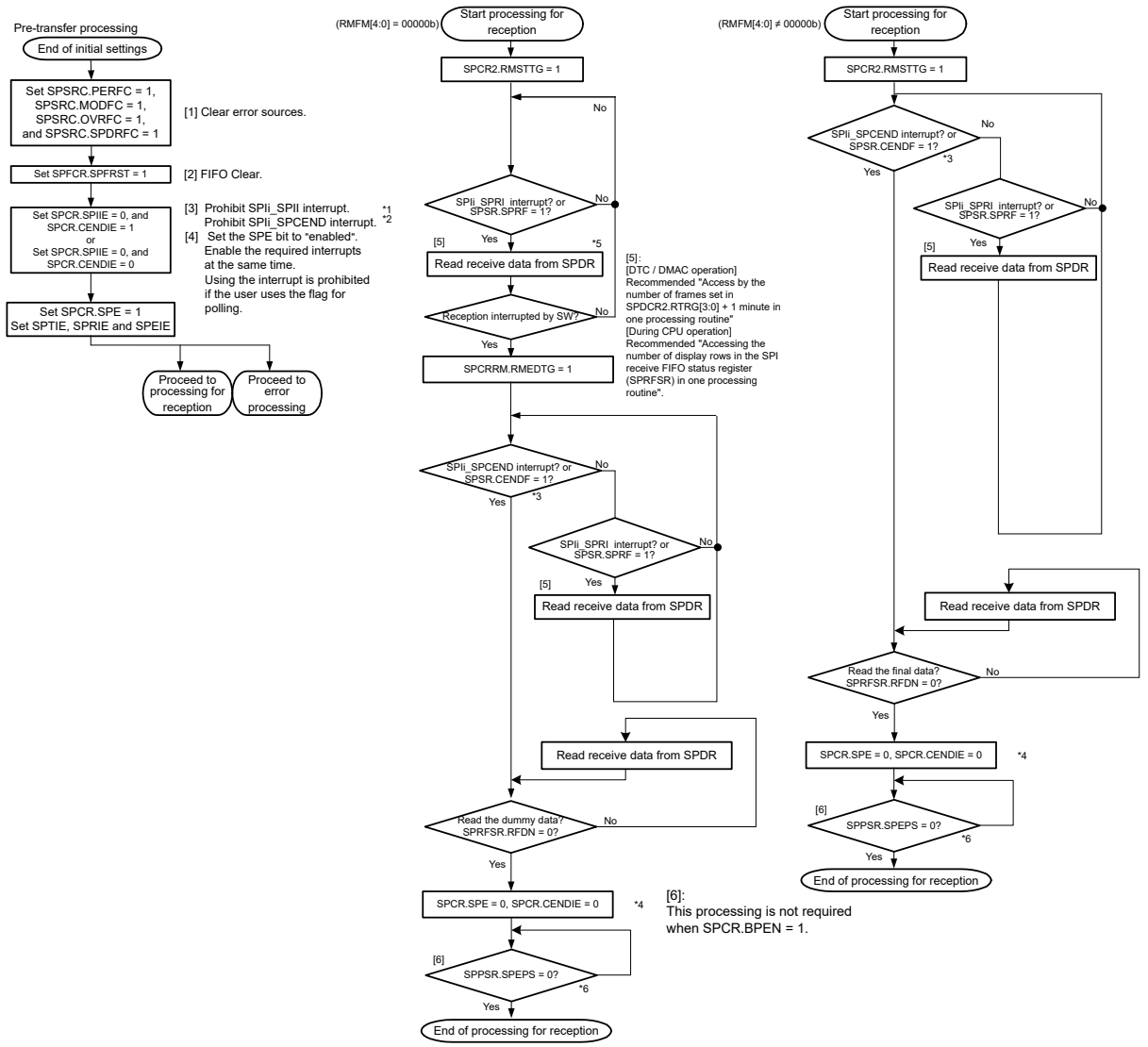
The SPI has receive only operation in slave mode.



Note 1. During CPU operation, write 1 to the SPSRC.SPRFC bit (SPI receive buffer full flag clear bit) after one processing routine.

Figure 30.65 Reception flow in master mode

Receive only processing flow in master mode



Note 1. The end of communication can be determined by the SPIi_SPCEND interrupt (CENDF flag). Use the SPIi_SPCEND interrupt (CENDF flag).
Note 2. Disable SPIi_SPCEND interrupt (SPCR.CENDIE = 0) during flag polling.
Note 3. A communication completion interrupt at reception only operation (Master) occurs at a predetermined timing under any of the following conditions.
- Frame number processing set by RMFM completed (RMFM[4:0] ≠ 00000b)
- RMEDTG = 1, when processing of the number of frames set by RMFM is not completed (including RMFM[4:0] = 00000b)

Note 4. Internal trigger is also cleared.
Note 5. During CPU operation, write 1 to the SPI receive buffer full flag clear bit (SPSRC.SPRFC) after one processing routine.

Figure 30.66 Software Processing Flowchart in Master Mode (Reception-only)

Error processing flow

The SPI detects the following errors:

- Mode fault error
- Underrun error
- Overrun error
- Parity error

When a mode fault error is generated, the SPCR.SPE bit is automatically cleared, stopping operations for transmission and reception. For errors from other sources, the SPCR.SPE bit is not cleared and operations for transmission and reception continue. Therefore, Renesas recommends clearing the SPCR.SPE bit to stop operations for errors other than mode fault errors. Not doing so leads to updating of the SPDCR2.SPECM[2:0] bits.

When an error is detected using an interrupt, clear the ICU.IELSRn.IR flag in the error processing routine. If this is not done, the ICU.IELSRn.IR flag might continue to indicate the SPIi_SPTI or SPIi_SPRI interrupt request. If the SPIi_SPRI interrupt request is indicated, read the receive buffer and initialize the sequencer in the SPI.

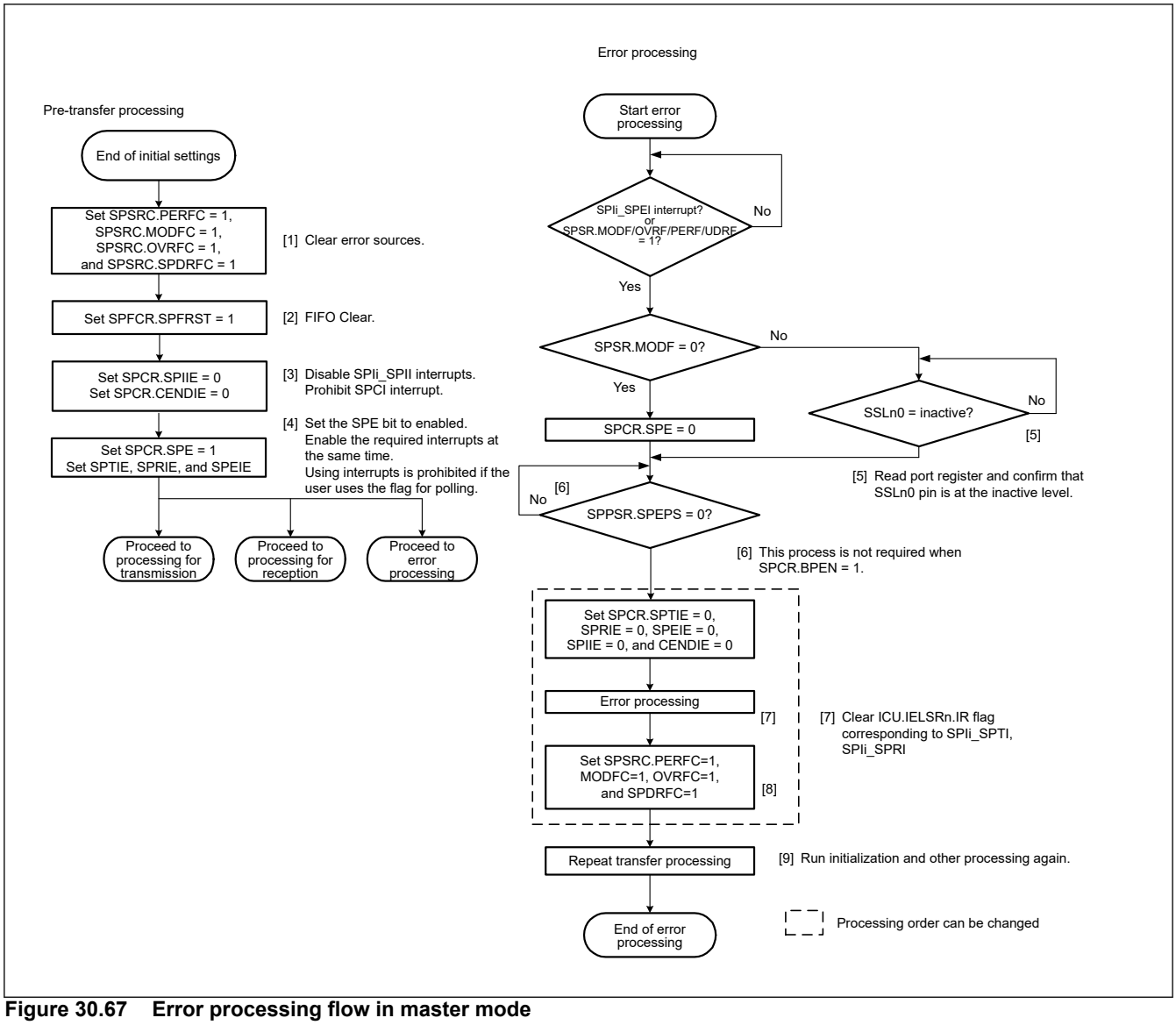


Figure 30.67 Error processing flow in master mode

30.3.12.2 Slave mode operation

(1) Starting a serial transfer

When the SPCMD0.CPHA bit is 0, if the SPI detects an SSLn0 input signal assertion, it must drive valid data to the MISOn output signal. For this reason, when the CPHA bit is 0, the assertion of the SSLn0 input signal triggers the start of a serial transfer.

When the CPHA bit is 1, if the SPI detects the first RSPCKn edge in an SSLn0 signal asserted condition, it must drive valid data to the MISOn output signal. For this reason, when the CPHA bit is 1, the first RSPCKn edge in an SSLn0 signal asserted condition triggers the start of a serial transfer.

Regardless of the CPHA bit setting, the SPI drives the MISOn output signal on SSLn0 signal assertion. The data that is output by the SPI is either valid or invalid, depending on the CPHA bit setting.

For details on the SPI transfer format, see section 30.3.5. Transfer Formats. The polarity of the SSLn0 input signal depends on the SPCR3.SSL0P setting.

(2) Terminating a serial transfer

Regardless of the SPCMD0.CPHA bit setting, the SPI terminates the serial transfer after detecting an RSPCKn edge corresponding to the final sampling timing. When the number of data stored in the receive FIFO is less than the number of FIFO stages, on termination of serial transfer, the SPI copies received data from the shift register to the receive buffer of the SPDR register. On termination of a serial transfer, the SPI changes the status of the shift register to empty, regardless of the receive buffer state. A mode fault error occurs if the SPI detects an SSLn0 input signal negation from the beginning of serial transfer to the end of serial transfer (see section 30.3.10. Error Detection).

The final sampling timing changes depending on the bit length of transfer data. In slave mode, the SPI data length is determined by the SPCMD0.SPB[4:0] bits setting. The polarity of the SSLn0 input signal is determined by the SPCR3.SSL0P bit setting. For details on the SPI transfer format, see section 30.3.5. Transfer Formats.

(3) Notes on single-slave operations

[In the Motorola-SPI case]

If the SPCMD0.CPHA bit is 0, the SPI starts serial transfers when it detects the assertion edge for an SSLn0 input signal. In the configuration shown in Figure 30.8, if the SPI is used in single-slave mode, the SSLn0 signal is fixed at an active state. Therefore, when the CPHA bit is set to 0, the SPI cannot correctly start a serial transfer. For the SPI to correctly execute transmit and receive operations in slave mode when the SSLn0 input signal is fixed at an active state, the CPHA bit must be set to 1. Do not fix the SSLn0 input signal if there is a requirement for setting the CPHA bit to 0.

[In the TI-SSP case]

When SPI is used as a single slave in the configuration shown in the Figure 30.8, the SSLn0 input signal is always fixed to the inactive state, so the SPI cannot start the serial transfer.

When using a single slave, use the configuration shown in the example in Figure 30.7.

(4) Burst transfer

[In the Motorola-SPI case]

If the SPCMD0.CPHA bit is 1, continuous serial transfer (burst transfer) can be executed while retaining the assertion state for the SSLn0 input signal. When the CPHA bit is 1, the serial transfer period is the period from the first RSPCKn edge to the sampling timing for the reception of the final bit in an SSLn0 signal active state. Even when the SSLn0 input signal remains at the active level, the SPI can accommodate burst transfers, because it can detect the start of an access.

When the CPHA bit is 0, the second and subsequent serial transfers during burst transfer cannot be executed correctly.

[In the TI-SSP case]

In serial transfer, data transfer starts after the SSLn0 input signal is asserted for RSPCK 1 cycle. Since frame transfer starts from the SSL input signal, SSLn0 must be asserted between frames.

(5) Initialization flow

Figure 30.68 shows an example of initialization flow for SPI operation when the SPI is in slave mode. For a description of how to set up the ICU, DTC, and I/O ports, see the descriptions given in the individual blocks.

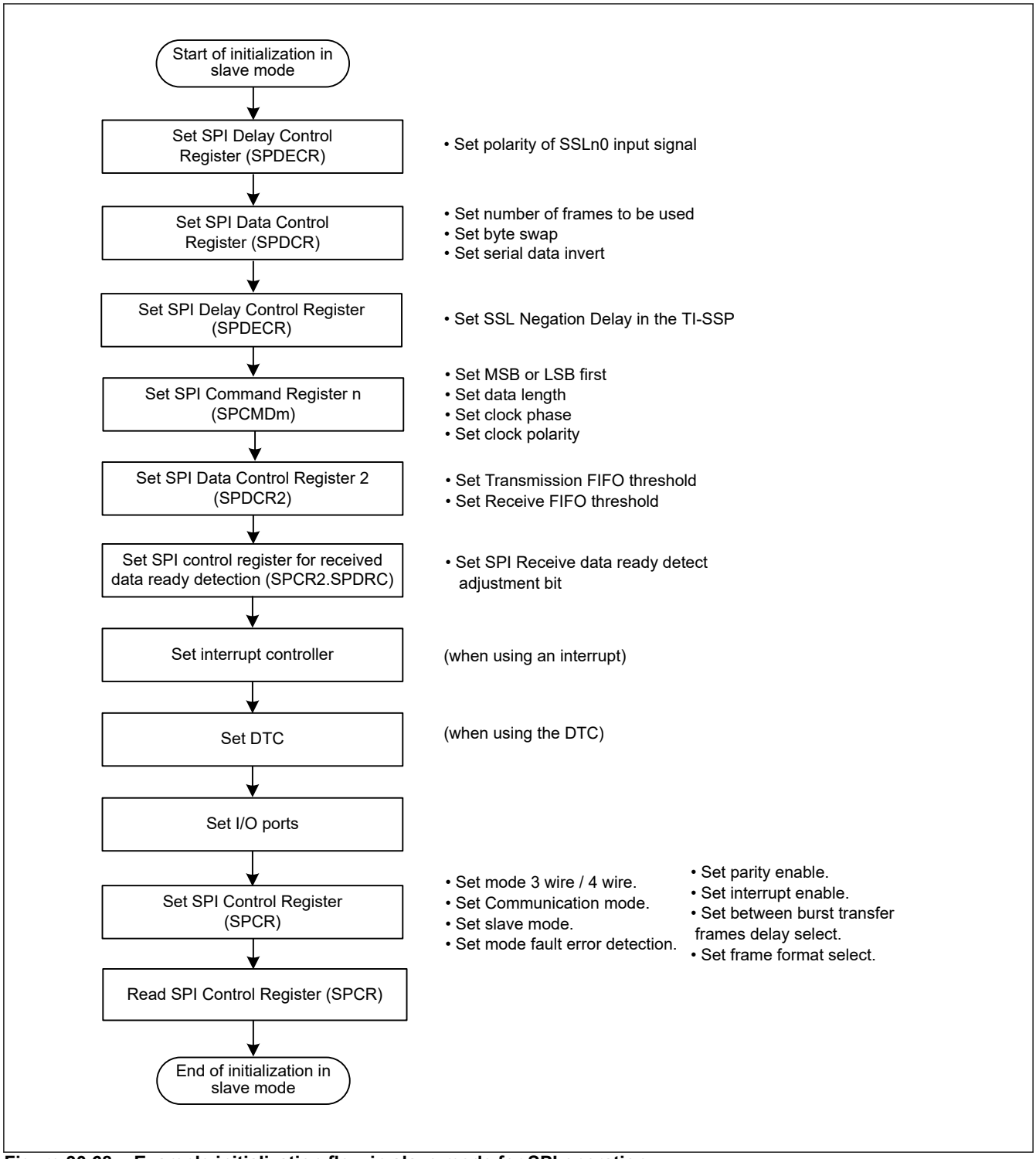
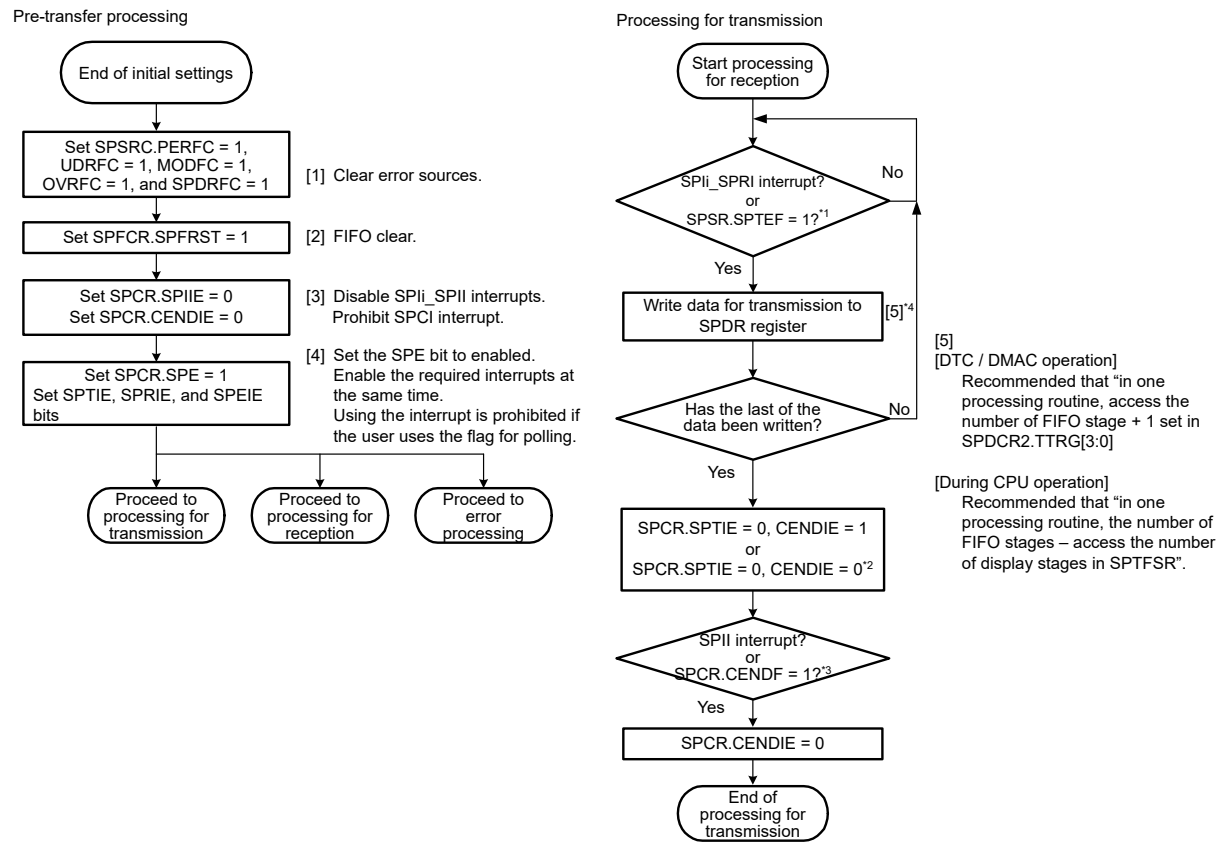


Figure 30.68 Example initialization flow in slave mode for SPI operation

(6) Software processing flow

Figure 30.69 to Figure 30.72 show examples of the flow of software processing.

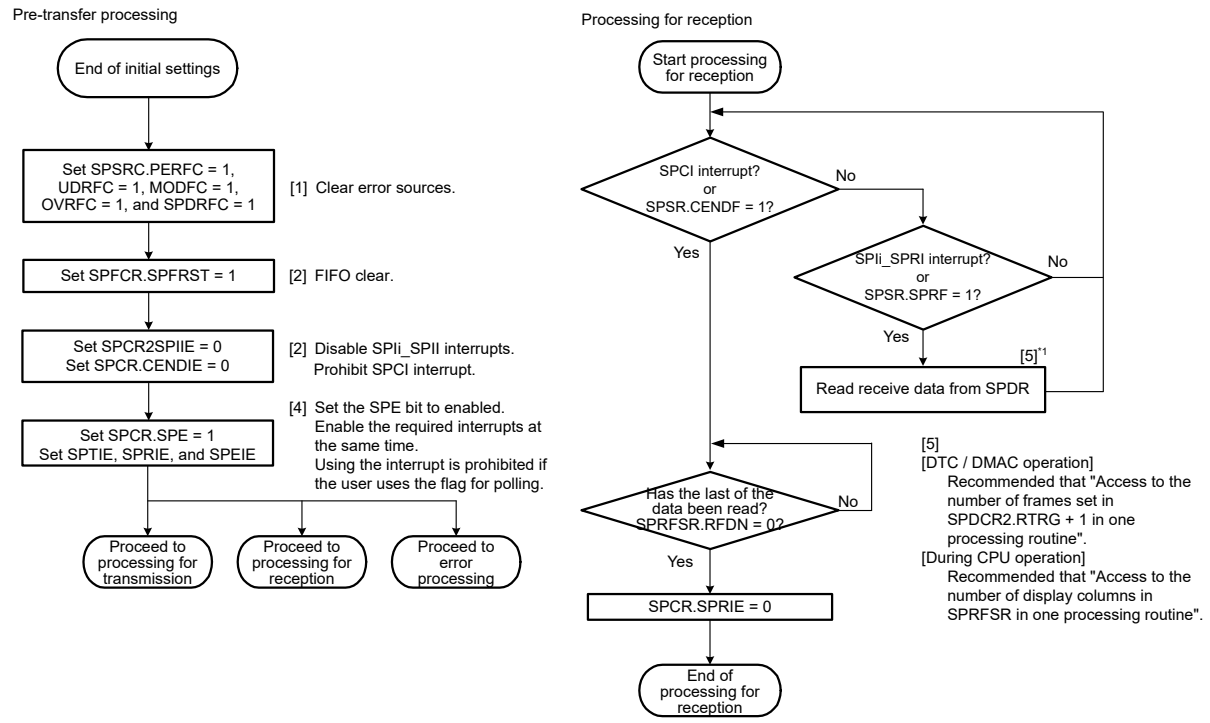
Transmit processing flow



Note 1. Before writing data for transmission to SPDR register, check the transmit buffer empty interrupt by reading SPSR.SPTEF flag, if the flag for polling is used.
Note 2. Set SPIi_SPCEND interrupt prohibited (SPCR.CENDIE = 0), if the user uses the flag for polling.
Note 3. Wait more than 1 PCLK, after writing data for transmission to SPDR, and before starting polling of SPSR.IDLNF or SPSR.CENDF, if the user uses the flag for polling.
Note 4. During CPU operation, write 1 to SPI transmit buffer empty flag clear bit (SPSRC.SPTEFC) after one processing routine.

Figure 30.69 Transmission flow in slave mode

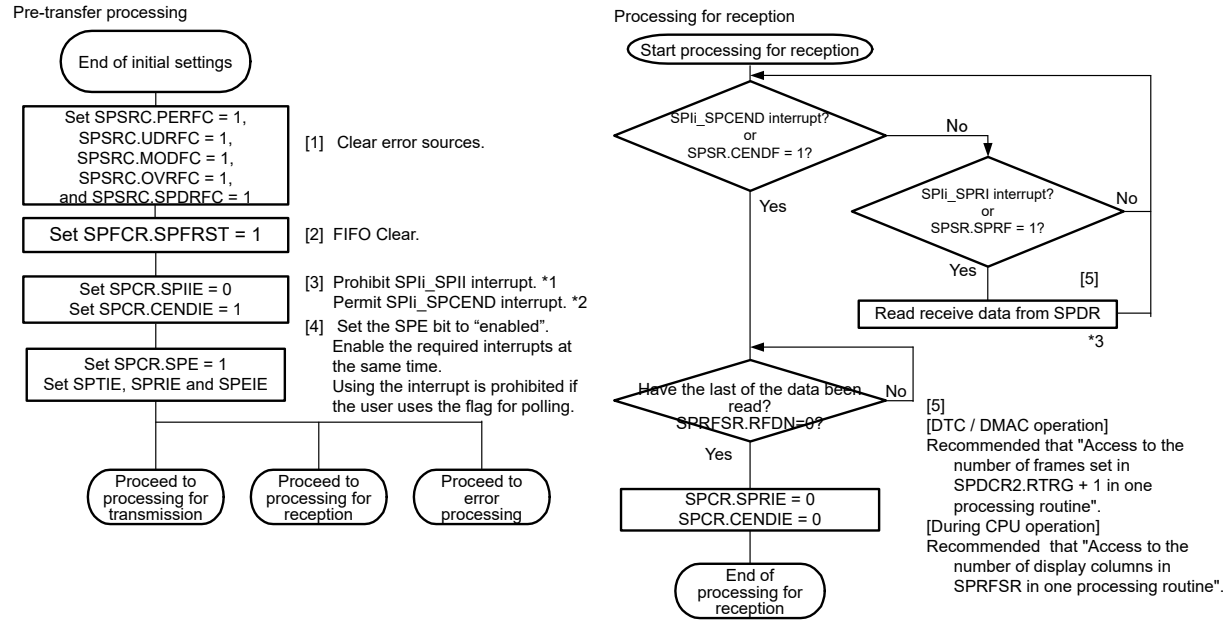
Receive processing flow



Note 1. During CPU operation, write 1 to the SPI receive buffer full flag clear bit (SPSRC.SPTEFC) after one processing routine.

Figure 30.70 Reception flow in slave mode

Master Reception-only processing flow



Note 1. The end of communication can be determined by the SPIi_SPCEND interrupt (CENDF flag). Use the SPIi_SPCEND interrupt (CENDF flag).
Note 2. Disable the SPIi_SPCEND interrupt (SPCR.CENDIE = 0) during flag polling.
Note 3. During CPU operation, write 1 to the SPI receive buffer full flag clear bit (SPSRC.SPTEFC) after one processing routine.

Figure 30.71 Software Processing Flowchart in Master Mode (Reception-only)

Error processing flow

In slave mode operation, even when a mode fault error is generated, the SPSR.MODF flag can be cleared regardless of the state of the SSLn0 pin.

When an error is detected by using an interrupt, clear the ICU.IELSRn.IR flag in the error processing routine. If this is not done, the ICU.IELSRn.IR flag might continue to indicate the SPIi_SPTI or SPIi_SPRI interrupt request. If the SPIi_SPRI interrupt request is indicated, read the receive buffer and initialize the sequencer in the SPI.

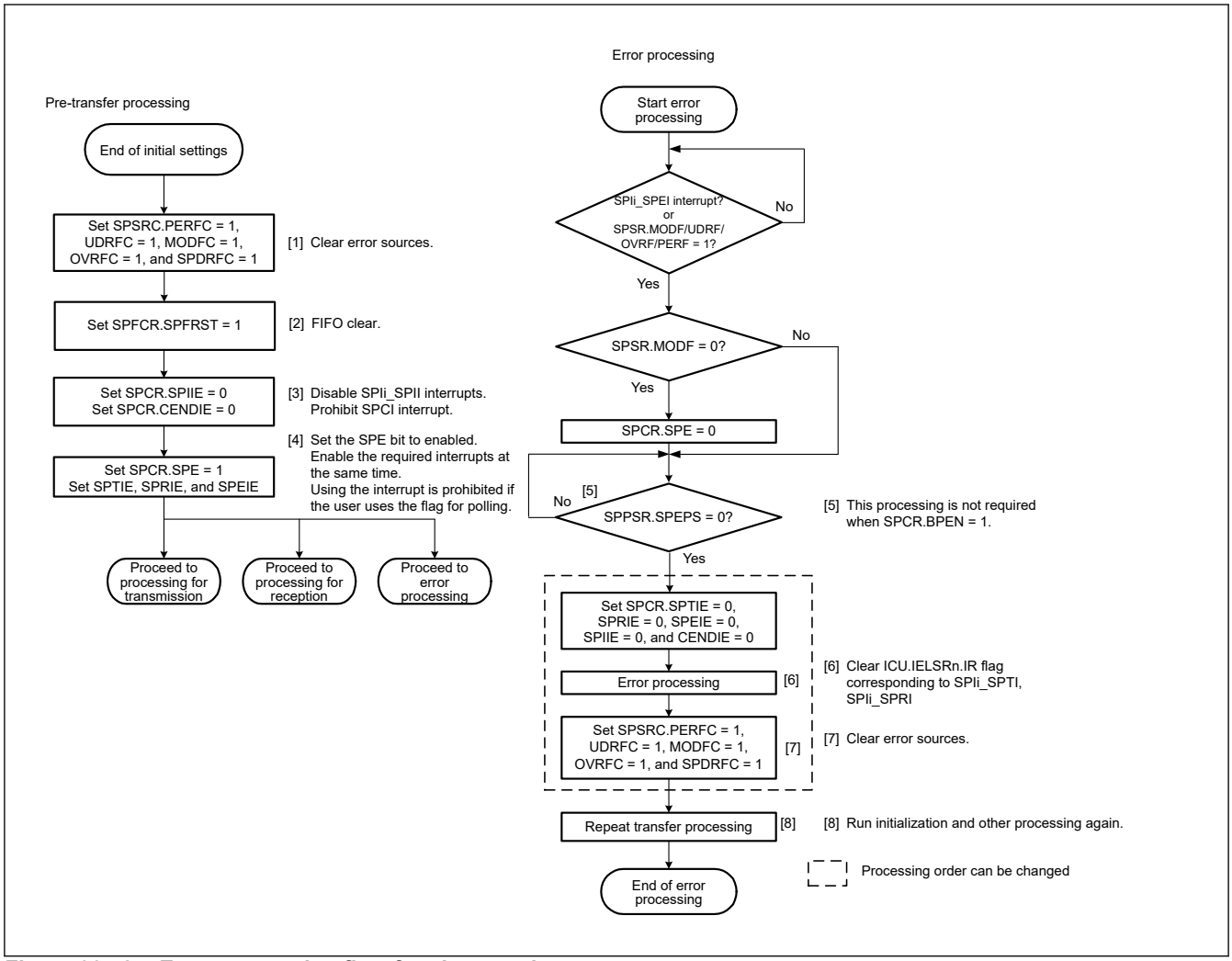


Figure 30.72 Error processing flow for slave mode

30.3.13 Clock Synchronous Operation

Setting the SPCR.SPMS bit to 1 selects clock synchronous operation of the SPI. In clock synchronous operation, the SSLni pin is not used, and the RSPCKn, MOSIn, and MISOn pins handle communications. All SSLni pins are available as I/O port pins.

Although clock synchronous operation does not require the use of the SSLni pin, operation of the module is the same as in SPI operation. In both master mode and slave mode operations, communications can be performed with the same flow as in SPI operation. However, mode fault errors are not detected, because the SSLni pin is not used.

Additionally, do not perform operation if clock synchronous operation is enabled when the SPCMDm.CPHA bit is set to 0 in slave mode (SPCR.MSTR = 0).

30.3.13.1 Master mode operation

(1) Starting serial transfer

When data is written to the SPI data register (SPDR) while the next transfer data is not set in the transmit FIFO, the SPI updates the transmit buffer (SPTXn, n = 0 to 3) data in SPDR.While the shift register is empty, the SPI copies transmit buffer data to the shift register to start serial transfer. After the SPI copies transmit data to the shift register, it changes the shift register status to full. Upon completion of serial transfer, the SPI changes the shift register status to empty. The shift register status cannot be monitored.

For details about the SPI transfer format, see section 30.3.5. Transfer Formats. In clock synchronous operation, however, the SSLn0 output signal is not used for communication.

(2) Terminating serial transfer

The SPI terminates the serial transfer after transmitting an RSPCKn edge corresponding to the sampling timing. If the number of data stored in the receive FIFO < the number of FIFO stages, on termination of serial transfer, the SPI copies data from the shift register to the receive buffer of the SPI Data Register (SPDR).

The final sampling timing varies depending on the bit length of transfer data. In master mode, the SPI data length depends on the SPCMDm.SPB[4:0] bits setting. Transfer in clock synchronous operation is conducted without the SSLn0 output signal. For details on the SPI transfer format, see section 30.3.5. Transfer Formats.

(3) Sequence control

The transfer format used in master mode is determined by the SPCR3, SPCMDm, and SPDECR registers. Although the SSLni signals are not output in clock synchronous operation, these settings are valid.

The SPCR3.SPSLN[2:0] bits determine the sequence configuration for serial transfers that are executed by the SPI in master mode. The following parameters are specified in the SPCMDm register:

- SSLni output signal value
- MSB or LSB first
- Data length
- Some of the bit rate settings
- RSPCKn polarity and phase
- Whether SPDECR.SCKDL is to be referenced
- Whether SPDECR.SLNDL is to be referenced
- Whether SPDECR.SPNDL is to be referenced

SPCR3.SPBR holds some of the bit rate settings such as SPDECR.SCKDL, an SPI clock delay value, SPDECR.SLNDL, an SSL negation delay, and SPDECR.SPNDL, a next-access delay value.

Based on the sequence length that is assigned to SPCR3, the SPI makes up a sequence comprised of a part or all of SPCMDm register. The SPI contains a pointer to the SPCMDm register that makes up the sequence. The value of this pointer can be checked by reading the SPDCR2.SPCP[2:0] bits. When the SPCR.SPE bit is set to 1 and the SPI function is enabled, the SPI loads the pointer to the commands in SPCMD0 register, and incorporates the SPCMD0 register setting into the transfer format at the beginning of serial transfer. The SPI increments the pointer each time the next-access delay period for a data transfer ends. On completion of the serial transfer that corresponds to the final command comprising the sequence, the SPI sets the pointer to the SPCMD0 register, and in this manner the sequence is executed repeatedly.

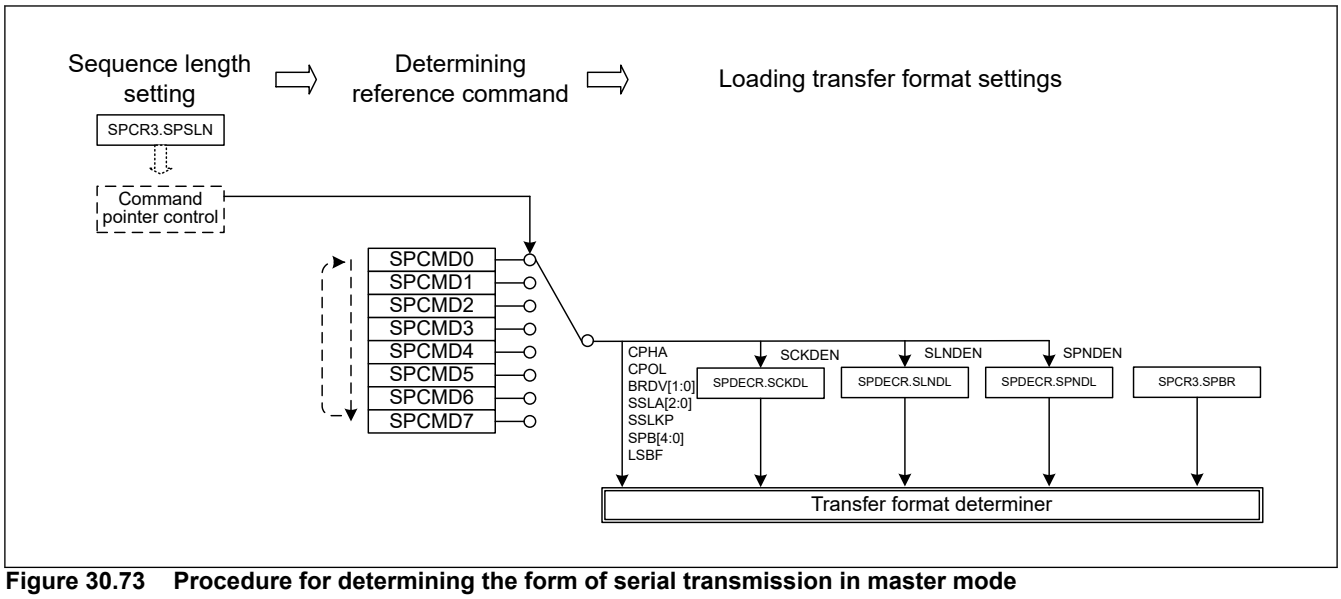


Figure 30.73 Procedure for determining the form of serial transmission in master mode

In this section, a frame is the combination of the data (SPDR) and the settings (SPCMDm).

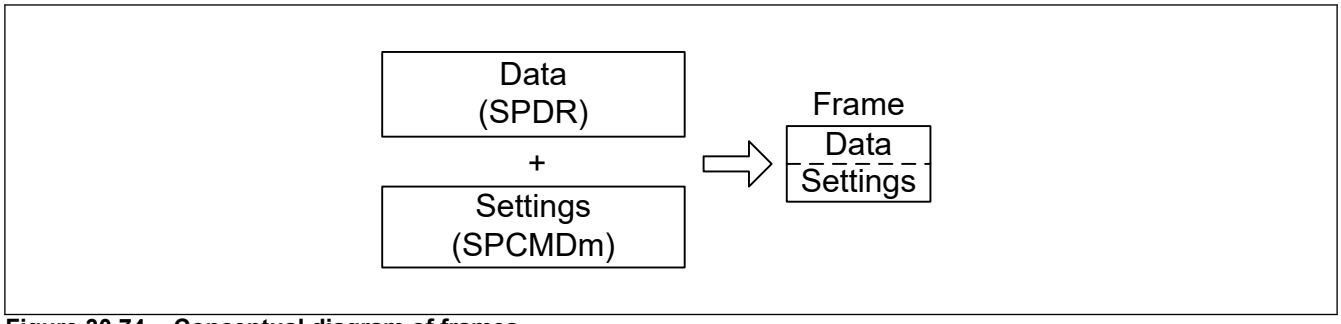


Figure 30.74 Conceptual diagram of frames

Figure 30.75 shows the relationship between the command and the transmit and receive buffers in the sequence of operations specified by the settings.

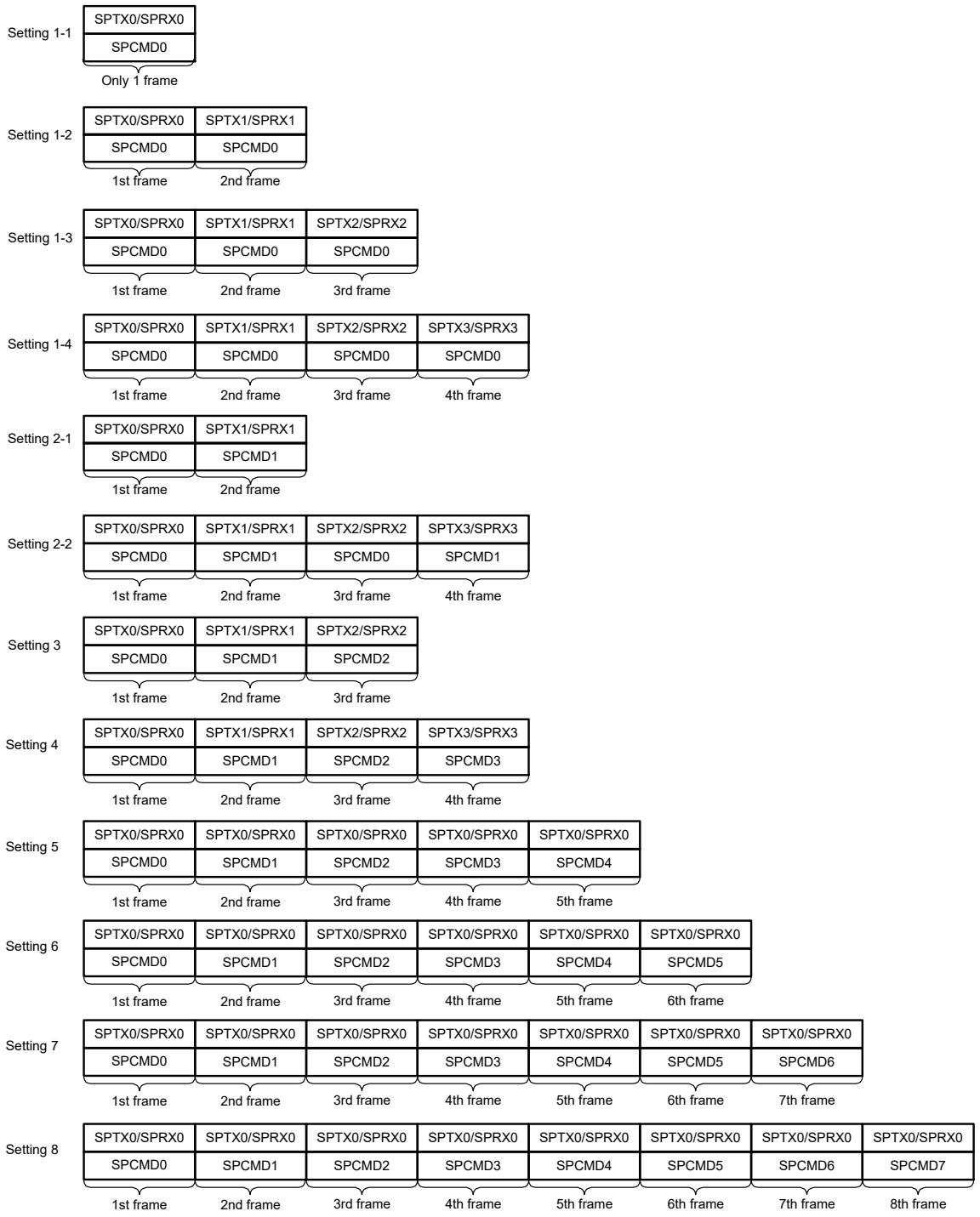


Figure 30.75 Correspondence between SPI Command Register and transmit and receive buffers in sequence operations

(4) Initialization flow

Figure 30.76 shows an example of initialization flow for clock synchronous operation when the SPI is used in master mode. For information on how to set up the ICU, DMAC or DTC, and I/O ports, see the descriptions given in the individual blocks.

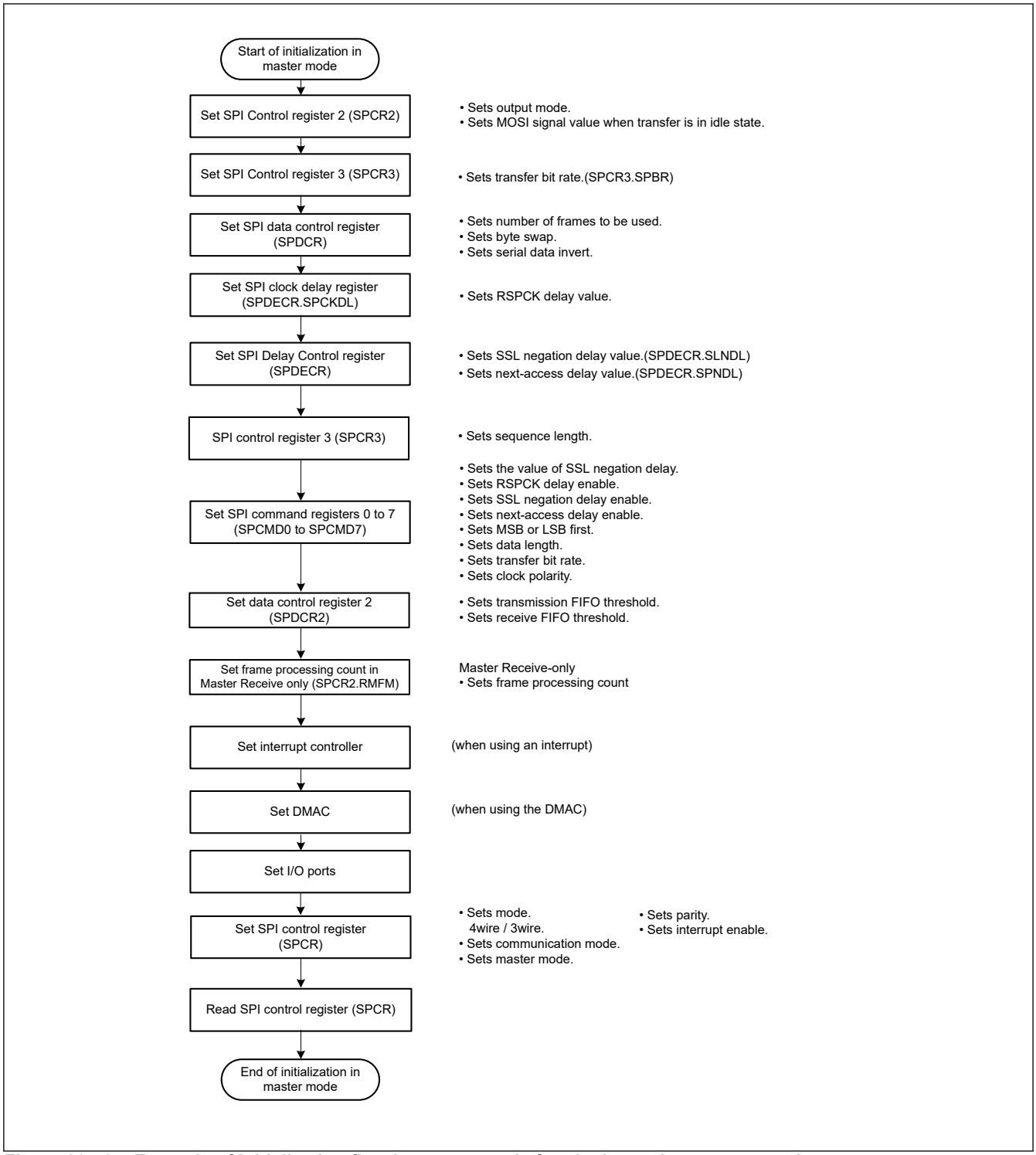


Figure 30.76 Example of initialization flow in master mode for clock synchronous operation

(5) Software processing flow

Software processing during clock synchronous master operation is the same as that for SPI master operation. For details, see (9) Software processing flow in section 30.3.12.1. Master mode operation. Mode fault errors do not occur in clock synchronous operation.

30.3.13.2 Slave mode operation

(1) Starting serial transfer

When the SPCR.SPMS bit is 1, the first RSPCKn edge triggers the start of a serial transfer in the SPI, and the SPI drives the MISOn output signal. The SSLn0 input signal is not used in clock synchronous operation. For details on the SPI transfer format, see section 30.3.5. Transfer Formats.

(2) Terminating serial transfer

The SPI terminates the serial transfer after detecting an RSPCKn edge corresponding to the final sampling timing. When the number of data stored in the receive FIFO < the number of FIFO stages, on termination of serial transfer, the SPI copies received data from the shift register to the receive buffer of the SPDR register. On termination of a serial transfer, the SPI changes the status of the shift register to empty regardless of the receive buffer.

The final sampling timing changes depending on the bit length of transfer data. In slave mode, the SPI data length depends on the SPCMD0.SPB[4:0] bits setting. For details on the SPI transfer format, see section 30.3.5. Transfer Formats.

(3) Initialization flow

Figure 30.77 shows an example of initialization flow for clock synchronous operation when the SPI is used in slave mode. For a description of how to set up the ICU, DTC, and I/O ports, see the descriptions given in the individual blocks.

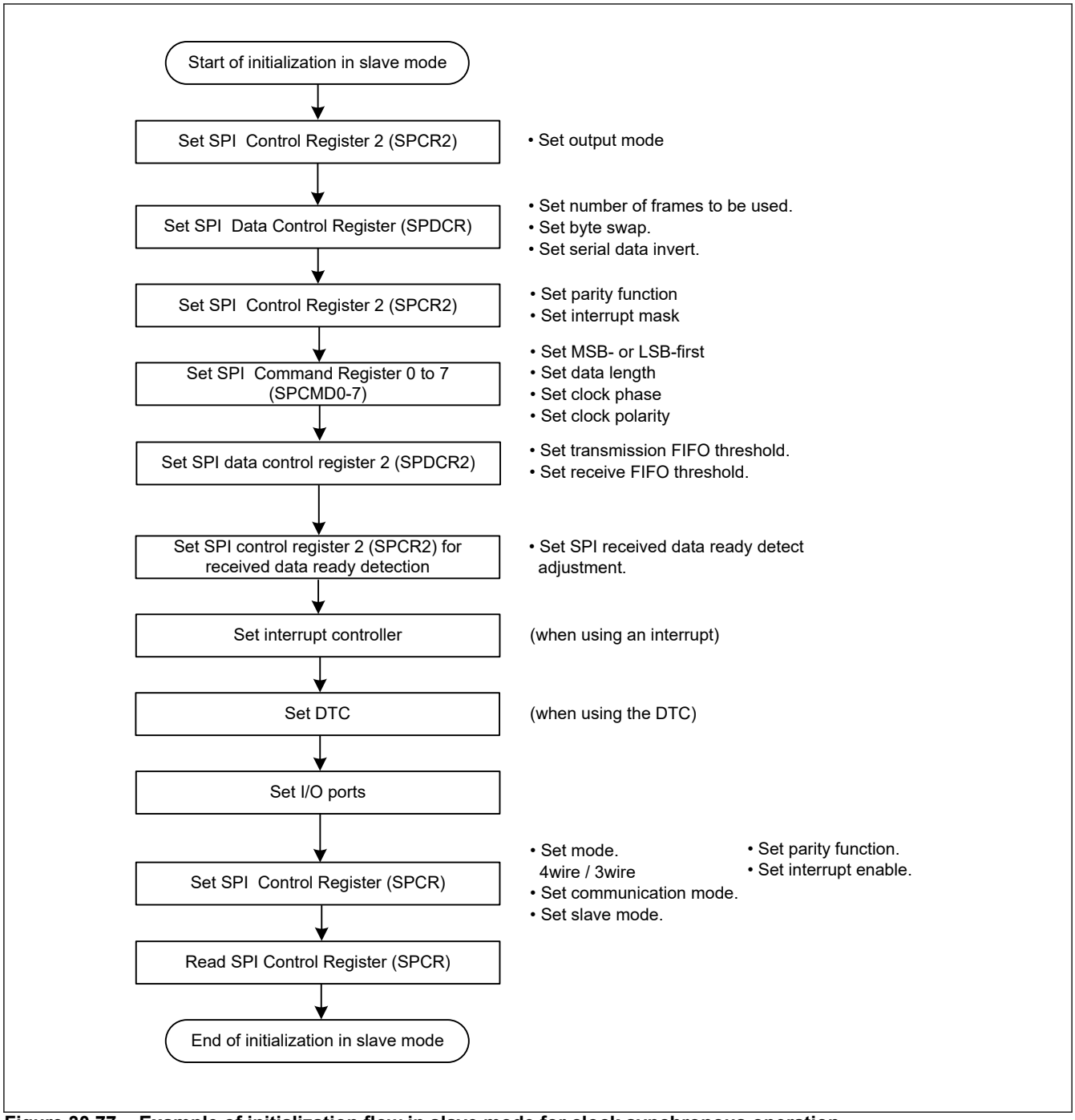


Figure 30.77 Example of initialization flow in slave mode for clock synchronous operation

(4) Software processing flow

Software processing during clock synchronous slave operation is the same as that for SPI slave operation. For details, see (6)Software processing flow. Mode fault errors do not occur in clock synchronous mode.

30.3.14 Loopback Mode

When 1 is written to the SPCR2.SPLP2 bit or SPCR2.SPLP bit, the SPI shuts off the path between the MISOn pin and the shift register if the SPCR.MSTR bit is 1, or between the MOSIn pin and the shift register if the SPCR.MSTR bit is 0, and connects the input and output paths of the shift register, establishing a loopback mode. The SPI does not shut off the path between the MOSIn pin and the shift register if the SPCR.MSTR bit is 1, or between the MISOn pin and the shift register if the SPCR.MSTR bit is 0. This is called loopback mode. When a serial transfer is executed in loopback mode, the transmit data for the SPI or the reversed transmit data becomes the received data for the SPI.

Table 30.13 lists the relationship between the SPLP2 and SPLP bits and the received data. Figure 30.78 shows the configuration of the shift register I/O paths when the SPI in master mode is set to loopback mode (SPCR2.SPLP2 = 0, SPCR2.SPLP = 1).

Table 30.13 SPLP2 and SPLP bit settings and received data

SPCR2.SPLP2 bit	SPCR2.SPLP bit	Received data
0	0	Input data from the MOSIn pin or MISOn pin
0	1	Inverted transmit data
1	0	Transmit data
1	1	Transmit data

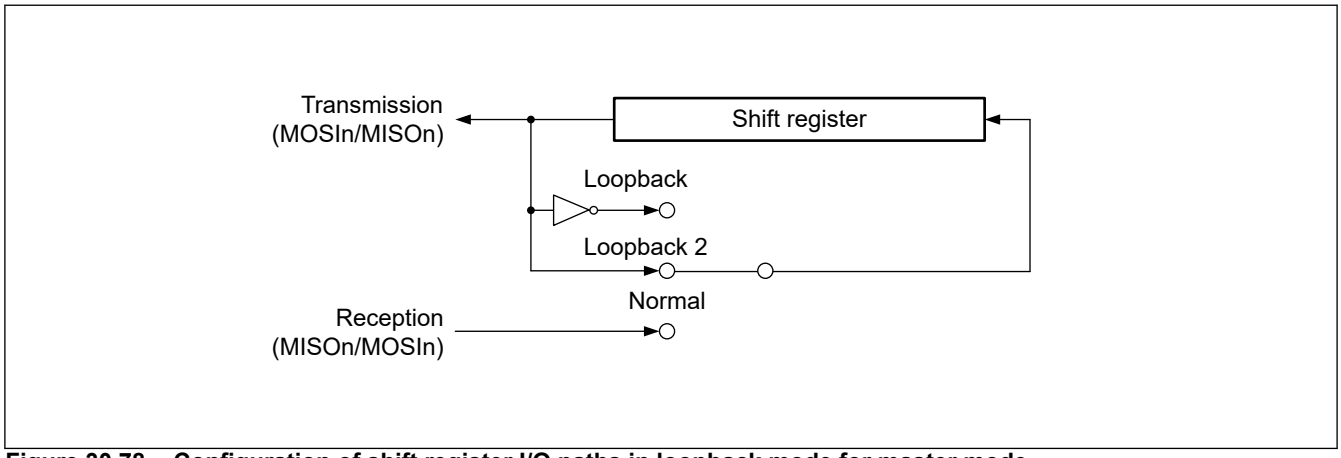


Figure 30.78 Configuration of shift register I/O paths in loopback mode for master mode

30.3.15 Self-Diagnosis of Parity Bit Function

The parity circuit consists of a parity bit adding unit used for transmit data and an error detecting unit used for received data. To detect defects in the parity bit adding unit and error detecting unit, the parity circuit performs self-diagnosis as shown in Figure 30.79.

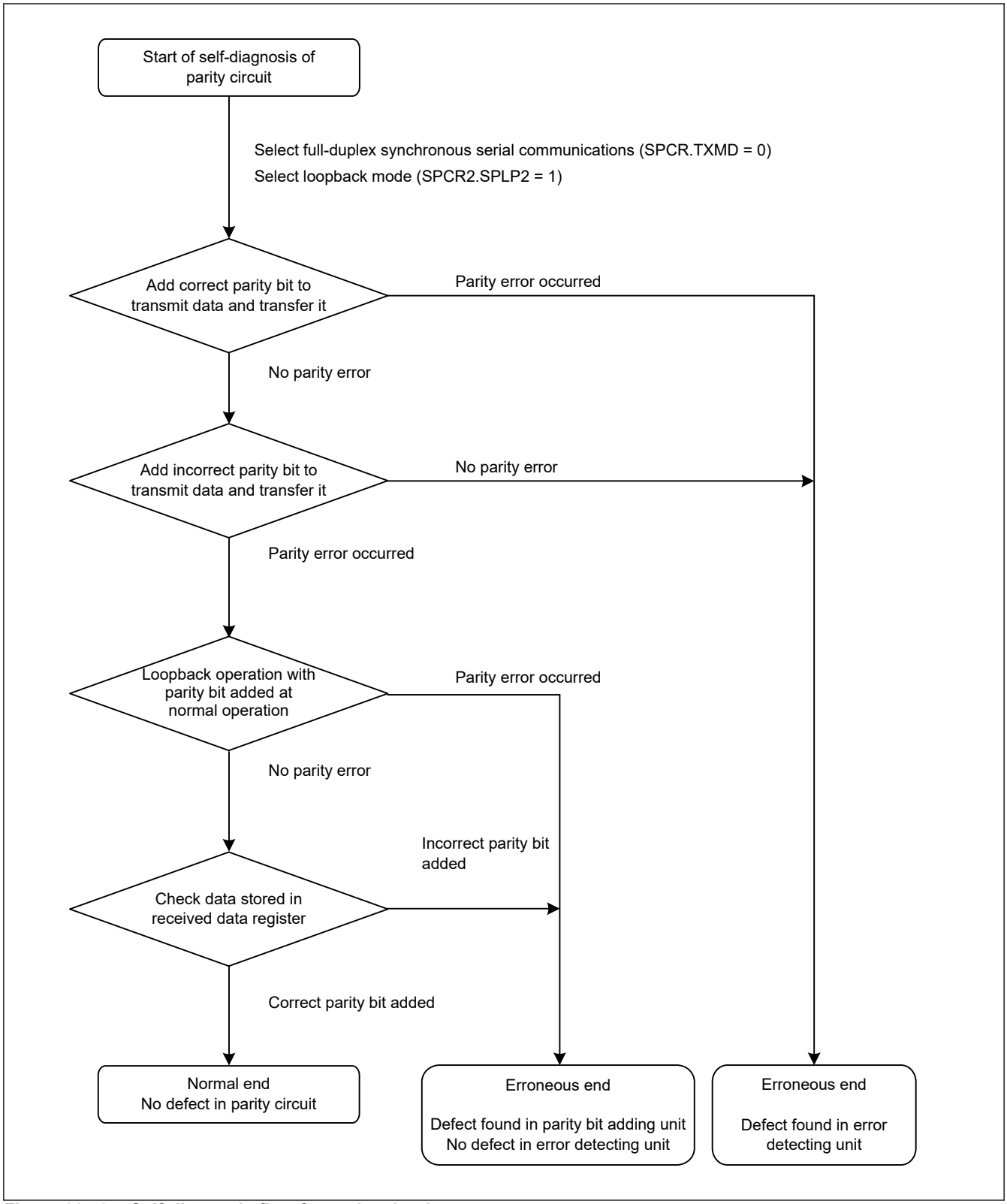


Figure 30.79 Self-diagnosis flow for parity circuit

30.3.16 Interrupt Sources

The SPI has the following interrupt sources:

- Receive buffer full
- Transmit buffer empty

- SPI error (mode-fault, underrun, overrun, or parity error)
- SPI idle
- Communication-end

The DMAC or DTC can be activated by the receive buffer full or transmit buffer empty interrupt to perform data transfer.

Because the vector address for the SPIi_SPEI (SPI error interrupt) is allocated to interrupt requests on mode-fault, underrun, overrun, and parity errors, the actual interrupt source must be determined from the flags. Interrupt sources for the SPI are listed in Table 30.14. An interrupt is generated on satisfaction of one of the interrupt conditions in Table 30.14. Clear the receive buffer full and transmit buffer empty sources through a data transfer.

When using the DMAC or DTC to perform data transmission and reception, you must first set up the DMAC or DTC to be in a transfer-enabled status before setting the SPI. For information on setting up the DMAC or DTC, see section 15, DMA Controller (DMAC) and section 16, Data Transfer Controller (DTC).

If the conditions for generating a transmit buffer empty or receive buffer full interrupt occur while the ICU.IELSRn.IR flag is 1, the interrupt is not output as a request for the ICU but is retained internally (the capacity for retention is one request per source). A retained interrupt request is output when the ICU.IELSRn.IR flag becomes 0. A retained interrupt request is automatically discarded when it is output as an actual interrupt request. The interrupt enable bit (the SPCR.SPTIE or SPCR.SPRIE bit) for an internally retained interrupt request can also be set to 0.

Table 30.14 SPI interrupt sources

Interrupt source	Symbol	Interrupt condition	DTC/DMAC activation
Receive buffer full	SPIi_SPRI	The receive buffer becomes full (SPSR.SPRF flag is 1) while the SPCR.SPRIE bit is 1 or The receive data become ready (SPSR.SPDRF flag is 1) while the SPCR.SPDRES bit is 0	Possible
Transmit buffer empty	SPIi_SPTI	The transmit buffer becomes empty (SPSR.SPTEF flag is 1) while the SPCR.SPTIE bit is 1	Possible
SPI error (mode-fault, underrun, overrun, or parity error)	SPIi_SPEI	The SPSR.MODF, OVRF,or PERF flag sets to 1, or the SPSR.SPDRES and SPDRF flag set to 1 while the SPCR.SPEIE bit is 1	Impossible
SPI idle	SPIi_SPII	The SPSR.IDLNF flag sets to 0 while the SPCR.SPIIE bit is 1	Impossible
Communication-end	SPIi_SPCEND	CENDIE = 1 and CENDF = 1	Impossible

30.4 Event Link Controller Event Output

The Event Link Controller (ELC) can produce the following event output signals:

- Receive buffer full event output
- Transmit buffer empty event output
- Mode-fault, underrun, overrun, or parity error event output
- SPI idle event output
- Transmission-completed event output

The event link output signal is output regardless of the interrupt enable bit setting.

30.4.1 Receive Buffer Full Event Output

When the number of data stored in the receive FIFO > the threshold value, or when the number of data stored in the receive FIFO ≤ the threshold value and SPDRES = 0 has elapsed after writing to the receive FIFO, and the SPDRC [7:0] has elapsed outputs an event.

30.4.2 Transmit Buffer Empty Event Output

An event is output when the number of empty transmission FIFO stage > the threshold or when the SPCR.SPE bit changes from 0 to 1.

30.4.3 Mode-Fault, Underrun, Overrun, Parity Error, or received data ready Event Output

This event signal is output when mode-fault, underrun, overrun, or parity error is detected. See section 30.5.4. Constraints on Mode-Fault, Underrun, Overrun, Parity Error, or Receive Data Ready Event Output if using this event signal.

(1) Mode-fault

Table 30.15 lists the conditions for occurrence of a mode-fault event.

Table 30.15 Conditions for mode-fault occurrence

SPI mode	SPCR.MODFEN bit	SSLn0 pin	Remarks
SPI operation (SPMS = 0) Slave (SPCR.MSTR = 0) Motorola-SPI (SPCR.SPFRF = 0)	1	Not active	Event is output only when the SSLn0 pin is deactivated during transmission
SPI operation (SPMS = 0) Slave (SPCR.MSTR = 0) TI-SSP (SPCR.SPFRF = 1)	1	active	Event is output only when the SSLn0 pin is activated during transmission

(2) Underrun

This event signal is output in response to an underrun when a serial transfer starts while the transmission data is not ready, and the value of the SPCR.MSTR bit is 0 and the SPCR.SPE bit is 1. Under these conditions, the MODF and UDRF flags are set to 1.

(3) Overrun

This event signal is output in response to an overrun when a serial transfer completes while the receive buffer contains unread data and the value of the SPCR.TXMD[1:0] bits are 00b or 10b. Under these conditions, the OVRF flag is set to 1.

(4) Parity error

This event signal is output in response to a parity error detected on completion of a serial transfer while the value of the SPPE bit in SPCR is 1.

(5) Receive Data Ready

When TXMD[1:0] of SPCR = 00b or 10b and SPDRES = 1 as the receive data ready event output condition, the number of data stored in the receive FIFO is received after writing the receive FIFO. An event will be output when the set value of SPDRC[7:0] has elapsed while the number is less than the FIFO threshold.

30.4.4 SPI Idle Event Output

(1) In master mode

In Transmit-Receive / Transmit-only master mode, an event is output when the IDLNF flag in SPSR changes from 1 to 0.

The IDLNF flag changes from 1 to 0 only when either of the conditions 1) and 2) below is met.

1. The SPE bit in SPCR is cleared to 0 (SPI initialized) during transmission.
2. All of the following three conditions are met.
   - The transmit buffer (SPTXn, n = 0 to 3) is empty (next transfer data has not been set).
   - The SPCP [2:0] bits in SPSR are 000b (at the start of sequence control).
   - Operation completed by the next access delay (when the master main state machine transitions to the idle state).

In receive only master mode

Any of the following 2 conditions is met.

1. SPE bit of SPCR is 0 (SPI initialization)
2. When any of the following is met
   - When RMFM [4:0] = 0x0, after writing 1 to RMEDTG, operation completed by the next access delay (when the master main state machine transitions to the idle state).

- When RMFM [4:0] ≠ 0x0, after writing 1 to RMEDTG, operation completed by the next access delay (when the master main state machine transitions to the idle state).
- When RMFM [4:0] ≠ 0x0, the SPI internal sequencer transitions to the idle state after operation completed by the next access delay (when the master main state machine transitions to the idle state).

(2) In slave mode

In slave mode, an event is output when the SPCR.SPE bit is set to 0 (SPI is initialized).

30.4.5 Communication End Event Output

In master mode, an event is output when the IDLNF flag (SPI idle flag) changes from 1 to 0. In slave mode, an event occurs with conditions shown in Table 30.16 and Table 30.17

Table 30.16 Communication End Event Generating Conditions (transmit-receive/transmit slave mode)

	Transmit Buffer Status	Shift Register Status	Others
SPI operation (SPMS = 0, SPFRF = 0)	Empty	Empty	SSLn0 input is negated
SPI operation (SPMS = 0, SPFRF = 1)	Empty	Empty	SSL negation delay completed
Clock synchronous operation (SPMS = 1)	Empty	Empty	The last even edge of RSPCK of last data was detected (CPHA = 1)

Table 30.17 Communication End Event Generating Conditions (receive only slave mode)

	Others
SPI operation (SPMS = 0, SPFRF = 0)	After storing the frames corresponding to the SPFC setting value in the receive buffer, negate SSLn0 input.
SPI operation (SPMS = 0, SPFRF = 1)	After storing the frames corresponding to the SPFC setting value in the receive buffer, SSL negation delay completed
Clock synchronous operation (SPMS = 1)	RSPCK last even edge detection when receiving the last frame for the SPFC set value (CPHA = 1)

Regardless of master mode or slave mode, no event is output when 0 is written to the SPCR.SPE bit during transmission or when the SPCR.SPE bit is cleared due to a mode fault error or an underrun error.

A communication end event is output at the following timing. The communication end event output timing in master operation is omitted because it is output at the same timing as an idle event.

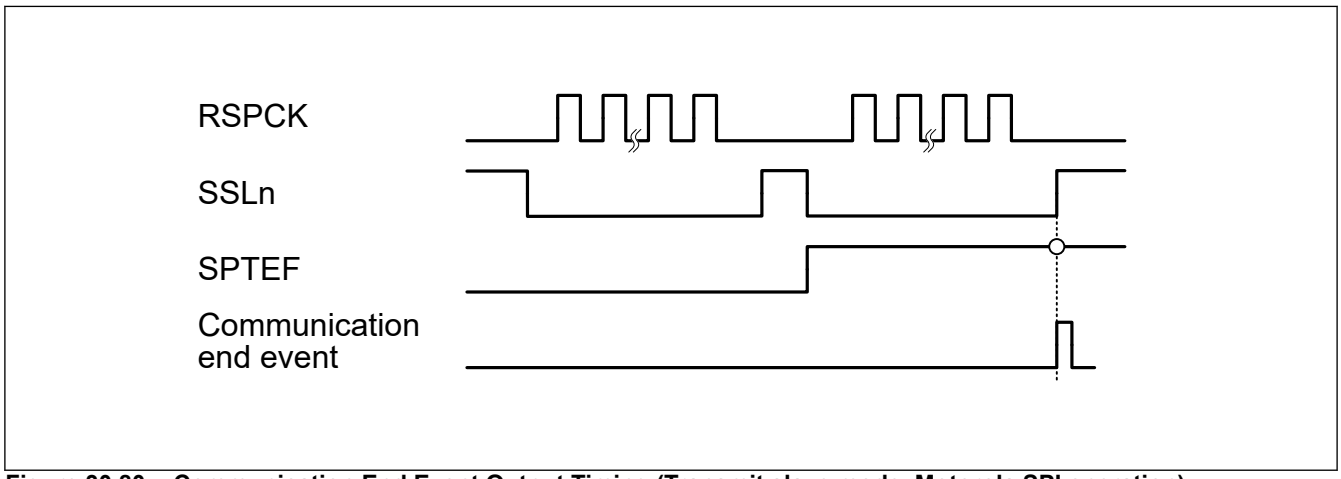


Figure 30.80 Communication End Event Output Timing (Transmit slave mode, Motorola SPI operation)

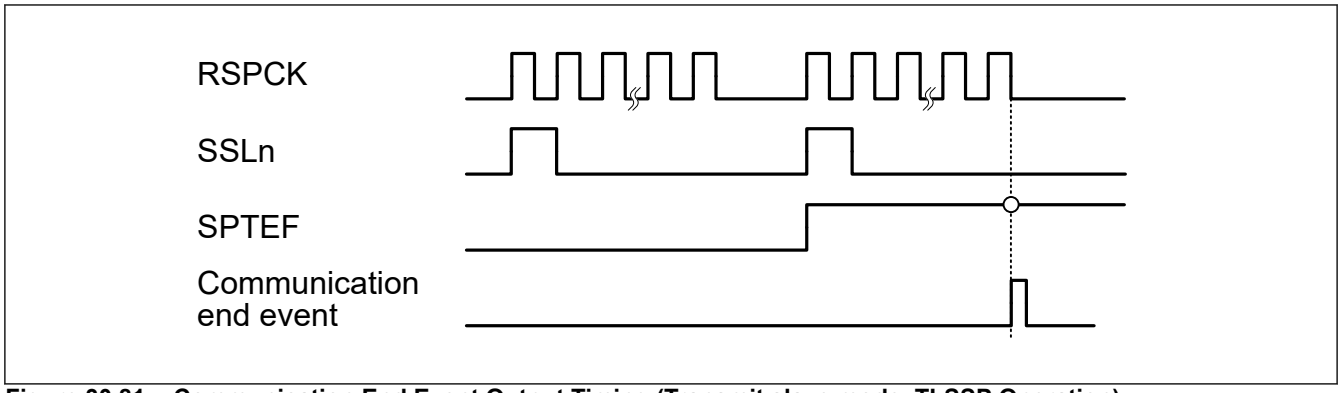


Figure 30.81 Communication End Event Output Timing (Transmit slave mode, TI-SSP Operation)

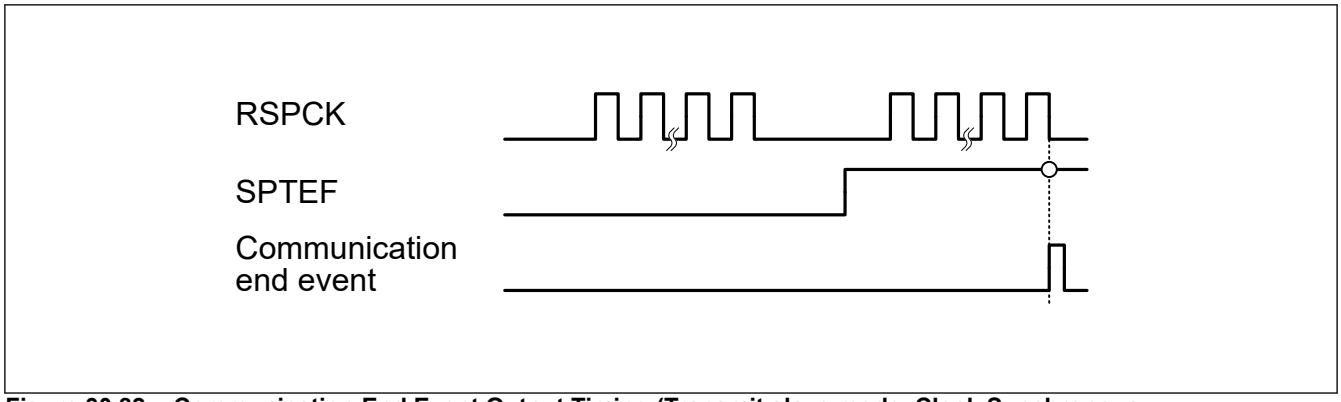


Figure 30.82 Communication End Event Output Timing (Transmit slave mode, Clock Synchronous Operation)

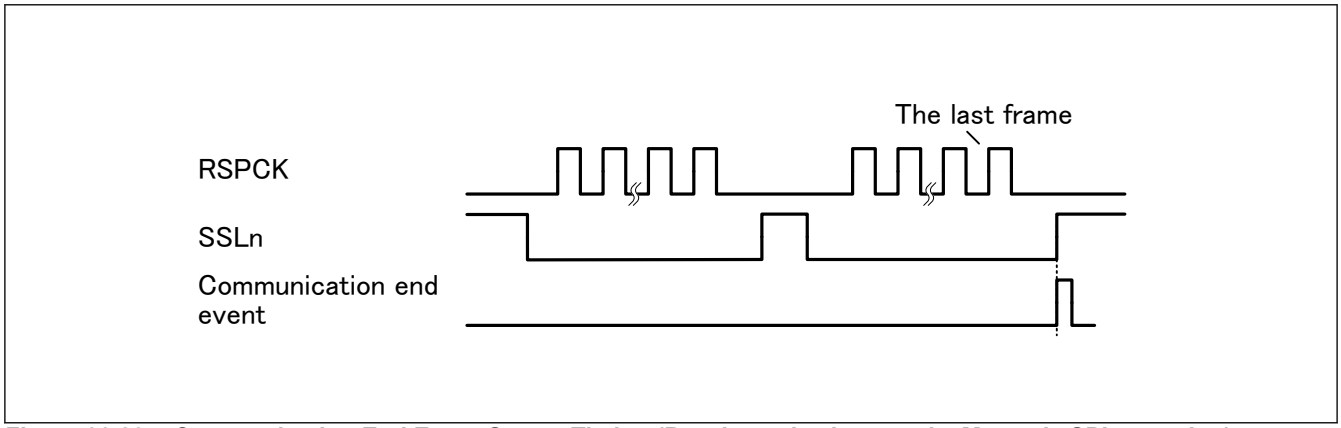


Figure 30.83 Communication End Event Output Timing (Receive only slave mode, Motorola SPI operation)

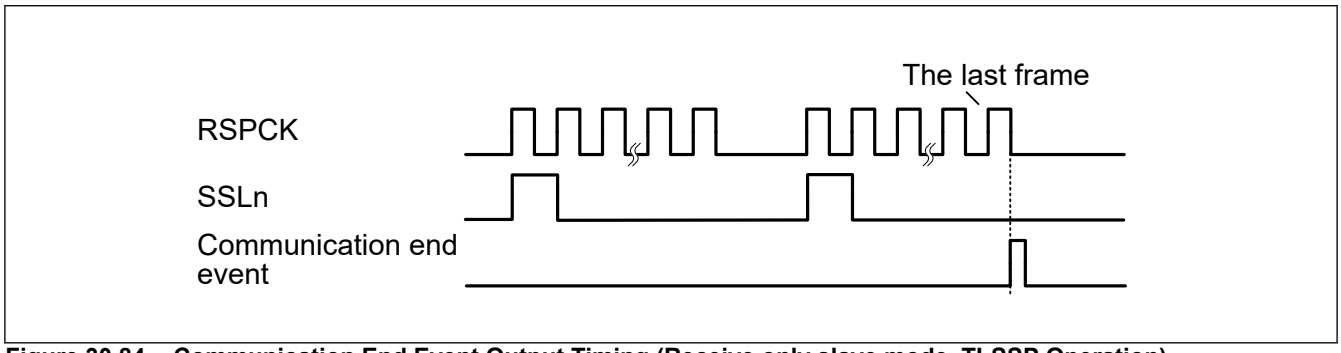


Figure 30.84 Communication End Event Output Timing (Receive only slave mode, TI-SSP Operation)

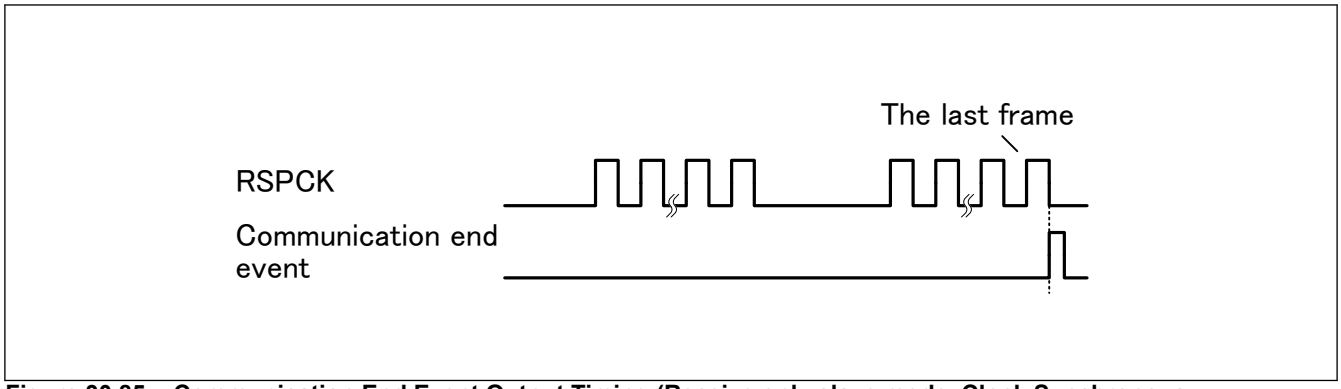


Figure 30.85 Communication End Event Output Timing (Receive only slave mode, Clock Synchronous Operation)

30.4.6 Synchronization bypass function

This IP has an internal clock (PCLK) and an operation clock (TCLK), and each has its own operation circuit. Therefore, a synchronization circuit is inserted between the signals between different clocks, and a signal delay between different clocks requires a synchronization delay time (2 to 3 PCLK or 2 to 3 TCLK).

However, the synchronization circuit can be bypassed by the BPEN = 1 of the SPI control register (SPCR) only when the same clock is input as the internal bus clock and the operation clock. In this case, the synchronization delay time is excluded, and responsiveness is improved.

In addition, this IP has a synchronization circuit between the communication clock (RSPCK) and the operation clock (TCLK), but this synchronization circuit cannot be bypassed.

30.5 Usage Notes

30.5.1 Settings for the Module-Stop State

The Module Stop Control Register B (MSTPCRB) can enable or disable the SPI operation. The SPI is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details on the Module Stop Control Register B, see section 10, Low Power Modes.

30.5.2 Constraint on Low-Power Functions

When using the module-stop function and entering a low-power mode other than Sleep mode, set the SPCR.SPE bit to 0 before completing communication.

30.5.3 Constraints on Starting Transfer

If the ICU.IELSRn.IR flag is 1 when transfer starts, the interrupt request is internally retained, which can lead to unanticipated behavior of the ICU.IELSRn.IR flag.

To prevent this, use the following procedure to clear interrupt requests before enabling operations (by setting the SPCR.SPE bit to 1):

1. Confirm that transfer stopped (the SPCR.SPE bit is 0).
2. Set the associated interrupt enable bit (SPCR.SPTIE bit or SPCR.SPRIE bit) to 0.
3. Read the associated interrupt enable bit (SPCR.SPTIE bit or SPCR.SPRIE bit) and confirm that its value is 0.
4. Set the ICU.IELSRn.IR flag to 0.

30.5.4 Constraints on Mode-Fault, Underrun, Overrun, Parity Error, or Receive Data Ready Event Output

Using the mode-fault, underrun, overrun, parity error, or receive data ready event is prohibited if the SPI is in multi-master mode (when the SPCR.SPMS bit is 0, the SPCR.MSTR bit is 1, and the SPCR.MODFEN bit is 1).

30.5.5 Constraints on the SPSR.SPRF and SPSR.SPTEF Flags

If the polling flags, SPRF and SPTEF, are used, using the interrupts is prohibited, and you must set the SPCR.SPRIE and SPCR.SPTIE bits to 0. Either the interrupts or the flags can be used, but not both.

31. Cyclic Redundancy Check (CRC)

31.1 Overview

The Cyclic Redundancy Check (CRC) generates CRC codes to detect errors in the data. The bit order of CRC calculation results can be switched for LSB-first or MSB-first communication. Additionally, various CRC-generation polynomials are available. The snoop function allows to monitor the access to specific addresses. This function is useful in applications that require CRC code to be generated automatically in certain events, such as monitoring writes to the serial transmit buffer and reads from the serial receive buffer.

Table 31.1 lists the CRC calculator specifications and Figure 31.1 shows a block diagram.

Table 31.1 CRC calculator specifications

Item	Description	
Data size	8-bit	32-bit
Data for CRC calculation*1	CRC code generated for data in 8n-bit units (where n is a natural number)	CRC code generated for data in 32n-bit units (where n is a natural number)
CRC processor unit	Operation executed on 8 bits in parallel	Operation executed on 32 bits in parallel
CRC generating polynomial	One of three generating polynomials that is selectable: [8-bit CRC] • $X^8 + X^2 + X + 1$ (CRC-8) [16-bit CRC] • $X^{16} + X^{15} + X^2 + 1$ (CRC-16) • $X^{16} + X^{12} + X^5 + 1$ (CRC-CCITT).	One of two generating polynomials that is selectable: [32-bit CRC] • $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$ (CRC-32) • $X^{32} + X^{28} + X^{27} + X^{26} + X^{25} + X^{23} + X^{22} + X^{20} + X^{19} + X^{18} + X^{14} + X^{13} + X^{11} + X^{10} + X^9 + X^8 + X^6 + 1$ (CRC-32C).
CRC calculation switching	The bit order of CRC calculation results can be switched for LSB-first or MSB-first communication	
Module-stop function	Module-stop state can be set to reduce power consumption	
CRC snoop	Monitor reads from and writes to a certain register address	
TrustZone Filter	Security attribution can be set	

Note 1. This function cannot divide data used in CRC calculations. Write data in 8-bit or 32-bit units.

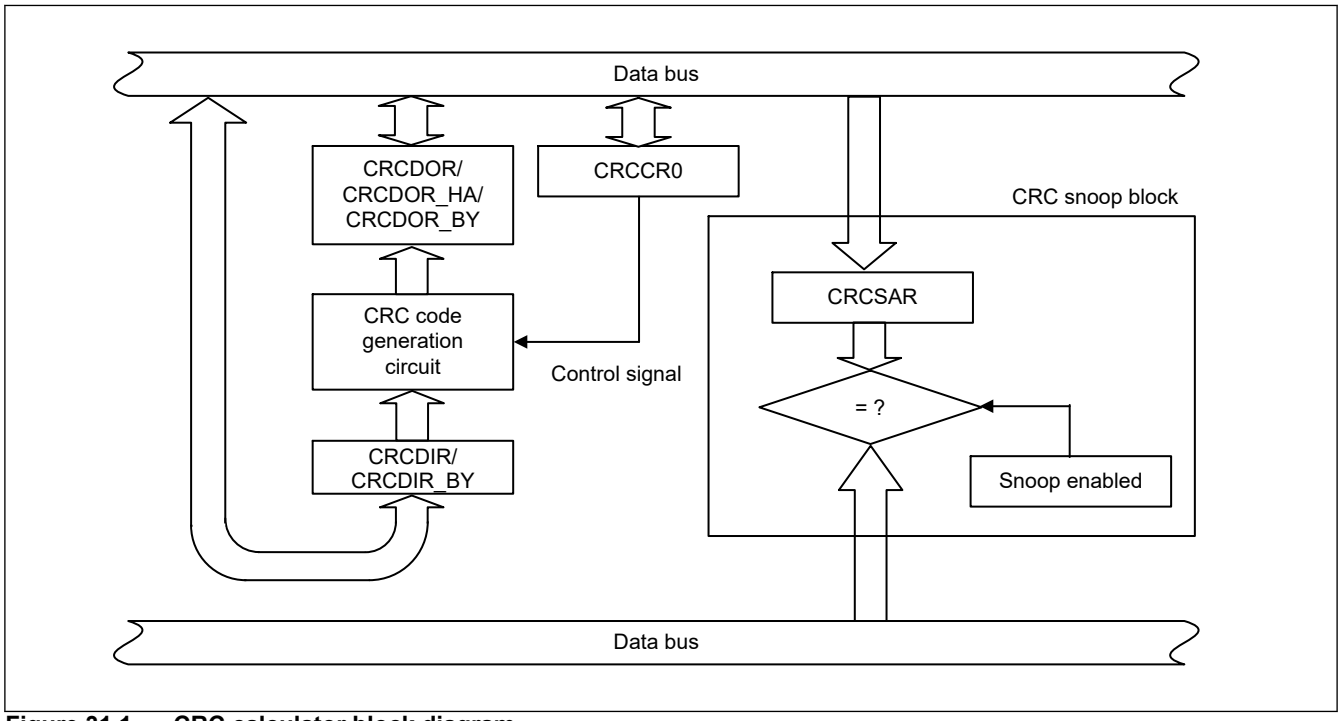


Figure 31.1 CRC calculator block diagram

31.2 Register Descriptions

31.2.1 CRCCR0 : CRC Control Register 0

Base address: CRC = 0x4010_8000

Offset address: 0x00

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DORCLR	LMS	—	—	—	GPS[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	GPS[2:0]	CRC Generating Polynomial Switching 0 0 1: 8-bit CRC-8 ($X^8 + X^2 + X + 1$) 0 1 0: 16-bit CRC-16 ($X^{16} + X^{15} + X^2 + 1$) 0 1 1: 16-bit CRC-CCITT ($X^{16} + X^{12} + X^5 + 1$) 1 0 0: 32-bit CRC-32 ($X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$) 1 0 1: 32-bit CRC-32C ($X^{32} + X^{28} + X^{27} + X^{26} + X^{25} + X^{23} + X^{22} + X^{20} + X^{19} + X^{18} + X^{14} + X^{13} + X^{11} + X^{10} + X^9 + X^8 + X^6 + 1$) Others: No calculation is executed	R/W
5:3	—	These bits are read as 0. The write value should be 0.	R/W
6	LMS	CRC Calculation Switching 0: Generate CRC code for LSB-first communication 1: Generate CRC code for MSB-first communication	R/W
7	DORCLR	CRCDOR/CRCDOR_HA/CRCDOR_BY Register Clear 0: No effect 1: Clear the CRCDOR/CRCDOR_HA/CRCDOR_BY register	W

GPS[2:0] bits (CRC Generating Polynomial Switching)

The GPS[2:0] bits select the CRC generating polynomial.

LMS bit (CRC Calculation Switching)

The LMS bit selects the bit order of generated CRC code. Transmit the lower byte of the CRC code first for LSB-first communication and the upper byte first for MSB-first communication. For details on transmitting and receiving CRC code, see section 31.3. Operation.

DORCLR bit (CRCDOR/CRCDOR_HA/CRCDOR_BY Register Clear)

Write 1 to the DORCLR bit to set the CRCDOR/CRCDOR_HA/CRCDOR_BY register to 0x00000000. This bit is read as 0. Only 1 can be written to it.

31.2.2 CRCCR1 : CRC Control Register 1

Base address: CRC = 0x4010_8000

Offset address: 0x01

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

Bit	Symbol	Function	R/W
5:0	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
6	CRCSWR	Snoop-On-Write/Read Switch 0: Snoop-on-read 1: Snoop-on-write	R/W
7	CRCSEN	Snoop Enable 0: Disabled 1: Enabled	R/W

CRCSWR bit (Snoop-On-Write/Read Switch)

The CRCSWR bit selects the direction of access in the CRC snoop function.

When this bit is set to 0 (initial value), the CRC snoop operation to read a specific register is enabled. Similarly, when this bit is set to 1, the CRC snoop operation to write a specific register is enabled.

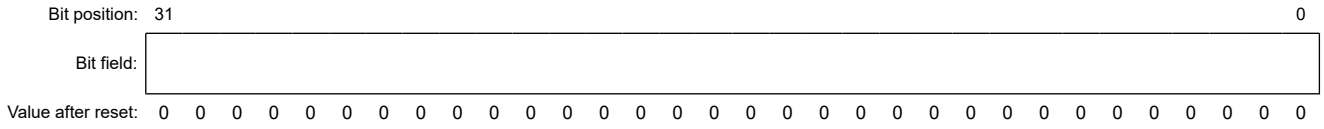
CRCSEN bit (Snoop Enable)

When the CRCSEN bit is set to 1, the CRC snoop operation is enabled. When this bit is set to 0, the CRC snoop operation is disabled.

31.2.3 CRCDIR/CRCDIR_BY : CRC Data Input Register

Base address: CRC = 0x4010_8000

Offset address: 0x04

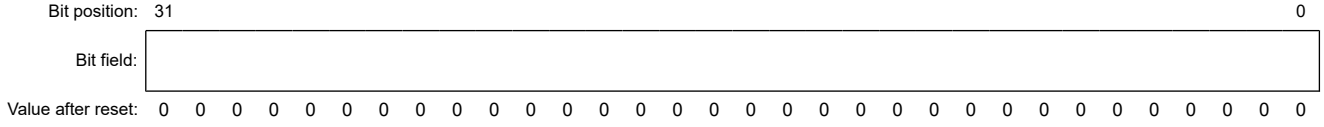


Bit	Symbol	Function	R/W
31:0	n/a	CRC input data The CRCDIR register is a 32-bit read/write register to write data for CRC-32 or CRC-32C calculation. The CRCDIR_BY (CRCDIR[31:24]) is an 8-bit read/write register to write data for CRC-8, CRC-16, or CRC-CCITT calculation.	R/W

31.2.4 CRCDOR/CRCDOR_HA/CRCDOR_BY : CRC Data Output Register

Base address: CRC = 0x4010_8000

Offset address: 0x08



Bit	Symbol	Function	R/W
31:0	n/a	CRC output data The CRCDOR register is a 32-bit read/write register for CRC-32 or CRC-32C calculation. The CRCDOR_HA (CRCDOR[31:16]) register is a 16-bit read/write register for CRC-16 or CRC-CCITT calculation. The CRCDOR_BY (CRCDOR[31:24]) register is an 8-bit read/write register for CRC-8 calculation. Because its initial value is 0x00000000, rewrite the CRCDOR/CRCDOR_HA/CRCDOR_BY register to perform the calculations using a value other than the initial value. Data written to the CRCDIR/CRCDIR_BY register is CRC calculated and the result is stored in the CRCDOR/CRCDOR_HA/CRCDOR_BY register. If the CRC code is calculated following the transferred data and the result is 0x00000000, there is no CRC error.	R/W

31.2.5 CRCSAR : Snoop Address Register

Base address: CRC = 0x4010_8000

Offset address: 0x0C

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

Bit	Symbol	Function	R/W
13:0	CRCSA[13:0]	Register Snoop Address These bits store the TDR or RDR address in the SCI module to snoop	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W

CRCSA[13:0] bits (Register Snoop Address)

The CRCSA[13:0] bits specify the lower address 14 bits of the register monitored by the CRC snoop operation.

Only the following addresses can be used for the CRCSA[13:0] bits:

- 0x4011_8004: SCI0.TDR, 0x4011_8000:SCI0.RDR
- 0x4011_8104: SCI1.TDR, 0x4011_8100:SCI1.RDR
- 0x4011_8204: SCI2.TDR, 0x4011_8200:SCI2.RDR
- 0x4011_8304: SCI3.TDR, 0x4011_8300:SCI3.RDR
- 0x4011_8404: SCI4.TDR, 0x4011_8400:SCI4.RDR
- 0x4011_8904: SCI9.TDR, 0x4011_8900:SCI9.RDR

31.3 Operation

31.3.1 Basic Operation

The CRC calculator generates CRC codes for use in LSB-first or MSB-first transfer.

The following examples show CRC code generation for input data (0xF0) using the 16-bit CRC-CCITT generating polynomial ($X^{16} + X^{12} + X^5 + 1$). In these examples, the value of the CRC Data Output Register (CRCDOR_HA) is cleared before CRC calculation.

When an 8-bit CRC (with the polynomial $X^8 + X^2 + X + 1$) is in use, the valid bits of the CRC code are obtained in CRCDOR_BY. When a 32-bit CRC is in use, the valid bits of the CRC code are obtained in CRCDOR.

Figure 31.2 and Figure 31.3 show the LSB-first and MSB-first data transmission examples respectively. Figure 31.4 and Figure 31.5 show the LSB-first and MSB-first data reception examples.

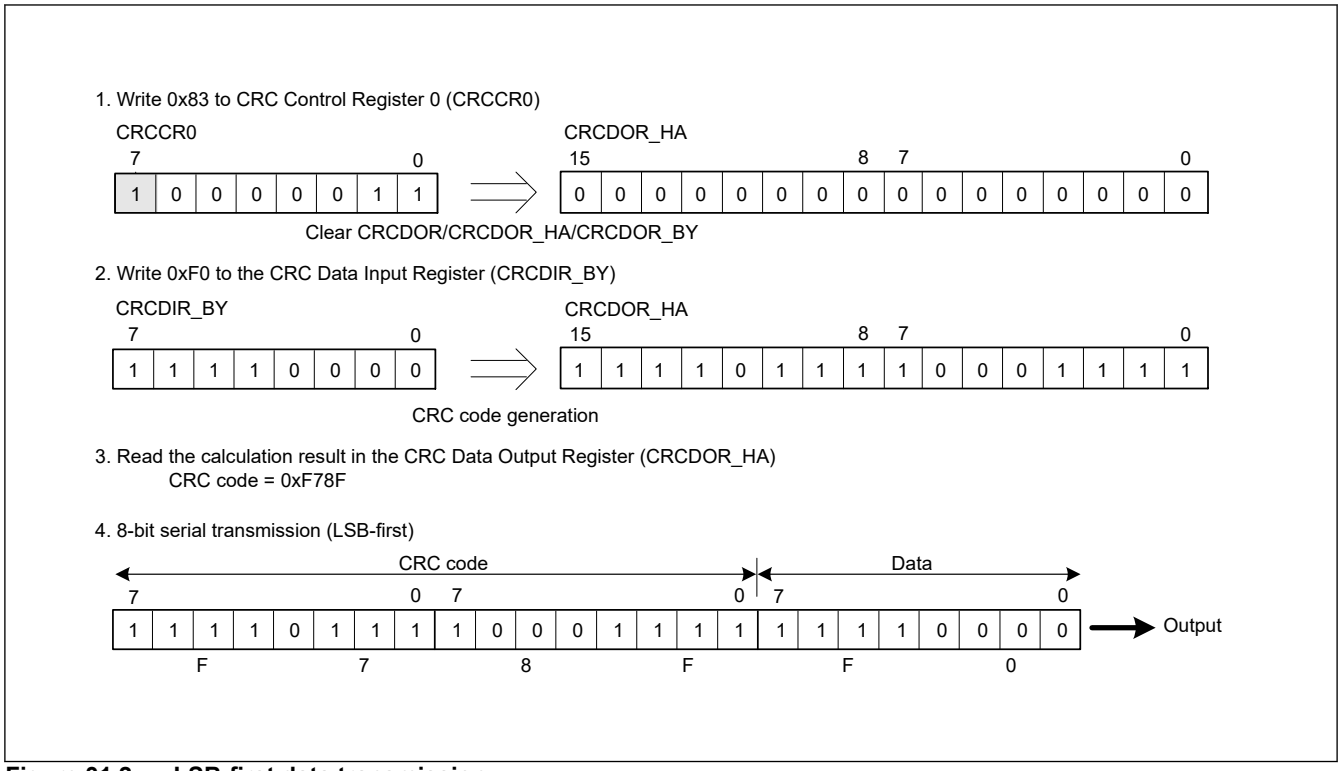


Figure 31.2 LSB-first data transmission

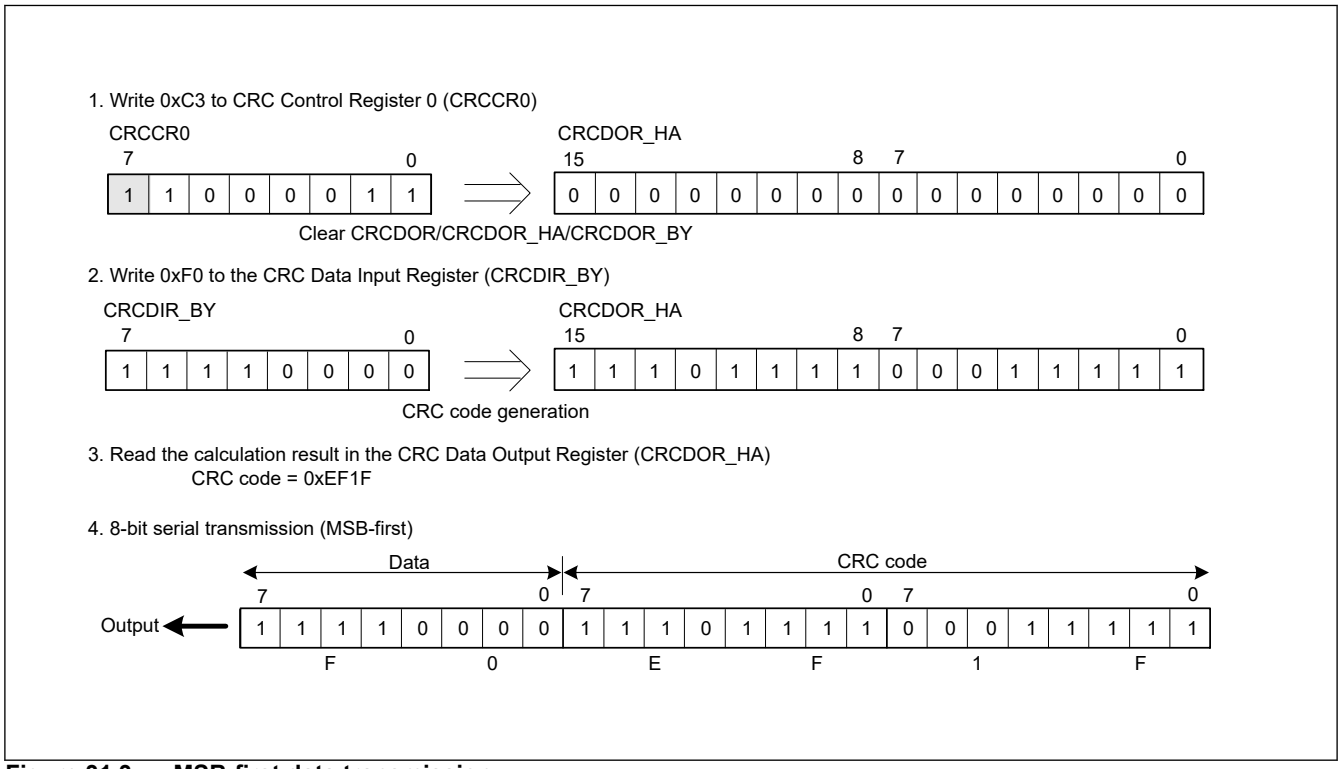


Figure 31.3 MSB-first data transmission

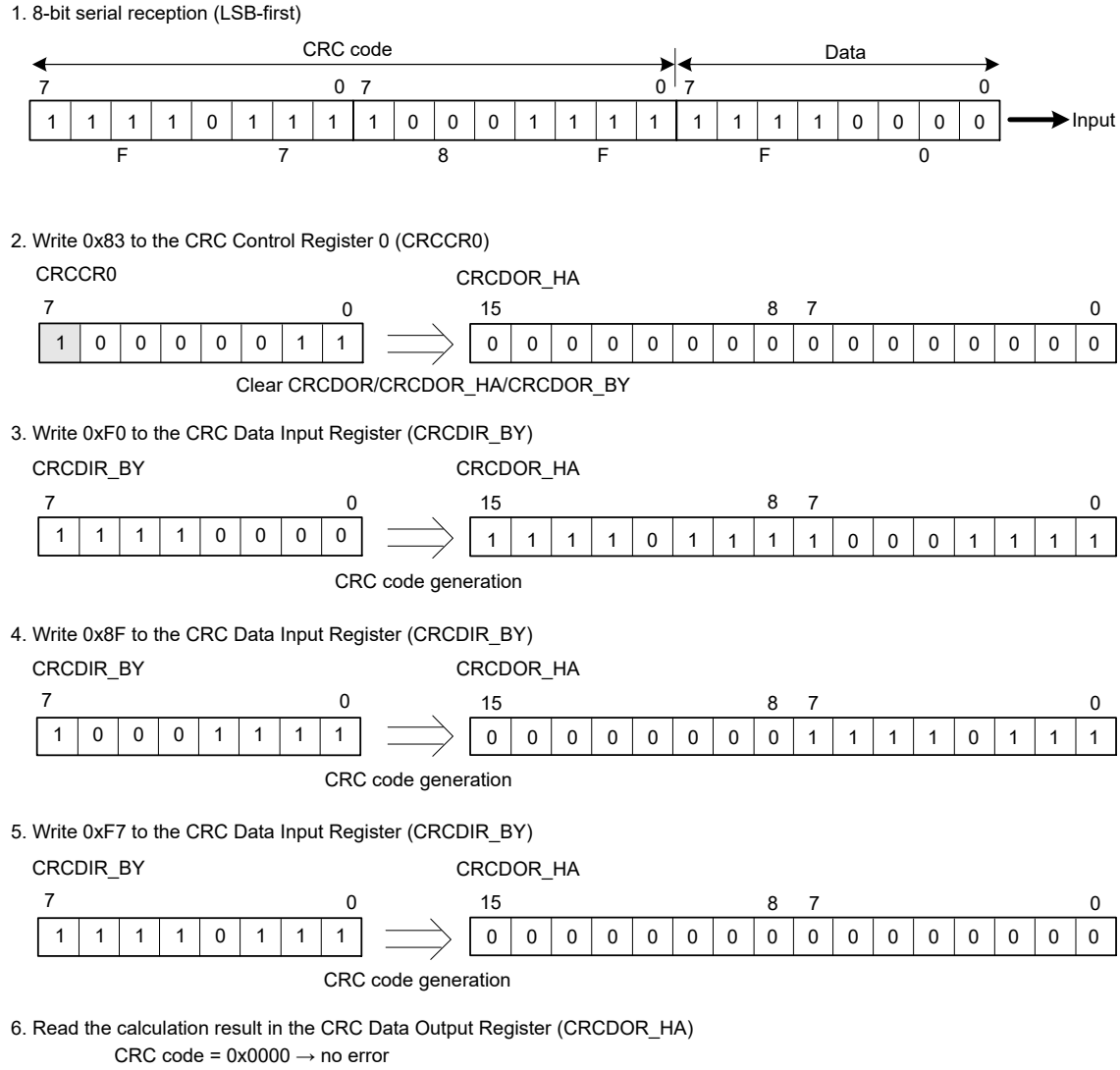


Figure 31.4 LSB-first data reception

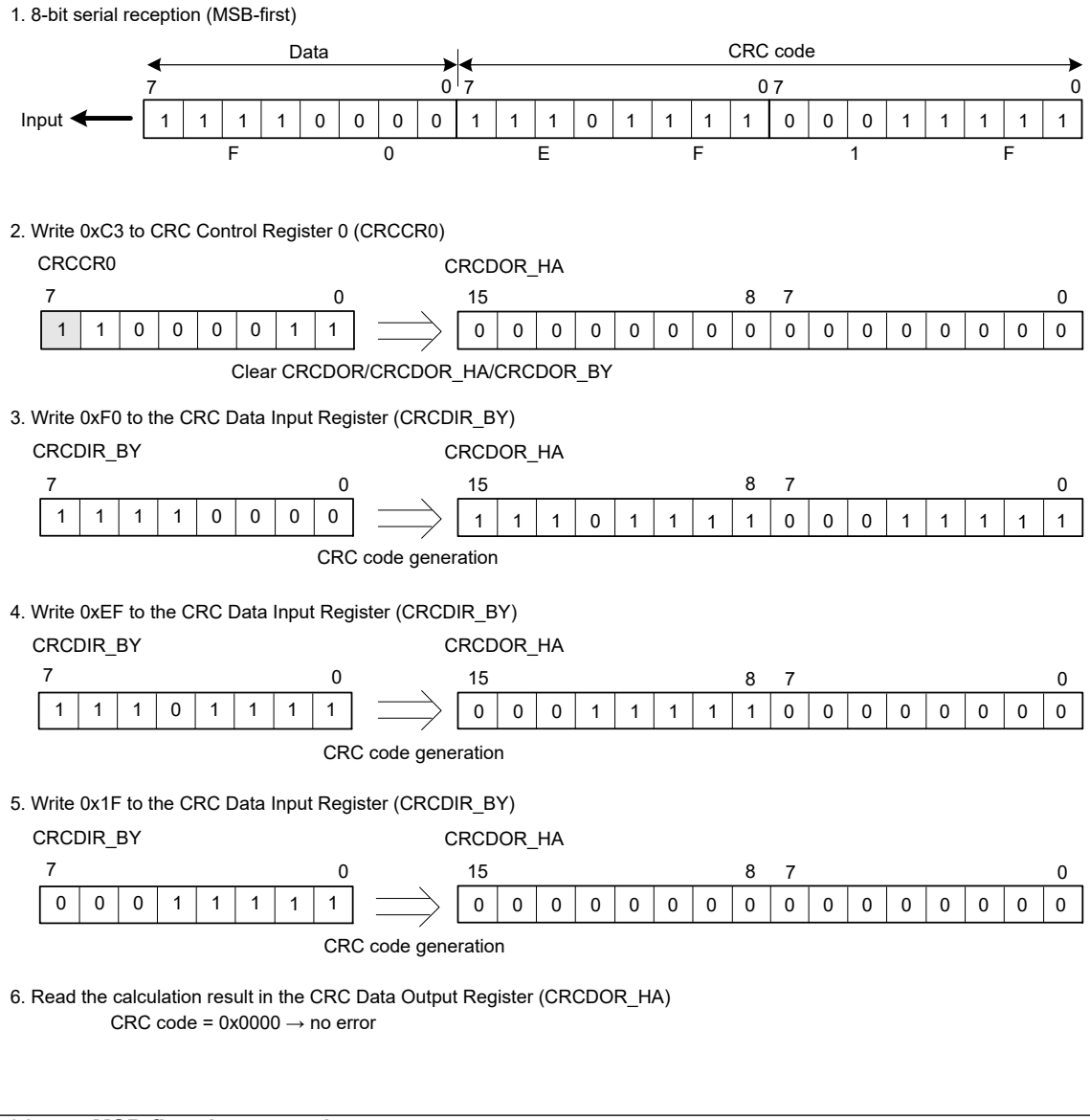


Figure 31.5 MSB-first data reception

31.3.2 CRC Snoop Function

The CRC snoop function monitors reads from and writes to a specific register and performs CRC calculation on the monitored data automatically. Because the CRC snoop function recognizes writes to and reads from a specific register address as a trigger to automatically perform CRC calculation, there is no need to write data to the CRCDIR register. All I/O register specified in the section 31.2.5. CRCSAR : Snoop Address Register are subject to the CRC snoop. The CRC snoop is useful in monitoring writes to the SCIn.TDR (n = 0 to 4, 9) register, and reads from the SCIn.RDR (n = 0 to 4, 9) register.

To use this function, write the lower address 14 bits of a specific register to bits CRCSA13 to CRCSA0 in the CRCSAR register, and set CRCSEN bit in the CRCCR1 register to 1. Then, set the CRCSWR bit in the CRCCR1 register to 1 to enable snooping on writes to the target register, or set the CRCSWR bit in the CRCCR1 register to 0 to enable snooping on reads from the target register. It is possible that access to a target I/O register may be executed before the CRCSWR bit write completed. In this case, the data is not stored in the CRCDIR register. To avoid this issue, before accessing I/O register, read back the CRCSWR bit that was written to confirm that the write completed.

When both the CRCSEN and CRCSWR bits are set to 1, and data is written to a target register in a bus master module such as the CPU, DMAC, and DTC, the CRC calculator stores the data in the CRCDIR register and performs CRC calculation. Similarly, when the CRCSEN bit is set to 1, CRCSWR bit to 0, and data is read from a target register in a bus master

module such as the CPU, DMAC, and DTC, the CRC calculator stores the data in the CRCDIR register and performs CRC calculations.

When the CRC code is generated by using CRC-8, CRC-16, and CRC-CCITT generating polynomial, the target register is accessed in 1 byte (8 bits). Similarly, when the CRC code is generated by using CRC-32 and CRC-32C generating polynomial, the target register is accessed in words (32 bits).

When CPU is halted, CRC snoop operation is invalid.

When CRC is marked as secure by PSARC.PSARC1 bit, the CRC snoop function is available for secure access to the specified I/O registers. When CRC is marked as non-secure by PSARC.PSARC1 bit, the CRC snoop function is available for non-secure access to the specified I/O registers.

31.4 Usage Notes

31.4.1 Settings for the Module-Stop State

The Module Stop Control Register C (MSTPCRC) can enable or disable CRC calculator operation. The CRC calculator is initially stopped after a reset. Releasing the module-stop state enables access to the registers. For details, see section 10, Low Power Modes.

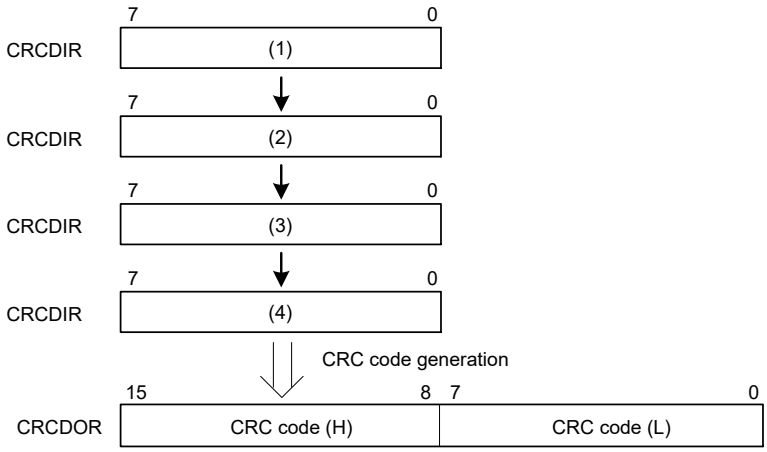
31.4.2 Note on Transmission

The transmission sequence for the CRC code differs based on whether the transmission is LSB-first or MSB-first. Figure 31.6 shows an LSB-first and MSB-first data transmission.

When transmitting 32-bit data (for operation executed on 8 bits in parallel)

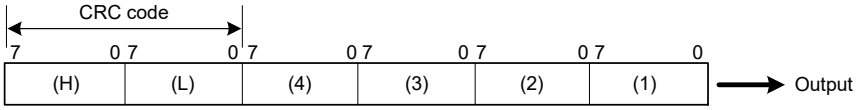
1. CRC code

After specifying the method for generation calculation, write data to CRCDIR in order of (1), (2), (3), and (4).



2. Transmit data

(i) When transmission is LSB-first



(ii) When transmission is MSB-first

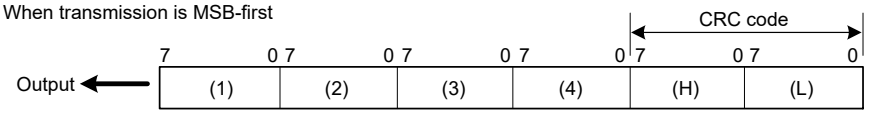


Figure 31.6 LSB-first and MSB-first data transmission

32. Trigonometric Function Unit (TFU)

32.1 Overview

An trigonometric Function Unit (TFU) handles the high-speed calculation for sinf, cosf, atan2f, and hypotf functions.

The ICLK is used as the operating clock for the TFU.

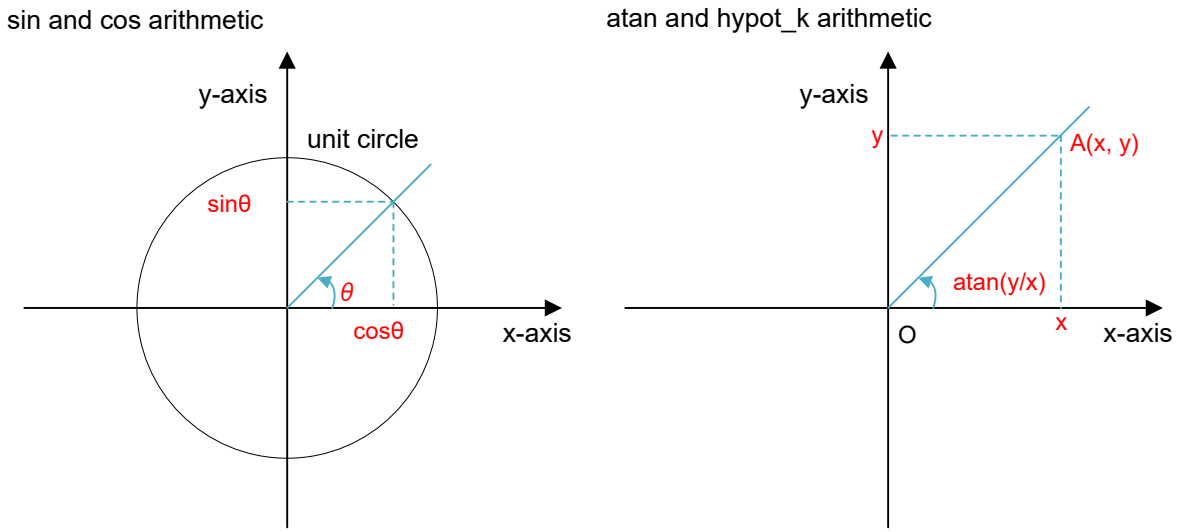
Table 32.1 lists the specifications of the TFU.

Table 32.1 TFU specifications

Item	Description				
Arithmetic Processing	Calculation of sine, cosine, arctangent, and hypot_k ($\sqrt{x^2+y^2}/k$) ● A sine and cosine can be simultaneously calculated. ● An arctangent and hypot_k can be simultaneously calculated.				
Range and Unit of Values	**Arithmetic Processing**	**I/O**		**Range**	**Unit**
	Calculating sine	Input	Angle θ	$-\text{float\_max} \leq \theta \leq \text{float\_max}$ *1	radian
		Output	$\sin\theta$	$-1.0 \leq \sin\theta \leq 1.0$	—
	Calculating cosine	Input	Angle θ	$-\text{float\_max} \leq \theta \leq \text{float\_max}$ *1	radian
		Output	$\cos\theta$	$-1.0 \leq \cos\theta \leq 1.0$	—
	Calculating arctangent	Input	x and y coordinates	$-\text{float\_max} \leq x \leq \text{float\_max}$ *1 $-\text{float\_max} \leq y \leq \text{float\_max}$ *1	—
		Output	$\text{atan}(y/x)$	$-\pi \leq \text{atan}(y/x) \leq \pi$	radian
	Calculating hypot_k	Input	x and y coordinates	$-\text{float\_max} \leq x \leq \text{float\_max}$ *1 $-\text{float\_max} \leq y \leq \text{float\_max}$ *1	—
		Output	$\sqrt{x^2+y^2}/\text{k}$	$0 \leq \sqrt{x^2+y^2}/\text{k} \leq \infty$	—
Data Type for Processing	Single-precision floating-point				
Number of cycles for calculation	Sine: 14 Cosine: 14 Arctangent: 14 hypot_k: 14				

Note: k is a constant. See section 32.3.1. Arithmetic Processing.

Note 1. float_max is the maximum value that can be expressed as single-precision floating-point: $(2 - 2^{-23}) \times 2^{127}$.



Note: sin and cos arithmetic:
Operation to find sinθ and cosθ, which are the triangular ratios of the input angle values θ.
atan arithmetic:
Operation to find the angle value between the vector OA (O is the origin and A is the point indicated by the input coordinates x and y) and the positive part of the x-axis.
hypot_k arithmetic:
Operation to find a value proportional to the distance from the point A indicated by the input coordinates x and y to the origin O.

Note: By multiplying the TFU output result hypot_k by the constant k,
the distance between OA ($\sqrt{x^2 + y^2}$) can be obtained. For details about k, see section 32.3.1. Arithmetic Processing.

Figure 32.1 Explanation for operations

32.1.1 Precautions on Use of the Trigonometric Function Unit

This section describes precautions on use of the trigonometric function unit.

32.1.1.1 General Precautions

If another operation is started after the operation is started and before the result is read, the result of the preceding operation will be discarded.

32.2 Register Descriptions

32.2.1 TRGSTS : Trigonometric Status Register

Base address: TFU = 0x4002_1000

Offset address: 0x08

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

Bit	Symbol	Function	R/W
0	BSYF	Calculation in progress flag 0: No calculating 1: Calculating	R

Bit	Symbol	Function	R/W
1	ERRF	Input error flag 0: No input error occurred 1: Input error occurred	R
7:2	—	These bits are read as 0. The write value should be 0.	R

Note: Writing to this register is invalid.

BSYF bit (Calculation in progress flag)

The BSYF bit specifies the calculation in progress flag.

[Setting condition]

- When calculation started

[Clearing condition]

- When calculation completed

ERRF bit (Input error flag)

The ERRF bit specifies the input error flag.

[Setting condition]

- When input error occurred. For details on input error, see Table 32.6.

[Clearing condition]

- When the next calculation started

32.2.2 SCDT0 : Sine Cosine Data Register 0

Base address: TFU = 0x4002_1000

Offset address: 0x10

Bit position:	31																												0
Bit field:	SCDT0[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	SCDT0[31:0]	Sine Cosine Data Register 0 (single-precision floating-point)	R/W

For the sincos operation, the SCDT0 register is dedicated for the output value cosθ of the trigonometric function unit. See Table 32.2 for detail. Writing to this register during operation is prohibited.

Writing to this register is prohibited even during the atanhypot_k operation.

If read access is made to this register during the calculation, the result is read after the calculation is completed. At this time, the bus access is forced to wait until the operation is completed.

See Table 32.2 for how to use this register.

32.2.3 SCDT1 : Sine Cosine Data Register 1

Base address: TFU = 0x4002_1000

Offset address: 0x14

Bit position:	31																												0
Bit field:	SCDT1[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	SCDT1[31:0]	Sine Cosine Data Register 1 (single-precision floating-point)	R/W

For the sincos operation, the SCDT1 register is shared for the input angle value θ and the output value sinθ of the trigonometric function unit. See Table 32.2 for detail. Writing to this register starts the sincos operation. Writing to this register during operation is prohibited.

Writing to this register is prohibited even during the atanhypot_k operation.

If read access is made to this register during the calculation, the result is read after the calculation is completed. At this time, the bus access is forced to wait until the operation is completed.

See Table 32.2 for how to use this register.

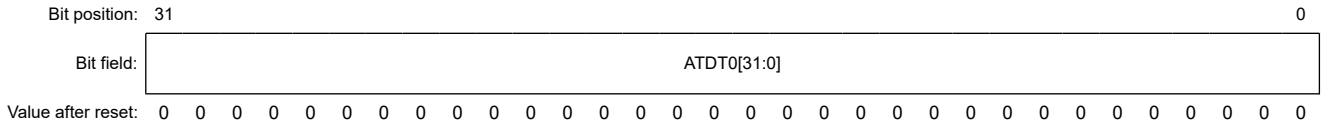
Table 32.2 Input/Output value of SCDT0 and SCDT1

Register	Input value	Output value
SCDT0	—	cosθ
SCDT1	Angle θ	sinθ

32.2.4 ATDT0 : Arctangent Data Register 0

Base address: TFU = 0x4002_1000

Offset address: 0x18



Bit	Symbol	Function	R/W
31:0	ATDT0[31:0]	Arctangent Data Register 0 (single-precision floating-point)	R/W

For the atanhypot_k operation, the ATDT0 register is shared for the input coordinates value x and the hypot_k output value of the trigonometric function unit. See Table 32.3 for detail. Writing to this register during operation is prohibited.

Writing to this register is prohibited even during the sincos operation.

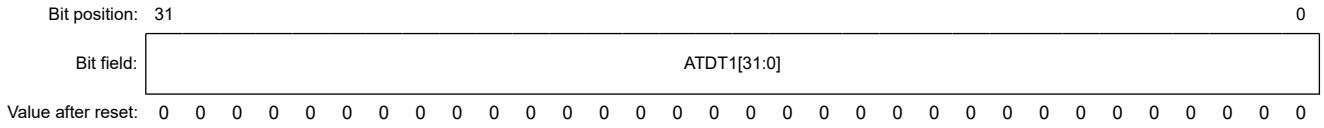
If read access is made to this register during the calculation, the result is read after the calculation is completed. At this time, the bus access is forced to wait until the operation is completed.

See Table 32.3 for how to use this register.

32.2.5 ATDT1 : Arctangent Data Register 1

Base address: TFU = 0x4002_1000

Offset address: 0x1C



Bit	Symbol	Function	R/W
31:0	ATDT1[31:0]	Arctangent Data Register 1 (single-precision floating-point)	R/W

For the atanhypot_k operation, the ATDT1 register is shared for the input coordinates value y and the atan(y/x) output value of the trigonometric function unit. See Table 32.3 for detail. Writing to this register starts the atanhypot_k operation. Writing to this register during operation is prohibited.

Writing to this register is prohibited even during the sincos operation.

If read access is made to this register during the calculation, the result is read after the calculation is completed. At this time, the bus access is forced to wait until the operation is completed.

See Table 32.3 for how to use this register.

Table 32.3 Input/Output value of ATDT0 and ATDT1

Register	Input value	Output value
ATDT0	Input coordinates x	hypot_k
ATDT1	Input coordinates y	atan(y/x)

32.3 Operation

32.3.1 Arithmetic Processing

The trigonometric function unit has two arithmetic operations, the sincos operation and the atanhypot_k operation. See Table 32.4 for detail.

Table 32.4 Arithmetic processing

Operation	Input value	Output value
sincos	Angle value θ	cosθ and sinθ
atanhypot_k	Coordinates x and y	atan(y/x) and hypot_k

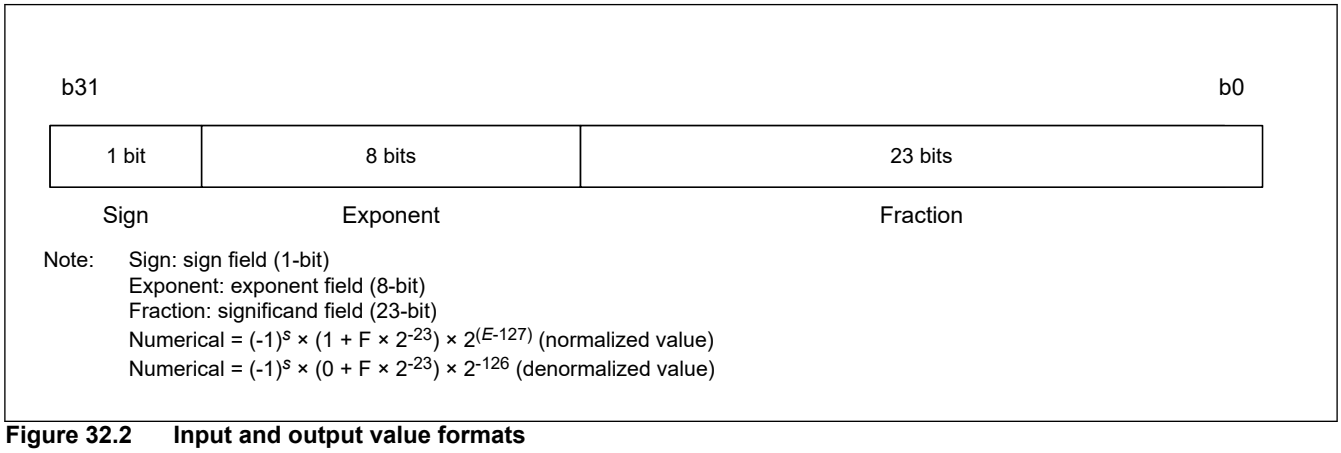
The value of scaling factor k is:

$$k = \prod_{i=0}^{\infty} \frac{1}{\sqrt{1 + 2^{-2i}}} \approx 0.607252935008881256169 4$$

32.3.2 Input and Output Value Formats

The input/output values of TFU support only single-precision floating-point as shown in Table 32.5.

- Floating-point
  Support single-precision floating-point specified by the IEEE754 standard.
  - Single-precision floating-point



Note: Sign: sign field (1-bit)
Exponent: exponent field (8-bit)
Fraction: significand field (23-bit)
Numerical = $(-1)^s \times (1 + F \times 2^{-23}) \times 2^{(E-127)}$ (normalized value)
Numerical = $(-1)^s \times (0 + F \times 2^{-23}) \times 2^{-126}$ (denormalized value)

Figure 32.2 Input and output value formats

Table 32.5 Support single-precision floating-point (1 of 2)

S	E	F	Numerical
Any value	0 < E < 255	Any value	$(-1)^s \times 1.F \times 2^{(E-127)}$ (normalized value — Normal Numbers)
Any value	E = 0	F > 0	$(-1)^s \times 0.F \times 2^{-126}$ (denormalized value — Subnormal Numbers)
S = 0	E = 0	F = 0	$(-1)^0 \times 0.0$ (positive zero — +0)

Table 32.5 Support single-precision floating-point (2 of 2)

S	E	F	Numerical
S = 1	E = 0	F = 0	$(-1)^{-1} \times 0.0$ (negative zero — -0)
S = 0	E = 255	F = 0	(positive infinity — +∞)
S = 1	E = 255	F = 0	(negative infinity — -∞)
Any value	E = 255	$2^{22} > F > 0$	(non-number — SNaN: Signaling Not a Number)
Any value	E = 255	$F \geq 2^{22}$	(non-number — QNaN: Quiet Not a Number)

32.3.3 Relationship Between Input and Output Values for Sincos Operation

When the input values in the sincos operation are ±0, ±∞, SNaN (Signaling Not a Number), and QNaN (Quiet Not a Number), a fixed value is output as shown in Table 32.6.

Table 32.6 Relationship between special input value and its output value (for sincos operation)

Input (θ)	Output (cos)	Output (sin)
-∞	QNaN	QNaN
-0	+1	-0
+0	+1	+0
+∞	QNaN	QNaN
SNaN/QNaN	QNaN	QNaN

Note: The output value of QNaN is 0xFFC0_0000.

32.3.4 Relationship Between Input and Output Values for Atan Operation

When either the input values x or y in the atan operation are ±0, ±∞, SNaN (Signaling Not a Number), and QNaN (Quiet Not a Number), a fixed value is output as shown in Table 32.7.

If both input values are ±0, it is determined as an input error.

Table 32.7 Relationship between special input value and its output value (for atan operation)

		x						
		-∞	Negative value	-0	+0	Positive value	+∞	SNaN/QNaN
y	-∞	QNaN	QNaN	QNaN	QNaN	QNaN	QNaN	QNaN
	Negative value	QNaN	—	-π/2	-π/2	—	QNaN	QNaN
	-0	QNaN	-π	QNaN[*1]	QNaN[*1]	-0	QNaN	QNaN
	+0	QNaN	+π	QNaN[*1]	QNaN[*1]	+0	QNaN	QNaN
	Positive value	QNaN	—	+π/2	+π/2	—	QNaN	QNaN
	+∞	QNaN	QNaN	QNaN	QNaN	QNaN	QNaN	QNaN
	SNaN/QNaN	QNaN	QNaN	QNaN	QNaN	QNaN	QNaN	QNaN

Note 1. An input error occurs for the special input value, and the input error flag (TRGSTS.ERRF) is set.

32.3.5 Relationship Between Input and Output Values for hypot_k Operation

When either the input values x or y in the hypot_k operation are ±0, ±∞, SNaN (Signaling Not a Number), and QNaN (Quiet Not a Number), or when both x and y are ± 0, a fixed value is output as shown in Table 32.8.

If both input values are ±0, it is determined as an input error.

Table 32.8 Relationship between special input value and its output value (for hypot_k operation)

		x						
		-∞	**Negative value**	**-0**	**+0**	**Positive value**	**+∞**	**SNaN/QNaN**
y	**-∞**	QNaN	QNaN	QNaN	QNaN	QNaN	QNaN	QNaN
	Negative value	QNaN	—	—	—	—	QNaN	QNaN
	-0	QNaN	—	+0*1	+0*1	—	QNaN	QNaN
	+0	QNaN	—	+0*1	+0*1	—	QNaN	QNaN
	Positive value	QNaN	—	—	—	—	QNaN	QNaN
	+∞	QNaN	QNaN	QNaN	QNaN	QNaN	QNaN	QNaN
	SNaN/QNaN	QNaN	QNaN	QNaN	QNaN	QNaN	QNaN	QNaN

Note 1. An input error occurs for the special input value, and the input error flag (TRGSTS.ERRF) is set.

32.3.6 Procedure for Trigonometric Function Operation

Figure 32.3 shows the procedure for sincos operation. There are two procedures. The advantages and disadvantages of each are shown in Table 32.9.

Figure 32.4 shows the procedure for atanhypot_k operation. There are two procedures. The advantages and disadvantages of each are shown in Table 32.9.

Table 32.9 Advantages and disadvantages of sincos and atanhypot_k operations

Method	Advantages	Disadvantages
Procedure 1	Not occupy the bus	Operation end determination is required by checking TRGSTS.BSY
Procedure 2	Operation end determination is not required by checking TRGSTS.BSY (reduced number of execution cycles)	Waiting for the number of execution cycles is required to read the result of the operation

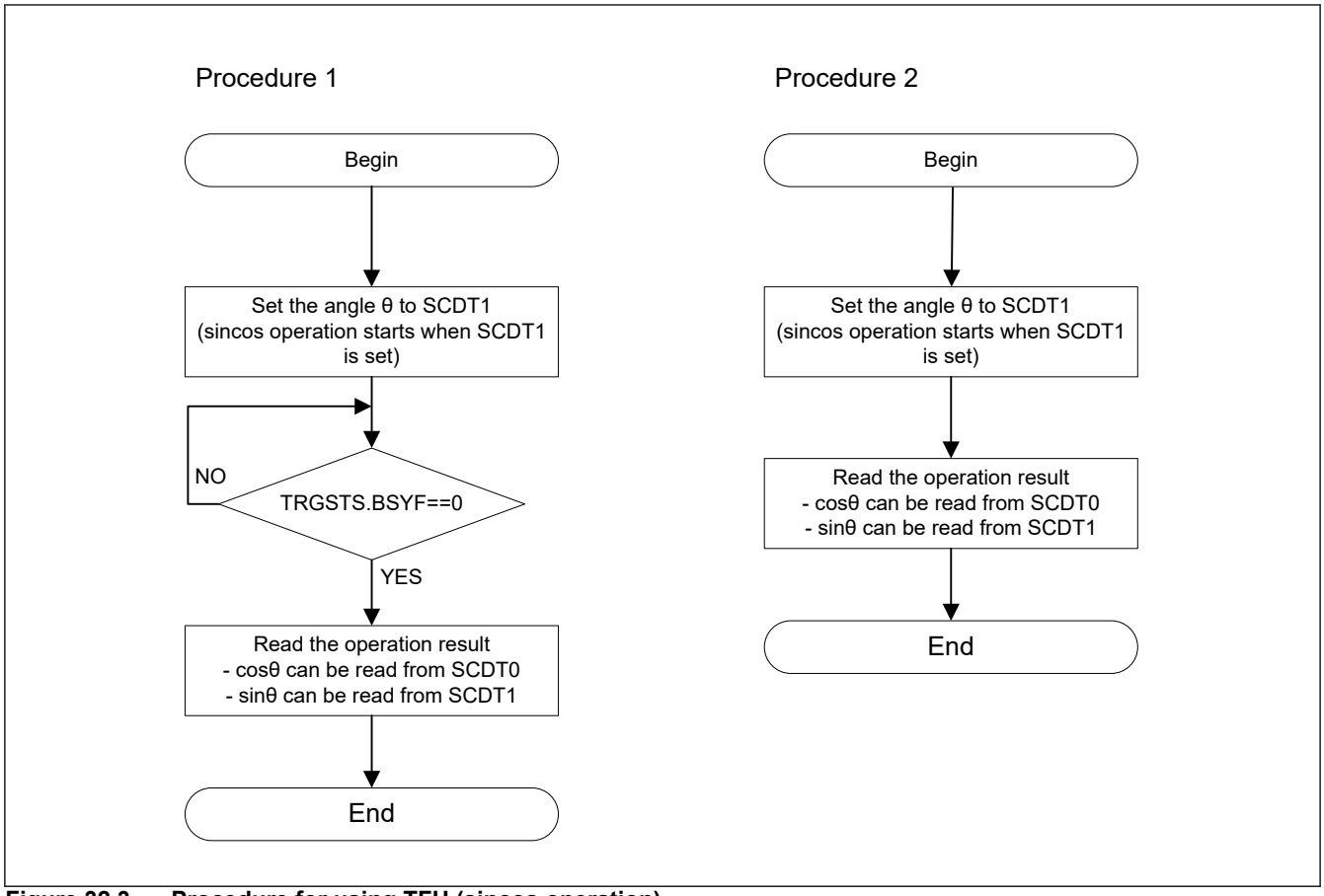


Figure 32.3 Procedure for using TFU (sincos operation)

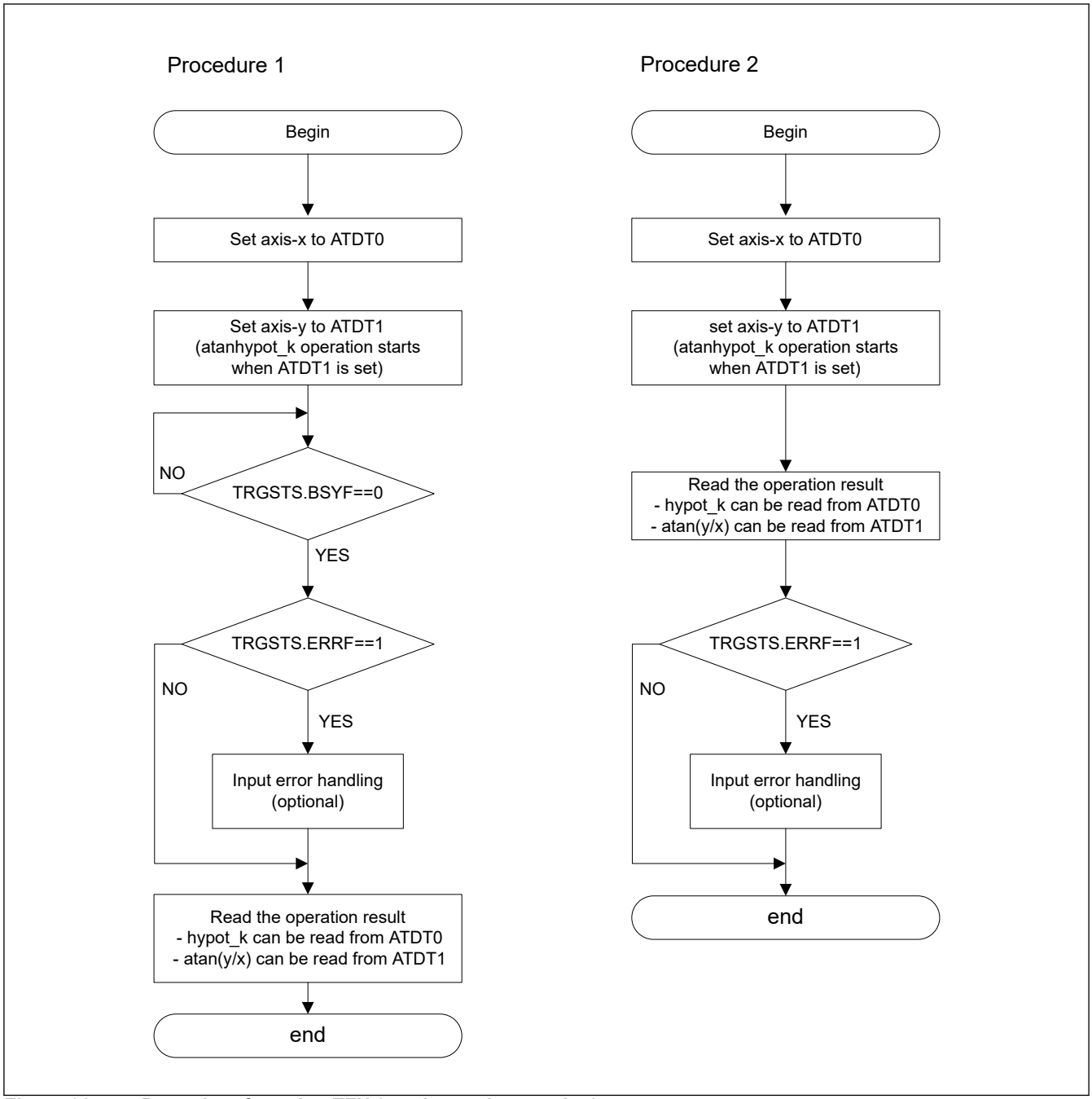


Figure 32.4 Procedure for using TFU (atanhypot_k operation)

33. IIR Filter Accelerator (IIRFA)

33.1 Overview

This IIR filter accelerator performs cascade connected Direct form II transposed biquad IIR (Infinite Impulse Response) filter operation as shown in Figure 33.1.

One biquad IIR filter operation is called a stage. Cascade connected stages are called channels.

The following settings are available:

- Any number of stages up to 32 stages can be connected.
- The stages to be cascade connected can be selected for each channel.
- The coefficients (a_1, a_2, b_0, b_1, b_2) and delay data (D_0, D_1) can be set for each stage.

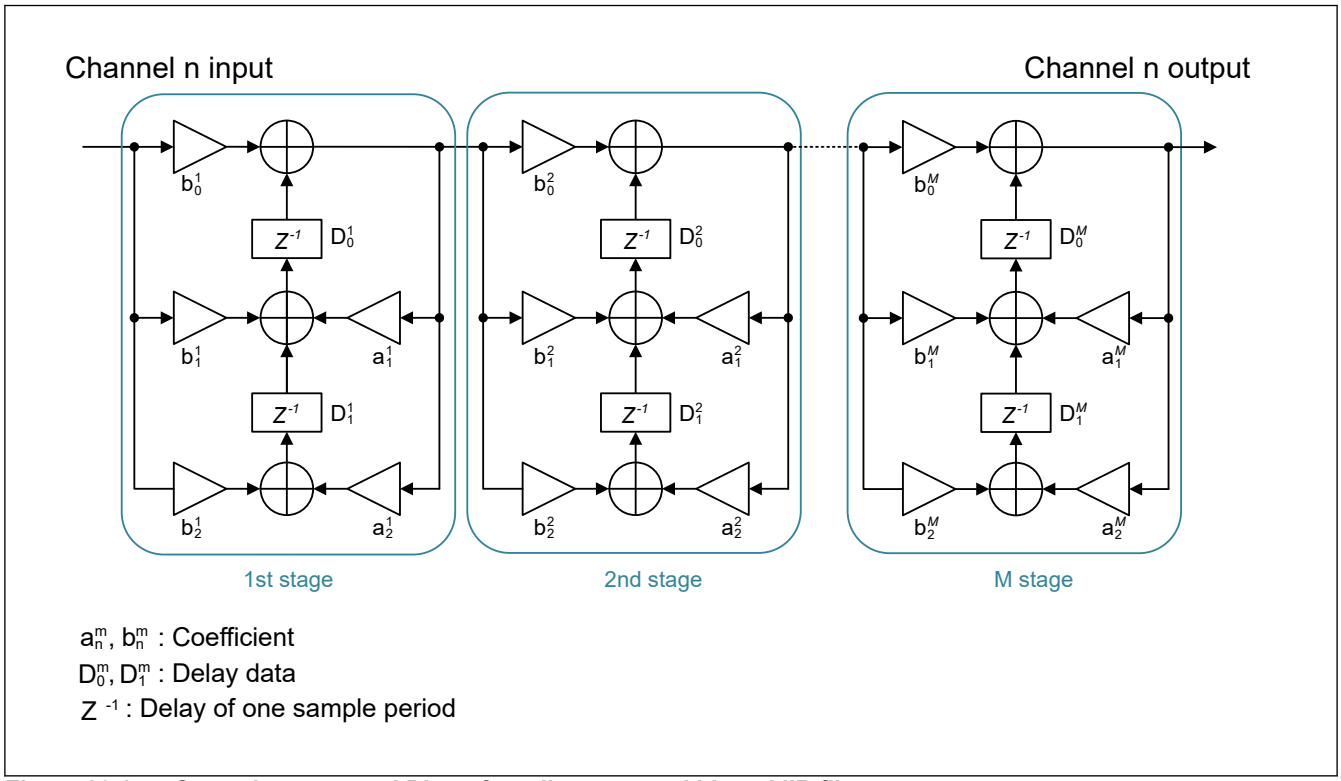


Figure 33.1 Cascade connected Direct form II transposed biquad IIR filter

Table 33.1 shows the specification of the IIR filter accelerator.

Table 33.1 IIR filter accelerator specification (1 of 2)

Parameter	Description
Operation processing	biquad IIR filter
Data type	Single precision floating-point defined in the IEEE754 standard (Negative denormalized numbers are treated as −0, positive denormalized numbers as +0, and NaN (Not a number) as ∞.)
Rounding mode	Round to nearest, round toward zero.
Maximum number of stages	32
Operation circuit	Multiple channels can be processed by time-slicing and one channel can be operated at the same time.
Number of I/O channels	16 channels
Number of cascade connected stages	1 to 32 stages can be connected per channel.
Number of operation cycles	2 cycles per stage until completion of output preparation

Table 33.1 IIR filter accelerator specification (2 of 2)

Parameter	Description
Interrupt	• The following interrupt can be generated – Output data preparation completion interrupt – Process completion interrupt – Operation error interrupt • ECC error interrupts can be generated.
Coefficient/delay data storage area	The coefficient and delay data are retained in local RAM and error correction and detection can be performed by ECC. (with 1-bit error correction and 2-bit error detection)

33.2 Register Descriptions

33.2.1 Register List

Table 33.2 lists the I/O registers.

The meaning of the R/W column in the subsequent register descriptions is as follows:

R/W: Indicates that the read and write access operations are possible.

R: Indicates that only the read access operation is possible. The write access operation is ignored.

W: Indicates that the write access operation is possible. Unless otherwise specified, the read value is the value after reset.

Table 33.2 I/O register list (base address: IIRFA = 0x4002_0000) (1 of 2)

Register name	Symbol	Access width	Value after reset	Offset address	R/W
Channel processing status register	IIRCPRCS	32	0x00000000	0x000	R
Channel processing completion flag register	IIRCPRCFF	32	0x00000000	0x004	R
Output data preparation completion flag register	IIRORDYF	32	0x00000000	0x008	R
Operation error flag register	IIRCERRF	32	0x00000000	0x00C	R
Operation control register	IIROPCNT	32	0x00000000	0x010	R/W
ECC control register	IIRECCCNT	32	0x00000000	0x020	R/W
ECC interrupt enable register	IIRECCINT	32	0x00000000	0x028	R/W
ECC error flag register	IIRECCEF	32	0x00000000	0x030	R
ECC error flag clear register	IIRECCEFCLR	32	0x00000000	0x034	W
ECC 1-bit error address register	IIRESEADR	32	0x00000000	0x038	R
ECC 2-bit error address register	IIREDEADR	32	0x00000000	0x03C	R
Channel n input register (n = 0 to 15)	IIRCHnINP	32	Undefined	0x100 + 0x10 × n	W
Channel n output register (n = 0 to 15)	IIRCHnOUT	32	0x00000000	0x104 + 0x10 × n	R
Channel n control register (n = 0 to 15)	IIRCHnCNT	32	0x00000000	0x108 + 0x10 × n	R/W
Channel n interrupt enable register (n = 0 to 15)	IIRCHnINT	8	0x00	0x10C + 0x10 × n	R/W
Channel n status register (n = 0 to 15)	IIRCHnSTS	8	0x00	0x10D + 0x10 × n	R
Channel n flag clear register (n = 0 to 15)	IIRCHnFCLR	8	0x00	0x10E + 0x10 × n	W
Stage m coefficient b0 register (m = 0 to 31)	IIRSTGmB0	32	Undefined	0x400 + 0x20 × m	R/W
Stage m coefficient b1 register (m = 0 to 31)	IIRSTGmB1	32	Undefined	0x404 + 0x20 × m	R/W
Stage m coefficient b2 register (m = 0 to 31)	IIRSTGmB2	32	Undefined	0x408 + 0x20 × m	R/W
Stage m coefficient a1 register (m = 0 to 31)	IIRSTGmA1	32	Undefined	0x40C + 0x20 × m	R/W
Stage m coefficient a2 register (m = 0 to 31)	IIRSTGmA2	32	Undefined	0x410 + 0x20 × m	R/W
Stage m delay data D0 register (m = 0 to 31)	IIRSTGmD0	32	Undefined	0x414 + 0x20 × m	R/W

Table 33.2 I/O register list (base address: IIRFA = 0x4002_0000) (2 of 2)

Register name	Symbol	Access width	Value after reset	Offset address	R/W
Stage m delay data D1 register (m = 0 to 31)	IIRSTGmD1	32	Undefined	0x418 + 0x20 × m	R/W

Note: The register cannot be accessed with an access width other than that described.

33.2.1.1 IIRCPRCS : Channel Processing Status Register

Base address: IIRFA = 0x4002_0000

Offset address: 0x000

Bit position:	31																15													0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CPRCS[15:0]														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	CPRCS[15:0]	Channel processing status bit Bit 0 corresponds to channel 0, bit 1 to channel 1, …, and bit 15 to channel 15. 0: The channel processing of the corresponding channel is not being performed. 1: The channel processing of the corresponding channel is being performed.	R
31:16	—	These bits are read as 0.	R

When the CPRCS[n] bit is read, IIRCHnSTS.CPRCS flag value is read (n = 0 to 15).

33.2.1.2 IIRCPRCFF : Channel Processing Completion Flag Register

Base address: IIRFA = 0x4002_0000

Offset address: 0x004

Bit position:	31																15													0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CPRCFF[15:0]														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	CPRCFF[15:0]	Channel processing completion flag Bit 0 corresponds to channel 0, bit 1 to channel 1, …, and bit 15 to channel 15. 0: The channel processing of the corresponding channel is not completed. 1: The channel processing of the corresponding channel is completed.	R
31:16	—	These bits are read as 0.	R

When the CPRCFF[n] bit is read, IIRCHnSTS.CPRCFF flag value is read (n = 0 to 15).

33.2.1.3 IIRORDYF : Output Data Preparation Completion Flag Register

Base address: IIRFA = 0x4002_0000

Offset address: 0x008

Bit position:	31																15													0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ORDYF[15:0]														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	ORDYF[15:0]	Output data preparation completion flag Bit 0 corresponds to channel 0, bit 1 to channel 1, …, and bit 15 to channel 15. 0: The output data preparation of the corresponding channel is not completed. 1: The output data preparation of the corresponding channel is completed.	R
31:16	—	These bits are read as 0.	R

When the ORDYF[n] bit is read, IIRCHnSTS.ORDYF flag value is read (n = 0 to 15).

33.2.1.4 IIRCERRF : Operation Error Flag Register

Base address: IIRFA = 0x4002_0000

Offset address: 0x00C

Bit position:	31															15															0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CERRF[15:0]														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	CERRF[15:0]	Operation error flag Bit 0 corresponds to channel 0, bit 1 to channel 1, …, and bit 15 to channel 15. 0: No operation error has occurred in the corresponding channel. 1: An operation error has occurred in the corresponding channel.	R
31:16	—	These bits are read as 0.	R

When the CERRF[n] bit is read, IIRCHnSTS.CERRF flag value is read (n = 0 to 15).

33.2.1.5 IIROPCNT : Operation Control Register

Base address: IIRFA = 0x4002_0000

Offset address: 0x010

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

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

Bit	Symbol	Function	R/W
2:0	RND[2:0]	Setting for the rounding mode for addition and multiplication 0 0 0: Round to nearest 0 0 1: Round toward zero Others: Setting prohibited.	R/W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

The IIROPCNT register switches the rounding mode for addition and multiplication results during channel processing.

Set this register before starting the channel processing. When this register is rewritten during channel processing, the change does not affect the channel processing that is being executed. The change affects from the next channel processing.

33.2.1.6 IIRECCCNT : ECC Control Register

Base address: IIRFA = 0x4002_0000

Offset address: 0x020

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

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

Bit	Symbol	Function	R/W
0	ECCMD	ECC setting bit 0: The ECC error detection/correction function is disabled. 1: The ECC error detection/correction function is enabled.	R/W
1	ECCWBDIS	ECC-corrected data write-back disable bit 0: The error-corrected data write-back is enabled. 1: The error-corrected data write-back is disabled.	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

This register sets the ECC error detection/correction function.

Set this register before starting the channel processing. If the value is changed during the channel processing, operation is not guaranteed.

ECCMD bit (ECC setting bit)

The ECCMD bit enables or disables the error detection/correction function.

If the ECC error detection/correction function is enabled, the ECC code is updated when data is written to the coefficient/delay data storage area.

If the ECC error detection/correction function is disabled, the ECC code is not updated when data is written to the coefficient/delay data storage area.

ECCWBDIS bit (ECC-corrected data write-back disable bit)

The ECCWBDIS bit controls the ECC 1-bit error-corrected data write-back when the ECC error detection/correction function is enabled.

33.2.1.7 IIRECCINT : ECC Interrupt Enable Register

Base address: IIRFA = 0x4002_0000

Offset address: 0x028

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

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

Bit	Symbol	Function	R/W
0	ESEIE	ECC 1-bit error interrupt enable bit 0: The generation of ECC 1-bit error interrupt requests is disabled. 1: The generation of ECC 1-bit error interrupt requests is enabled.	R/W
1	EDEIE	ECC 2-bit error interrupt enable bit 0: The generation of ECC 2-bit error interrupt requests is disabled. 1: The generation of ECC 2-bit error interrupt requests is enabled.	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

ESEIE bit (ECC 1-bit error interrupt enable bit)

The ESEIE bit enables or disables the ECC 1-bit error interrupt requests (IIRFA_ERR) when the IIRECCEF.ESEF flag is 1.

EDEIE bit (ECC 2-bit error interrupt enable bit)

The EDEIE bit enables or disables the ECC 2-bit error interrupt requests (IIRFA_ERR) when the IIRECCEF.EDEF flag is 1.

33.2.1.8 IIRECCEF : ECC Error Flag Register

Base address: IIRFA = 0x4002_0000

Offset address: 0x030

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

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

Bit	Symbol	Function	R/W
0	ESEF	ECC 1-bit error flag 0: No 1-bit ECC error is detected. 1: 1-bit ECC error is detected.	R
1	EDEF	ECC 2-bit error flag 0: No 2-bit ECC error is detected. 1: 2-bit ECC error is detected.	R
31:2	—	These bits are read as 0.	R

ESEF bit (ECC 1-bit error flag)

[Setting condition]

- 1-bit ECC error is detected.

[Clearing condition]

- When 1 is written to the IIRECCEFCLR.ESEFCLR bit.

Note: When the setting condition and clearing condition occur at the same time, the clearing condition takes precedence.

EDEF bit (ECC 2-bit error flag)

[Setting condition]

- 2-bit ECC error is detected.

[Clearing condition]

- When 1 is written to the IIRECCEFCLR.EDEFCLR bit.

Note: When the setting condition and clearing condition occur at the same time, the clearing condition takes precedence.

33.2.1.9 IIRECCEFCLR : ECC Error Flag Clear Register

Base address: IIRFA = 0x4002_0000

Offset address: 0x034

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

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

Bit	Symbol	Function	R/W
0	ESEFCLR	ECC 1-bit error flag clear bit 0: No effect 1: Clears the ESEF flag of the IIRECCEF register.	W
1	EDEFCLR	ECC 2-bit error status flag clear bit 0: No effect 1: Clears the EDEF flag of the IIRECCEF register.	W
31:2	—	The write value should be 0.	W

The IIRECCEFCLR register clears the ESEF and EDEF flags of the IIRECCEF register.

33.2.1.10 IIRESEADR : ECC 1-bit Error Address Register

Base address: IIRFA = 0x4002_0000

Offset address: 0x038

Bit position:	31	10	0
Bit field:	—	SEADR[10:0]	
Value after reset:	0	0	0

Bit	Symbol	Function	R/W
10:0	SEADR[10:0]	Error address Retains a part of the address of the coefficient/delay data that has detected an ECC 1-bit error. The address to be retained is from bit 10 to bit 2. The lower 2 bits are always 0.	R
31:11	—	These bits are read as 0.	R

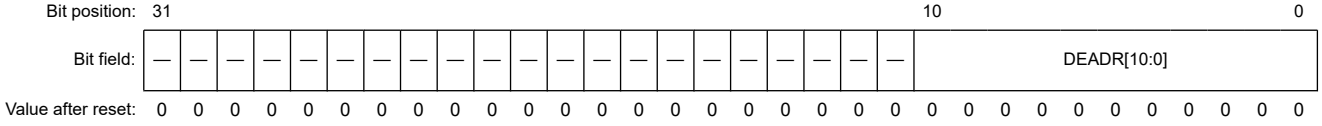
A part of the address of the coefficient/delay data that has detected an ECC 1-bit error is retained. When an ECC 1-bit error is detected in a state where the IIRECCEF.ESEF flag is 0, the SEADR[10:0] bits are updated. When the IIRECCEF.ESEF flag is 1, the SEADR[10:0] bits are not updated.

When the base address (IIRFA) is added to the value of this register, the address of the I/O register corresponding to the coefficient/delay data that has detected an ECC 1-bit error is obtained.

33.2.1.11 IIREDEADR : ECC 2-bit Error Address Register

Base address: IIRFA = 0x4002_0000

Offset address: 0x03C



Bit	Symbol	Function	R/W
10:0	DEADR[10:0]	Error address Retains a part of the address of the coefficient/delay data that has detected an ECC 2-bit error. The address to be retained is from bit 10 to bit 2. The lower 2 bits are always 0.	R
31:11	—	These bits are read as 0.	R

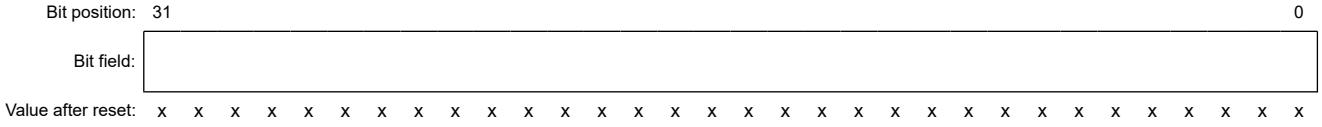
A part of the address of the coefficient/delay data that has detected an ECC 2-bit error is retained. When an ECC 2-bit error is detected in a state where the IIRECCEF.EDEF flag is 0, the DEADR[10:0] bits are updated. When the IIRECCEF.EDEF flag is 1, the DEADR[10:0] bits are not updated.

When the base address (IIRFA) is added to the value of this register, the address of the I/O register corresponding to the coefficient/delay data that has detected an ECC 2-bit error is obtained.

33.2.1.12 IIRCHnINP : Channel n Input Register (n = 0 to 15)

Base address: IIRFA = 0x4002_0000

Offset address: 0x100 + 0x10 × n



Bit	Symbol	Function	R/W
31:0	n/a	Input data (single precision floating-point) setting of channel n The channel processing of channel n is started.	W

Set the input data to this register. When a value is written to this register, the channel processing of channel n is started.

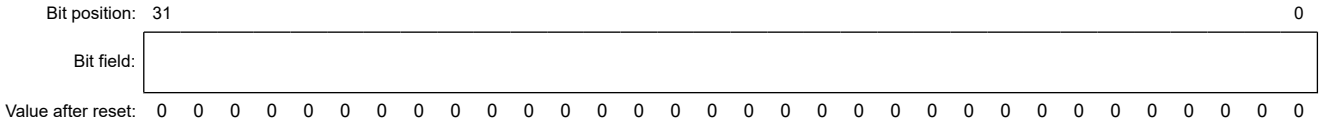
The read value is undefined when a read access to this register is performed.

If a write access to this register is performed during the channel processing of any channel, bus access is forced to wait until the channel processing is completed. A value is written to this register after the channel processing is complete.

33.2.1.13 IIRCHnOUT : Channel n Output Register (n = 0 to 15)

Base address: IIRFA = 0x4002_0000

Offset address: 0x104 + 0x10 × n



Bit	Symbol	Function	R/W
31:0	n/a	Output data (single precision floating-point) of channel n	R

This register reads the output data of channel n.

If a read access is performed to this register before the output data preparation of channel n is completed, bus access is forced to wait until the output data preparation is completed. The value is read after the output data preparation is complete.

33.2.1.14 IIRCHnCNT : Channel n Control Register (n = 0 to 15)

Base address: IIRFA = 0x4002_0000

Offset address: 0x108 + 0x10 × n

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

Bit	Symbol	Function	R/W
31:0	STGSEL[31:0]	Stage selection bit The stage to be used for channel n is selected. Bit 0 corresponds to stage 0, bit 1 to stage 1, …, and bit 31 to stage 31. 0: The corresponding stage is not used for channel n. 1: The corresponding stage is used for channel n.	R/W

This register selects the stage to use for channel n.

Set the stage to be used using this register before starting the channel processing. Operation is not guaranteed when the channel processing of channel n is started with the setting that does not use any stage, that is, the value of this register is 0.

When multiple stages are selected, the cascade connected biquad IIR filter operation is performed for the selected stages. In this case, the operation of the stages is performed in ascending order of stage number.

Note:
- Writing 1 to the bit corresponding to the stage used for another channel is ignored. The stage with the same number cannot be used between different channels.
- Whether 1 is written to the bit corresponding to the stage used for another channel can be determined by reading the IIRCHnCNT immediately after writing to the IIRCHnCNT and checking whether the written value matches with the read value. When not matched, this means that 1 is written to the bit corresponding to the stage used for another channel.
- Do not rewrite the corresponding IIRCHnCNT register during the channel processing of channel n. Operation is not guaranteed if the bit is rewritten.

33.2.1.15 IIRCHnINT : Channel n Interrupt Enable Register (n = 0 to 15)

Base address: IIRFA = 0x4002_0000

Offset address: 0x10C + 0x10 × n

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

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
1	CPRCFIE	Channel processing completion interrupt enable bit 0: The generation of channel processing completion interrupt requests is disabled. 1: The generation of channel processing completion interrupt requests is enabled.	R/W
2	ORDYIE	Output data preparation completion interrupt enable bit 0: The generation of output data preparation completion interrupt requests is disabled. 1: The generation of output data preparation completion interrupt requests is enabled.	R/W

Bit	Symbol	Function	R/W
3	CERRIE	Operation error interrupt enable bit 0: The generation of operation error interrupt requests is disabled. 1: The generation of operation error interrupt requests is enabled.	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W

CPRCFIE bit (Channel processing completion interrupt enable bit)

The CPRCFIE bit enables or disables the channel processing completion interrupt requests of channel n when the IRCHnSTS.CPRCFF flag is 1.

ORDYIE bit (Output data preparation completion interrupt enable bit)

The ORDYIE bit enables or disables the output data preparation completion interrupt requests of channel n when the IIRCHnSTS.ORDYF flag is 1.

CERRIE bit (Operation error interrupt enable bit)

The CERRIE bit enables or disables the operation error interrupt requests of channel n when the IIRCHnSTS.CERRF flag is 1.

33.2.1.16 IIRCHnSTS : Channel n Status Register (n = 0 to 15)

Base address: IIRFA = 0x4002_0000

Offset address: 0x10D + 0x10 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	CERRF	ORDYF	CPRCFF	CPRCS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CPRCS	Channel processing status flag 0: The channel processing is not being performed. 1: The channel processing is being performed.	R
1	CPRCFF	Channel processing completion flag 0: The channel processing is not completed. 1: The channel processing is completed.	R
2	ORDYF	Output data preparation completion flag 0: The output data preparation is not completed. 1: The output data preparation is completed.	R
3	CERRF	Operation error flag 0: No operation error has occurred. 1: An operation error has occurred.	R
7:4	—	These bits are read as 0.	R

CPRCS bit (Channel processing status flag)

[Setting condition]

- The channel processing of channel n is started.

[Clearing condition]

- The channel processing of channel n is completed.

Note: When the setting condition and clearing condition occur at the same time, the setting condition takes precedence.

CPRCFF bit (Channel processing completion flag)

[Setting condition]

- The channel processing of channel n is completed.

[Clearing condition]

- When 1 is written to the IIRCHnFCLR.CPRCFFCLR bit or the channel processing of channel n is started.

Note: When the setting condition and clearing condition occur at the same time, the clearing condition takes precedence.

ORDYF bit (Output data preparation completion flag)

[Setting condition]

- The output data preparation of channel n is completed.

[Clearing condition]

- The IIRCHnOUT register is read or the channel processing of channel n is started.

Note: When the setting condition and clearing condition occur at the same time, the clearing condition takes precedence.

CERRF bit (Operation error flag)

[Setting condition]

- An operation error occurs when the output data preparation of channel n is completed.

[Clearing condition]

- When 1 is written to the IIRCHnFCLR.CERRFCLR bit.

Note: When the setting condition and clearing condition occur at the same time, the clearing condition takes precedence.

33.2.1.17 IIRCHnFCLR : Channel n Flag Clear Register (n = 0 to 15)

Base address: IIRFA = 0x4002_0000

Offset address: 0x10E + 0x10 × n

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

Bit	Symbol	Function	R/W
0	—	The write value should be 0.	W
1	CPRCFFCLR	Channel processing completion flag clear bit 0: No effect 1: Clears the CPRCFF flag of the IIRCHnSTS register.	W
2	—	The write value should be 0.	W
3	CERRFCLR	Operation error flag clear bit 0: No effect 1: Clears the CERRF flag of the IIRCHnSTS register.	W
7:4	—	The write value should be 0.	W

This register clears the CERRF and CPRCFF flags of the IIRCHnSTS register.

33.2.1.18 IIRSTGmB0 : Stage m Coefficient b0 Register (m = 0 to 31)

Base address: IIRFA = 0x4002_0000

Offset address: 0x400 + 0x20 × m

Bit position:	31																												0
Bit field:																													
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
31:0	n/a	Coefficient data (single precision floating-point) of stage m	R/W

Each register corresponds to the coefficient data with the same symbol as shown in Figure 33.2.

If an access to this register is performed during the channel processing of any channel, bus access is forced to wait until the channel process is completed. A write access or read access is performed after the channel processing is complete.

When a write access to this register is performed, the write value is saved to the coefficient/delay data storage area. When a read access to this register is performed, the read value is read from the coefficient/delay data storage area.

33.2.1.19 IIRSTGmB1 : Stage m Coefficient b1 Register (m = 0 to 31)

Base address: IIRFA = 0x4002_0000

Offset address: 0x404 + 0x20 × m

Bit position:	31																												0
Bit field:																													
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
31:0	n/a	Coefficient data (single precision floating-point) of stage m	R/W

Each register corresponds to the coefficient data with the same symbol as shown in Figure 33.2.

If an access to this register is performed during the channel processing of any channel, bus access is forced to wait until the channel process is completed. A write access or read access is performed after the channel processing is complete.

When a write access to this register is performed, the write value is saved to the coefficient/delay data storage area. When a read access to this register is performed, the read value is read from the coefficient/delay data storage area.

33.2.1.20 IIRSTGmB2 : Stage m Coefficient b2 Register (m = 0 to 31)

Base address: IIRFA = 0x4002_0000

Offset address: 0x408 + 0x20 × m

Bit position:	31																												0
Bit field:																													
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
31:0	n/a	Coefficient data (single precision floating-point) of stage m	R/W

Each register corresponds to the coefficient data with the same symbol as shown in Figure 33.2.

If an access to this register is performed during the channel processing of any channel, bus access is forced to wait until the channel process is completed. A write access or read access is performed after the channel processing is complete.

When a write access to this register is performed, the write value is saved to the coefficient/delay data storage area. When a read access to this register is performed, the read value is read from the coefficient/delay data storage area.

33.2.1.21 IIRSTGmA1 : Stage m Coefficient a1 Register (m = 0 to 31)

Base address: IIRFA = 0x4002_0000

Offset address: 0x40C + 0x20 × m

Bit position:	31																												0
Bit field:																													
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
31:0	n/a	Coefficient data (single precision floating-point) of stage m	R/W

Each register corresponds to the coefficient data with the same symbol as shown in Figure 33.2.

If an access to this register is performed during the channel processing of any channel, bus access is forced to wait until the channel process is completed. A write access or read access is performed after the channel processing is complete.

When a write access to this register is performed, the write value is saved to the coefficient/delay data storage area. When a read access to this register is performed, the read value is read from the coefficient/delay data storage area.

33.2.1.22 IIRSTGmA2 : Stage m Coefficient a2 Register (m = 0 to 31)

Base address: IIRFA = 0x4002_0000

Offset address: 0x410 + 0x20 × m

Bit position:	31																												0
Bit field:																													
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
31:0	n/a	Coefficient data (single precision floating-point) of stage m	R/W

Each register corresponds to the coefficient data with the same symbol as shown in Figure 33.2.

If an access to this register is performed during the channel processing of any channel, bus access is forced to wait until the channel process is completed. A write access or read access is performed after the channel processing is complete.

When a write access to this register is performed, the write value is saved to the coefficient/delay data storage area. When a read access to this register is performed, the read value is read from the coefficient/delay data storage area.

33.2.1.23 IIRSTGmD0 : Stage m Delay Data D0 Register (m = 0 to 31)

Base address: IIRFA = 0x4002_0000

Offset address: 0x414 + 0x20 × m

Bit position:	31																												0
Bit field:																													
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
31:0	n/a	Delay data (single precision floating-point) of stage m	R/W

Each register corresponds to the delay data with the same symbol as shown in Figure 33.2.

The value of the delay data D0 register is updated when the operation of the stage corresponding to the register is executed.

If an access to this register is performed during the channel processing of any channel, bus access is forced to wait until the channel process is completed. A write access or read access is performed after the channel processing is complete.

When a write access to this register is performed, the write value is saved to the coefficient/delay data storage area. When a read access to this register is performed, the read value is read from the coefficient/delay data storage area.

33.2.1.24 IIRSTGmD1 : Stage m Delay Data D1 Register (m = 0 to 31)

Base address: IIRFA = 0x4002_0000

Offset address: 0x418 + 0x20 × m

Bit position:	31																												0
Bit field:																													
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
31:0	n/a	Delay data (single precision floating-point) of stage m	R/W

Each register corresponds to the delay data with the same symbol as shown in Figure 33.2.

The value of the delay data D1 register is updated when the operation of the stage corresponding to the register is executed.

If an access to this register is performed during the channel processing of any channel, bus access is forced to wait until the channel process is completed. A write access or read access is performed after the channel processing is complete.

When a write access to this register is performed, the write value is saved to the coefficient/delay data storage area. When a read access to this register is performed, the read value is read from the coefficient/delay data storage area.

33.3 Operation

33.3.1 Overview

IIRFA has a cascade connected biquad IIR (Infinite Impulse Response) filter operation function.

The formula for the transfer function of cascade connected M biquad IIR filters is as follows:

$$H(z) = \prod_{m=1}^{M} \frac{b_0^m + b_1^m z^{-1} + b_2^m z^{-2}}{1 - a_1^m z^{-1} - a_2^m z^{-2}}$$

b_0^m , b_1^m , and b_2^m , are feed-forward coefficients, and a a_1^m and a_2^m are feed-back coefficients.

The operation for each biquad IIR filter is defined by the following difference equation:

$$y(n) = b_0 x(n) + b_1 x(n-1) + b_2 x(n-2) + a_1 y(n-1) + a_2 y(n-2)$$

$x(n)$ is input data, $y(n)$ is output data, $x(n-1)$ and $y(n-1)$ are input data and output data of 1 sample period delay, and $x(n-2)$ and $y(n-2)$ are input data and output data of 2 sample period delay.

IIRFA performs the cascade connected Direct form II transposed biquad IIR filter operation shown in Figure 33.2.

One biquad IIR filter operation is called a stage. Any number of stages up to 32 stages that are cascade connected are called channels.

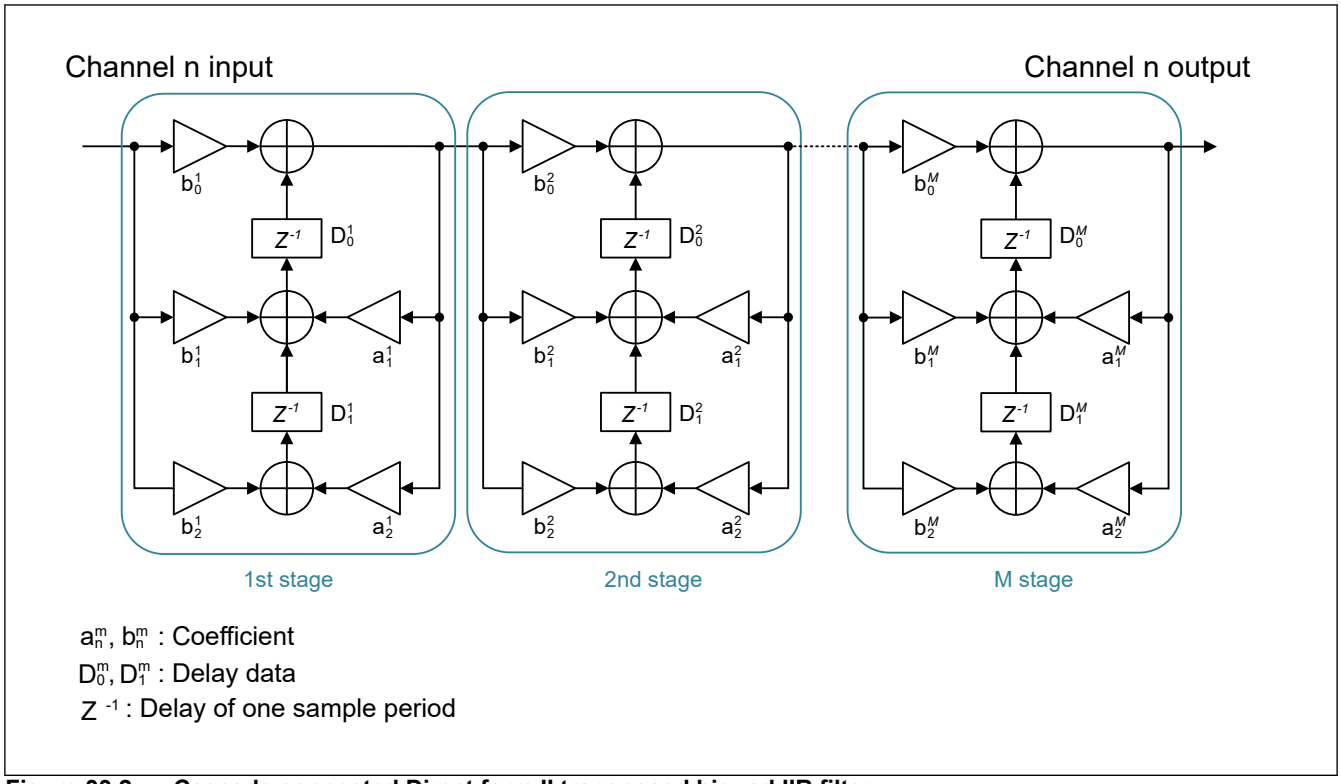


Figure 33.2 Cascade connected Direct form II transposed biquad IIR filter

The stages to be cascade connected can be selected for each channel. In addition, the coefficient (a_1, a_2, b_0, b_1, b_2) and delay data (D_0, D_1) can be set for each stage.

Figure 33.3 shows an example of stage selection for each channel. In this example, channel 0 performs the cascade connected biquad IIR filter operation for stage 0, stage 2, and stage 3. For the operation, see section 33.3.2. Channel Processing Operation.

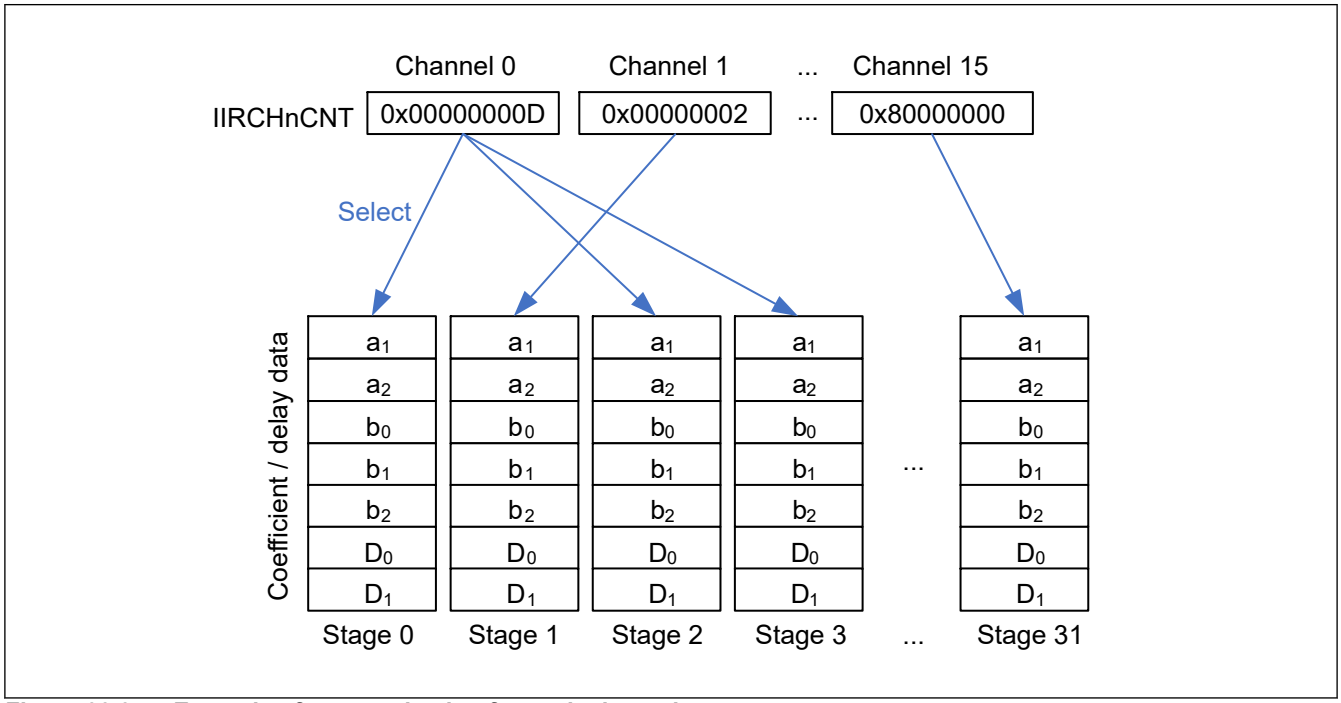


Figure 33.3 Example of stage selection for each channel

The coefficient and delay data for each stage are stored to the coefficient/delay data storage area. The coefficient/delay data storage area has an error detection/correction function by ECC. When the ECC error detection/correction function is

enabled (IIRECCCNT.ECCMD = 1), an error that has occurred in the data in the coefficient/delay data storage area can be detected and corrected. For details on the ECC error, see section 33.3.4. Operation on ECC Error Detection.

The I/O data, stage coefficient/delay data are retained in the single precision floating-point format specified in the IEEE754 standard. For details on the single precision floating-point, see Figure 33.4.

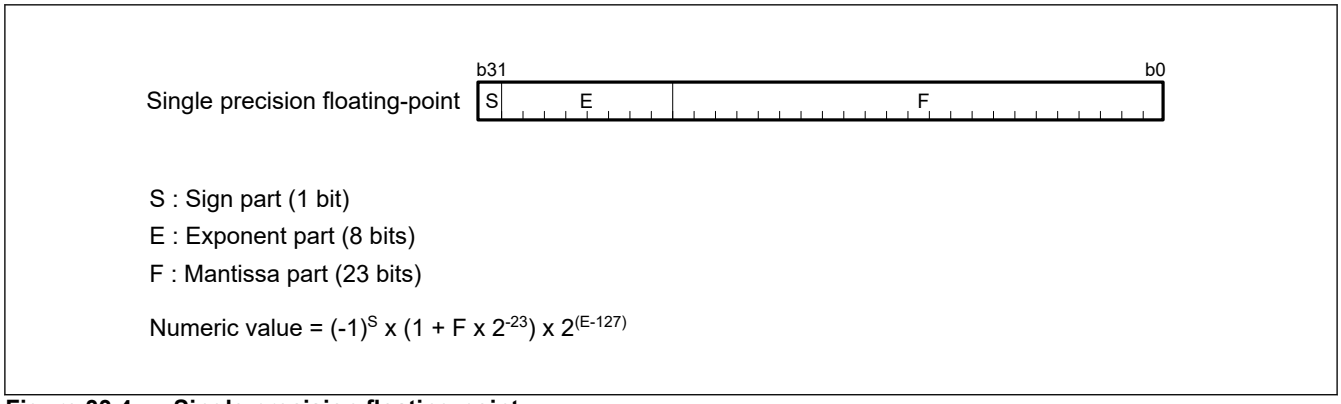


Figure 33.4 Single precision floating-point

The single precision floating-point format supports the following values:

- 0 < E < 255 (normal numbers)
- E = 0 and F = 0 (signed zero)
- E = 0 and F > 0 (denormalized numbers)
- E = 255 and F = 0 (infinity)
- E = 255 and F > 0 (NaN: Not a Number)
  MSB of F is 0. (SNaN: Signaling NaN)
  MSB of F is 1. (QNaN: Quiet NaN)

IIRFA treats the input as +0 if a positive denormalized number is input, −0 if a negative denormalized number is input, and infinity if a NaN (Not a Number) is input.

IIRFA performs addition and multiplication of single precision floating-points multiple times in the cascade connected biquad IIR filter operation. If the result of each addition and multiplication is a positive denormalized number, it is treated as +0. If the result is a negative denormalized number, it is treated as -0. If the result is a NaN (Not a Number), it is treated as infinity. In addition, the rounding mode for each addition and multiplication result can be selected by IIROPCNT.

Note: The output data of channel n may be infinity depending on the channel processing. At this time, an operation error occurs. For details on the operation error, see section 33.3.3. Operation When an Operation Error Occurs.

33.3.2 Channel Processing Operation

Channel processing is a series of operations that are executed when a write access is performed to the input register of a channel. Operations for all stages used by the channel are performed sequentially during the channel processing. If the output data operation is completed in the middle of the channel processing (before the IIRCHnSTS.CPRFF flag is 1), the IIRCHnSTS.ORDYF flag is 1 and the output data (IIRCHnOUT register) can be read.

Figure 33.5 shows an example of channel processing operation.

1. Input data is written.
   When input data is written to the IIR channel n input register (IIRCHnINP), the IIR filter operation is performed in accordance with the setting of the IIR channel n control register (IIRCHnCNT). At this time, the IIRCHnSTS.CPRCS bit is 1. In addition, the delay data registers for stage m (IIRSTGmD0, IIRSTGmD1) used for the operation are updated by the operation result.
2. Operation of the output data is completed.
   When the operation of the output data is completed, the IIRCHnSTS.ORDYF flag is 1 and the operation result is stored to the IIR channel n output register (IIRCHnOUT).
3. Output data is read.
   When the value is read from the IIRCHnOUT register, the IIRCHnSTS.ORDYF flag is 0.

4. Channel processing is completed.
   When the channel processing is completed, the IIRCHnSTS.CPRCFF flag is 1. In addition, the IIRCHnSTS.CPRCS bit is 0.

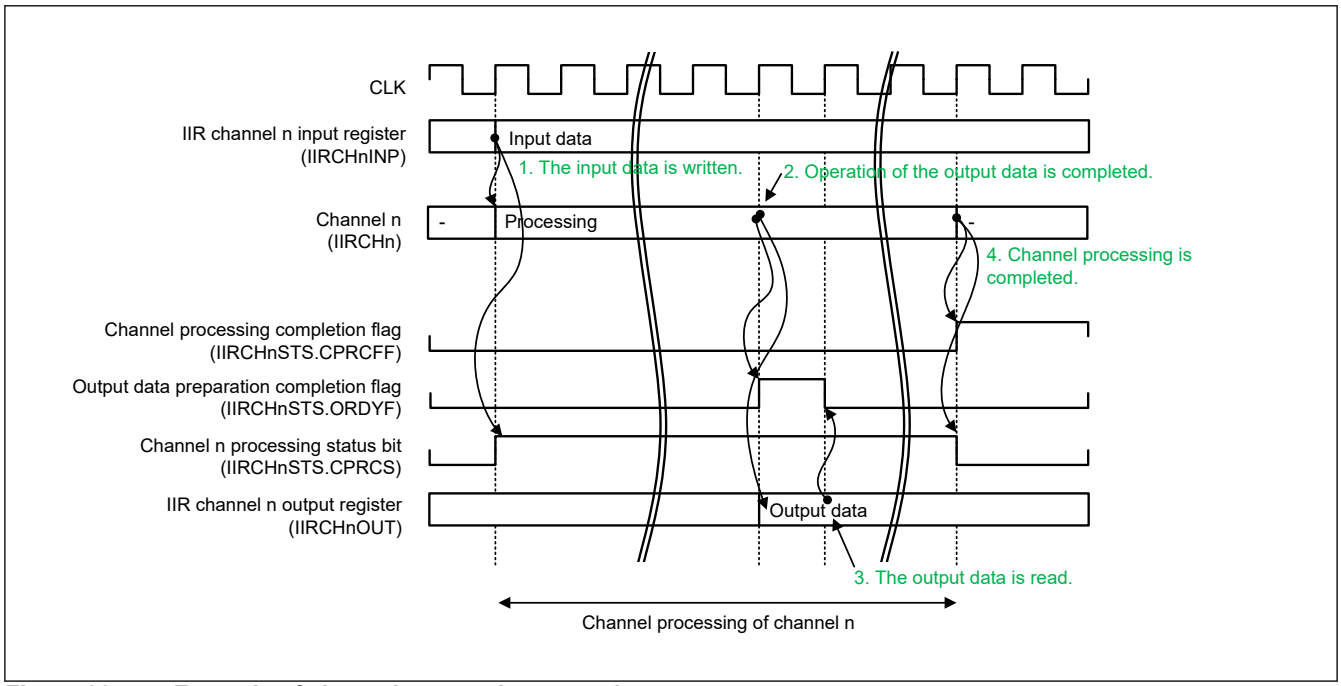


Figure 33.5 Example of channel processing operation

Note: The maximum number of channels that can be processed at the same time is 1. If a write access to the IIRCHnINP register is performed during the channel processing of any channel, bus access is forced to wait until the channel processing that is being performed is completed.

33.3.3 Operation When an Operation Error Occurs

An operation error shows that the result of the cascade connected biquad IIR filter operation is infinity. If the value of the IIR channel n output register (IIRCHnOUT) is positive or negative infinity, an operation error occurs at the same time as when the output data preparation is completed. If an operation error occurs, the IIRCHnSTS.CERRF flag is 1.

A positive infinity and a negative infinity are collectively referred to as infinity or ∞. In addition, +∞ indicates positive infinity and −∞ indicates negative infinity.

The value of the IIRCHnOUT register is the value that is obtained by performing the addition and multiplication of single precision floating-points multiple times. When the result of any addition or multiplication is infinity, this value is propagated and the value of the IIRCHnOUT register ultimately becomes infinity.

The following examples show the conditions in which the result of the addition and multiplication becomes infinity.

(1) An overflow occurs in the addition or multiplication.

If the result of addition or multiplication exceeds the maximum finite value represented by a single precision floating-point, an overflow occurs. If an overflow occurs, the result of the relevant addition or multiplication becomes infinity.

(2) The input value of the addition or multiplication includes SNaN, QNaN, or infinity.

When the input value of the addition or multiplication includes SNaN, QNaN, or infinity, the result of the relevant addition or multiplication becomes infinity.

Table 33.3 and Table 33.4 show the calculation results for input values at the time of addition and multiplication.

Table 33.3 Calculation results for input values at the time of addition (x + y)

		y				
		Normal number	+0, +Denormalized number	−0, −Denormalized number	+∞, +SNaN, +QNaN	−∞, −SNaN, −QNaN
x	Normal number	Addition*1			+∞	−∞
	+0, +Denormalized number	Addition*1	+0		+∞	−∞
	−0, −Denormalized number	Addition*1	+0	−0	+∞	−∞
	+∞, +SNaN, +QNaN	+∞				
	+∞, +SNaN, −QNaN	−∞			+∞	−∞

Note 1. If an overflow occurs, the calculation result is +∞ or −∞.

Table 33.4 Calculation results for input values at the time of multiplication (x × y)

		y					
		+Normal number	−Normal number	+0, +Denormalized number	−0, −Denormalized number	+∞, +SNaN, +QNaN	−∞, −SNaN, −QNaN
x	+Normal number	Multiplication*1				+∞	−∞
	−Normal number	Multiplication*1				−∞	+∞
	+0, +Denormalized number	Multiplication*1		+0	−0	+∞	
	−0, −Denormalized number	Multiplication*1		−0	+0	+∞	
	+∞, +SNaN, +QNaN	+∞	−∞	+∞			−∞
	+∞, +SNaN,−QNaN	−∞	+∞			−∞	+∞

Note 1. If an overflow occurs, the calculation result is +∞ or −∞.

33.3.4 Operation on ECC Error Detection

The coefficient/delay data storage area has an ECC error detection/correction function and a corrected data write-back function.

The ECC specification is SEC-DED (Single-Error Correction/Double-Error Detection). A 7-bit ECC code is generated for one coefficient/delay data (32-bit data).

(1) ECC error detection function

This ECC error detection function detects the ECC error (ECC 1-bit error, ECC 2-bit error) if an error occurs in the data in the coefficient/delay data storage area. An ECC error is detected when a read access to the coefficient/delay data register is performed and when the coefficient/delay data is used during the channel processing.

An ECC error may be detected multiple times during one channel processing. Table 33.5 shows the conditions for detecting an ECC error.

Table 33.5 Conditions for detecting an ECC error

ECC error detection/correction function	Number of bits for the error that occurred	ECC error
Enable (IIRECCCNT.ECCMD = 1)	0 bits	No ECC error is detected.
	1 bit	An ECC 1-bit error is detected.
	2 bits	An ECC 2-bit error is detected.
	3 bits or more	Whether an ECC error is detected is undefined.
Disable (IIRECCCNT.ECCMD = 0)	Don't Care	No ECC error is detected.

When an ECC 1-bit error is detected, the IIRECCEF.ESEF flag is 1. When an ECC 2-bit error is detected, the IIRECCEF.EDEF flag is 1.

(2) ECC error correction function

The ECC error correction function corrects errors in the values read from the coefficients/delayed data register or the data used for the operation. When an ECC 1-bit error is detected, the error correction is performed.

(3) Corrected data write-back function

The corrected data write-back function writes back the value after an ECC error correction to the data in the coefficient/delay data storage area. If the ECC error correction is performed when the corrected data write-back function is enabled (IIRECCCNT.ECCWBDIS = 0), the corrected data is written back.

When the ECC error correction is performed during the channel processing, the channel processing continues using the data after ECC error correction. At this time, the number of channel processing cycles is not extended.

When the corrected data is written back during the channel processing, the channel processing continues. At this time, the number of channel processing cycles is extended in accordance with the number of write-back times.

33.3.5 Operating Procedure

33.3.5.1 Initial Settings

Perform the initial settings of IIRFA before use. Table 33.6 shows an example of the initial setting procedure.

When the ECC error detection/correction function is enabled (IIRECCCNT.ECCMD = 1), write the initial values to all coefficient/delay data registers of all stages. The data in the coefficient/delay data storage area is undefined when the power is turned on. Therefore, when the ECC error detection/correction function is enabled (IIRECCCNT.ECCMD = 1), if the data is read from the coefficient/delay data storage area without initializing or if the channel processing is started without initializing, an ECC error occurs.

Table 33.6 Example of initialization procedures

Step	Description	Register to be set
1	Set the ECC function of the coefficient/delay data storage area.	IIRECCCNT
2	Set what stages are to be cascade connected for use in each channel.	IIRCHnCNT
3	Set the initial values (single precision floating-points) to the coefficient/delay data registers.	IIRSTGmB0, IIRSTGmB1, IIRSTGmB2,IIRSTGmA1, IIRSTGmA2, IIRSTGmD0, IIRSTGmD1
4	Enable or disable the ECC error interrupt and channel n interrupt.	IIRECCINT IIRCHnINT

33.3.5.2 Procedure for Channel Processing Execution

This section shows an example procedure of the channel process execution.

There are three methods to execute the channel processing. The difference between the three methods is the procedure for output data reading after the channel processing starts.

Table 33.7 shows the procedure for output data reading of each method.

Table 33.7 Procedure for output data reading of each method

	Single data processing	Multiple data processing
Method 1	Read without waiting for output data preparation completion.	
Method 2	Read after polling the output data preparation completion flag.	Read after polling the channel processing completion flag.
Method 3	Read after accepting the output data preparation completion interrupt.	Read after accepting the channel processing completion interrupt.

The single data processing and multiple data processing classify the channel processing procedure in accordance with the number of data to be processed. Single data processing means that the channel processing is performed for one input value

and no other channel processing is performed after the processing is completed. Multiple data processing means that the channel processing is performed consecutively for multiple input values.

Method 2 and method 3 have different procedures for single data processing and multiple data processing. IIRFA completes the output data preparation before the channel processing is completed. Therefore, to reduce the number of cycles, the procedure for single data processing is to read the output data when the output data preparation is completed, instead of when the channel processing is completed.

Points to note for methods 1, 2, and 3 are as follows:

- Method 1
  - Since the flag determination processing is not required, the overhead is small.
  - When a read access to the IIRCHnOUT register is performed, bus access is forced to wait until the output data preparation is completed.
- Method 2
  - Since the flag determination processing is required, the overhead is large.
  - When a read access to the IIRCHnOUT register is performed, bus access is not forced to wait.
- Method 3
  - The overhead is relatively large due to processing at the time of interrupt acceptance.
  - When a read access to the IIRCHnOUT register is performed, bus access is not forced to wait.
  - After the channel processing starts, the CPU can perform other process until the output data preparation completion interrupt or the channel processing completion interrupt occurs.

In accordance with the number of stages used for the channel processing, the number of cycles increases from the start of the channel processing to the completion of output data preparation and channel processing. Therefore, the recommended method may vary depending on the number of stages to be used.

Method 1 is recommended when the number of stages used for the channel processing is small. When this method is used, the bus is occupied until the output data preparation is completed. Therefore, bus access to other bus master that uses the same bus is forced to wait. In addition, the CPU does not accept any interrupt while the bus access is waited.

Methods 2 and 3 are recommended when the number of stages used for the channel processing is large. When these methods are used, the overhead is large due to the flag determination processing or interrupt processing. Therefore, when the number of stages used for the channel processing is small, the effect of the overhead is large.

Figure 33.6, Table 33.7, and Figure 33.8 show an example procedure for each method.

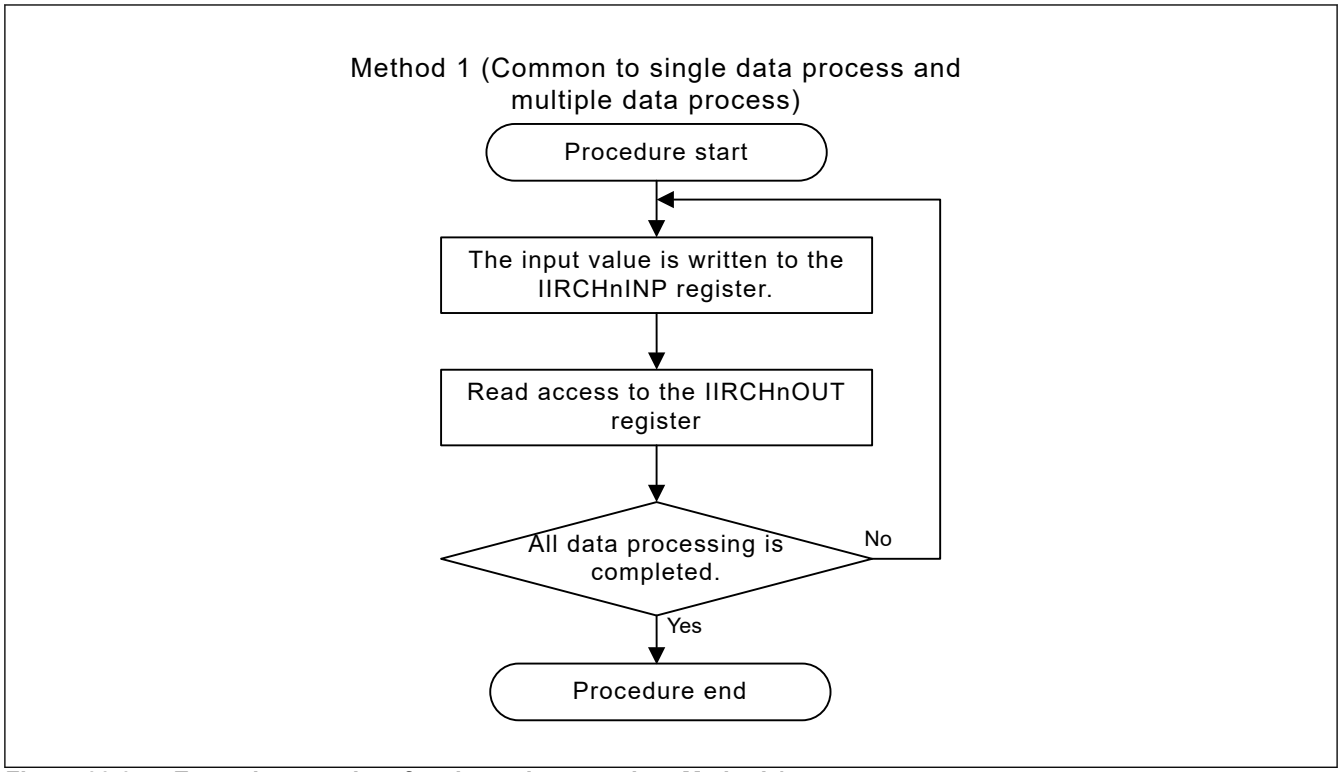


Figure 33.6 Example procedure for channel processing: Method 1

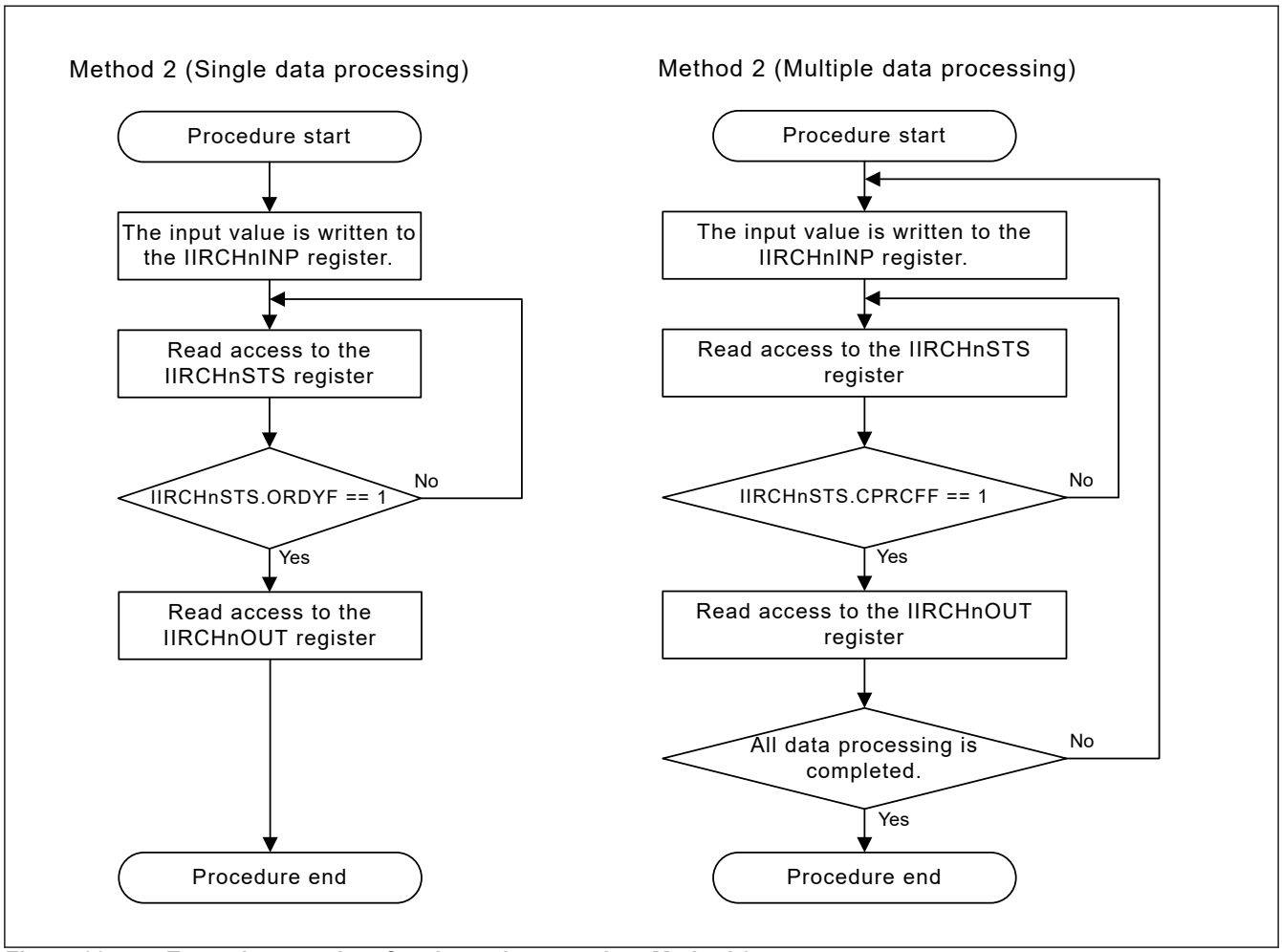
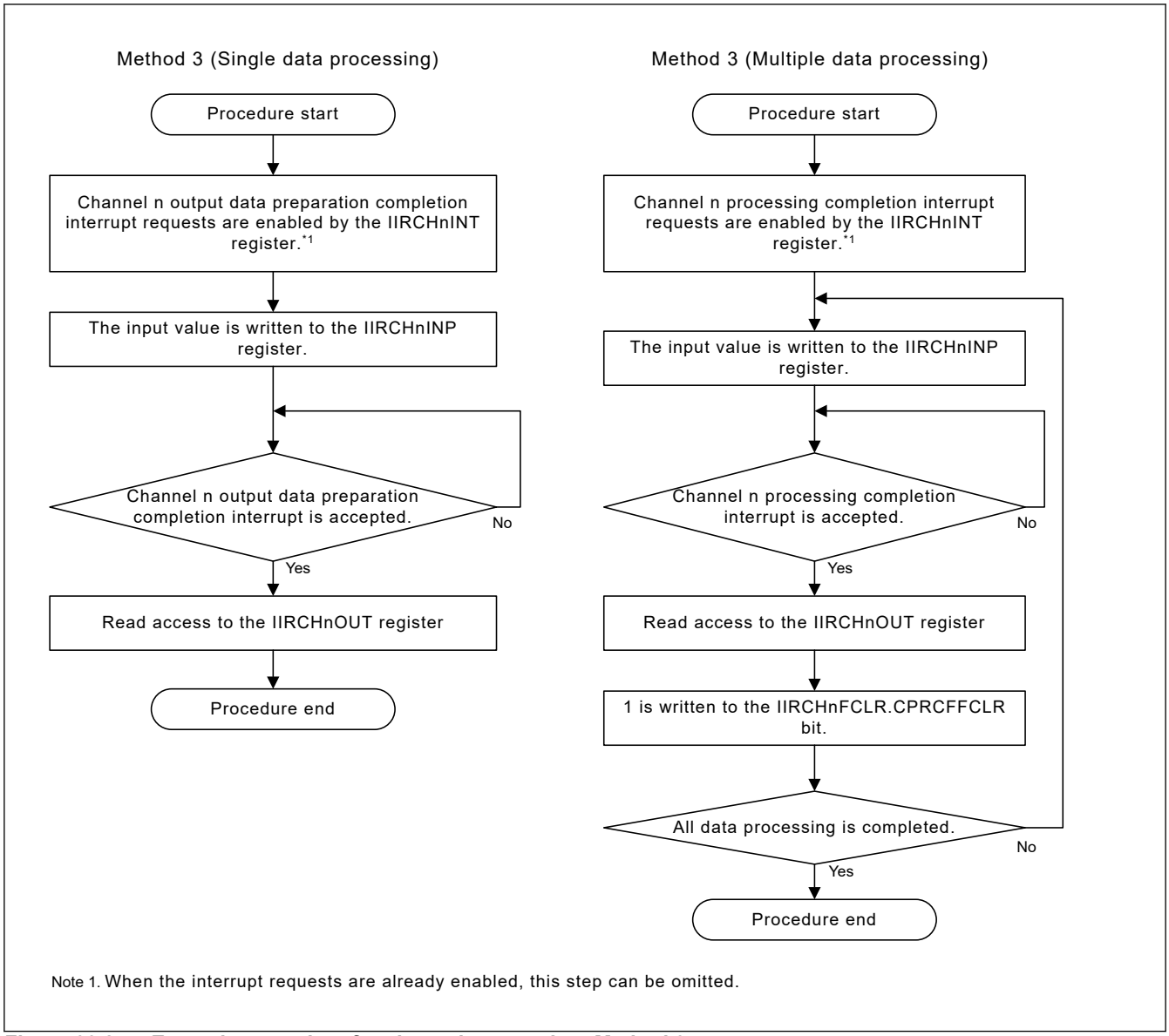


Figure 33.7 Example procedure for channel processing: Method 2



Note 1. When the interrupt requests are already enabled, this step can be omitted.

Figure 33.8 Example procedure for channel processing: Method 3

33.3.5.3 Procedure for Intentionally Detecting ECC Errors

This section describes the procedure for intentionally detecting ECC errors (ECC 1-bit error, ECC 2-bit error). To check the behavior of a software routine after an ECC error is detected, perform these procedures.

If the ECC error detection/correction function is disabled (IIRECCCNT.ECCMD = 0), any data can be written to the coefficient/delay data register without updating the ECC code when data is written to the coefficient/delay data register. Therefore, the data error occurrence status of the coefficient/delay data storage area can be reproduced by reversing an appropriate bit of the coefficient/delay data. After the corrupted data has been generated, a read access to the coefficient/delay data register is performed to detect an ECC 1-bit error or ECC 2-bit error in accordance with the number of reversed bits.

The following steps show an example procedure.

1. Perform the initial settings with the ECC error detection/correction function enabled (IIRECCCNT.ECCMD = 1).
2. Write any data to a coefficient/delay data register in any stage.
3. Disable the ECC error detection/correction function (IIRECCCNT.ECCMD = 0).
4. Write a value to detect an ECC 1-bit error or ECC 2-bit error to the address to which data is written in step 2.

Example: When 0x00000000 is written in step 2, writing 0x00000001 reproduces a state where a 1-bit error occurs in the coefficient/delay data and writing 0x00000003 reproduces a state where a 2-bit error occurs in the coefficient/delay data.

5. Enable the ECC error detection/correction function (IIRECCCNT.ECCMD = 1).
6. Read the data from the address to which data is written in steps 2 and 4.
   At this time, an ECC 1-bit error or ECC 2-bit error is detected depending on the value of the data written in steps 2 and 4.

33.4 Interrupt Sources

Table 33.8 shows a list of interrupt sources.

Table 33.8 Interrupt sources

Name	Interrupt source	Interrupt flag	Interrupt generation condition
IIRFA_ORDYn (n = 0 to 2)	Channel n output data preparation is completed.	IIRCHnSTS.ORDYF	IIRCHnSTS.ORDYF = 1 and IIRCHnINT.ORDYIE = 1
IIRFA_ORDY3 (m = 3 to 15)	Output data preparation is completed in any of channel m.	IIRCHmSTS.ORDYF	IIRCHmSTS.ORDYF = 1 and IIRCHmINT.ORDYIE = 1
IIRFA_CPRCFn (n = 0 to 2)	Channel n processing is completed.	IIRCHnSTS.CPRCFF	IIRCHnSTS.CPRCFF = 1 and IIRCHnINT.OPRCFIE = 1
IIRFA_CPRCF3 (m = 3 to 15)	Processing is completed in any of channel m.	IIRCHmSTS.CPRCFF	IIRCHmSTS.CPRCFF = 1 and IIRCHmINT.CPRCFIE = 1
IIRFA_ERR (x = 0 to 15)	An operation error has occurred in any of channel.	IIRCHxSTS.CERRF	IIRCHxSTS.CERRF = 1 and IIRCHxINT.CERRIE = 1
	An ECC 1-bit error has occurred.	IIRECCEF.ESEF	IIRECCEF.ESEF = 1 and IIRECCINT.ESEIE = 1
	An ECC 2-bit error has occurred.	IIRECCEF.EDEF	IIRECCEF.EDEF = 1 and IIRECCINT.EDEIE = 1

IIRFA_ORDY3 is an output data preparation completion interrupt of wire-ORed channels 3 to 15. To check the channel in which an interrupt is generated, see section 33.2.1.3. IIRORDYF : Output Data Preparation Completion Flag Register.

IIRFA_CPRCF3 is an processing completion interrupt of wire-ORed channels 3 to 15. To check the channel in which an interrupt is generated, see section 33.2.1.2. IIRCPRCFF : Channel Processing Completion Flag Register.

IIRFA_ERR is a wire-ORed interrupt for the all channel of operation error, ECC 1-bit error, and ECC 2-bit error. To check which error interrupts are generated, see section 33.2.1.4. IIRCERRF : Operation Error Flag Register, and section 33.2.1.8. IIRECCEF : ECC Error Flag Register.

34. Boundary Scan

34.1 Overview

The boundary scan function provides a serial I/O interface based on the JTAG (Joint Test Action Group), IEEE Std.1149.1, and IEEE Standard Test Access Port and Boundary Scan Architecture. Table 34.1 lists the boundary scan specifications, Figure 34.1 shows a block diagram, and Table 34.2 lists the I/O pins.

Table 34.1 Boundary scan specifications

Parameter	Specifications
Execution condition	Boundary scan must be executed when the RES pin is driven low.
Test modes	• BYPASS mode • EXTEST mode • SAMPLE/PRELOAD mode • CLAMP mode • HIGHZ mode • IDCODE mode

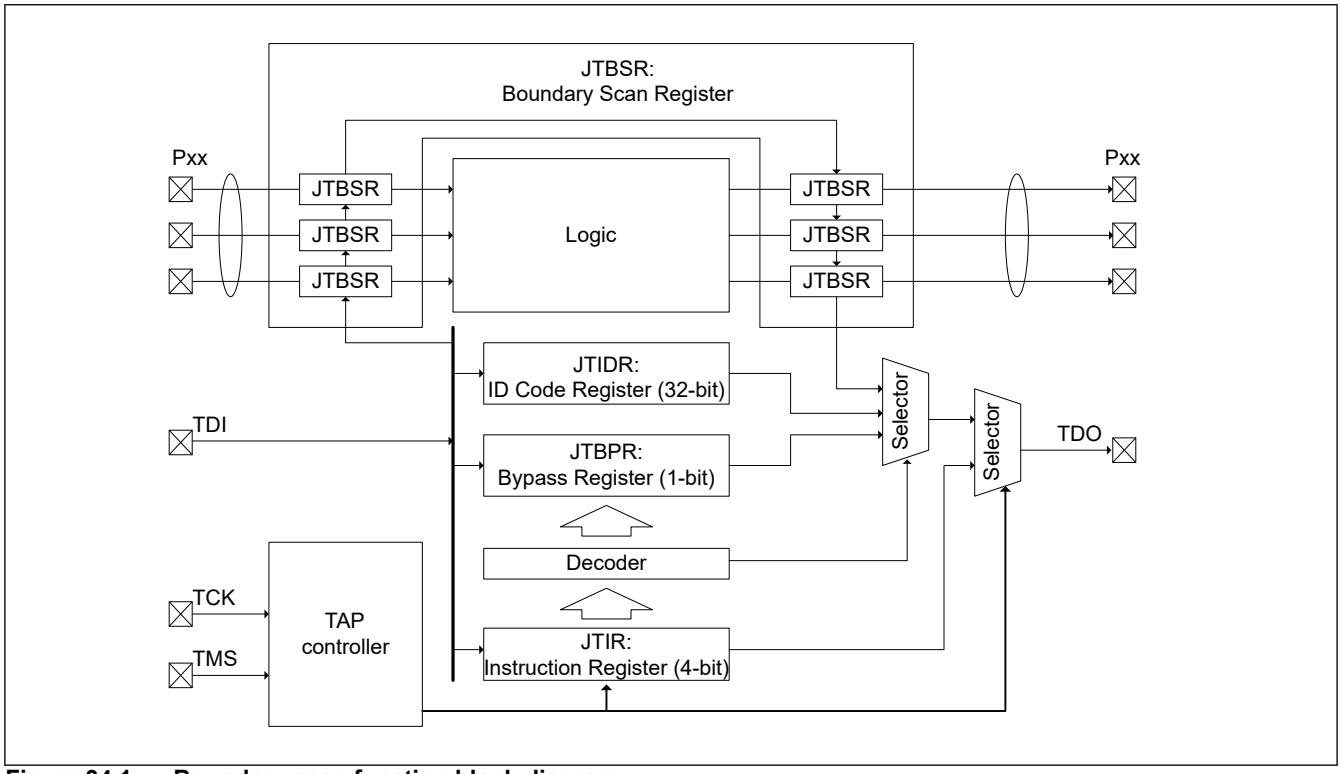


Figure 34.1 Boundary scan function block diagram

Table 34.2 Boundary scan I/O pins

Pin name	I/O	Description
TCK	Input	Test clock input pin Clock signal for boundary scan. The input clock duty cycle is 50% when the boundary scan function is used.
TMS	Input	Test mode select pin
TDI	Input	Test data input pin
TDO	Output	Test data output pin

Note: This device does not support the TRST pin for the JTAG interface.

34.2 Register Descriptions

Table 34.3 lists the boundary scan registers.

Table 34.3 Boundary scan registers

Register name	Symbol	Value after reset
Instruction Register	JTIR	0xE
ID Code Register	JTIDR	0x0841_F447
Bypass Register	JTBPR	Undefined
Boundary Scan Register	JTBSR	Undefined

Usage notes for the boundary scan registers:

- Instructions can be input to the Instruction Register (JTIR) through the TDI pin by serial transfer.
- The Bypass Register (JTBPR), which is a 1-bit register, is connected between the TDI and TDO pins in BYPASS mode.
- The Boundary Scan Register (JTBSR), which is configured according to the BSDL description, is connected between the TDI and TDO pins when test data is being shifted in.

Table 34.4 shows the availability of serial transfer for the registers.

Table 34.4 Serial transfer for registers

Register name	Serial input	Serial output
Instruction Register (JTIR)	Available	Available
ID Code Register (JTIDR)	Available	Available
Bypass Register (JTBPR)	Available	Available
Boundary Scan Register (JTBSR)	Available	Available

34.2.1 JTIR : Instruction Register

Bit position:	3	2	1	0
Bit field:	TS[3:0]			
Value after reset:	1	1	1	0

Bit	Symbol	Function	R/W
3:0	TS[3:0]	Test Bit Set The command configuration for these bits	—

TS[3:0]	Instruction
0x0	EXTEST
0x1	SAMPLE/PRELOAD
0x3	IDCODE (Renesas code)
0x5	CLAMP
0x6	HIGHZ
0xF	BYPASS
Others	Reserved

JTAG instructions can be transferred to the JTIR register by serial input from the TDI pin. The JTIR register is initialized when a power-on reset occurs, or when the TAP controller is in the Test-Logic-Reset state.

34.2.2 JTIDR : ID Code Register

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

Bit	Symbol	Function	R/W
31:0	DID[31:0]	Device ID These bits store the fixed value that indicates the device IDCODE (0x0841_F447).	—

The JTIDR register data is output from the TDO pin when the IDCODE instruction is executed. After a reset release, the DID[31:0] of JTIDR changes into the Arm® debug code. See the *Arm® CoreSight™ SoC-400 Technical Reference Manual* (ARM DDI 0480F).

34.2.3 JTBPR : Bypass Register

The JTBPR register is a 1-bit register and is connected between the TDI and TDO pins when the JTIR register is set to BYPASS mode. The JTBPR register cannot be read from or written to by the CPU.

34.2.4 JTBSR : Boundary Scan Register

The JTBSR register is a shift register for controlling the external input and output pins of this device, and is distributed across the pads. To apply the JTBSR register in boundary-scan testing, issue the EXTEST, SAMPLE/PRELOAD, CLAMP, and HIGHZ instructions. The BSDL file describes the associations between the JTBSR register bits and the pins of this device. The value after reset is undefined.

34.3 Operation

During a reset, the JTAG ports, TCK, TMS, TDI, and TDO, are assigned as default pin functions. The TCK, TMS, and TDI pins are pulled up by the pull-up resistors. Boundary scan testing can be executed after the setup time elapses when POR is negated and RES is driven low.

34.3.1 TAP Controller

Figure 34.2 shows the state transition diagram of the TAP controller. All transitions are controlled by the TMS signal.

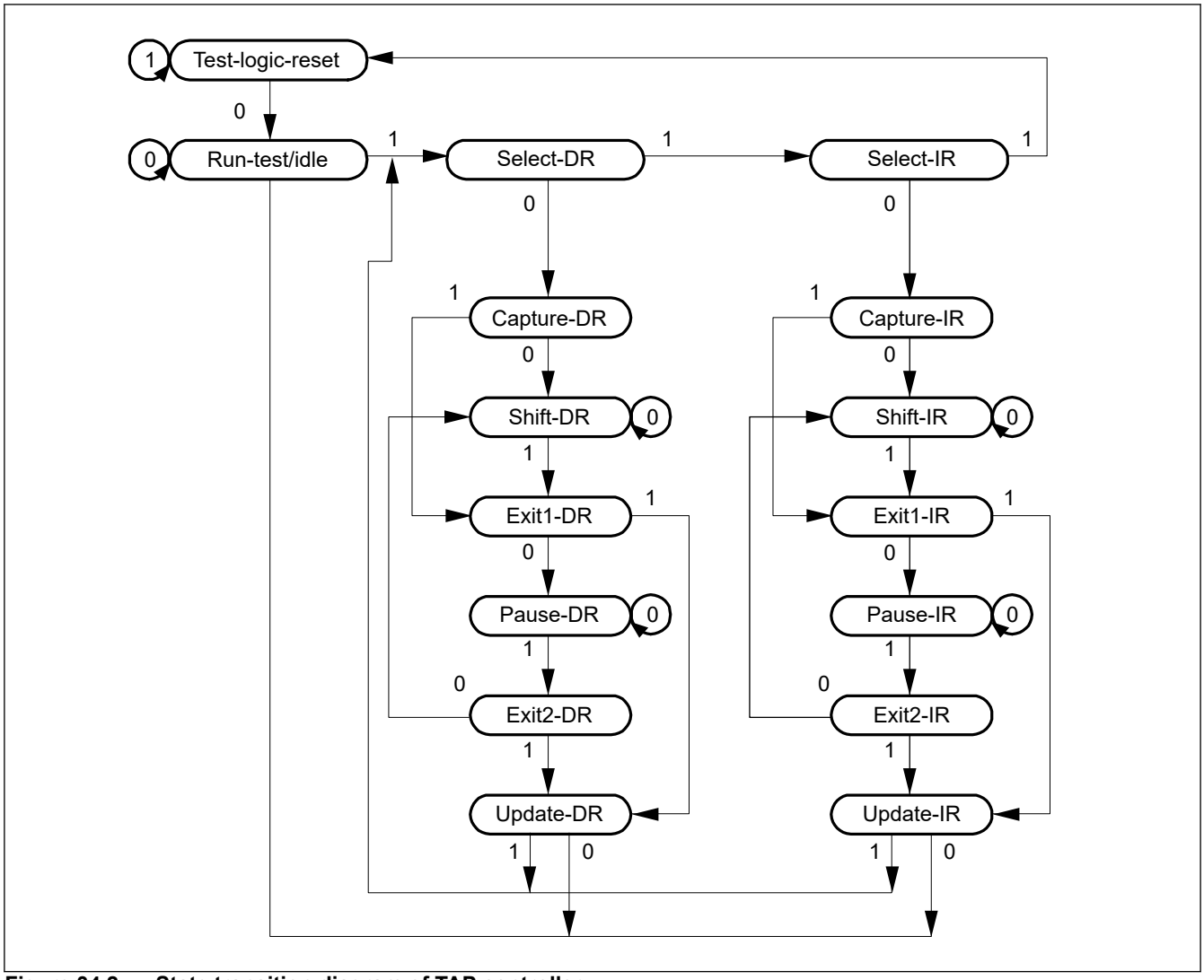


Figure 34.2 State transition diagram of TAP controller

34.3.2 Commands

(1) BYPASS

The BYPASS instruction drives the Bypass Register (JTBPR). This instruction shortens the shift path, facilitating the transfer of serial data to other LSIs on a printed circuit board at higher speeds. While this instruction is being executed, the test circuit has no effect on the system circuits.

The JTBPR register is connected between the TDI and TDO pins. Bypass operation is initiated from the Shift-DR operation. The TDO is low in the first clock cycle in the Shift-DR state. In the subsequent clock cycles, values input to the TDI pin are output from the TDO pin.

(2) EXTEST

The EXTEST instruction is used to test external circuits when this device is installed on the printed circuit board. If this instruction is executed, output pins are used to output test data (specified in the SAMPLE/PRELOAD instruction) from the Boundary Scan Register (JTBSR) to the other devices, and input pins are used to input the test result.

(3) SAMPLE/PRELOAD

The SAMPLE/PRELOAD instruction is used to input data from the internal circuits of this device to the JTBSR register, output data from the scan path, and reload the data to the scan path. While this instruction is executed, input signals are directly input to this device and output signals are also directly output to the external circuits. This device system circuit is not affected by this instruction.

In SAMPLE operation, the JTBSR register latches a snapshot of the data transferred from the input pins to the internal circuit or data transferred from the internal circuit to the output pins. The latched data is read from the scan path. The JTBSR register latches the data snapshot on the rising edge of the TCK pin in the Capture-DR state. The data snapshot is only transferred from the internal circuit to the output pins during a reset.

In PRELOAD operation, the initial value is written from the scan path to the parallel output latch of the JTBSR register prior to the EXTEST instruction execution. If EXTEST is executed without executing this PRELOAD operation, undefined values are output from the beginning to the end (transfer to the output latch) of the EXTEST sequence. (In the EXTEST instruction, output parallel latches are always output to the output pins.)

(4) IDCODE

When the IDCODE instruction is selected, the ID Code Register (JTIDR) value is output to the TDO pin in the Shift-DR state of the TAP controller. In this case, the JTIDR register value is output LSB-first. During this instruction execution, the test circuit does not affect the system circuit.

(5) CLAMP

When the CLAMP instruction is selected, output pins output the JTBSR register value that was specified in the SAMPLE/PRELOAD instruction in advance. While the CLAMP instruction is selected, the status of the JTBSR register is maintained regardless of the TAP controller state.

The JTBPR register is connected between the TDI and TDO pins, leading to the same operation as when the BYPASS instruction is selected.

(6) HIGHZ

When the HIGHZ instruction is selected, all output pins enter high-impedance state. While the HIGHZ instruction is selected, the JTBSR register is maintained regardless of the state of the TAP controller.

The JTBPR register is connected between the TDI and TDO pins, leading to the same operation as when the BYPASS instruction is selected.

34.4 Usage Notes

The boundary scan function is subject to the following constraints:

- The boundary scan must be executed when the RES pin is driven low
- Serial data input/output is in LSB order, as shown in Figure 34.3

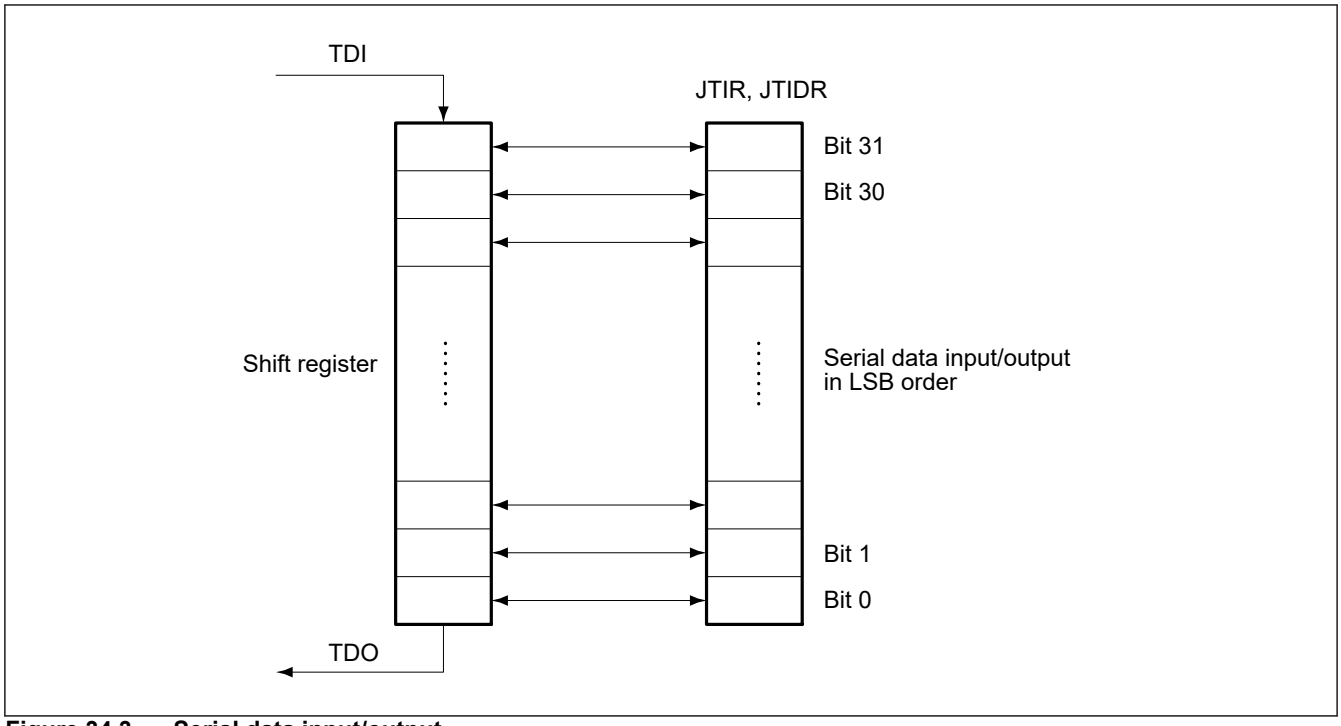


Figure 34.3 Serial data input/output

The following pins cannot be boundary-scanned:

- Power supply pins (VCC, VCL, VSS, AVCC0, AVSS0)
- Analog reference pins (AVREFH0, AVREFL0)
- Clock pins (EXTAL, XTAL)
- Reset pin (RES)
- The boundary-scan pins (TCK, TMS, TDI, and TDO).

35. Secure Cryptographic Engine (SCE5)

This is the SCE5_B version of the SCE5 peripheral module.

SCE5_B is referred to as SCE5 in this chapter.

35.1 Overview

The Secure Cryptographic Engine (SCE5) consists of the access management circuit, encryption engine, and random number generation circuit. In combination with the SCE5 library, the SCE5 can prevent eavesdropping (to maintain confidentiality), falsification of information (to ensuret integrity), and impersonation (to verify authenticity).

Because key information required for encryption and decryption is stored only in the SCE5 and all accesses from the outside can be blocked, SCE5 enables building a more robust security system.

Table 35.1 lists the SCE5 specifications. Figure 35.1 shows the SCE5 block diagram.

Table 35.1 SCE5 specifications

Parameter	Specifications
Access control	Access management circuit ● In case of irregular access to the SCE5 due to a tampered program or CPU runaway, this circuit blocks all subsequent accesses and stops data output from the SCE5
Encryption engine	AES: Compliant with NIST FIPS PUB 197 ● Key length: 128 or 256 bits ● Data block size: 128 bits ● Encryption usage modes ECB, CBC, CTR: Compliant with NIST SP 800-38A CMAC: Compliant with NIST SP 800-38B GCM: Compliant with NIST SP 800-38D XTS: Compliant with NIST SP 800-38E GCTR ● Throughput for 128-bit data 44 PCLKA cycles for 128-bit key 61 PCLKA cycles for 256-bit key*1 AES-GCM ● AES-GCM is realized by combining AES-GCTR and GHASH. Key management ● Wrapped keys are only valid within the SCE5
Random number generation	32-bit true random number generation circuit
Hardware Unique Key	● A read-only, 128-bit Hardware Unique Key (HUK). ● Key derivation functions (KDFs) combine the Hardware Unique Key with the key generation information. The derived keys implement the key wrapping for user key secure storage. ● The HUK uniqueness prevents the illicit cloning and copying of keys to another MCU of the MCU group. ● The HUK itself is stored in wrapped (encrypted, non-plain) format in an isolated memory area. Therefore it is protected from illicit access and copy.
Unique ID	● A read-only, 128-bit ID unique to an MCU (Unique ID) is accessible from the access management circuit. ● Key derivation functions (KDFs) combine the Unique ID with the key generation information. Such derived keys are used to unwrap the HUK within the SCE boundary.
Low power consumption	Setting of the module-stop state is possible

Note 1. This does not include the overhead of calling SCE5 library functions.

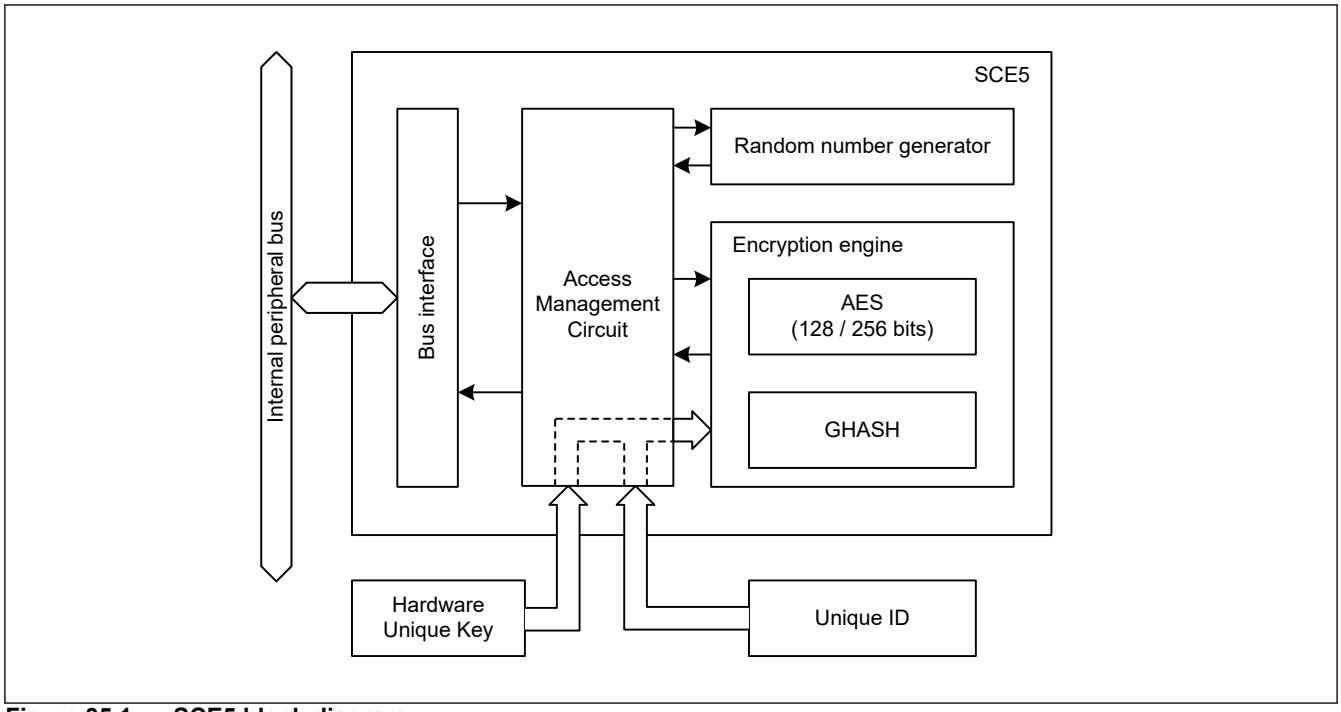


Figure 35.1 SCE5 block diagram

35.2 Operation

35.2.1 Encryption Engine

Figure 35.2 shows conceptual diagram of the encryption engine installed in the SCE5.

The encryption engine uses the key generation information, and converts the plaintext data to ciphertext or ciphertext data to plaintext through the hardware.

The encryption/decryption process can be completed without exposing the key data and the process's intermediate data to the outside of the SCE5.

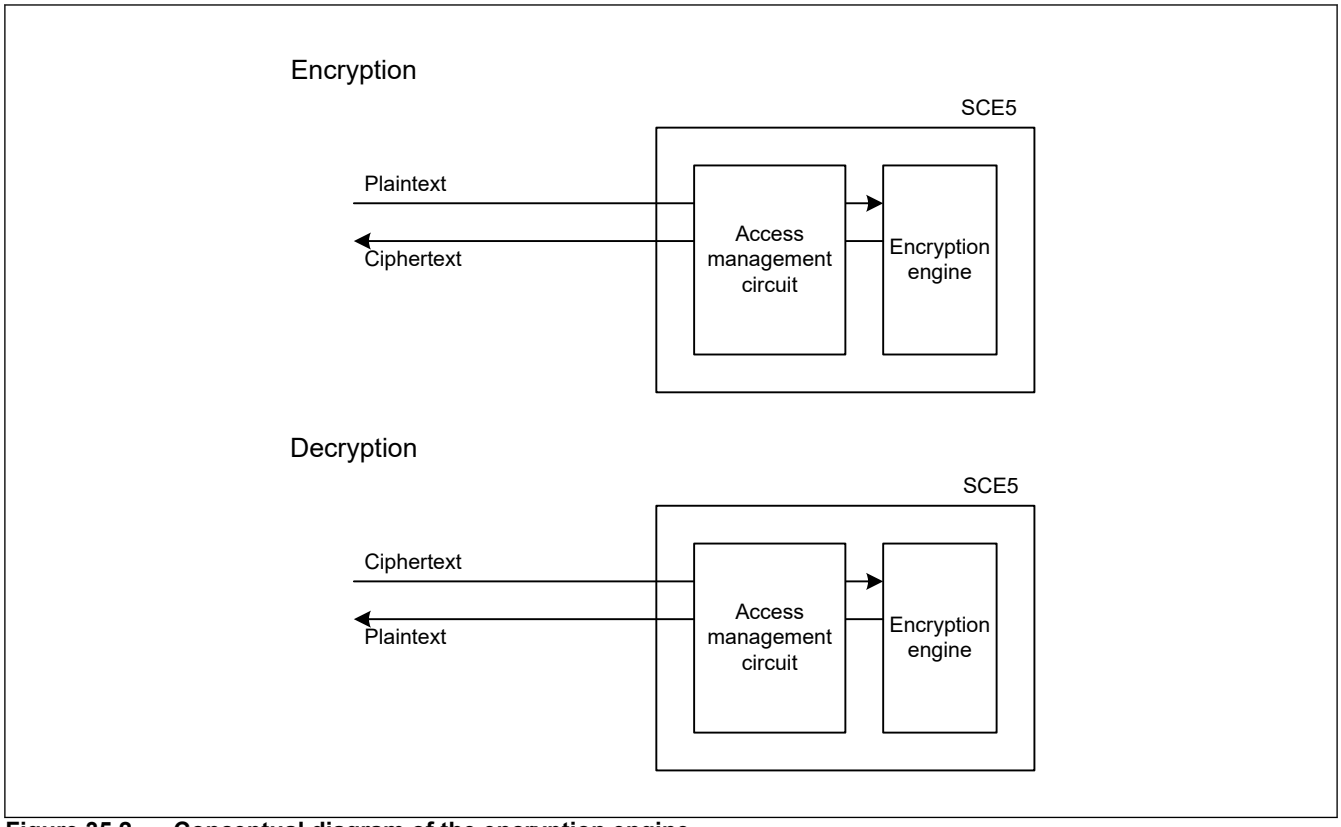


Figure 35.2 Conceptual diagram of the encryption engine

35.2.2 Encryption and Decryption

Follow the procedure below to encrypt and decrypt the data:

1. Enter the key generation information to the SCE5 and restore the key data.
2. Enter the target data to the SCE5. Plaintext data is converted to ciphertext and ciphertext data to plaintext.
3. Read the converted data.

The encryption engine has input and output buffers, and can perform encryption/decryption in parallel with data input/output.

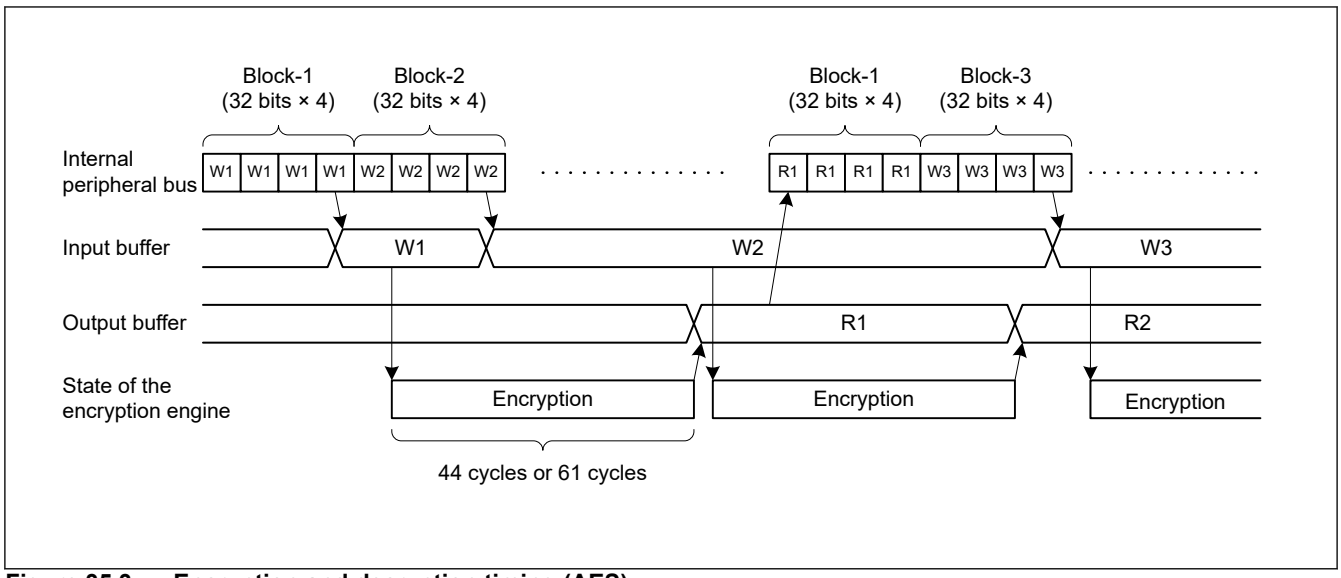


Figure 35.3 Encryption and decryption timing (AES)

35.3 Usage Notes

35.3.1 Software Standby Mode

When Software Standby mode is entered while the encryption engine is in process, proper processing cannot be resumed after cancelling Software Standby mode. Software Standby mode should therefore be entered while the encryption engine is not running.

35.3.2 Module-Stop Function Setting

SCE5 operation can be disabled or enabled using Module Stop Control Register C (MSTPCRC). The SCE5 module is initially stopped after reset. Releasing the module-stop state enables access to the registers.

36. A/D Converter

This is the ADC_B version of the ADC peripheral module.

ADC_B is referred to as ADC in this chapter.

36.1 Overview

This MCU contains two units of Noise-Shaping SAR-type A/D converters (ADC) that is the hybrid architecture with features of SAR-type and Delta-Sigma modulation-type. The A/D converter unit 0 (ADC0) can select up to 21 channels of analog inputs. The A/D converter unit 1 (ADC1) can select up to 17 channels of analog inputs. The temperature sensor, internal reference voltage, and D/A converters can be A/D-converted by A/D converter unit 0 or unit 1. A/D conversion data can be selected from 16-bit, 14-bit, 12-bit, and 10-bit length data formats.

The ADC has the following features.

- Resolution: Up to 16-bit
- Fast conversion: Up to 6.25 Msps (0.16 µs per channel) (when A/D conversion clock ADCLK = 50 MHz)
- Input channels: Up to 29 analog input channels
- Support single-ended input or differential inputs
- Self-calibration Function
- Built-in channel-dedicated sample-and-hold circuit (S&H)
- Built-in Programmable Gain Amplifier (PGA)

Table 36.1 shows the ADC specifications, Table 36.2 shows the correspondence between ADC functions and ADC operation mode, and Table 36.3 shows the ADC I/O pins.

Figure 36.1 shows the block diagram of ADC. Figure 36.2 shows the analog channel structure for A/D converter unit 0. Figure 36.3 shows the analog channel structure for A/D converter unit 1.

Table 36.5 shows the ADC channel configuration of the extended analog function.

Table 36.1 ADC specifications (1 of 4)

Item	Description
Number of units	Two units (unit 0 and unit 1)
Input channels	• Support single-ended input or differential inputs • Up to 29 analog input channels – A/D converter unit 0: Up to 21 analog input channels at Single-ended input, 12 analog input channels (6 pairs) of which support Differential inputs. – A/D converter unit 1: Up to 17 analog input channels at Single-ended input, 8 analog input channels (4 pairs) of which support Differential inputs. – 9 analog input channels are shared by A/D converter unit 0 and unit 1
Extended analog function	Self-diagnosis, temperature sensor, internal reference voltage, D/A converter (DA0 to DA3)
A/D conversion method	• Successive approximation register method in SAR mode • Noise-Shaping Successive approximation register method in Oversampling mode and Hybrid mode
Resolution of A/D converter	• Up to 12-bits in SAR mode • Up to 16-bits in Oversampling mode or Hybrid mode
Conversion time	0.16 µs per channel (when A/D conversion clock ADCLK = 50 MHz).
A/D conversion clock	The A/D conversion clock (ADCLK) can be set by selecting the clock source and the division ratio as follows: • The clock source: the peripheral module clock PCLKC*1, the peripheral module clock PCLKA*1, the GPT clock GPTCLK*1. • The division ratio: 1/2/3/4/5/6/7/8 A/D conversion clock (ADCLK) can operate between 25 MHz at a minimum and 60 MHz at a maximum.
A/D conversion data	• A/D conversion results are stored in data register or FIFO. • A/D conversion results are available in 16-, 14-, 12-, and 10-bit data formats.

Table 36.1 ADC specifications (2 of 4)

<table>
<tr><th>Item</th><th>Description</th></tr>
<tr><td>Operation mode</td><td>
● SAR mode
– A/D converter samples the signal source once, and convert by Successive Approximation Register method
– Fast A/D conversion
– Support up to 8 channels per one scan group
– Support only single-ended input (excluding self-diagnosis function)
● Oversampling mode
– A/D converter oversamples the signal source, and converts analog-to-digital by Noise Shaping Successive Approximation Register method
– High-accuracy A/D conversion
– Support up to 8 channels per one scan group
– Support single-ended input and differential input
● Hybrid mode
– A/D converter oversamples the signal source, and converts analog-to-digital by Noise Shaping Successive Approximation Register method
– High-accuracy A/D conversion
– High data rate in continuous scanning operation
– Support up to 4 channels per one scan group
– Support single-ended input and differential input
</td></tr>
<tr><td>Scan mode</td><td>
● Single scan mode
– Assign any selected analog input or analog channel of the extended analog function to any scan group, and convert the selected analog input only once per scan group for A/D conversion.
– By selecting the scan start conditions for each scan group individually, A/D conversion for each scan group can be started at different times.
● Continuous scan mode
– Assign any selected analog input or analog channel of the extended analog function to any scan group and repeat A/D conversion in scan group units.
● Background continuous scan mode
– Assign any selected analog input or analog channel of the extended analog function to any scan group and repeat A/D conversion in scan group units.
– The A/D conversion is performed in the background and, if the A/D conversion start trigger is entered, the A/D conversion data at that point is acquired.
</td></tr>
<tr><td>Combination of operation mode and scan mode</td><td>
<table>
<tr><th rowspan="2">Scan mode</th><th colspan="3">Operation mode of A/D converter</th></tr>
<tr><th>SAR mode</th><th>Oversampling mode</th><th>Hybrid mode</th></tr>
<tr><td>Single scan mode</td><td>✓</td><td>✓</td><td>✓</td></tr>
<tr><td>Continuous scan mode</td><td>✓</td><td>✓</td><td>✓</td></tr>
<tr><td>Background continuous scan mode</td><td>—</td><td>—</td><td>✓</td></tr>
</table>
Note: ✓: available, —: not available
</td></tr>
<tr><td>Conditions for A/D conversion start</td><td>
● Software trigger (for simultaneous activation of scan group: max. 9 triggers)
● Software trigger (for individual scan group activation: up to 9 triggers)
● Trigger from Event Link Controller: 6 triggers
● Triggers from GPT: 20 triggers
● External trigger input: 2 triggers (ADTRGn input (n = 0, 1))
</td></tr>
</table>

Table 36.1 ADC specifications (3 of 4)

Item	Description
Functions	● Virtual Channel function (37 virtual channels) ● Scan Group function (up to 9 scan groups) ● Variable sampling time (select from 16 tables per virtual channel) ● Channel-dedicated sample-and-hold circuit (SH) (3 SH units for A/D converter unit 0, 3 SH units for A/D converter unit 1) ● Self-diagnosis function for A/D converter ● Selectable A/D-converted value addition mode or average mode ● Analog input disconnection detection assist function (discharge function/precharge function) ● Selectable the data format from 16- / 14- / 12- / 10-bit. ● Selectable the signed-data or the unsinged-data format (support only Oversampling mode and Hybrid mode) ● Limiter Clip Function (Up to 8 tables) ● Compare Match Function (Up to 8 tables) ● Self-calibration function ● User's Gain adjustment function ● User's Offset adjustment function ● Built-in FIFO (8 stages per each scan group) ● Built-in digital filter (2 type) (support only Oversampling mode and Hybrid mode) ● Multiple A/D converters Unit-to-unit synchronous operation function
Programmable Gain Amplifier (PGA)	● Analog input signals can be amplified by programmable gain amplifier (PGA), and A/D conversion can be performed. (3 PGA for A/D converter unit 0, 1 PGA for A/D converter unit 1) ● Support the single-ended input or the pseudo-differential input ● Monitor function via pin for PGA output
Interrupt sources	● A/D scan end interrupt – Generates the interrupt requests and the ELC events at the end of A/D scan operation for the scan group i (ADC_ADIi (i = 0 to 4)). The interrupt requests are independent for each scan group. – Generates the interrupt request and the ELC event at the end of A/D scan operation for any of the scan groups 5 to 8 (ADC_ADI5678). The interrupt request is shared for scan groups 5 to 8. ● FIFO data read request interrupt – Generates the interrupt requests when the number of vacant stages of FIFO for the scan group i becomes less than or equal to the specified value (ADC_FIFOREQi (i = 0 to 4)). The interrupt requests are independent for each scan group. – Generates the interrupt request or the ELC event when the number of vacant stages of FIFO for any of the scan groups 5 to 8 becomes less than or equal to the specified value (ADC_FIFOREQ5678). The interrupt request is shared for scan group 5 to 8. ● FIFO data overflow interrupt – Generates the interrupt request when the overflow occurs in any of the FIFO for the scan groups 0 to 8 (ADC_FIFOOVF). ● Limiter clip interrupt – Generates the interrupt request when the limiter clip using the limiter tables 0 to 7 is occurred for A/D conversion results (ADC_LIMCLPI). ● Compare match interrupt – Generates the interrupt requests when the compare match using the compare match tables j are occurred for A/D conversion results (ADC_CMPIj (j = 0 to 3)). The interrupt requests are independent for each compare match table. ● Composite compare match interrupt – Generates the interrupt requests and the ELC events when the compare match of the composite condition using compare match tables 0 to 7 occurs (ADC_CCMPMm (m = 0, 1)). ● A/D converter error interrupt – Generates the interrupt requests when the operational error is detected in A/D converter Unit j (ADC_ERRj (j = 0, 1)). ● A/D conversion overflow interrupt – Generates the interrupt request when the A/D conversion result overflow is occurred (ADC_RESOVFj (j = 0, 1)) The interrupt requests are independent for A/D converter unit j. ● A/D converter calibration end interrupt – Generates the interrupt requests at the end of calibration operation of A/D converter unit j (ADC_CALENDj (j = 0, 1)).
ELC interface	● Trigger Input – Scan can be started by the trigger from the ELC. ● Event Generation – An event is generated at the end of each scan operation for the scan group 0 to 4. – An event is generated at the end of scan operation for any of the scan group 5 to 8. – An event is generated when Complex Compare Match is occurred.

Table 36.1 ADC specifications (4 of 4)

Item	Description
Reference voltage	VREFH0 is the analog reference voltage. VREFL0 is the analog reference ground.
Module-stop function	Module-stop state can be set to reduce power consumption.*3

Note 1. For details, see section 8, Clock Generation Circuit.
Note 2. Up to 8 channels can be assigned per scan group.
Note 3. For details, see section 10, Low Power Modes.

Table 36.2 Association between ADC functions and ADC operation mode

Function		SAR mode		Oversampling mode		Hybrid mode		
		Single scan mode	Continuous scan mode	Single scan mode	Continuous scan mode	Single scan mode	Continuous scan mode	Background continuous scan mode
Analog input	Single-ended input	✓	✓	✓	✓	✓	✓	✓
	Differential input	—*1	—*1	✓	✓	✓	✓	✓
	Programmable gain amplifier	✓	✓	✓	✓	✓	✓	✓
	Channel-dedicated sample-and-hold circuit	✓	—	—	—	✓	✓	✓
Diagnosis function/ extended analog function	Disconnection detection assist function	✓	✓	✓	✓	✓	✓	—
	Self-diagnosis	✓	✓	✓	—	✓	—	—
	Internal reference voltage	✓	✓	✓	✓	✓	✓	✓
	Temperature sensor	✓	✓	✓	✓	✓	✓	✓
	D/A converter (DA0 to DA3)	✓	✓	✓	✓	✓	✓	✓
Scanning operation	Group priority operation	✓	✓	—	—	—	—	—
	Multiple A/D converter synchronous operation	✓	✓	✓	✓	✓	✓	✓
Digital calculation	Digital filter function	—	—	✓*2	✓*2	✓*2	✓*2	✓*2
	User's gain/offset adjustment function	✓	✓	✓	✓	✓	✓	✓
	Addition/average function	✓	✓	✓	✓	✓	✓	✓
	Limiter clip function	✓	✓	✓	✓	✓	✓	✓
	Compare match function	✓	✓	✓	✓	✓	✓	✓
FIFO function		✓	✓	✓	✓	✓	✓	✓

Note 1. As an exception, only in the self-diagnosis operation, the Differential input mode is supported.
Note 2. Use of the digital filter function is required when in Oversampling mode and Hybrid mode.

Table 36.3 ADC I/O pins (1 of 2)

Pin name	I/O	Function
AVCC0	Input	Analog block power supply pin

Table 36.3 ADC I/O pins (2 of 2)

Pin name	I/O	Function
AVSS0	Input	Analog block power supply ground pin
VREFH0	Input	Analog reference voltage supply pin
VREFL0	Input	Analog reference ground pin
AN000 to AN028	Input	Analog input pins
PGAIN0 to PGAIN3	Input	Analog input pins for programmable gain amplifier
PGAVSS0 to PGAVSS3	Input	Reference ground input pins for programmable gain amplifier
PGAOUT0 to PGAOUT3	Output	Voltage monitor pins for programmable gain amplifier
SHIN0P to SHIN2P, SHIN4P to SHIN6P	Input	Analog input pins for channel-dedicated sample-and-hold circuit ● For single-ended input: Connect to signal source ● For differntial input: Functions as the non-inverting inputs (+) terminal for differential input
SHIN0N to SHIN2N, SHIN4N to SHIN6N	Input	Analog input pins for channel-dedicated sample-and-hold circuit ● For single-ended input: Connect to signal source ● For differntial input: Functions as the inverting inputs (-) terminal for differential input
ADTRG0, ADTRG1	Input	External trigger input pin for starting A/D conversion, active-low.

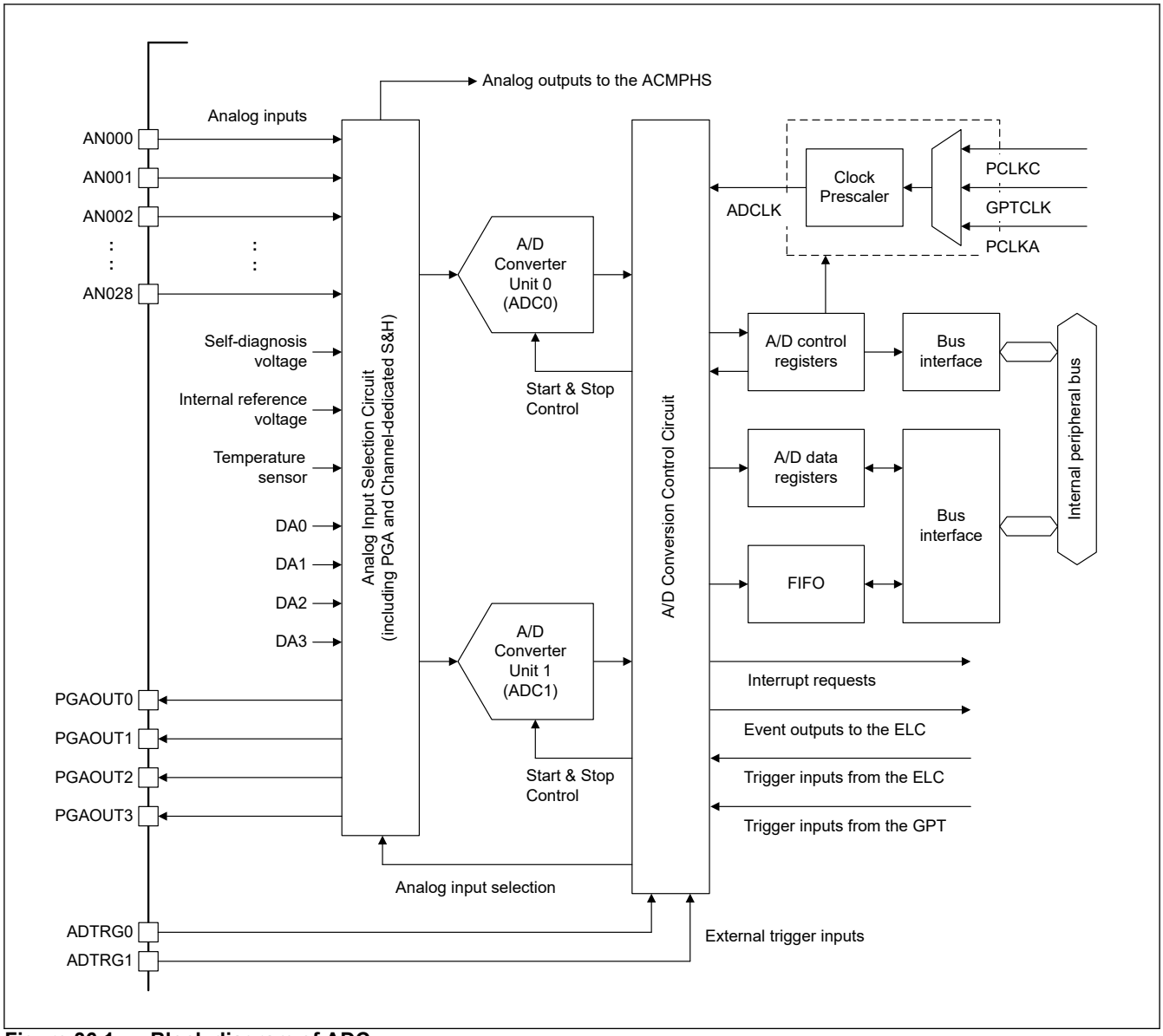


Figure 36.1 Block diagram of ADC

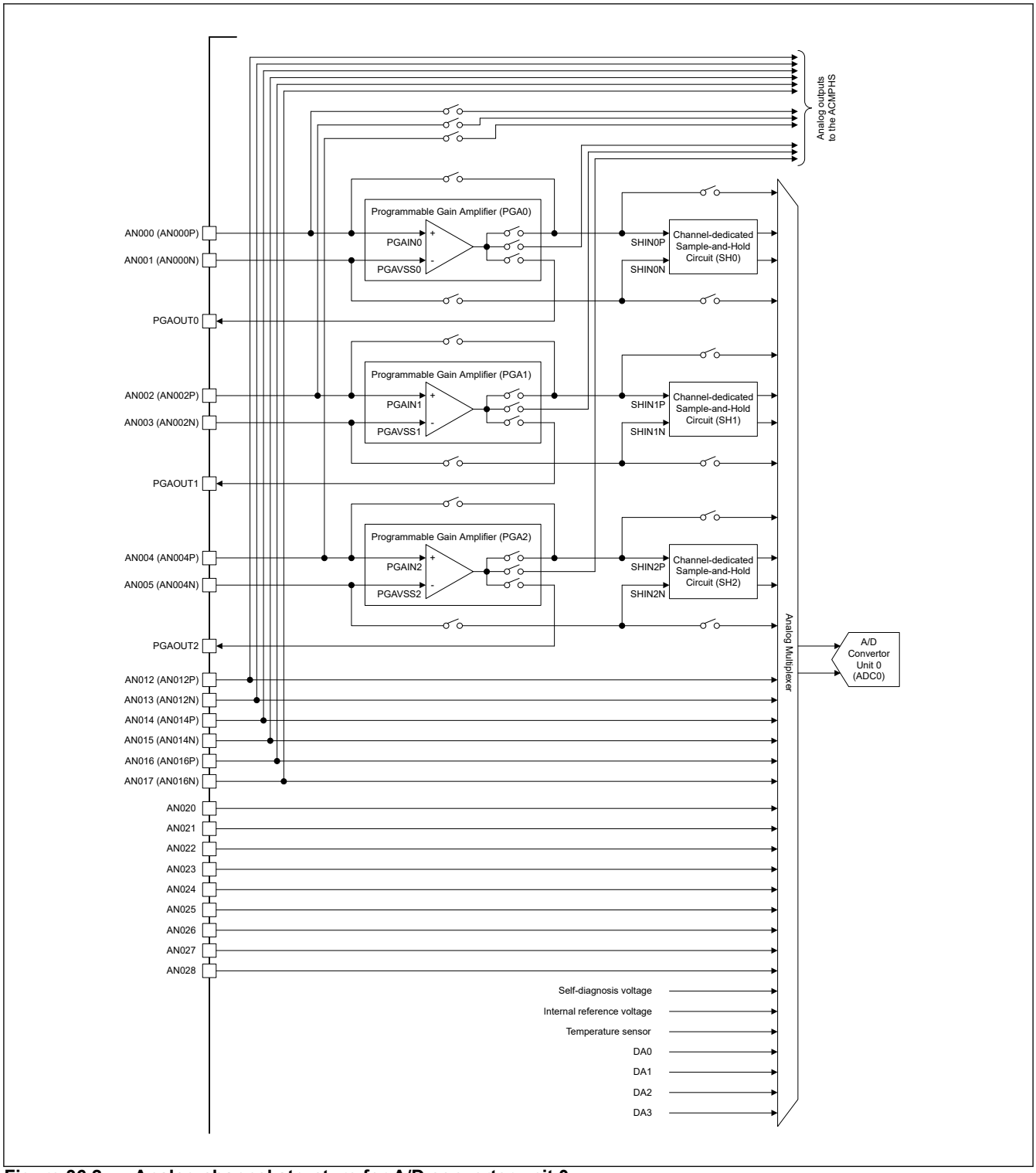


Figure 36.2 Analog channel structure for A/D converter unit 0

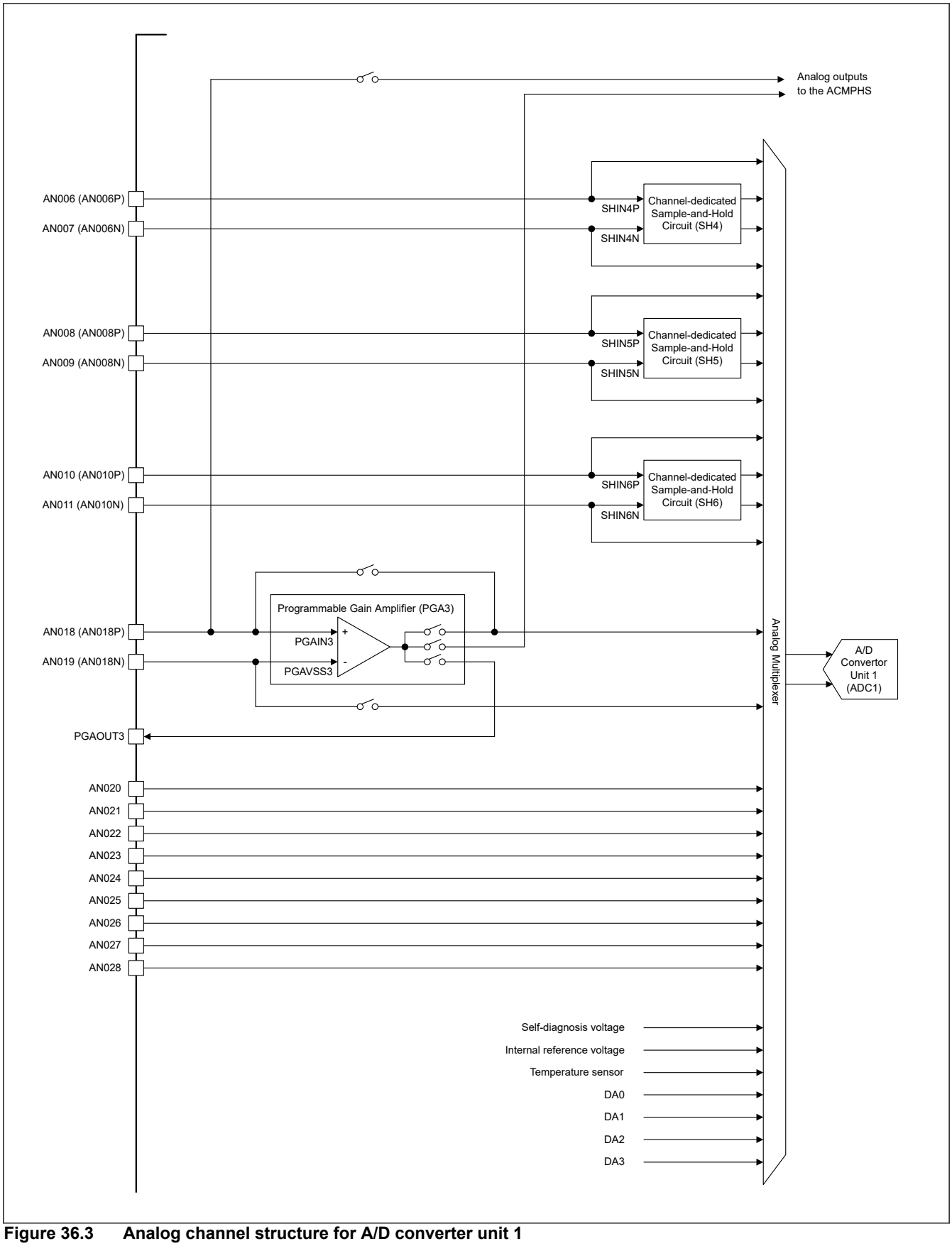


Figure 36.3 Analog channel structure for A/D converter unit 1

Table 36.4 shows the ADC channel configuration of the analog inputs.

Table 36.4 ADC channel configuration of the analog inputs

Analog channel number	Source of analog signal			A/D converter unit 0		A/D converter unit 1	
	Analog input	Programmable Gain Amplifier (PGA)	Channel-dedicated Sample-and-hold Circuit (S&H)*5	Single-ended input	Differential input*6	Single-ended input	Differential input*6
0	AN000 (AN000P)*3	PGAIN0	SHIN0P*3	✓	✓	—	—
1	AN001 (AN000N)*2	PGAVSS0	SHIN0N*4	✓		—	
2	AN002 (AN002P)*3	PGAIN1	SHIN1P*3	✓	✓	—	—
3	AN003 (AN002N)*2	PGAVSS1	SHIN1N*4	✓		—	
4	AN004 (AN004P)*3	PGAIN2	SHIN2P*3	✓	✓	—	—
5	AN005 (AN004N)*2	PGAVSS2	SHIN2N*4	✓		—	
6	AN006 (AN006P)	—	SHIN4P	—	—	✓	✓
7	AN007 (AN006N)	—	SHIN4N	—		✓	
8	AN008 (AN008P)	—	SHIN5P	—	—	✓	✓
9	AN009 (AN008N)	—	SHIN5N	—		✓	
10	AN010 (AN010P)	—	SHIN6P	—	—	✓	✓
11	AN011 (AN010N)	—	SHIN6N	—		✓	
12	AN012 (AN012P)	—	—	✓	✓	—	—
13	AN013 (AN012N)	—	—	✓		—	—
14	AN014 (AN014P)	—	—	✓	✓	—	—
15	AN015 (AN014N)	—	—	✓		—	—
16	AN016 (AN016P)	—	—	✓	✓	—	—
17	AN017 (AN016N)	—	—	✓		—	—
18	AN018 (AN018P)*3	PGAIN3*3	—	—	—	✓	✓
19	AN019 (AN018N)*2	PGAVSS3	—	—		✓*4	
20	AN020*1	—	—	✓	—	✓	—
21	AN021*1	—	—	✓		✓	
22	AN022*1	—	—	✓	—	✓	—
23	AN023*1	—	—	✓		✓	
24	AN024*1	—	—	✓	—	✓	—
25	AN025*1	—	—	✓		✓	
26	AN026*1	—	—	✓	—	✓	—
27	AN027*1	—	—	✓		✓	
28	AN028*1	—	—	✓	—	✓	—

Note 1. Do not perform A/D conversions of the same signal source from both A/D converter unit 0 and unit 1.
Note 2. Only when the Programmable Gain Amplifier (PGA) is disabled and the PGA is set to single-ended input, the ANx can be used for A/D conversion as an analog input channel. When the PGA is enabled, ANx functions as PGAVSS pin. Do not perform A/D conversion of ANx. Regardless of the PGA enable/disable setting, when the PGA is set to pseudo-differential input, A/D conversion of the ANx is not possible.
Note 3. When the Programmable Gain Amplifier (PGA) is used in single-ended input mode or pseudo-differential input mode, and A/D converting the output of the PGA, the A/D conversion channel corresponding to PGAINn (n = 0 to 3) should be set to single-ended input.
Note 4. When using the Programmable Gain Amplifier (PGA) or when the PGA is set to pseudo-differential mode, the SHINxN (inverting input (−)) of channel-dedicated sample-and-hold circuit cannot be used.
Note 5. When the channel-dedicated sample-and-hold circuit is used in single-ended input, the SHINxP (non-inverting input (+)) channel and the SHINxN (inverting input (−)) channel must not be assigned to the same scan group (x = 0 to 2, 4 to 6).

Note 6. When performing A/D conversion with differential-input, set the even-numbered channel as the A/D conversion target. For differential input, the even-numbered channels function as non-inverting inputs (+), and the odd-numbered channels function as inverting inputs (−).

Table 36.5 ADC channel configuration of the extended analog function

Analog channel number	Source of analog signal*1	A/D converter unit 0		A/D converter unit 1	
		Single-ended input*2	Differential input*3	Single-ended input*2	Differential input*3
96	Self-diagnosis	—	✓	—	✓
97	Temperature sensor	✓	—	✓	—
98	Internal reference voltage	✓	—	✓	—
101	D/A converter channel 0	✓	—	✓	—
102	D/A converter channel 1	✓	—	✓	—
103	D/A converter channel 2	✓	—	✓	—
104	D/A converter channel 3	✓	—	✓	—

Note 1. Do not perform A/D conversions of the same signal source from both A/D converter unit 0 and unit 1.
Note 2. The extend analog function (except for the self-diagnosis function) is only supported with the single-ended input. Do not set to the differential input.
Note 3. The self-diagnosis function is only supported with differential input.

36.2 Register Descriptions

36.2.1 System

36.2.1.1 ADCLKENR : A/D Conversion Clock Enable Register

Base address: ADC_B = 0x4017_0000

Offset address: 0x000

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

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

Bit	Symbol	Function	R/W
0	CLKEN	ADCLK Operating Enable 0: Stop ADCLK 1: Supply ADCLK	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

The ADCLKENR controls the supply of the A/D conversion clock (ADCLK).

36.2.1.2 ADCLKSR : A/D Conversion Clock Status Register

Base address: ADC_B = 0x4017_0000

Offset address: 0x004

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

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

Bit	Symbol	Function	R/W
0	CLKSR	ADCLK status 0: ADCLK is stopped 1: ADCLK is in supply	R
31:1	—	These bits are read as 0.	R

The ADCLKSR register indicates the supply status of the A/D conversion clock (ADCLK).

36.2.1.3 ADCLKCR : A/D Conversion Clock Control Register

Base address: ADC_B = 0x4017_0000

Offset address: 0x008

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

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

Bit	Symbol	Function	R/W
1:0	CLKSEL[1:0]	ADCLK Clock Source Select 0 0: Peripheral Module Clock C (PCLKC) 0 1: GPT clock (GPTCLK) 1 0: Peripheral Module Clock A (PCLKA) 0 1: Setting prohibited	R/W
15:2	—	These bits are read as 0. The write value should be 0.	R/W
18:16	DIVR[2:0]	Clock Division Ratio Select 0 0 0: 1/1 0 0 1: 1/2 0 1 0: 1/3 0 1 1: 1/4 1 0 0: 1/5 1 0 1: 1/6 1 1 0: 1/7 1 1 1: 1/8	R/W
31:19	—	These bits are read as 0. The write value should be 0.	R/W

The ADCLKCR register selects the frequency of A/D conversion clock (ADCLK). The frequency of ADCLK should be set within the range specified in the section 46, Electrical Characteristics.

CLKSEL[1:0] bits (ADCLK Clock Source Select)

CLKSEL[1:0] bits select the clock source of A/D conversion clock (ADCLK).

DIVR[2:0] bits (Clock Division Ratio Select)

DIVR[2:0] bits select the division ratio for the clock source selected by CLKSEL[1:0] bits. The clock divided by the clock source selected by CLKSEL[1:0] bits at the ratio set by DIVR[2:0] bits becomes the A/D conversion operating clock (ADCLK).

36.2.1.4 ADSYCR : A/D Converter Synchronous Operation Control Register

Base address: ADC_B = 0x4017_0000

Offset address: 0x00C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADSYDIS1	ADSYDIS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
10:0	ADSYCYC[10:0]	Synchronous Operation Period Cycle These bits should be set in the range from 2 to 1023. The settings of 0 or 1 are prohibited.	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W
16	ADSYDIS0	ADC0 Synchronous Operation Select 0: Enable ADC0 synchronous operation 1: Disable ADC0 synchronous operation	R/W
17	ADSYDIS1	ADC1 Synchronous Operation Select 0: Enable ADC1 synchronous operation 1: Disable ADC1 synchronous operation	R/W
31:18	—	These bits are read as 0. The write value should be 0.	R/W

The ADSYCR register controls the synchronous operation function.

ADSYCYC[10:0] bits (Synchronous Operation Period Cycle)

These bits are selected the synchronous operation period cycles.

These bits set in the range from 2 to 1023 regardless of whether the synchronous operation function is used or not. The settings of 0 or 1 are prohibited.

ADSYDIS0 bit (ADC0 Synchronous Operation Select)

ADSYDIS0 bit selects enable/disable the synchronous operation of ADC0. If the synchronous operation is enabled, the ADC0 operates synchronously with other A/D converter that has synchronous operation enabled.

ADSYDIS1 bit (ADC1 Synchronous Operation Select)

ADSYDIS1 bit selects enables/disables to the synchronous operation of ADC1. If the synchronous operation is enabled, the ADC1 operates synchronously with other A/D converter that has synchronous operation enabled.

36.2.1.5 ADERINTCR : A/D Conversion Error Interrupt Enable Register

Base address: ADC_B = 0x4017_0000

Offset address: 0x020

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADEIE1	ADEIE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ADEIE0	ADC0 A/D Conversion Error Interrupt Enable 0: Disable A/D conversion error interrupt for ADC0 1: Enable A/D conversion error interrupt for ADC0	R/W
1	ADEIE1	ADC1 A/D Conversion Error Interrupt Enable 0: Disable A/D conversion error interrupt for ADC1 1: Enable A/D conversion error interrupt for ADC1	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

The ADERINTCR register controls the enable/disable to the A/D conversion error interrupt.

36.2.1.6 ADOVFINTCR : A/D Conversion Overflow Interrupt Enable Register

Base address: ADC_B = 0x4017_0000

Offset address: 0x024

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADOVFIE1	ADOVFIE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ADOVFIE0	ADC0 A/D Conversion Overflow Interrupt Enable 0: Disable A/D conversion overflow interrupt for ADC0 1: Enable A/D conversion overflow interrupt for ADC0	R/W
1	ADOVFIE1	ADC1 A/D Conversion Overflow Interrupt Enable 0: Disable A/D conversion overflow interrupt for ADC1 1: Enable A/D conversion overflow interrupt for ADC1	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

The ADOVFINTCR register controls the enable/disable to the A/D conversion overflow interrupt.

36.2.1.7 ADCALINTCR : Calibration interrupt Enable Register

Base address: ADC_B = 0x4017_0000

Offset address: 0x028

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CALENDIE1	CALENDIE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
15:0	—	These bits are read as 0. The write value should be 0.	R/W
16	CALENDIE0	ADC0 Calibration End Interrupt Enable 0: Disable Calibration End Interrupt for ADC0 1: Enable Calibration End Interrupt for ADC0	R/W
17	CALENDIE1	ADC1 Calibration End Interrupt Enable 0: Disable Calibration End Interrupt for ADC1 1: Enable Calibration End Interrupt for ADC1	R/W
31:18	—	These bits are read as 0. The write value should be 0.	R/W

The ADCALINTCR register controls the enable/disable to the Calibration End Interrupt.

36.2.1.8 ADMDR : A/D Converter Mode Selection Register

Base address: ADC_B = 0x4017_0000

Offset address: 0x040

Bit position:	31	11	8	3	0
Bit field:	—	ADMD1[3:0]	—	ADMD0[3:0]	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0				

Bit	Symbol	Function	R/W
3:0	ADMD0[3:0]	ADC0 Mode Selection 0x0: SAR mode – Single scan mode 0x1: SAR mode – Continuous scan mode 0x4: Oversampling mode – Single scan mode 0x5: Oversampling mode – Continuous scan mode 0x8: Hybrid mode – Single scan mode 0x9: Hybrid mode – Continuous scan mode 0xA: Hybrid mode – Background continuous scan mode Others: Setting prohibited	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
11:8	ADMD1[3:0]	ADC1 Mode Selection 0x0: SAR mode – Single scan mode 0x1: SAR mode – Continuous scan mode 0x4: Oversampling mode – Single scan mode 0x5: Oversampling mode – Continuous scan mode 0x8: Hybrid mode – Single scan mode 0x9: Hybrid mode – Continuous scan mode 0xA: Hybrid mode – Background continuous scan mode Others: Setting prohibited	R/W

Bit	Symbol	Function	R/W
31:12	—	These bits are read as 0. The write value should be 0.	R/W

The ADMDR register selects the operation mode and the scan mode for each A/D converter.

36.2.2 Scan Group

36.2.2.1 ADGSPCR : A/D Group scan Priority Control Register

Base address: ADC_B = 0x4017_0000

Offset address: 0x044

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	GRP1	LGRRS1	RSCN1	PGS1	—	—	—	—	GRP0	LGRRS0	RSCN0	PGS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PGS0	ADC0 Group Priority Control Setting 0: ADC0 operation is without group priority control 1: ADC0 operation is with group priority control in SAR mode. Setting prohibited when other than SAR mode.	R/W
1	RSCN0	ADC0 Group Priority Control Setting 2 0: Set when PGS0 is set to 0. 1: Set when PGS0 is set to 1.	R/W
2	LGRRS0	ADC0 Group Priority Control Setting 3 0: Set when PGS0 is set to 0. 1: Set when PGS0 is set to 1.	R/W
3	GRP0	ADC0 Group Priority Control Setting 4 0: Set the following case: ● When PGS0 is set to 0 ● When PGS0 is set to 1 and ADC0 is SAR mode – Single scan mode.*1 1: Set when PGS0 is set to 1 and ADC0 is SAR mode – Continuous scan mode.	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
8	PGS1	ADC1 Group Priority Control Setting 0: ADC1 operation is without group priority control 1: ADC1 operation is with group priority control in SAR mode. Setting prohibited when other than SAR mode.	R/W
9	RSCN1	ADC1 Group Priority Control Setting 2 0: Set when PGS1 is set to 0. 1: Set when PGS1 is set to 1.	R/W
10	LGRRS1	ADC1 Group Priority Control Setting 3 0: Set when PGS1 is set to 0. 1: Set when PGS1 is set to 1.	R/W
11	GRP1	ADC1 Group Priority Control Setting 4 0: Set the following case: ● When PGS1 is set to 0 ● When PGS1 is set to 1 and ADC1 is SAR mode – Single scan mode.*1 1: Set when PGS1 is set to 1 and ADC1 is SAR mode – Continuous scan mode.	R/W
31:12	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. To perform the group priority operation by assigning 3 or more scan groups to the A/D converter Unit n (n = 0, 1), GRPn must be set to 1.

The ADGSPCR register controls the group priority operation to each A/D converter. For details on setting this register, see section 36.3.18. Group Priority Operation.

36.2.2.2 ADSGER : Scan Group Enable Register

Base address: ADC_B = 0x4017_0000

Offset address: 0x048

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	SGRE8	SGRE7	SGRE6	SGRE5	SGRE4	SGRE3	SGRE2	SGRE1	SGRE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	SGRE0 to SGRE8	Scan Group n Enable The suffix number of each bit symbol corresponds to the scan group number n. 0: Disable the scan group n 1: Enable the scan group n	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

The ADSGER register selects the enable/disable to each scan group.

SGREn bit (Scan Group n Enable) (n = 0 to 8)

The SGREn bit selects the enable/disable to the scan group n. When SGREn is set to 1, A/D conversion with the scan group n is available. When SGREn is set to 0, A/D conversion with the scan group n is not available, even if the trigger corresponding to the scan group n is input.

36.2.2.3 ADSGCR0 : Scan Group Control Register 0

Base address: ADC_B = 0x4017_0000

Offset address: 0x04C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	SGADS3[1:0]		—	—	—	—	—	—	SGADS2[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	SGADS1[1:0]		—	—	—	—	—	—	SGADS0[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	SGADS0[1:0]	Scan Group 0 A/D Converter Selection 0 0: Select ADC0 0 1: Select ADC1 Others: Setting prohibited	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
9:8	SGADS1[1:0]	Scan Group 1 A/D Converter Selection 0 0: Select ADC0 0 1: Select ADC1 Others: Setting prohibited	R/W

Bit	Symbol	Function	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
17:16	SGADS2[1:0]	Scan Group 2 A/D Converter Selection 0 0: Select ADC0 0 1: Select ADC1 Others: Setting prohibited	R/W
23:18	—	These bits are read as 0. The write value should be 0.	R/W
25:24	SGADS3[1:0]	Scan Group 3 A/D Converter Selection 0 0: Select ADC0 0 1: Select ADC1 Others: Setting prohibited	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

The ADSGCR0 register selects which A/D converter is used for each scan group.

36.2.2.4 ADSGCR1 : Scan Group Control Register 1

Base address: ADC_B = 0x4017_0000

Offset address: 0x050

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	SGADS7[1:0]		—	—	—	—	—	—	SGADS6[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	SGADS5[1:0]		—	—	—	—	—	—	SGADS4[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	SGADS4[1:0]	Scan Group 4 A/D Converter Selection 0 0: Select ADC0 0 1: Select ADC1 Others: Setting prohibited	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
9:8	SGADS5[1:0]	Scan Group 5 A/D Converter Selection 0 0: Select ADC0 0 1: Select ADC1 Others: Setting prohibited	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
17:16	SGADS6[1:0]	Scan Group 6 A/D Converter Selection 0 0: Select ADC0 0 1: Select ADC1 Others: Setting prohibited	R/W
23:18	—	These bits are read as 0. The write value should be 0.	R/W
25:24	SGADS7[1:0]	Scan Group 7 A/D Converter Selection 0 0: Select ADC0 0 1: Select ADC1 Others: Setting prohibited	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

The ADSGCR1 register selects which A/D converter is used for each scan group.

36.2.2.5 ADSGCR2 : Scan Group Control Register 2

Base address: ADC_B = 0x4017_0000

Offset address: 0x054

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SGADS8[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	SGADS8[1:0]	Scan Group 8 A/D Converter Selection 0 0: Select ADC0 0 1: Select ADC1 Others: Setting prohibited	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

The ADSGCR2 register selects which A/D converter is used for each scan group.

36.2.2.6 ADINTCR : Scan End Interrupt Enable Register

Base address: ADC_B = 0x4017_0000

Offset address: 0x05C

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	ADIE8	ADIE7	ADIE6	ADIE5	ADIE4	ADIE3	ADIE2	ADIE1	ADIE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	ADIE0 to ADIE8	Scan Group n Scan End Interrupt Enable The suffix number of each bit symbol corresponds to the scan group number n. 0: Disable scan end interrupt 1: Enable scan end interrupt	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

The ADINTCR register enables/disables the scan end interrupt.

36.2.2.7 ADTRGEXTn : External Trigger Enable Register n (n = 0 to 8)

Base address: ADC_B = 0x4017_0000

Offset address: 0x0C0 + 0x10 × n

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TRGEXT1	TRGEXT0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TRGEXT0	External Trigger Input 0 (ADTRG0) Enable 0: Disable ADTRG0 1: Enable ADTRG0	R/W
1	TRGEXT1	External Trigger Input 1 (ADTRG1) Enable 0: Disable ADTRG1 1: Enable ADTRG1	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

The ADTRGEXTn register enables/disables the External Trigger input 0 and 1 (ADTRG0 and ADTRG1) as the starting conditions for A/D conversion of scan group n.

36.2.2.8 ADTRGELCn : ELC Trigger Enable Register n (n = 0 to 8)

Base address: ADC_B = 0x4017_0000

Offset address: 0x0C4 + 0x10 × n

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	TRGELC5	TRGELC4	TRGELC3	TRGELC2	TRGELC1	TRGELC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	TRGELC0 to TRGELC5	ELC Trigger m Enable The suffix number of each bit symbol corresponds to the ELC trigger number m. 0: Disable ELC Trigger m 1: Enable ELC Trigger m	R/W
31:6	—	These bits are read as 0. The write value should be 0.	R/W

The ADTRGELCn register enables/disables the ELC Trigger m as the starting conditions for A/D conversion of scan group n.

36.2.2.9 ADTRGGPTn : GPT Trigger Enable Register n (n = 0 to 8)

Base address: ADC_B = 0x4017_0000

Offset address: 0x0C8 + 0x10 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	TRGGPTB9	TRGGPTB8	TRGGPTB7	TRGGPTB6	TRGGPTB5	TRGGPTB4	TRGGPTB3	TRGGPTB2	TRGGPTB1	TRGGPTB0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	TRGGPTA9	TRGGPTA8	TRGGPTA7	TRGGPTA6	TRGGPTA5	TRGGPTA4	TRGGPTA3	TRGGPTA2	TRGGPTA1	TRGGPTA0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	TRGGPTA0 to TRGGPTA9	GPT channel m A/D Conversion Starting Request A Enable The suffix number of each bit symbol corresponds to the GPT channel number m. 0: Disable the A/D conversion stating request A from GPT channel m 1: Enable the A/D conversion stating request A from GPT channel m	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
25:16	TRGGPTB0 to TRGGPTB9	GPT channel m A/D Conversion Starting Request B Enable The suffix number of each bit symbol corresponds to the GPT channel number m. 0: Disable the A/D conversion stating request B from GPT channel m 1: Enable the A/D conversion stating request B from GPT channel m	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

The ADTRGGPTn register enables/disables the A/D conversion starting request A/B from GPT channel m as the starting conditions for A/D conversion of scan group n.

36.2.2.10 ADTRGENR : A/D Conversion Start Trigger Enable Register

Base address: ADC_B = 0x4017_0000

Offset address: 0xC08

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	STTRGEN8	STTRGEN7	STTRGEN6	STTRGEN5	STTRGEN4	STTRGEN3	STTRGEN2	STTRGEN1	STTRGEN0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	STTRGEN0 to STTRGEN8	Scan Group n A/D Conversion Start Trigger Enable The suffix number of each bit symbol corresponds to the scan group number n. 0: Disable the A/D conversion start trigger 1: Enable the A/D conversion start trigger	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

The ADTRGENR register enables/disables the trigger from the peripheral modules for starting the A/D conversion of Scan group n. The triggers for each scan group are selected in the ADTRGEXTn, ADTRGELCn, and ADTRGGPTn registers.

STTRGENn bit (Scan Group n A/D Conversion Start Trigger Enable)

The STTRGENn bit selects whether the trigger inputs from the peripheral module, which are selected in the ADTRGEXTn, ADTRGELCn, and ADTRGGPTn registers, is used for the A/D conversion start condition.

When the STTRGENn bit is set to 1, the A/D conversion of scan group n can be started by trigger inputs from the peripheral module.

When the STTRGENn bit is set to 0, the A/D conversion of scan group n cannot be started by trigger inputs from the peripheral module. In this setting, the A/D conversion of scan group n can be only started by writing to the ADSYSTR or ADSTRn registers.

36.2.2.11 ADTRGDLR0 : A/D Conversion Start Trigger Delay Register 0

Base address: ADC_B = 0x4017_0000

Offset address: 0x1C0

Bit position:	31								23								16								7								0
Bit field:	—	—	—	—	—	—	—	—	TRGDLY1[7:0]								—	—	—	—	—	—	—	—	TRGDLY0[7:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	TRGDLY0[7:0]	Scan Group 0 Trigger Input Delay Configuration	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W
23:16	TRGDLY1[7:0]	Scan Group 1 Trigger Input Delay Configuration	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

The ADTRGDLR0 register controls the input delay added to the External trigger inputs, ELC triggers and GPT triggers which are the starting conditions for the A/D conversion of each scan group. The input delay value is given as the register setting value × the cycle of the A/D conversion clock (ADCLK).

36.2.2.12 ADTRGDLR1 : A/D Conversion Start Trigger Delay Register 1

Base address: ADC_B = 0x4017_0000

Offset address: 0x1C4

Bit position:	31								23								16								7								0
Bit field:	—	—	—	—	—	—	—	—	TRGDLY3[7:0]								—	—	—	—	—	—	—	—	TRGDLY2[7:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

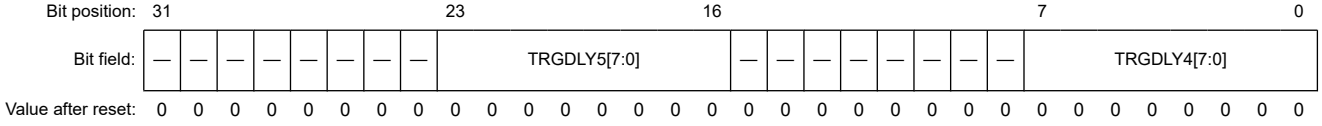
Bit	Symbol	Function	R/W
7:0	TRGDLY2[7:0]	Scan Group 2 Trigger Input Delay Configuration	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W
23:16	TRGDLY3[7:0]	Scan Group 3 Trigger Input Delay Configuration	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

The ADTRGDLR1 register controls the input delay added to the External trigger inputs, ELC triggers and GPT triggers which are the starting conditions for the A/D conversion of each scan group. The input delay value is given as the register setting value × the cycle of the A/D conversion clock (ADCLK).

36.2.2.13 ADTRGDLR2 : A/D Conversion Start Trigger Delay Register 2

Base address: ADC_B = 0x4017_0000

Offset address: 0x1C8



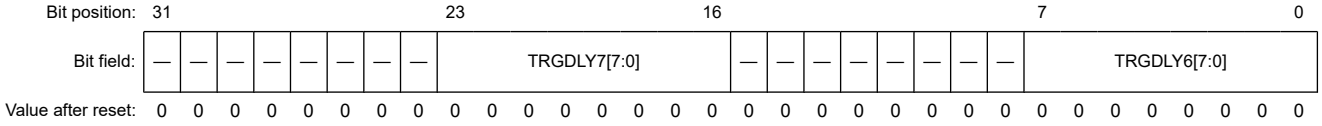
Bit	Symbol	Function	R/W
7:0	TRGDLY4[7:0]	Scan Group 4 Trigger Input Delay Configuration	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W
23:16	TRGDLY5[7:0]	Scan Group 5 Trigger Input Delay Configuration	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

The ADTRGDLR2 register controls the input delay added to the External trigger inputs, ELC triggers and GPT triggers which are the starting conditions for the A/D conversion of each scan group. The input delay value is given as the register setting value × the cycle of the A/D conversion clock (ADCLK).

36.2.2.14 ADTRGDLR3 : A/D Conversion Start Trigger Delay Register 3

Base address: ADC_B = 0x4017_0000

Offset address: 0x1CC



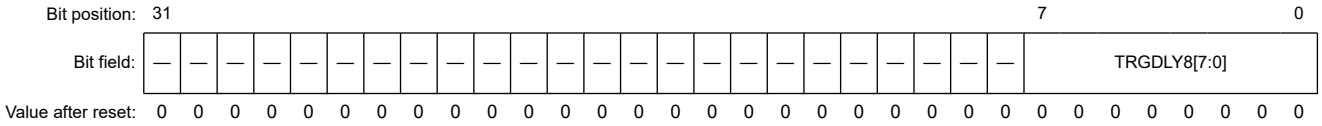
Bit	Symbol	Function	R/W
7:0	TRGDLY6[7:0]	Scan Group 6 Trigger Input Delay Configuration	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W
23:16	TRGDLY7[7:0]	Scan Group 7 Trigger Input Delay Configuration	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

The ADTRGDLR3 register controls the input delay added to the External trigger inputs, ELC triggers and GPT triggers which are the starting conditions for the A/D conversion of each scan group. The input delay value is given as the register setting value × the cycle of the A/D conversion clock (ADCLK).

36.2.2.15 ADTRGDLR4 : A/D Conversion Start Trigger Delay Register 4

Base address: ADC_B = 0x4017_0000

Offset address: 0x1D0



Bit	Symbol	Function	R/W
7:0	TRGDLY8[7:0]	Scan Group 8 Trigger Input Delay Configuration	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

The ADTRGDLR4 register controls the input delay added to the External trigger inputs, ELC triggers and GPT triggers which are the starting conditions for the A/D conversion of each scan group. The input delay value is given as the register setting value × the cycle of the A/D conversion clock (ADCLK).

36.2.3 Virtual Channel

36.2.3.1 ADCHCRn : A/D Conversion Channel Configuration Register n (n = 0 to 36)

Base address: ADC_B = 0x4017_0000

Offset address: 0x600 + 0x10 × n

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

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

Bit	Symbol	Function	R/W
4:0	SGSEL[4:0]	Scan Group Selection 0x00: Not select any scan group 0x01: Select the scan group 0 0x02: Select the scan group 1 0x03: Select the scan group 2 0x04: Select the scan group 3 ⋮ 0x09: Select the scan group 8 Others: Setting prohibited	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
14:8	CNVCS[6:0]	A/D Conversion Channel Selection 0x00: AN000 0x01: AN001 ⋮ 0x1C: AN028 0x60: Self-diagnosis 0x61: Temperature sensor 0x62: Internal reference voltage 0x65: D/A converter channel 0 0x66: D/A converter channel 1 0x67: D/A converter channel 2 0x68: D/A converter channel 3 Others: Setting prohibited	R/W
15	AINMD	Analog Input mode selection*1 0: Single-ended input 1: Differential input	R/W
19:16	SSTSEL[3:0]	Sampling State Table Selection 0x0: Select the sampling state table 0 0x1: Select the sampling state table 1 0x2: Select the sampling state table 2 0x3: Select the sampling state table 3 ⋮ 0xF: Select the sampling state table 15	R/W
31:20	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Select Differential input when self-diagnosis is selected (CNVCS[6:0] = 0x60).

The ADCHCRn register selects the Virtual channel configuration for A/D conversion.

SGSEL[4:0] bits (Scan Group Selection)

The SGSEL[4:0] bits select which scan group to assign the selected channel in CNVCS[6:0]. If SGSEL[4:0] is set to 0x00, the channel selected by CNVCS[6:0] is not available for A/D conversion. If SGSEL[4:0] is set to other than 0x00, the channel selected by CNVCS[6:0] is available for A/D conversion in the scan operation of the scan group corresponding to the setting value.

Up to 8 A/D conversion channels can be assigned per scan group in the SAR mode or Oversampling mode; Up to 4 A/D conversion channels can be assigned per scan group in the Hybrid mode. Do not assign more than specified number of A/D conversion channels per scan group.

CNVCS[6:0] bits (A/D Conversion Channel Selection)

The CNVCS[6:0] bits select the analog signal source for the A/D conversion. The CNVCS[6:0] bits should be set for the channel number corresponding to the Analog Input or the Extended Analog function. For A/D conversion of the analog signal source in Differential input mode, the CNVCS[6:0] bits should be set the even-numbered channel; When the odd-numbered channel is set in Differential input mode, A/D conversion is not guaranteed.

To A/D convert the channel selected by the CNVCS[6:0] bits, it should be assigned to one of the scan groups by the SGSEL[4:0] bits.

AINMD bits (Analog Input mode selection)

The AINMD bit selects the input method for A/D conversion of the analog signal source selected by the CNVCS[6:0] bits. When the AINMD bit is set to 0, A/D conversion is performed in single-ended input. When the AINMD bit is set to 1, A/D conversion is performed in differential input.

If the self-diagnosis channel is selected with the CNVCS[6:0] bits, a differential input should be selected.

SSTSEL[3:0] bits (Sampling State Table Selection)

The SSTSEL[3:0] bits select the sampling state table for A/D conversion of the analog signal source selected by the CNVCS[6:0] bits. The number of sampling states is the value set in the selected sampling state table.

36.2.3.2 ADDOPCRAn : A/D Conversion Data Operation Control A Register n (n = 0 to 36)

Base address: ADC_B = 0x4017_0000

Offset address: 0x604 + 0x10 × n

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

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

Bit	Symbol	Function	R/W
2:0	DFSEL[2:0]	Digital Filter Selection*1 *2 0x0: Not use the digital filter 0x1: Use the 1st digital filter 0x2: Use the 2nd digital filter 0x3: Use the 3rd digital filter 0x4: Use the 4th digital filter Others: Setting prohibited	R/W
15:3	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
19:16	GAINSEL[3:0]	User Gain Table Selection 0x0: Not use the User Gain Table 0x1: Use the User Gain Table 0 0x2: Use the User Gain Table 1 0x3: Use the User Gain Table 2 ⋮ 0x8: Use the User Gain Table 7 Others: Setting prohibited	R/W
23:20	—	These bits are read as 0. The write value should be 0.	R/W
27:24	OFSETSEL[3:0]	User Offset Table Selection 0x0: Not use the user offset table 0x1: Use the user offset table 0 0x2: Use the user offset table 1 0x3: Use the user offset table 2 ⋮ 0x8: Use the user offset table 7 Others: Setting prohibited	R/W
31:28	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Whenever A/D conversion is performed in Oversampling mode or Hybrid mode, the digital filter must be used.
Note 2. Whenever A/D conversion is performed in SAR mode, the digital filter must not be used.

The ADDOPCRAn register is one of the registers that selects the data calculation function for the A/D conversion data of virtual channel n.

DFSEL[2:0] bits (Digital Filter Selection)

The DFSEL[2:0] bits select the digital filter for inputting A/D converted data of virtual channel n. When 000b is set in the DFSEL[2:0] bits, the digital filter is not used. When 001b, 010b, 011b, or 100b is set in the DFSEL[2:0] bits, the digital filter corresponding to the setting of the DFSEL[2:0] bits is selected. The characteristics of the digital filters are specified by ADDFSRm (m = 0, 1) register.

For SAR mode, select 000b for the DFSEL[2:0] bits. For Oversampling mode or Hybrid mode, set the DFSEL[2:0] bits to 001b, 010b, 011b, or 100b.

GAINSEL[3:0] bits (User Gain Table Selection)

The GAINSEL[3:0] bits select the User Gain Table for adjusting the A/D conversion data of virtual channel n.

When the GAINSEL[3:0] bits are set to 0x0, no gain adjustment is performed. When the GAINSEL[3:0] bits are set to any of 0x1 to 0x8, the gain adjustment is performed according to the User Gain Table selected by the GAINSEL[3:0] bits.

OFSETSEL[3:0] bits (User Offset Table Selection)

The OFSETSEL[3:0] bits select the user offset table for adjusting the A/D conversion data of virtual channel n. When the OFSETSEL[3:0] bits are set to 0x0, no offset adjustment is performed. When the OFSETSEL[3:0] bits are set to any of 0x1 to 0x8, the offset adjustment is performed according to the User Offset Table selected by the OFSETSEL[3:0] bits.

36.2.3.3 ADDOPCRBn : A/D Conversion Data Operation Control B Register n (n = 0 to 36)

Base address: ADC_B = 0x4017_0000

Offset address: 0x608 + 0x10 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	CMPTBLE7	CMPTBLE6	CMPTBLE5	CMPTBLE4	CMPTBLE3	CMPTBLE2	CMPTBLE1	CMPTBLE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
1:0	AVEMD[1:0]	Addition/Averaging Mode Selection 0 0: Not use Addition/Averaging mode 0 1: Addition mode 1 0: Averaging mode 1 1: Setting prohibited	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
11:8	ADC[3:0]	Addition/Averaging Times Selection 0x0: 1-time conversion (no addition, same as normal conversion) 0x1: 2-time conversion (1 addition) 0x3: 4-time conversion (3 additions) 0x4: 8-time conversion (7 additions) 0x5: 16-time conversion (15 additions) 0x6: 32-time conversion (31 additions) 0x7: 64-time conversion (63 additions) 0x8: 128-time conversion (127 additions) 0x9: 256-time conversion (255 additions) 0xA: 512-time conversion (511 additions) 0xB: 1024-time conversion (1023 additions) Others: Setting prohibited	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W
23:16	CMPTBLE0 to CMPTBLE7	Compare Match Enable The suffix number of each bit symbol corresponds to the compare match table number m. 0: Disable the compare match with the compare match table m 1: Enable the compare match with the compare match table m	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

The ADDOPCRBn register is one of the registers that selects the data calculation function for the A/D conversion data of virtual channel n.

AVEMD[1:0] bits (Addition/Averaging Mode Selection)

The AVEMD[1:0] bits select the addition mode or the averaging mode for the A/D conversion of virtual channel n. When the AVEMD[1:0] bits are set to 00b, the addition/averaging operation is not performed. When the AVEMD[1:0] bits are set to 01b or 10b, the addition/averaging operation is performed according to the setting value of the ADC[3:0] bits.

ADC[3:0] bits (Addition/Averaging Times Selection)

The ADC[3:0] bits select the number of times of addition/averaging for the A/D conversion of virtual channel n. If the addition mode or the averaging mode is selected in the AVEMD[1:0] bits, the A/D conversion is performed a number of times according to the setting value of the ADC[3:0] bits.

CMPTBLEm bits (Compare Match Enable) (m = 0 to 7)

The CMPTBLEm bits enable/disable the compare match with the compare match table m for the A/D conversion data of virtual channel n.

36.2.3.4 ADDOPCRCn : A/D Conversion Data Operation Control C Register n (n = 0 to 36)

Base address: ADC_B = 0x4017_0000

Offset address: 0x60C + 0x10 × n

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

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

Bit	Symbol	Function	R/W
3:0	LIMTBLS[3:0]	Limiter Clip Table Selection 0x0: Not use the Limiter Clip Table 0x1: Use the Limiter Clip Table 0 0x2: Use the Limiter Clip Table 1 0x3: Use the Limiter Clip Table 2 ⋮ 0x8: Use the Limiter Clip Table 7 Others: Setting prohibited	R/W
15:4	—	These bits are read as 0. The write value should be 0.	R/W
17:16	ADPRC[1:0]	A/D Conversion Data Format Selection 0 0: Store the A/D conversion result as 16-bit data format 0 1: Store the A/D conversion result as 14-bit data format 1 0: Store the A/D conversion result as 12-bit data format 1 1: Store the A/D conversion result as 10-bit data format	R/W
19:18	—	These bits are read as 0. The write value should be 0.	R/W
20	SIGNSEL	A/D Conversion Data Signed/Un-signed Selection 0: Signed data format 1: Un-signed data format	R/W
31:21	—	These bits are read as 0. The write value should be 0.	R/W

The ADDOPCRCn register is one of the registers that selects the data calculation function for the A/D conversion data of virtual channel n.

LIMTBLS[3:0] bits (Limiter Clip Table Selection)

The LIMTBLS[3:0] bits select the limiter clip table for clipping the A/D conversion data of virtual channel n. When the LIMTBLS[3:0] bits are set to 0x0, the A/D conversion data is not clipped. When the LIMTBLS[3:0] bits are set to any of 0x1 to 0x8, the A/D conversion data is clipped according to the limiter clip table selected by the LIMTBLS[3:0] bits.

ADPRC[1:0] bits (A/D Conversion Data Format Selection)

The ADPRC[1:0] bits select the data length of A/D conversion data. The A/D conversion data of virtual channel n is stored to the A/D Data Register or the FIFO with selected data length.

SIGNSEL bit (A/D Conversion Data Signed/Un-signed Selection)

The SIGNSEL bit select the Signed/Unsigned data format of the A/D conversion data. The A/D conversion data of the virtual channel n is stored to the A/D Data Register or the FIFO with selected data format.

Select the Signed data format (SIGNSEL = 0) whenever the A/D conversion of the self-diagnosis channel is performed. Select the Unsigned data format (SIGNSEL = 1) whenever the A/D conversion in SAR mode, unless to convert the self-diagnosis channel, is performed. The signed data format or the unsigned data format is available for the A/D conversion in Oversampling mode or Hybrid mode unless converting to the self-diagnosis channel.

36.2.4 A/D Conversion Configuration

36.2.4.1 ADSGDCRn : Scan Group Diagnosis Function Control Register n (n = 0 to 8)

Base address: ADC_B = 0x4017_0000

Offset address: 0x200 + 0x04 × n

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

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

Bit	Symbol	Function	R/W
2:0	DIAGVAL[2:0]	Self-diagnosis Mode Selection 0 0 0: Set when any self-diagnosis channel are not included. Setting prohibited when any self-diagnosis channels are included. 1 0 0: Self-diagnosis mode 1 1 0 1: Self-diagnosis mode 2 1 1 0: Self-diagnosis mode 3 Others: Setting prohibited	R/W
15:3	—	These bits are read as 0. The write value should be 0.	R/W
16	ADDISEN	Disconnection Detection Assist Enable 0: Disable the disconnection detection assist function 1: Enable the disconnection detection assist function	R/W
19:17	—	These bits are read as 0. The write value should be 0.	R/W
20	ADDISP	Disconnection Detection Assist Mode Selection (for the even-numbered analog channel) 0: Discharge 1: Precharge	R/W
21	ADDISN	Disconnection Detection Assist Mode Selection (for the odd-numbered analog channel) 0: Discharge 1: Precharge	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W
27:24	ADNDIS[3:0]	Disconnection Detection Assist Period 0x0: Setting prohibited when the disconnection detection assist function is enabled 0x1: Setting prohibited 0x2: Setting prohibited Others: The number of states for the discharge or precharge period	R/W
31:28	—	These bits are read as 0. The write value should be 0.	R/W

The ADSGDCRn register controls the diagnosis function at the scanning operation of scan group n.

DIAGVAL[2:0] bits (Self-diagnosis Mode Selection)

The DIAGVAL[2:0] bits select the self-diagnosis mode for A/D converter.

The DIAGVAL[2:0] bits must be set to 000b when the scan group n does not include any virtual channels that selected the self-diagnosis channel. The DIAGVAL[2:0] bits must be set to 100b, 101b or 110b when the scan group n is including the virtual channel that selected the self-diagnosis channel.

ADDISEN bit (Disconnection Detection Assist Enable)

The ADDISEN bit enables/disables the disconnection detection assist function. When the ADDISEN bit is set to 0, the disconnection detection assist operation is not performed. When the ADDISEN bit is set to 1, the disconnection detection assist operation according to the setting of the ADDISP bit and the ADDISN bit is performed at the scanning operation of scan group n.

ADDISP bit / ADDISN bit (Disconnection Detection Assist Mode Selection)

The ADDISP bit and the ADDISN bit select the discharge/precharge operation for the disconnection detection.

The ADDISP bit selects the discharge/precharge operation for the even-numbered analog channel.

The ADDISN bit selects the discharge/precharge operation for the odd-numbered analog channel.

ADNDIS[3:0] bits (Disconnection Detection Assist Period)

The ADNDIS[3:0] bits select the discharge/precharge period. When the disconnection detection assist function is disabled (ADDISEN = 0), the ADNDIS[3:0] bits should be set to 0x0. When the disconnection detection assist function is enabled (ADDISEN = 1), the ADNDIS[3:0] bits should be set the value within 0x3 to 0xF as the number of state for the discharge/precharge period.

36.2.4.2 ADSSTR0 : Sampling State Table Register 0

Base address: ADC_B = 0x4017_0000

Offset address: 0x240

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	SST1[9:0]										—	—	—	—	—	—	SST0[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit	Symbol	Function	R/W
9:0	SST0[9:0]	Sampling State Table 0 These bits set the sampling time in the range from 2 to 1023 states.	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
25:16	SST1[9:0]	Sampling State Table 1 These bits set the sampling time in the range from 2 to 1023 states.	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

The ADSSTR0 register selects the sampling time at the A/D conversion that is using each sampling state table.

36.2.4.3 ADSSTR1 : Sampling State Table Register 1

Base address: ADC_B = 0x4017_0000

Offset address: 0x244

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	SST3[9:0]										—	—	—	—	—	—	SST2[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit	Symbol	Function	R/W
9:0	SST2[9:0]	Sampling State Table 2 These bits set the sampling time in the range from 2 to 1023 states.	R/W

Bit	Symbol	Function	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
25:16	SST3[9:0]	Sampling State Table 3 These bits set the sampling time in the range from 2 to 1023 states.	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

The ADSSTR1 register selects the sampling time at the A/D conversion that is using each sampling state table.

36.2.4.4 ADSSTR2 : Sampling State Table Register 2

Base address: ADC_B = 0x4017_0000

Offset address: 0x248

Bit position:	31						25								16						9								0
Bit field:	—	—	—	—	—	—	SST5[9:0]									—	—	—	—	—	—	SST4[9:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit	Symbol	Function	R/W
9:0	SST4[9:0]	Sampling State Table 4 These bits set the sampling time in the range from 2 to 1023 states.	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
25:16	SST5[9:0]	Sampling State Table 5 These bits set the sampling time in the range from 2 to 1023 states.	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

The ADSSTR2 register selects the sampling time at the A/D conversion that is using each sampling state table.

36.2.4.5 ADSSTR3 : Sampling State Table Register 3

Base address: ADC_B = 0x4017_0000

Offset address: 0x24C

Bit position:	31						25								16						9								0
Bit field:	—	—	—	—	—	—	SST7[9:0]									—	—	—	—	—	—	SST6[9:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit	Symbol	Function	R/W
9:0	SST6[9:0]	Sampling State Table 6 These bits set the sampling time in the range from 2 to 1023 states.	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
25:16	SST7[9:0]	Sampling State Table 7 These bits set the sampling time in the range from 2 to 1023 states.	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

The ADSSTR3 register selects the sampling time at the A/D conversion that is using each sampling state table.

36.2.4.6 ADSSTR4 : Sampling State Table Register 4

Base address: ADC_B = 0x4017_0000

Offset address: 0x250

Bit position:	31						25								16						9								0
Bit field:	—	—	—	—	—	—	SST9[9:0]									—	—	—	—	—	—	SST8[9:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit	Symbol	Function	R/W
9:0	SST8[9:0]	Sampling State Table 8 These bits set the sampling time in the range from 2 to 1023 states.	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
25:16	SST9[9:0]	Sampling State Table 9 These bits set the sampling time in the range from 2 to 1023 states.	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

The ADSSTR4 register selects the sampling time at the A/D conversion that is using each sampling state table.

36.2.4.7 ADSSTR5 : Sampling State Table Register 5

Base address: ADC_B = 0x4017_0000

Offset address: 0x254

Bit position:	31						25								16						9								0
Bit field:	—	—	—	—	—	—	SST11[9:0]									—	—	—	—	—	—	SST10[9:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit	Symbol	Function	R/W
9:0	SST10[9:0]	Sampling State Table 10 These bits set the sampling time in the range from 2 to 1023 states.	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
25:16	SST11[9:0]	Sampling State Table 11 These bits set the sampling time in the range from 2 to 1023 states.	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

The ADSSTR5 register selects the sampling time at the A/D conversion that is using each sampling state table.

36.2.4.8 ADSSTR6 : Sampling State Table Register 6

Base address: ADC_B = 0x4017_0000

Offset address: 0x258

Bit position:	31						25								16						9								0
Bit field:	—	—	—	—	—	—	SST13[9:0]									—	—	—	—	—	—	SST12[9:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit	Symbol	Function	R/W
9:0	SST12[9:0]	Sampling State Table 12 These bits set the sampling time in the range from 2 to 1023 states.	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
25:16	SST13[9:0]	Sampling State Table 13 These bits set the sampling time in the range from 2 to 1023 states.	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

The ADSSTR6 register selects the sampling time at the A/D conversion that is using each sampling state table.

36.2.4.9 ADSSTR7 : Sampling State Table Register 7

Base address: ADC_B = 0x4017_0000

Offset address: 0x25C

Bit position:	31						25								16								9								0
Bit field:	—	—	—	—	—	—	SST15[9:0]								—	—	—	—	—	—	SST14[9:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit	Symbol	Function	R/W
9:0	SST14[9:0]	Sampling State Table 14 These bits set the sampling time in the range from 2 to 1023 states.	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
25:16	SST15[9:0]	Sampling State Table 15 These bits set the sampling time in the range from 2 to 1023 states.	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

The ADSSTR7 register selects the sampling time at the A/D conversion that is using each sampling state table.

36.2.4.10 ADCNVSTR : A/D Conversion State Register

Base address: ADC_B = 0x4017_0000

Offset address: 0x260

Bit position:	31																		13				8		5						0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CST1[5:0]				—	—	CST0[5:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1	1

Bit	Symbol	Function	R/W
5:0	CST0[5:0]	A/D Converter Unit 0 (ADC0) A/D successive approximation time configuration These bits set the successive approximation time in the range from 3 to 63 states.	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
13:8	CST1[5:0]	A/D Converter Unit 1 (ADC1) A/D successive approximation time configuration These bits set the successive approximation time in the range from 3 to 63 states.	R/W
31:14	—	These bits are read as 0. The write value should be 0.	R/W

The ADCNVSTR register specifies the A/D successive approximation time, which required by the A/D converter Unit 0 or Unit 1 to convert the sampled analog input to the digital value, in the number of cycles based on A/D conversion clock (ADCLK). The A/D successive approximation time does not include the sampling time of the analog input.

The A/D successive approximation time must be set to meet the value specified in section 46, Electrical Characteristics. If the successive approximation time is set to a value over the range specified for the electrical characteristics, the A/D conversion result is not guaranteed. In addition, the A/D successive approximation time for each A/D converter should be set to be the same. (Set the same value to CST0[5:0] bits and CST1[5:0] bits).

36.2.5 S&H, PGA and Others

36.2.5.1 ADSHCR0 : Channel-dedicated Sample-and-hold Circuit Control Register 0

Base address: ADC_B = 0x4017_0000

Offset address: 0x280

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	SHMD2	SHMD1	SHMD0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	SHEN2	SHEN1	SHEN0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SHEN0	Channel-dedicated Sample-and-hold Circuit Unit 0 Select 0: Bypass the circuit unit 0 1: Use the circuit unit 0	R/W
1	SHEN1	Channel-dedicated Sample-and-hold Circuit Unit 1 Select 0: Bypass the circuit unit 1 1: Use the circuit unit 1	R/W
2	SHEN2	Channel-dedicated Sample-and-hold Circuit Unit 2 Select 0: Bypass the circuit unit 2 1: Use the circuit unit 2	R/W
15:3	—	These bits are read as 0. The write value should be 0.	R/W
16	SHMD0	Channel-dedicated Sample-and-hold Circuit Unit 0 Input Mode Select 0: Single-ended Input mode 1: Differential Input mode	R/W
17	SHMD1	Channel-dedicated Sample-and-hold Circuit Unit 1 Input Mode Select 0: Single-ended Input mode 1: Differential Input mode	R/W
18	SHMD2	Channel-dedicated Sample-and-hold Circuit Unit 2 Input Mode Select 0: Single-ended Input mode 1: Differential Input mode	R/W
31:19	—	These bits are read as 0. The write value should be 0.	R/W

The ADSHCR0 register controls the channel-dedicated sample-and-hold circuits (unit 0, unit 1 and unit 2) connected to the A/D converter unit 0 (ADC0).

SHENn bit (Channel-dedicated Sample-and-hold Circuit Bypass Select) (n = 0 to 2)

The SHENn bit selects whether to use or bypass the channel-dedicated sample-and-hold circuits unit n.

When the SHENn bit is set to 0, the channel-dedicated sample-and-hold circuits unit n is disabled. If A/D conversion of analog input channel which is connected to the channel-dedicated sample-and-hold circuit unit n is performed, the channel-dedicated sample-and-hold circuit is not used and is bypassed.

When the SHENn bit is set to 1, the channel-dedicated sample-and-hold circuits unit n is enabled. In this case, A/D conversion of analog input channel to which the channel-dedicated sample-and-hold circuit unit n is performed with the channel-dedicated sample-and-hold circuit used.

SHMDn bit (Channel-dedicated Sample-and-hold Circuit Input Mode Select) (n = 0 to 2)

The SHMDn bit selects the input mode of the channel-dedicated sample-and-hold circuits unit n.

When the SHMDn bit is set to 0, the channel-dedicated sample-and-hold circuits unit n is in Single-ended input mode.

When the SHMDn bit is set to 1, the channel-dedicated sample-and-hold circuits unit n is in Differential input mode.

36.2.5.2 ADSHCR1 : Channel-dedicated Sample-and-hold Circuit Control Register 1

Base address: ADC_B = 0x4017_0000

Offset address: 0x28C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	SHMD6	SHMD5	SHMD4
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	SHEN6	SHEN5	SHEN4
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SHEN4	Channel-dedicated Sample-and-hold Circuit Unit 4 Select 0: Bypass the circuit unit 4 1: Use the circuit unit 4	R/W
1	SHEN5	Channel-dedicated Sample-and-hold Circuit Unit 5 Select 0: Bypass the circuit unit 5 1: Use the circuit unit 5	R/W
2	SHEN6	Channel-dedicated Sample-and-hold Circuit Unit 6 Select 0: Bypass the circuit unit 6 1: Use the circuit unit 6	R/W
15:3	—	These bits are read as 0. The write value should be 0.	R/W
16	SHMD4	Channel-dedicated Sample-and-hold Circuit Unit 4 Input Mode Select 0: Single-ended Input mode 1: Differential Input mode	R/W
17	SHMD5	Channel-dedicated Sample-and-hold Circuit Unit 5 Input Mode Select 0: Single-ended Input mode 1: Differential Input mode	R/W
18	SHMD6	Channel-dedicated Sample-and-hold Circuit Unit 6 Input Mode Select 0: Single-ended Input mode 1: Differential Input mode	R/W
31:19	—	These bits are read as 0. The write value should be 0.	R/W

The ADSHCR1 register controls the channel-dedicated sample-and-hold circuits (unit 4, unit 5 and unit 6) connected to the A/D converter unit 1 (ADC1).

SHENn bit (Channel-dedicated Sample-and-hold Circuit Bypass Select) (n = 4 to 6)

The SHENn bit selects whether to use or bypass the channel-dedicated sample-and-hold circuits unit n.

When the SHENn bit is set to 0, the channel-dedicated sample-and-hold circuits unit n is disabled. If A/D conversion of analog input channel which is connected to the channel-dedicated sample-and-hold circuit unit n is performed, the channel-dedicated sample-and-hold circuit is not used and is bypassed.

When the SHENn bit is set to 1, the channel-dedicated sample-and-hold circuits unit n is enabled. In this case, A/D conversion of analog input channel to which the channel-dedicated sample-and-hold circuit unit n is performed with the channel-dedicated sample-and-hold circuit used.

SHMDn bit (Channel-dedicated Sample-and-hold Circuit Input Mode Select) (n = 4 to 6)

The SHMDn bit selects the input mode of the channel-dedicated sample-and-hold circuits unit n.

When the SHMDn bit is set to 0, the channel-dedicated sample-and-hold circuits unit n is in Single-ended input mode.

When the SHMDn bit is set to 1, the channel-dedicated sample-and-hold circuits unit n is in Differential input mode.

36.2.5.3 ADSHSTR0 : Channel-dedicated Sample-and-hold Circuit State Register 0

Base address: ADC_B = 0x4017_0000

Offset address: 0x288

Bit position:	31													18	16								7							0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	SHHST[2:0]		—	—	—	—	—	—	—	—	SHSST[7:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

Bit	Symbol	Function	R/W
7:0	SHSST[7:0]	Channel-dedicated Sample-and-hold Circuit Unit 0 to 2 Sampling Time Setting These bits set the sampling time in the range from 4 to 255 states.	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W
18:16	SHHST[2:0]	Channel-dedicated Sample-and-hold Circuit Unit 0 to 2 Hold Mode Switching Time Setting These bits set the hold mode switching time in the range from 2 to 7 states.	R/W
31:19	—	These bits are read as 0. The write value should be 0.	R/W

The ADSHSTR0 register specifies the sampling time and the hold mode switching time for the channel-dedicated sample-and-hold circuits (unit 0, unit 1, and unit 2) connected to the A/D converter unit 0 (ADC0). The value set in this register should be set to meet the value specified in section 46, Electrical Characteristics.

36.2.5.4 ADSHSTR1 : Channel-dedicated Sample-and-hold Circuit State Register 1

Base address: ADC_B = 0x4017_0000

Offset address: 0x294

Bit position:	31													18	16								7							0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	SHHST[2:0]		—	—	—	—	—	—	—	—	SHSST[7:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

Bit	Symbol	Function	R/W
7:0	SHSST[7:0]	Channel-dedicated Sample-and-hold Circuit Unit 4 to 6 Sampling Time Setting These bits set the sampling time in the range from 4 to 255 states.	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W
18:16	SHHST[2:0]	Channel-dedicated Sample-and-hold Circuit Unit 4 to 6 Hold Mode Switching Time Setting These bits set the hold mode switching time in the range from 2 to 7 states.	R/W
31:19	—	These bits are read as 0. The write value should be 0.	R/W

The ADSHSTR1 register specifies the sampling time and the hold mode switching time for the channel-dedicated sample-and-hold circuits (unit 4, unit 5, and unit 6) connected to the A/D converter unit 1 (ADC1). The value set in this register should be set to meet the value specified in section 46, Electrical Characteristics.

36.2.5.5 ADPGACRn : Programmable Gain Amplifier Control Register n (n = 0 to 3)

Base address: ADC_B = 0x4017_0000

Offset address: 0x2C0 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	PGAGAIN[3:0]				—	—	PGADG[1:0]		—	—	—	PGAGEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	PGAENAMP	PGASEL1	PGADEN	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0/1*1	0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
1	PGADEN	PGA Unit n Input Mode Select*1 0: Single-ended Input mode 1: Pseudo-differential Input mode	R/W
2	PGASEL1	PGA Unit n Amplifier Output Enable 0: Not output the signal in a path through the PGA 1: Output the signal in a path through the PGA	R/W
3	PGAENAMP	PGA Unit n Enable 0: Disable the PGA 1: Enable the PGA	R/W
15:4	—	These bits are read as 0. The write value should be 0.	R/W
16	PGAGEN	PGA Unit n Gain Setting Enable 0: Disable gain setting 1: Enable gain setting	R/W
19:17	—	These bits are read as 0. The write value should be 0.	R/W
21:20	PGADG[1:0]	PGA Unit n Differential Input Gain Setting When PGA is Single-ended Input mode, set 00b. When PGA is Pseudo-differential Input mode, select the following values: 0 0: × 1.500 0 1: × 2.333 1 0: × 4.000 1 1: × 5.667	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
27:24	PGAGAIN[3:0]	PGA Unit n Gain Setting 0x0: × 2.000 (PGA is Single-ended Input mode) Setting prohibited (PGA is Pseudo-differential Input mode) 0x1: × 2.500 (PGA is Single-ended Input mode) × 1.500 (PGA is Pseudo-differential Input mode) 0x2: × 2.667 (PGA is Single-ended Input mode) Setting prohibited (PGA is Pseudo-differential Input mode) 0x3: × 2.857 (PGA is Single-ended Input mode) Setting prohibited (PGA is Pseudo-differential Input mode) 0x4: × 3.077 (PGA is Single-ended Input mode) Setting prohibited (PGA is Pseudo-differential Input mode) 0x5: × 3.333 (PGA is Single-ended Input mode) × 2.333 (PGA is Pseudo-differential Input mode) 0x6: × 3.636 (PGA is Single-ended Input mode) Setting prohibited (PGA is Pseudo-differential Input mode) 0x7: × 4.000 (PGA is Single-ended Input mode) Setting prohibited (PGA is Pseudo-differential Input mode) 0x8: × 4.444 (PGA is Single-ended Input mode) Setting prohibited (PGA is Pseudo-differential Input mode) 0x9: × 5.000 (PGA is Single-ended Input mode) × 4.000 (PGA is Pseudo-differential Input mode) 0xA: × 5.714 (PGA is Single-ended Input mode) Setting prohibited (PGA is Pseudo-differential Input mode) 0xB: × 6.667 (PGA is Single-ended Input mode) × 5.667 (PGA is Pseudo-differential Input mode) 0xC: × 8.000 (PGA is Single-ended Input mode) Setting prohibited (PGA is Pseudo-differential Input mode) 0xD: × 10.000 (PGA is Single-ended Input mode) Setting prohibited (PGA is Pseudo-differential Input mode) 0xE: × 13.333 (PGA is Single-ended Input mode) Setting prohibited (PGA is Pseudo-differential Input mode) 0xF: Setting prohibited	R/W
31:28	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. The initial value after reset depends on the user setting of Option Function Select Register 1 in Option Setting Memory.

The ADPGACRn register controls the Programmable Gain Amplifier Unit n.

PGADEN bit (PGA Unit n Input Mode Select)

The PGADEN bit selects the Single-ended Input mode or the Pseudo-differential Input mode for PGA unit n.

PGASEL1 bit (PGA Unit n Amplifier Output Enable)

The PGASEL1 bit controls the output of the PGA unit n.

PGAENAMP bit (PGA Unit n Enable)

The PGAENAMP bit controls the power-on of the amplifier in the PGA unit n. When 1 is set to the PGAENAMP bit, the amplifier in the PGA unit n is enabled.

PGAGEN bit (PGA Unit n Gain Setting Enable)

The PGAGEN bit enables or disables the gain setting for PGA unit n.

PGADG[1:0] bits (PGA Unit n Differential Input Gain Setting)

The PGADG[1:0] bits specify the gain of the amplifier in the PGA Unit n in Pseudo-differential Input mode. These bits are used in combination with PGAGAIN[3:0].

PGAGAIN[3:0] bits (PGA Unit n Gain Setting)

The PGAGAIN[3:0] bits specify the gain of the amplifier in the PGA Unit n. In Pseudo-differential Input mode (PGADEN = 1 and PGAGEN = 1), these bits are used in combination with PGADG[1:0].

36.2.5.6 ADPGAMONCR : Programable Gain Amp Monitor Output Control Register

Base address: ADC_B = 0x4017_0000

Offset address: 0x300

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MONSEL3	MONSEL2	MONSEL1	MONSEL0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
2:0	PGAMON[2:0]	PGA Monitor Signal Selection 0x0: Not select monitor signal (Hi-Z) 0x1: PGA output Others: Setting prohibited	R/W
15:3	—	These bits are read as 0. The write value should be 0.	R/W
16	MONSEL0	PGA Unit 0 Monitor Output Enable 0: Disable monitor output 1: Enable monitor output	R/W
17	MONSEL1	PGA Unit 1 Monitor Output Enable 0: Disable monitor output 1: Enable monitor output	R/W
18	MONSEL2	PGA Unit 2 Monitor Output Enable 0: Disable monitor output 1: Enable monitor output	R/W
19	MONSEL3	PGA Unit 3 Monitor Output Enable 0: Disable monitor output 1: Enable monitor output	R/W
31:20	—	These bits are read as 0. The write value should be 0.	R/W

The ADPGAMONCR register controls the PGA monitor output function for the PGA analog input and output signals.

PGAMON[2:0] bits (PGA Monitor Signal Selection)

The PGAMON[2:0] bits select which analog signal is output from the I/O port pin. This setting is applied to all of the PGA units.

MONSELn bit (PGA Monitor Output Setting) (n = 0 to 3)

The MONSELn bit enables or disables the monitor output function of the PGA unit n. When the MONSELn bit is set to 0, no analog signal for the PGA unit n is monitored out. When the MONSELn bit is set to 1, the analog signal selected by PGAMON[2:0] for the PGA unit n can be monitored on the I/O port pin.

Note: It is also required to configure the I/O ports. For details, see section 18, I/O Ports.

36.2.5.7 ADREFCR : Internal Reference Voltage Monitor Enable Register

Base address: ADC_B = 0x4017_0000

Offset address: 0x320

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

Bit	Symbol	Function	R/W
0	VDE	Internal Reference Voltage A/D Conversion Select 0: Disable A/D conversion of internal reference voltage 1: Enable A/D conversion of internal reference voltage	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

The ADREFCR register controls the function of the internal reference voltage circuit. When the A/D conversion of the internal reference voltage is performed, the VDE bit must be set to 1.

36.2.6 Digital Filter

36.2.6.1 ADDFSRn : A/D Converter Digital Filter Selection Register n (n = 0, 1)

Base address: ADC_B = 0x4017_0000

Offset address: 0x340 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	DFSEL3[1:0]		—	—	—	—	—	—	DFSEL2[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	DFSEL1[1:0]		—	—	—	—	—	—	DFSEL0[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	DFSEL0[1:0]	A/D Converter Unit n the 1st Digital Filter Characteristic Selection 0 1: Sinc filter 1 0: Minimum phase filter Others: Setting prohibited	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
9:8	DFSEL1[1:0]	A/D Converter Unit n the 2nd Digital Filter Characteristic Selection 0 1: Sinc filter 1 0: Minimum phase filter Others: Setting prohibited	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
17:16	DFSEL2[1:0]	A/D Converter Unit n the 3rd Digital Filter Characteristic Selection 0 1: Sinc filter 1 0: Minimum phase filter Others: Setting prohibited	R/W
23:18	—	These bits are read as 0. The write value should be 0.	R/W
25:24	DFSEL3[1:0]	A/D Converter Unit n the 4th Digital Filter Characteristic Selection 0 1: Sinc filter 1 0: Minimum phase filter Others: Setting prohibited	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

The ADDFSRn register selects the characteristics of the digital filter for the A/D converter unit n. The selection of the digital filter to be used at A/D conversion is specified by the ADDOPCRAm register (m = 0 to 36). If the digital filter function is used with prohibited values set to the ADDFSRn.DFSELx[1:0] (x = 0 to 3) bits, the A/D conversion characteristics and A/D conversion results are not guaranteed.

36.2.7 Self-calibration

36.2.7.1 ADCALSTCR : A/D Converter Self-calibration State Register

Base address: ADC_B = 0x4017_0000

Offset address: 0x264

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

Bit	Symbol	Function	R/W
9:0	CALADSST[9:0]	A/D Converter Self-calibration Sampling Time Configuration These bits set the sampling time in the range from 2 to 1023 states.	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
21:16	CALADCST[5:0]	A/D Converter Self-calibration Successive Approximation Time Configuration. These bits set the successive approximation time in the range from 3 to 63 states.	R/W
31:22	—	These bits are read as 0. The write value should be 0.	R/W

The ADCALSTCR register specifies the sampling and the A/D successive approximation times at the self-calibration operation for all A/D converter (ADC0, ADC1). The sampling time and successive approximation time should be set by the number of clock cycles based on the A/D conversion clock (ADCLK) to meet the values specified in section 46, Electrical Characteristics.

36.2.7.2 ADCALSHCR : Channel-dedicated Sample-and-hold Circuit Self-calibration State Register

Base address: ADC_B = 0x4017_0000

Offset address: 0x2B0

Bit position:	31													18		16								7								0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	CALSHHST[2:0]		—	—	—	—	—	—	—	—	CALSHSST[7:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	

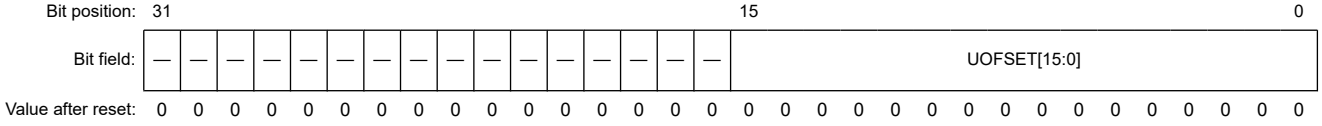
Bit	Symbol	Function	R/W
7:0	CALSHSST[7:0]	Channel-dedicated Sample-and-hold Circuit Self-calibration Sampling Time Configuration These bits set the sampling time in the range from 5 to 255 states.	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W
18:16	CALSHHST[2:0]	Channel-dedicated Sample-and-hold Circuit Self-calibration Hold Mode Switching Time Configuration These bits set the hold mode switching time in the range from 2 to 7 states.	R/W
31:19	—	These bits are read as 0. The write value should be 0.	R/W

The ADCALSHCR register specifies the sampling time and the hold mode switching time at the self-calibration operation for all channel-dedicated sample-and-hold circuits (SH0 to SH2, SH4 to SH6). The sampling time and the hold mode switching time should be set by the number of clock cycles based on the A/D conversion clock (ADCLK) to meet the values specified in section 46, Electrical Characteristics.

36.2.7.3 ADUOFTRn : User Offset Table Register n (n = 0 to 7)

Base address: ADC_B = 0x4017_0000

Offset address: 0x360 + 0x04 × n



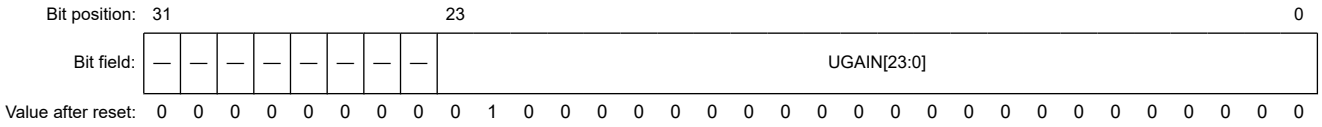
Bit	Symbol	Function	R/W
15:0	UOFSET[15:0]	User Offset Table n 0x7FFF: + 32767 0x7FFE: + 32766 ⋮ 0x0002: + 2 0x0001: + 1 0x0000: 0 (without user offset) 0xFFFF: - 1 0xFFFE: - 2 ⋮ 0x8001: - 32767 0x8000: - 32768	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

The ADUOFTRn register specifies the offset value in the User Offset Table n.

36.2.7.4 ADUGTRn : User Gain Table Register n (n = 0 to 7)

Base address: ADC_B = 0x4017_0000

Offset address: 0x380 + 0x04 × n



Bit	Symbol	Function	R/W
23:0	UGAIN[23:0]	User Gain Table n These bits set the gain value to be multiplied by the A/D conversion result. UGAIN[23:22]: Integer part of gain UGAIN[21:0]: Fractional part of gain	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

The ADUGTRn register specifies the gain value in the user gain table n.

36.2.8 Limiter Clip Function

36.2.8.1 ADLIMINTCR : Limiter Clip Interrupt Enable Register

Base address: ADC_B = 0x4017_0000

Offset address: 0x3A0

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	LIMIE8	LIMIE7	LIMIE6	LIMIE5	LIMIE4	LIMIE3	LIMIE2	LIMIE1	LIMIE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	LIMIE0 to LIMIE8	Limiter Clip Interrupt n Enable bit The suffix number of each bit symbol corresponds to the limiter clip interrupt number n. 0: Disable the limiter clip interrupt n 1: Enable the limiter clip interrupt n	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

The ADLIMINTCR register enables/disables the limiter clip interrupt n.

36.2.8.2 ADLIMTRn : Limiter Clip Table Register n (n = 0 to 7)

Base address: ADC_B = 0x4017_0000

Offset address: 0x3A4 + 0x04 × n

Bit position:	31	16	15	0
Bit field:	LIMU[15:0]		LIML[15:0]	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
15:0	LIML[15:0]	Limiter clip table n : Lower-side limit value	R/W
31:16	LIMU[15:0]	Limiter clip table n : Upper-side limit value	R/W

The ADLIMTRn register specifies the lower- and upper-side limit value of the limiter clip table n.

36.2.8.3 ADLIMGRSR : Limiter Clip Scan Group Status Register

Base address: ADC_B = 0x4017_0000

Offset address: 0xD28

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	LIMGRF8	LIMGRF7	LIMGRF6	LIMGRF5	LIMGRF4	LIMGRF3	LIMGRF2	LIMGRF1	LIMGRF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	LIMGRF0 to LIMGRF8	Scan Group n Limiter Clip Flag The suffix number of each bit symbol corresponds to the scan group number n. 0: Limiter clip for scan group n is not detected 1: Limiter clip for scan group n is detected	R
31:9	—	These bits are read as 0.	R

The ADLIMGRSR register indicates whether the limiter clip occurred in the scanning operation for scan group n. Each flag can be cleared in the ADLIMGRSCR.

36.2.8.4 ADLIMCHSR0 : Limiter Clip Channel Status Register 0

Base address: ADC_B = 0x4017_0000

Offset address: 0xD2C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	LIMCHF28	LIMCHF27	LIMCHF26	LIMCHF25	LIMCHF24	LIMCHF23	LIMCHF22	LIMCHF21	LIMCHF20	LIMCHF19	LIMCHF18	LIMCHF17	LIMCHF16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	LIMCHF15	LIMCHF14	LIMCHF13	LIMCHF12	LIMCHF11	LIMCHF10	LIMCHF9	LIMCHF8	LIMCHF7	LIMCHF6	LIMCHF5	LIMCHF4	LIMCHF3	LIMCHF2	LIMCHF1	LIMCHF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
28:0	LIMCHF0 to LIMCHF28	Analog Channel n: Limiter Clip Flag The suffix number of each bit symbol corresponds to the analog channel number n. 0: Limiter clip is not detected 1: Limiter clip is detected	R
31:29	—	These bits are read as 0.	R

The ADLIMCHSR0 register indicates whether the limiter clip occurred when the A/D conversion for the analog channel n is performed. Each flag can be cleared in ADLIMCHSCR0.

36.2.8.5 ADLIMEXSR : Extended Analog Limiter Clip Status Register

Base address: ADC_B = 0x4017_0000

Offset address: 0xD38

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	LIMEXF8	LIMEXF7	LIMEXF6	LIMEXF5	—	—	LIMEXF2	LIMEXF1	LIMEXF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	LIMEXF0	Self-diagnosis Channel: Limiter Clip Flag 0: Limiter clip is not detected 1: Limiter clip is detected	R

Bit	Symbol	Function	R/W
1	LIMEXF1	Temperature Sensor Channel: Limiter Clip Flag 0: Limiter clip is not detected 1: Limiter clip is detected	R
2	LIMEXF2	Internal Reference Voltage Channel: Limiter Clip Flag 0: Limiter clip is not detected 1: Limiter clip is detected	R
4:3	—	These bits are read as 0.	R
5	LIMEXF5	D/A Converter 0 Channel: Limiter Clip Flag 0: Limiter clip is not detected 1: Limiter clip is detected	R
6	LIMEXF6	D/A Converter 1 Channel: Limiter Clip Flag 0: Limiter clip is not detected 1: Limiter clip is detected	R
7	LIMEXF7	D/A Converter 2 Channel: Limiter Clip Flag 0: Limiter clip is not detected 1: Limiter clip is detected	R
8	LIMEXF8	D/A Converter 3 Channel: Limiter Clip Flag 0: Limiter clip is not detected 1: Limiter clip is detected	R
31:9	—	These bits are read as 0.	R

The ADLIMEXSR register indicates whether the limiter clip occurred when the A/D conversion for the extended analog function (which channel no. is 96 to 98, 101 to 104) is performed. Each flag can be cleared in the ADLIMEXSCR.

36.2.8.6 ADLIMGRSCR : Limiter Clip Scan Group Status Clear Register

Base address: ADC_B = 0x4017_0000

Offset address: 0xD3C

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	LIMGRC8	LIMGRC7	LIMGRC6	LIMGRC5	LIMGRC4	LIMGRC3	LIMGRC2	LIMGRC1	LIMGRC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	LIMGRC0 to LIMGRC8	Scan Group n Limiter Clip Flag Clear The suffix number of each bit symbol corresponds to the scan group number n. 0: No effect 1: ADLIMGRSR.LIMGRFn is cleared	W
31:9	—	The write value should be 0.	W

The ADLIMGRSCR register clears the limiter clip flag for scan group n (ADLIMGRSR.LIMGRFn).

36.2.8.7 ADLIMCHSCR0 : Limiter Clip Channel Status Clear Register 0

Base address: ADC_B = 0x4017_0000

Offset address: 0xD40

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	LIMCHC28	LIMCHC27	LIMCHC26	LIMCHC25	LIMCHC24	LIMCHC23	LIMCHC22	LIMCHC21	LIMCHC20	LIMCHC19	LIMCHC18	LIMCHC17	LIMCHC16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	LIMCHC15	LIMCHC14	LIMCHC13	LIMCHC12	LIMCHC11	LIMCHC10	LIMCHC9	LIMCHC8	LIMCHC7	LIMCHC6	LIMCHC5	LIMCHC4	LIMCHC3	LIMCHC2	LIMCHC1	LIMCHC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
28:0	LIMCHC0 to LIMCHC28	Analog Channel n Limiter Clip Flag Clear bit The suffix number of each bit symbol corresponds to the analog channel number n. 0: No effect 1: ADLIMCHSR0.LIMCHFn is cleared	W
31:29	—	The write value should be 0.	W

The ADLIMCHSCR0 register clears the limiter clip flag for the analog channel n (ADLIMCHSR0.LIMCHFn).

36.2.8.8 ADLIMEXSCR : Extended Analog Limiter Clip Status Clear Register

Base address: ADC_B = 0x4017_0000

Offset address: 0xD4C

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	LIMEXF8	LIMEXF7	LIMEXF6	LIMEXF5	—	—	LIMEXF2	LIMEXF1	LIMEXF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	LIMEXF0	Self-diagnosis Channel: Limiter Clip Flag Clear 0: No effect 1: ADLIMEXSR.LIMEXF0 is cleared	W
1	LIMEXF1	Temperature Sensor Channel: Limiter Clip Flag Clear 0: No effect 1: ADLIMEXSR.LIMEXF1 is cleared	W
2	LIMEXF2	Internal Reference Voltage Channel: Limiter Clip Flag Clear 0: No effect 1: ADLIMEXSR.LIMEXF2 is cleared	W
4:3	—	The write value should be 0.	W
5	LIMEXF5	D/A Converter 0 Channel: Limiter Clip Flag Clear 0: No effect 1: ADLIMEXSR.LIMEXF5 is cleared	W
6	LIMEXF6	D/A Converter 1 Channel: Limiter Clip Flag Clear 0: No effect 1: ADLIMEXSR.LIMEXF6 is cleared	W

Bit	Symbol	Function	R/W
7	LIMEXF7	D/A Converter 2 Channel: Limiter Clip Flag Clear 0: No effect 1: ADLIMEXSR.LIMEXF7 is cleared	W
8	LIMEXF8	D/A Converter 3 Channel: Limiter Clip Flag Clear 0: No effect 1: ADLIMEXSR.LIMEXF8 is cleared	W
31:9	—	The write value should be 0.	W

The ADLIMEXSCR register clears the limiter clip flag for the extended analog function.

36.2.9 Compare Match Function

36.2.9.1 ADCMPENR : Compare Match Enable Register

Base address: ADC_B = 0x4017_0000

Offset address: 0x400

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	CMPEN7	CMPEN6	CMPEN5	CMPEN4	CMPEN3	CMPEN2	CMPEN1	CMPEN0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	CMPEN0 to CMPEN7	Compare Match n Enable The suffix number of each bit symbol corresponds to the compare match number n. 0: Disable the compare match n 1: Enable the compare match n	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

The ADCMPENR register enables/disables the compare match n.

36.2.9.2 ADCMPINTCR : Compare Match Interrupt Enable Register

Base address: ADC_B = 0x4017_0000

Offset address: 0x404

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	CMPIE3	CMPIE2	CMPIE1	CMPIE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	CMPIE0 to CMPIE3	Compare Match Interrupt n Enable The suffix number of each bit symbol corresponds to the compare match interrupt number n. 0: Disable the compare match interrupt n 1: Enable the compare match interrupt n	R/W
31:4	—	These bits are read as 0. The write value should be 0.	R/W

The ADCMPINTCR register enables/disables the compare match interrupt n.

36.2.9.3 ADCCMPCRn : Composite Compare Match Configuration Register n (n = 0, 1)

Base address: ADC_B = 0x4017_0000

Offset address: 0x408 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	CCMPTBL7	CCMPTBL6	CCMPTBL5	CCMPTBL4	CCMPTBL3	CCMPTBL2	CCMPTBL1	CCMPTBL0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
1:0	CCMPCND[1:0]	Composite Compare Match Condition Selection 0 0: Logical disjunction (OR) conditions 0 1: Logical conjunction (AND) conditions 1 0: Logical exclusive disjunction (EXOR) conditions 1 1: Setting prohibited	R/W
15:2	—	These bits are read as 0. The write value should be 0.	R/W
23:16	CCMPTBL0 to CCMPTBL7	Composite Compare Match Condition Table Selection 0: Not use the compare match table m 1: Use the compare match table m	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

The ADCCMPCRn register configures the conditions for the composite compare match interrupt n.

CCMPCND[1:0] bits (Composite Compare Match Condition Selection)

The CCMPCND[1:0] bits select the generating condition for the composite compare match interrupt n.

CCMPTBLm bits (Composite Compare Match Condition Table Selection) (m = 0 to 7)

The CCMPTBLm bits select whether to use or not to use the compare match table m as the generating conditions of the composite compare match interrupt.

36.2.9.4 ADCMPMDR0 : Compare Match Mode Selection Register 0

Base address: ADC_B = 0x4017_0000

Offset address: 0x448

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	CMPMD3[1:0]		—	—	—	—	—	—	CMPMD2[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	CMPMD1[1:0]		—	—	—	—	—	—	CMPMD0[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	CMPMD0[1:0]	Compare Match 0 : Match Mode Selection 0 0: Generate the match event when high-side level or more 0 1: Generate the match event when low-side level or less 1 0: Generate the match event when high-side level or more, or low-side level or less 1 1: Generate the match event when low-side level or more and high-side level or less	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
9:8	CMPMD1[1:0]	Compare Match 1 : Match Mode Selection 0 0: Generate the match event when high-side level or more 0 1: Generate the match event when low-side level or less 1 0: Generate the match event when high-side level or more, or low-side level or less 1 1: Generate the match event when low-side level or more and high-side level or less	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
17:16	CMPMD2[1:0]	Compare Match 2 : Match Mode Selection 0 0: Generate the match event when high-side level or more 0 1: Generate the match event when low-side level or less 1 0: Generate the match event when high-side level or more, or low-side level or less 1 1: Generate the match event when low-side level or more and high-side level or less	R/W
23:18	—	These bits are read as 0. The write value should be 0.	R/W
25:24	CMPMD3[1:0]	Compare Match 3 : Match Mode Selection 0 0: Generate the match event when high-side level or more 0 1: Generate the match event when low-side level or less 1 0: Generate the match event when high-side level or more, or low-side level or less 1 1: Generate the match event when low-side level or more and high-side level or less	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

The ADCMPMDR0 register selects the generating condition of each compare match event.

36.2.9.5 ADCMPMDR1 : Compare Match Mode Selection Register 1

Base address: ADC_B = 0x4017_0000

Offset address: 0x44C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	CMPMD7[1:0]		—	—	—	—	—	—	CMPMD6[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	CMPMD5[1:0]		—	—	—	—	—	—	CMPMD4[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	CMPMD4[1:0]	Compare Match 4 : Match Mode Selection 0 0: Generate the match event when high-side level or more 0 1: Generate the match event when low-side level or less 1 0: Generate the match event when high-side level or more, or low-side level or less 1 1: Generate the match event when low-side level or more and high-side level or less	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
9:8	CMPMD5[1:0]	Compare Match 5 : Match Mode Selection 0 0: Generate the match event when high-side level or more 0 1: Generate the match event when low-side level or less 1 0: Generate the match event when high-side level or more, or low-side level or less 1 1: Generate the match event when low-side level or more and high-side level or less	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
17:16	CMPMD6[1:0]	Compare Match 6 : Match Mode Selection 0 0: Generate the match event when high-side level or more 0 1: Generate the match event when low-side level or less 1 0: Generate the match event when high-side level or more, or low-side level or less 1 1: Generate the match event when low-side level or more and high-side level or less	R/W
23:18	—	These bits are read as 0. The write value should be 0.	R/W
25:24	CMPMD7[1:0]	Compare Match 7 : Match Mode Selection 0 0: Generate the match event when high-side level or more 0 1: Generate the match event when low-side level or less 1 0: Generate the match event when high-side level or more, or low-side level or less 1 1: Generate the match event when low-side level or more and high-side level or less	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

The ADCMPMDR1 register selects the generating condition of each compare match event.

36.2.9.6 ADCMPTBRn : Compare Match Table Register n (n = 0 to 7)

Base address: ADC_B = 0x4017_0000

Offset address: 0x458 + 0x04 × n

Bit position:	31															16	15														0
Bit field:	CMPTBH[15:0]																CMPTBL[15:0]														
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	CMPTBL[15:0]	Compare Match Table n : Low-side level Set CMPTBH > CMPTBL	R/W
31:16	CMPTBH[15:0]	Compare Match Table n : High-side level Set CMPTBH > CMPTBL	R/W

The ADCMPTBRn register specifies the low-side level and the high-side level for the compare match table n.

36.2.9.7 ADCMPTBSR : Compare Match Table Status Register

Base address: ADC_B = 0x4017_0000

Offset address: 0xD00

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	CMPTBF7	CMPTBF6	CMPTBF5	CMPTBF4	CMPTBF3	CMPTBF2	CMPTBF1	CMPTBF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	CMPTBF0 to CMPTBF7	Compare Match Table n Match Flag The suffix number of each bit symbol corresponds to the compare match table number n. 0: Match event with compare match table n is not detected 1: Match event with compare match table n is detected	R
31:8	—	These bits are read as 0.	R

The ADCMPTBSR register indicates whether the match event with the Compare Match Table n occurred during the A/D conversion. Each flag can be cleared in the ADCMPTBSCR.

36.2.9.8 ADCMPTBSCR : Compare Match Table Status Clear Register

Base address: ADC_B = 0x4017_0000

Offset address: 0xD04

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	CMPTBC7	CMPTBC6	CMPTBC5	CMPTBC4	CMPTBC3	CMPTBC2	CMPTBC1	CMPTBC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	CMPTBC0 to CMPTBC7	Compare Match Table n: Match Flag Clear The suffix number of each bit symbol corresponds to the compare match table number n. 0: No effect 1: ADCMPTBSR.CMPTBFn is cleared	W
31:8	—	The write value should be 0.	W

The ADCMPTBSCR register clears the match flag for the compare match n (ADCMPTBSR.CMPTBFn).

36.2.9.9 ADCMPCHSR0 : Compare Match Channel Status Register 0

Base address: ADC_B = 0x4017_0000

Offset address: 0xD08

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	CMPCHF28	CMPCHF27	CMPCHF26	CMPCHF25	CMPCHF24	CMPCHF23	CMPCHF22	CMPCHF21	CMPCHF20	CMPCHF19	CMPCHF18	CMPCHF17	CMPCHF16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CMPCHF15	CMPCHF14	CMPCHF13	CMPCHF12	CMPCHF11	CMPCHF10	CMPCHF9	CMPCHF8	CMPCHF7	CMPCHF6	CMPCHF5	CMPCHF4	CMPCHF3	CMPCHF2	CMPCHF1	CMPCHF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
28:0	CMPCHF0 to CMPCHF28	Analog Channel n: Compare Match Flag The suffix number of each bit symbol corresponds to the analog channel number n. 0: Compare match is not detected 1: Compare match is detected	R
31:29	—	These bits are read as 0.	R

The ADCMPCHSR0 register indicates whether the compare match event for the Analog Channel n is detected. Each flag can be cleared in ADCMPCHSCR0.

36.2.9.10 ADCMPEXSR : Extended Analog Compare Match Status Register

Base address: ADC_B = 0x4017_0000

Offset address: 0xD14

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	CMPEXF8	CMPEXF7	CMPEXF6	CMPEXF5	—	—	CMPEXF2	CMPEXF1	CMPEXF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CMPEXF0	Self-diagnosis Channel: Compare Match Flag 0: Compare match is not detected 1: Compare match is detected	R
1	CMPEXF1	Temperature Sensor Channel: Compare Match Flag 0: Compare match is not detected 1: Compare match is detected	R
2	CMPEXF2	Internal Reference Voltage Channel: Compare Match Flag 0: Compare match is not detected 1: Compare match is detected	R
4:3	—	These bits are read as 0.	R
5	CMPEXF5	D/A Converter 0 Channel: Compare Match Flag 0: Compare match is not detected 1: Compare match is detected	R

Bit	Symbol	Function	R/W
6	CMPEXF6	D/A Converter 1 Channel: Compare Match Flag 0: Compare match is not detected 1: Compare match is detected	R
7	CMPEXF7	D/A Converter 2 Channel: Compare Match Flag 0: Compare match is not detected 1: Compare match is detected	R
8	CMPEXF8	D/A Converter 3 Channel: Compare Match Flag 0: Compare match is not detected 1: Compare match is detected	R
31:9	—	These bits are read as 0.	R

The ADCMPEXSR register indicates whether the compare match event for the extended analog function channel is detected. Each flag can be cleared in ADCMPEXSCR.

36.2.9.11 ADCMPCHSCR0 : Compare Match Channel Status Clear Register 0

Base address: ADC_B = 0x4017_0000

Offset address: 0xD18

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	CMPCHC28	CMPCHC27	CMPCHC26	CMPCHC25	CMPCHC24	CMPCHC23	CMPCHC22	CMPCHC21	CMPCHC20	CMPCHC19	CMPCHC18	CMPCHC17	CMPCHC16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CMPCHC15	CMPCHC14	CMPCHC13	CMPCHC12	CMPCHC11	CMPCHC10	CMPCHC9	CMPCHC8	CMPCHC7	CMPCHC6	CMPCHC5	CMPCHC4	CMPCHC3	CMPCHC2	CMPCHC1	CMPCHC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
28:0	CMPCHC0 to CMPCHC28	Analog Channel n: Compare Match Flag Clear bit The suffix number of each bit symbol corresponds to the analog channel number n. 0: No effect 1: ADCMPCHSR0.CMPCHFn is cleared	W
31:29	—	The write value should be 0.	W

The ADCMPCHSCR0 register clears the compare match flag for the Analog Channel n (ADCMPCHSR0.CMPCHFn).

36.2.9.12 ADCMPEXSCR : Extended Analog Compare Match Status Clear Register

Base address: ADC_B = 0x4017_0000

Offset address: 0xD24

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	CMPEXC8	CMPEXC7	CMPEXC6	CMPEXC5	—	—	CMPEXC2	CMPEXC1	CMPEXC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CMPEXC0	Self-diagnosis Channel: Compare Match Flag Clear 0: No effect 1: ADCMPEXSR.CMPEXF0 is cleared	W
1	CMPEXC1	Temperature Sensor Channel: Compare Match Flag Clear 0: No effect 1: ADCMPEXSR.CMPEXF1 is cleared	W
2	CMPEXC2	Internal Reference Voltage Channel: Compare Match Flag Clear 0: No effect 1: ADCMPEXSR.CMPEXF2 is cleared	W
4:3	—	The write value should be 0.	W
5	CMPEXC5	D/A Converter 0 Channel: Compare Match Flag Clear 0: No effect 1: ADCMPEXSR.CMPEXF5 is cleared	W
6	CMPEXC6	D/A Converter 1 Channel: Compare Match Flag Clear 0: No effect 1: ADCMPEXSR.CMPEXF6 is cleared	W
7	CMPEXC7	D/A Converter 2 Channel: Compare Match Flag Clear 0: No effect 1: ADCMPEXSR.CMPEXF7 is cleared	W
8	CMPEXC8	D/A Converter 3 Channel: Compare Match Flag Clear 0: No effect 1: ADCMPEXSR.CMPEXF8 is cleared	W
31:9	—	The write value should be 0.	W

The ADCMPEXSCR register clears the compare match flag for the Extended Analog Function Channel (ADCMPEXSR.CMPEXFn).

36.2.10 Start and Stop Control of A/D Conversion

36.2.10.1 ADCALSTR : A/D Converter Self-calibration Start Register

Base address: ADC_B = 0x4017_0000

Offset address: 0xC00

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	ADCALST1[2:0]			—	—	—	—	—	ADCALST0[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	ADCALST0[2:0]*1	A/D Converter Unit 0 (ADC0) Self-calibration Start Control [Function of each bit] b0: Start bit of the internal circuit calibration b1: Start bit of the gain and offset calibration b2: Start bit of the channel-dedicated sample-and-hold circuit calibration [Write value] 0: No effect (not start the calibration) 1: Start the calibration	W
7:3	—	The write value should be 0.	W

Bit	Symbol	Function	R/W
10:8	ADCALST1[2:0]*1	A/D Converter Unit 1 (ADC1) Self-calibration Start Control [Function of each bit] b0: Start bit of the internal circuit calibration b1: Start bit of the gain and offset calibration b2: Start bit of the channel-dedicated sample-and-hold circuit calibration [Write value] 0: No effect (not start the calibration) 1: Start the calibration	W
31:11	—	The write value should be 0.	W

Note 1. To perform a calibration operation, write 1 to each of the calibration start bits simultaneously.

The ADCALSTR register controls the start of the self-calibration for each A/D converter.

ADCALSTm[2:0] bits (A/D converter Unit m (ADCm) Self-calibration Start Control bits) (m = 0, 1)

The ADCALSTm[2:0] bits control the start of the self-calibration operation of the A/D converter Unit m. When any bit of ADCALSTm[2:0] is set to 1, the self-calibration operation corresponding to each bit is started. When multiple bits of ADCALSTm[2:0] are set to 1 simultaneously, each self-calibration operation is performed in the order as following:

1. Internal circuit calibration
2. A/D converter gain/offset calibration
3. Channel-specific sample and hold circuit gain/offset calibration

Note: The self-calibration operation corresponding to the bits set to 0 in ADCALSTm[2:0] is skipped.

The ADCALSTm[2:0] bits must be written when all A/D converters are stopped (ADSR.ADACTm = 0 and ADSR.CALACTm = 0);

Writing to the ADCALSTm[2:0] bits are prohibited when A/D converters are operating.

36.2.10.2 ADSYSTR : A/D Conversion Synchronous Software Start Register

Base address: ADC_B = 0x4017_0000

Offset address: 0xC10

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	ADSYST8	ADSYST7	ADSYST6	ADSYST5	ADSYST4	ADSYST3	ADSYST2	ADSYST1	ADSYST0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	ADSYST0 to ADSYST8	Scan Group n: A/D Conversion start The suffix number of each bit symbol corresponds to the scan group number n. 0: No effect 1: Start the A/D conversion of scan group n	W
31:9	—	The write value should be 0.	W

The ADSYSTR register controls the start of the A/D conversion of scan group n. This register is used to start the A/D conversions of several scan groups simultaneously by software.

ADSYSTn bit (Scan Group n: A/D Conversion start) (n = 0 to 8)

The ADSYSTn bit controls the start of the A/D conversion of scan group n. When the ADSYSTn bit is set to 1, the A/D conversion of scan group n is started. Setting 0 to the ADSYSTn bit has no effect on the operation. Set 1 simultaneously to the ADSYSTn bit of the scan group to start A/D conversion simultaneously.

36.2.10.3 ADSTRn : A/D Conversion Software Start Register n (n = 0 to 8)

Base address: ADC_B = 0x4017_0000

Offset address: 0xC20 + 0x04 × n

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

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

Bit	Symbol	Function	R/W
0	ADST	Scan Group n A/D Conversion Start 0: No effect 1: Start the A/D conversion of scan group n	W
31:1	—	The write value should be 0.	W

The ADSTRn register controls the start of the A/D conversion of scan group n. This register is used to start the A/D conversion of a scan group by software.

If A/D conversions of several scan groups are to be started simultaneously by software, the ADSYSTR register should be used.

ADST bit (Scan Group n A/D Conversion Start)

The ADST bit controls the start of the A/D conversion of scan group n. When the ADST bit is set to 1, the A/D conversion of the scan group n is started. Setting 0 to the ADST bit has no effect on the operation.

36.2.10.4 ADSTOPR : A/D Conversion Stop Register

Base address: ADC_B = 0x4017_0000

Offset address: 0xC60

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	ADST OP1	—	—	—	—	—	—	—	ADST OP0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ADSTOP0	A/D Converter Unit 0 Force Stop 0: No effect 1: Force stop the operation of A/D converter unit 0	W
7:1	—	The write value should be 0.	W

Bit	Symbol	Function	R/W
8	ADSTOP1	A/D Converter Unit 1 Force Stop 0: No effect 1: Force stop the operation of A/D converter unit 1	W
31:9	—	The write value should be 0.	W

The ADSTOPR register forces each A/D converter to stop the operation. When the A/D converter is stopped with this register, the A/D conversion results with that A/D converter are not guaranteed.

ADSTOP0 bit (A/D Converter Unit 0 Force Stop)

The ADSTOP0 bit forces the A/D converter unit 0 to stop the A/D conversion operation. Setting 0 to the ADSTOP0 bit has no effect on the operation. When the ADSTOP0 bit is set to 1, the A/D converter unit 0 is forced to stop the operation. If the A/D converter unit 0 is forced to stop, the A/D conversion results with the A/D converter unit 0 are not guaranteed.

ADSTOP1 bit (A/D Converter Unit 1 Force Stop)

The ADSTOP1 bit forces the A/D converter unit 1 to stop the A/D conversion operation. Setting 0 to the ADSTOP1 bit has no effect on the operation. When the ADSTOP1 bit is set to 1, the A/D converter unit 1 is forced to stop the operation. If the A/D converter unit 1 is forced to stop, the A/D conversion results with the A/D converter unit 1 are not guaranteed.

36.2.11 Status Registers

36.2.11.1 ADSR : A/D Conversion Status Register

Base address: ADC_B = 0x4017_0000

Offset address: 0xC80

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CALACT1	CALACT0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADACT1	ADACT0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ADACT0	A/D Converter Unit 0 (ADC0) A/D Conversion Status 0: ADC0 is not in A/D conversion 1: ADC0 is in A/D conversion	R
1	ADACT1	A/D Converter Unit 1 (ADC1) A/D Conversion Status 0: ADC1 is not in A/D conversion 1: ADC1 is in A/D conversion	R
15:2	—	These bits are read as 0.	R
16	CALACT0	A/D Converter Unit 0 (ADC0) : Calibration Status 0: ADC0 is not in the calibration operation 1: ADC0 is in the calibration operation	R
17	CALACT1	A/D Converter Unit 1 (ADC1) : Calibration Status 0: ADC1 is not in the calibration operation 1: ADC1 is in the calibration operation	R
31:18	—	These bits are read as 0.	R

The ADSR register indicates the status for each A/D converter operation.

36.2.11.2 ADGRSR : Scan Group Status Register

Base address: ADC_B = 0x4017_0000

Offset address: 0xC84

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	ACTGR8	ACTGR7	ACTGR6	ACTGR5	ACTGR4	ACTGR3	ACTGR2	ACTGR1	ACTGR0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	ACTGR0 to ACTGR8*1	Scan Group n Status The suffix number of each bit symbol corresponds to the scan group number n. 0: Scan group n is idle 1: Scan group n is in the scanning operation	R
31:9	—	These bits are read as 0.	R

Note 1. In the group priority operation, if the scanning operation of a low priority group is interrupted, the ACTGRn bit of the corresponding scan group is set to 1.

The ADGRSR register indicates the operating status for each scan group.

36.2.11.3 ADSCANENDSR : Scan End Status Register

Base address: ADC_B = 0x4017_0000

Offset address: 0xD50

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	SCENDF8	SCENDF7	SCENDF6	SCENDF5	SCENDF4	SCENDF3	SCENDF2	SCENDF1	SCENDF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	SCENDF0 to SCENDF8*1	Scan Group n Scan End Flag The suffix number of each bit symbol corresponds to the scan group number n. 0: Scan group n has not been scanned 1: End of scan for scan group n is detected	R
31:9	—	These bits are read as 0.	R

Note 1. If the A/D conversion operation is stopped with the ADSTOPR register, the SCENDFn bit of the scan group in which the scan operation was stopped is not changed. (It is not set to 1)

The ADSCANENDSR register indicates whether the scanning operation for each scan group has been done. Each flag can be cleared in ADSCANENDSCR register.

36.2.11.4 ADSCANENDSCR : Scan End Status Clear Register

Base address: ADC_B = 0x4017_0000

Offset address: 0xD54

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	SCENDC8	SCENDC7	SCENDC6	SCENDC5	SCENDC4	SCENDC3	SCENDC2	SCENDC1	SCENDC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	SCENDC0 to SCENDC8	Scan Group n Scan End Flag Clear The suffix number of each bit symbol corresponds to the scan group number n. 0: No effect 1: ADSCANENDSR.SCENDFn is cleared	W
31:9	—	The write value should be 0.	W

The ADSCANENDSCR register clears the scan end flag for the scan group n.

36.2.11.5 ADERSR : A/D Conversion Error Status Register

Base address: ADC_B = 0x4017_0000

Offset address: 0xC88

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADERF1	ADERF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ADERF0	A/D Converter Unit 0 (ADC0) Error Flag 0: Error is not detected 1: Error is detected	R
1	ADERF1	A/D Converter Unit 1 (ADC1) Error Flag 0: Error is not detected 1: Error is detected	R
31:2	—	These bits are read as 0.	R

The ADERSR register indicates whether the operating error has occurred in each A/D converter. Each flag can be cleared in ADERSCR register.

The A/D conversion data is not guaranteed in the A/D converter operation with an error detected.

36.2.11.6 ADERSCR : A/D Conversion Error Status Clear Register

Base address: ADC_B = 0x4017_0000

Offset address: 0xC8C

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADERCLR1	ADERCLR0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ADERCLR0	A/D Converter Unit 0 Error Flag Clear 0: No effect 1: ADERSR.ADERF0 is cleared	W
1	ADERCLR1	A/D Converter Unit 1 Error Flag Clear 0: No effect 1: ADERSR.ADERF1 is cleared	W
31:2	—	The write value should be 0.	W

The ADERSCR register clears the error flag for A/D converter unit 0 or unit 1.

36.2.11.7 ADCALENDSR : A/D Converter Calibration End Status Register

Base address: ADC_B = 0x4017_0000

Offset address: 0xC98

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CALENDF1	CALENDF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CALENDF0	A/D Converter Unit 0 Calibration End flag 0: End of the calibration is not detected 1: End of the calibration is detected	R
1	CALENDF1	A/D Converter Unit 1 Calibration End flag 0: End of the calibration is not detected 1: End of the calibration is detected	R
31:2	—	These bits are read as 0.	R

The ADCALENDSR register indicates the end of the calibration operation for the A/D converter unit 0 or unit 1. Each flag can be cleared in ADCALENDSCR register.

36.2.11.8 ADCALENDSCR : A/D Converter Calibration End Status Clear Register

Base address: ADC_B = 0x4017_0000

Offset address: 0xC9C

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CALENDC1	CALENDC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CALENDC0	A/D Converter Unit 0 Calibration End Flag Clear 0: No effect 1: ADCALENDSR.CALENDF0 is cleared	W
1	CALENDC1	A/D Converter Unit 1 Calibration End Flag Clear 0: No effect 1: ADCALENDSR.CALENDF1 is cleared	W
31:2	—	The write value should be 0.	W

The ADCALENDSCR register clears the Calibration End flag for the A/D converter unit 0 or unit 1.

36.2.11.9 ADOVFERSR : A/D Conversion Overflow Error Status Register

Base address: ADC_B = 0x4017_0000

Offset address: 0xCA0

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADOVFEF1	ADOVFEF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ADOVFEF0	A/D Converter Unit 0 (ADC0) Overflow Error Flag 0: ADC0 overflow error is not detected 1: ADC0 overflow error is detected	R
1	ADOVFEF1	A/D Converter Unit 1 (ADC1) Overflow Error Flag 0: ADC1 overflow error is not detected 1: ADC1 overflow error is detected	R
31:2	—	These bits are read as 0.	R

The ADOVFERSR register indicates whether the overflow error has occurred in an A/D conversion using A/D converter unit 0 or unit 1. Each flag can be cleared in ADOVFERSCR register.

36.2.11.10 ADOVFCHSR0 : A/D Conversion Overflow Channel Status Register 0

Base address: ADC_B = 0x4017_0000

Offset address: 0xCA4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	OVFCHF28	OVFCHF27	OVFCHF26	OVFCHF25	OVFCHF24	OVFCHF23	OVFCHF22	OVFCHF21	OVFCHF20	OVFCHF19	OVFCHF18	OVFCHF17	OVFCHF16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	OVFCHF15	OVFCHF14	OVFCHF13	OVFCHF12	OVFCHF11	OVFCHF10	OVFCHF9	OVFCHF8	OVFCHF7	OVFCHF6	OVFCHF5	OVFCHF4	OVFCHF3	OVFCHF2	OVFCHF1	OVFCHF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
28:0	OVFCHF0 to OVFCHF28	Analog Channel n: Overflow Flag The suffix number of each bit symbol corresponds to the analog channel number n. 0: Overflow is not detected 1: Overflow is detected	R
31:29	—	These bits are read as 0.	R

The ADOVFCHSR0 register indicates whether the overflow has occurred in the A/D conversion of the analog channel n. Each flag can be cleared in ADOVFCHSCR0 register.

36.2.11.11 ADOVFEXSR : Extended Analog A/D Conversion Overflow Status Register

Base address: ADC_B = 0x4017_0000

Offset address: 0xCB0

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	OVFEXF8	OVFEXF7	OVFEXF6	OVFEXF5	—	—	OVFEXF2	OVFEXF1	OVFEXF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	OVFEXF0	Self-diagnosis Channel: Overflow Flag 0: Overflow is not detected 1: Overflow is detected	R
1	OVFEXF1	Temperature Sensor Channel: Overflow Flag 0: Overflow is not detected 1: Overflow is detected	R
2	OVFEXF2	Internal Reference Voltage Channel: Overflow Flag 0: Overflow is not detected 1: Overflow is detected	R
4:3	—	These bits are read as 0.	R
5	OVFEXF5	D/A Converter 0 Channel: Overflow Flag 0: Overflow is not detected 1: Overflow is detected	R

Bit	Symbol	Function	R/W
6	OVFEXF6	D/A Converter 1 Channel: Overflow Flag 0: Overflow is not detected 1: Overflow is detected	R
7	OVFEXF7	D/A Converter 2 Channel: Overflow Flag 0: Overflow is not detected 1: Overflow is detected	R
8	OVFEXF8	D/A Converter 3 Channel: Overflow Flag 0: Overflow is not detected 1: Overflow is detected	R
31:9	—	These bits are read as 0.	R

The ADOVFEXSR register indicates whether the overflow occurred when the A/D conversion for the extended analog function is performed. Each flag can be cleared in ADOVFEXSCR.

36.2.11.12 ADOVFERSCR : A/D Conversion Overflow Error Status Clear Register

Base address: ADC_B = 0x4017_0000

Offset address: 0xCB4

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADOVFEC1	ADOVFEC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ADOVFEC0	A/D Converter Unit 0 (ADC0) Overflow Error Flag Clear 0: No effect 1: ADOVFERSR.ADOVFEF0 is cleared	W
1	ADOVFEC1	A/D Converter Unit 1 (ADC1) Overflow Error Flag Clear 0: No effect 1: ADOVFERSR.ADOVFEF1 is cleared	W
31:2	—	The write value should be 0.	W

The ADOVFERSCR register clears the Overflow Error flag for the A/D converter unit 0 or unit 1.

36.2.11.13 ADOVFCHSCR0 : A/D Conversion Overflow Channel Status Clear Register 0

Base address: ADC_B = 0x4017_0000

Offset address: 0xCB8

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	OVFCHC28	OVFCHC27	OVFCHC26	OVFCHC25	OVFCHC24	OVFCHC23	OVFCHC22	OVFCHC21	OVFCHC20	OVFCHC19	OVFCHC18	OVFCHC17	OVFCHC16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	OVFCHC15	OVFCHC14	OVFCHC13	OVFCHC12	OVFCHC11	OVFCHC10	OVFCHC9	OVFCHC8	OVFCHC7	OVFCHC6	OVFCHC5	OVFCHC4	OVFCHC3	OVFCHC2	OVFCHC1	OVFCHC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
28:0	OVFCHC0 to OVFCHC28	Analog Channel n: Overflow Flag Clear The suffix number of each bit symbol corresponds to the analog channel number n. 0: No effect 1: ADOVFCHSR0.OVFCHFn is cleared	W
31:29	—	The write value should be 0.	W

The ADOVFCHSCR0 register clears the Overflow flag for the analog channel n.

36.2.11.14 ADOVFEXSCR : Extended Analog A/D Conversion Overflow Status Clear Register

Base address: ADC_B = 0x4017_0000

Offset address: 0xCC4

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	OVFEXC8	OVFEXC7	OVFEXC6	OVFEXC5	—	—	OVFEXC2	OVFEXC1	OVFEXC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	OVFEXC0	Self-diagnosis Channel: Overflow Flag Clear 0: No effect 1: ADOVFEXSR.OVFEXF0 is cleared	W
1	OVFEXC1	Temperature Sensor Channel: Overflow Flag Clear 0: No effect 1: ADOVFEXSR.OVFEXF1 is cleared	W
2	OVFEXC2	Internal Reference Voltage Channel: Overflow Flag Clear 0: No effect 1: ADOVFEXSR.OVFEXF2 is cleared	W
4:3	—	The write value should be 0.	W
5	OVFEXC5	D/A Converter 0 Channel: Overflow Flag Clear 0: No effect 1: ADOVFEXSR.OVFEXF5 is cleared	W
6	OVFEXC6	D/A Converter 1 Channel: Overflow Flag Clear 0: No effect 1: ADOVFEXSR.OVFEXF6 is cleared	W
7	OVFEXC7	D/A Converter 2 Channel: Overflow Flag Clear 0: No effect 1: ADOVFEXSR.OVFEXF7 is cleared	W
8	OVFEXC8	D/A Converter 3 Channel: Overflow Flag Clear 0: No effect 1: ADOVFEXSR.OVFEXF8 is cleared	W
31:9	—	The write value should be 0.	W

The ADOVFEXSCR register clears the overflow flag for the extended analog function channels.

36.2.12 FIFO

36.2.12.1 ADFIFOCR : FIFO Control Register

Base address: ADC_B = 0x4017_0000

Offset address: 0x4C0

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	FIFOEN8	FIFOEN7	FIFOEN6	FIFOEN5	FIFOEN4	FIFOEN3	FIFOEN2	FIFOEN1	FIFOEN0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	FIFOEN0 to FIFOEN8	Scan Group n FIFO Enable The suffix number of each bit symbol corresponds to the scan group number n. 0: Disable scan group n FIFO function 1: Enable scan group n FIFO function	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

The ADFIFOCR register enables/disables the FIFO function of scan group n.

FIFOENn bit (Scan Group n FIFO Enable) (n = 0 to 8)

The FIFOENn bit enables or disables the FIFO function of scan group n. When the FIFOENn bit is set to 1, the FIFO function of scan group n is enabled, and the result of A/D conversion is stored to the FIFO. When the FIFOENn bit is set to 0, the FIFO function of scan group n is disabled.

The A/D conversion result can also be read from the A/D data or extended data registers.

36.2.12.2 ADFIFOINTCR : FIFO Interrupt Control Register

Base address: ADC_B = 0x4017_0000

Offset address: 0x4C4

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	FIFOIE8	FIFOIE7	FIFOIE6	FIFOIE5	FIFOIE4	FIFOIE3	FIFOIE2	FIFOIE1	FIFOIE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	FIFOIE0 to FIFOIE8	Scan Group n FIFO Interrupt Enable The suffix number of each bit symbol corresponds to the scan group number n. 0: Disable scan group n FIFO interrupt 1: Enable scan group n FIFO interrupt	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

The ADFIFOINTCR register enables/disables the FIFO data read request interrupt and FIFO overflow interrupt for scan group n.

36.2.12.3 ADFIFOINTLR0 : FIFO Interrupt Generation Level Register 0

Base address: ADC_B = 0x4017_0000

Offset address: 0x4C8

Bit position:	31												19		16												3		0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	FIFOILV1[3:0]			—	—	—	—	—	—	—	—	—	—	—	FIFOILV0[3:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	FIFOILV0[3:0]	Scan Group 0 FIFO Interrupt Output Timing Setting These bits set the generation timing of FIFO interrupt. The FIFO data read request interrupt is generated when the number of vacant stages in FIFO becomes less than or equal to the value set by these bits. Set a value in the range from 0 to 7. Setting a value in the range from 8 to 15 is prohibited.	R/W
15:4	—	These bits are read as 0. The write value should be 0.	R/W
19:16	FIFOILV1[3:0]	Scan Group 1 FIFO Interrupt Output Timing Setting These bits set the generation timing of FIFO interrupt. The FIFO data read request interrupt is generated when the number of vacant stages in FIFO becomes less than or equal to the value set by these bits. Set a value in the range from 0 to 7. Setting a value in the range from 8 to 15 is prohibited.	R/W
31:20	—	These bits are read as 0. The write value should be 0.	R/W

The ADFIFOINTLR0 register specifies the generation timing of the FIFO data read request interrupt for the scan group 0 and 1.

36.2.12.4 ADFIFOINTLR1 : FIFO Interrupt Generation Level Register 1

Base address: ADC_B = 0x4017_0000

Offset address: 0x4CC

Bit position:	31												19		16												3		0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	FIFOILV3[3:0]			—	—	—	—	—	—	—	—	—	—	—	FIFOILV2[3:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	FIFOILV2[3:0]	Scan Group 2 FIFO Interrupt Output Timing Setting These bits set the generation timing of FIFO interrupt. The FIFO data read request interrupt is generated when the number of vacant stages in FIFO becomes less than or equal to the value set by these bits. Set a value in the range from 0 to 7. Setting a value in the range from 8 to 15 is prohibited.	R/W
15:4	—	These bits are read as 0. The write value should be 0.	R/W
19:16	FIFOILV3[3:0]	Scan Group 3 FIFO Interrupt Output Timing Setting These bits set the generation timing of FIFO interrupt. The FIFO data read request interrupt is generated when the number of vacant stages in FIFO becomes less than or equal to the value set by these bits. Set a value in the range from 0 to 7. Setting a value in the range from 8 to 15 is prohibited.	R/W
31:20	—	These bits are read as 0. The write value should be 0.	R/W

The ADFIFOINTLR1 register specifies the generation timing of the FIFO data read request interrupt for the scan group 2 and 3.

36.2.12.5 ADFIFOINTLR2 : FIFO Interrupt Generation Level Register 2

Base address: ADC_B = 0x4017_0000

Offset address: 0x4D0

Bit position:	31												19		16												3		0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	FIFOILV5[3:0]			—	—	—	—	—	—	—	—	—	—	—	FIFOILV4[3:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	FIFOILV4[3:0]	Scan Group 4 FIFO Interrupt Output Timing Setting These bits set the generation timing of FIFO interrupt. The FIFO data read request interrupt is generated when the number of vacant stages in FIFO becomes less than or equal to the value set by these bits. Set a value in the range from 0 to 7. Setting a value in the range from 8 to 15 is prohibited.	R/W
15:4	—	These bits are read as 0. The write value should be 0.	R/W
19:16	FIFOILV5[3:0]	Scan Group 5 FIFO Interrupt Output Timing Setting These bits set the generation timing of FIFO interrupt. The FIFO data read request interrupt is generated when the number of vacant stages in FIFO becomes less than or equal to the value set by these bits. Set a value in the range from 0 to 7. Setting a value in the range from 8 to 15 is prohibited.	R/W
31:20	—	These bits are read as 0. The write value should be 0.	R/W

The ADFIFOINTLR2 register specifies the generation timing of the FIFO data read request interrupt for the scan group 4 and 5.

36.2.12.6 ADFIFOINTLR3 : FIFO Interrupt Generation Level Register 3

Base address: ADC_B = 0x4017_0000

Offset address: 0x4D4

Bit position:	31												19		16												3		0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	FIFOILV7[3:0]			—	—	—	—	—	—	—	—	—	—	—	FIFOILV6[3:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	FIFOILV6[3:0]	Scan Group 6 FIFO Interrupt Output Timing Setting These bits set the generation timing of FIFO interrupt. The FIFO data read request interrupt is generated when the number of vacant stages in FIFO becomes less than or equal to the value set by these bits. Set a value in the range from 0 to 7. Setting a value in the range from 8 to 15 is prohibited.	R/W
15:4	—	These bits are read as 0. The write value should be 0.	R/W
19:16	FIFOILV7[3:0]	Scan Group 7 FIFO Interrupt Output Timing Setting These bits set the generation timing of FIFO interrupt. The FIFO data read request interrupt is generated when the number of vacant stages in FIFO becomes less than or equal to the value set by these bits. Set a value in the range from 0 to 7. Setting a value in the range from 8 to 15 is prohibited.	R/W
31:20	—	These bits are read as 0. The write value should be 0.	R/W

The ADFIFOINTLR3 register specifies the generation timing of the FIFO data read request interrupt for the scan group 6 and 7.

36.2.12.7 ADFIFOINTLR4 : FIFO Interrupt Generation Level Register 4

Base address: ADC_B = 0x4017_0000

Offset address: 0x4D8

Bit position:	31																										3	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FIFOILV8[3:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	FIFOILV8[3:0]	Scan Group 8 FIFO Interrupt Output Timing Setting These bits set the generation timing of FIFO interrupt. The FIFO data read request interrupt is generated when the number of vacant stages in FIFO becomes less than or equal to the value set by these bits. Set a value in the range from 0 to 7. Setting a value in the range from 8 to 15 is prohibited.	R/W
31:4	—	These bits are read as 0. The write value should be 0.	R/W

The ADFIFOINTLR4 register specifies the generation timing of the FIFO data read request interrupt for the scan group 8.

36.2.12.8 ADFIFOSR0 : FIFO Status Register 0

Base address: ADC_B = 0x4017_0000

Offset address: 0xCD0

Bit position:	31												19		16												3	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	FIFOST1[3:0]			—	—	—	—	—	—	—	—	—	—	—	FIFOST0[3:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Bit	Symbol	Function	R/W
3:0	FIFOST0[3:0]	Number of vacant stages in FIFO for Scan Group 0	R
15:4	—	These bits are read as 0.	R
19:16	FIFOST1[3:0]	Number of vacant stages in FIFO for Scan Group 1	R
31:20	—	These bits are read as 0.	R

The ADFIFOSR0 register indicates the number of vacant stages in FIFO for scan group 0 and 1.

36.2.12.9 ADFIFOSR1 : FIFO Status Register 1

Base address: ADC_B = 0x4017_0000

Offset address: 0xCD4

Bit position:	31												19		16												3	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	FIFOST3[3:0]			—	—	—	—	—	—	—	—	—	—	—	FIFOST2[3:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Bit	Symbol	Function	R/W
3:0	FIFOST2[3:0]	Number of vacant stages in FIFO for Scan Group 2	R
15:4	—	These bits are read as 0.	R
19:16	FIFOST3[3:0]	Number of vacant stages in FIFO for Scan Group 3	R
31:20	—	These bits are read as 0.	R

The ADFIFOSR1 register indicates the number of vacant stages in FIFO for scan group 2 and 3.

36.2.12.10 ADFIFOSR2 : FIFO Status Register 2

Base address: ADC_B = 0x4017_0000

Offset address: 0xCD8

Bit position:	31												19		16												3		0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	FIFOST5[3:0]				—	—	—	—	—	—	—	—	—	—	—	FIFOST4[3:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Bit	Symbol	Function	R/W
3:0	FIFOST4[3:0]	Number of vacant stages in FIFO for Scan Group 4	R
15:4	—	These bits are read as 0.	R
19:16	FIFOST5[3:0]	Number of vacant stages in FIFO for Scan Group 5	R
31:20	—	These bits are read as 0.	R

The ADFIFOSR2 register indicates the number of vacant stages in FIFO for scan group 4 and 5.

36.2.12.11 ADFIFOSR3 : FIFO Status Register 3

Base address: ADC_B = 0x4017_0000

Offset address: 0xCDC

Bit position:	31												19		16												3		0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	FIFOST7[3:0]				—	—	—	—	—	—	—	—	—	—	—	FIFOST6[3:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Bit	Symbol	Function	R/W
3:0	FIFOST6[3:0]	Number of vacant stages in FIFO for Scan Group 6	R
15:4	—	These bits are read as 0.	R
19:16	FIFOST7[3:0]	Number of vacant stages in FIFO for Scan Group 7	R
31:20	—	These bits are read as 0.	R

The ADFIFOSR3 register indicates the number of vacant stages in FIFO for scan group 6 and 7.

36.2.12.12 ADFIFOSR4 : FIFO Status Register 4

Base address: ADC_B = 0x4017_0000

Offset address: 0xCE0

Bit position:	31																										3		0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FIFOST8[3:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Bit	Symbol	Function	R/W
3:0	FIFOST8[3:0]	Number of vacant stages in FIFO for Scan Group 8	R
31:4	—	These bits are read as 0.	R

The ADFIFOSR4 register indicates the number of vacant stages in FIFO for scan group 8.

36.2.12.13 ADFIFODCR : FIFO Data Clear Register

Base address: ADC_B = 0x4017_0000

Offset address: 0xCF0

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	FIFODC8	FIFODC7	FIFODC6	FIFODC5	FIFODC4	FIFODC3	FIFODC2	FIFODC1	FIFODC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	FIFODC0 to FIFODC8	Scan Group n FIFO Data Clear The suffix number of each bit symbol corresponds to the scan group number n. 0: No effect 1: Clear the data of scan group n FIFO	W
31:9	—	The write value should be 0.	W

The ADFIFODCR register clears the data in FIFO for each scan group.

FIFODCn bit (Scan Group n FIFO Data Clear) (n = 0 to 8)

The FIFODCn bit clears the data in FIFO for scan group n. Writing 1 to the FIFODCn clears the data in the FIFO of scan group n. Writing 0 has no effect on the operation, and the data in FIFO is not cleared.

36.2.12.14 ADFIFOERSR : FIFO Error Status Register

Base address: ADC_B = 0x4017_0000

Offset address: 0xCF4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	FIFOFLF8	FIFOFLF7	FIFOFLF6	FIFOFLF5	FIFOFLF4	FIFOFLF3	FIFOFLF2	FIFOFLF1	FIFOFLF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	FIFOOVF8	FIFOOVF7	FIFOOVF6	FIFOOVF5	FIFOOVF4	FIFOOVF3	FIFOOVF2	FIFOOVF1	FIFOOVF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	FIFOOVF0 to FIFOOVF8	Scan Group n FIFO Overflow Flag The suffix number of each bit symbol corresponds to the scan group number n. 0: No overflow 1: FIFO overflow is detected	R
15:9	—	These bits are read as 0.	R
24:16	FIFOFLF0 to FIFOFLF8	Scan Group n FIFO Data Read Request Flag The suffix number of each bit symbol corresponds to the scan group number n. 0: FIFO Data Read Request is not detected. 1: FIFO Data Read Request is detected.	R
31:25	—	These bits are read as 0.	R

The ADFIFOERSR register indicates the status of the FIFO. Each flag can be cleared in ADFIFOERSCR.

FIFOOVFn bit (Scan Group n FIFO Overflow Flag) (n = 0 to 8)

The FIFOOVFn bit indicates whether the overflow occurred in the FIFO of the scan group n.

When FIFOOVFn = 0, an overflow is not detected in the FIFO. When FIFOOVFn = 1, an overflow occurred in the FIFO and the A/D conversion result cannot be stored to the FIFO. FIFOOVFn can be cleared in ADFIFOERSCR.

FIFOFLFn bit (Scan Group n FIFO Data Read Request Flag) (n = 0 to 8)

The FIFOFLFn bit indicates whether a FIFO data read request in scan group n has been detected. The setting and clearing conditions of FIFOFLFn are shown as follows:

[Setting condition]

When the condition of ADFIFOSRm.FIFOSTn[3:0] ≤ ADFIFOINTLRm.FIFOILVn[3:0] is detected.

[Clearing condition]

When ADFIFOERSCR.FIFOFLCn is written to 1 under the condition of ADFIFOSRm.FIFOSTn[3:0] > ADFIFOINTLRm.FIFOILVn[3:0].

When the condition of ADFIFOSRm.FIFOSTn[3:0] > ADFIFOINTLRm.FIFOILVn[3:0] is occurred by the read access to ADFIFODRn register with DMAC or DTC.

Note: m = 0 to 4, n = 0 to 8

36.2.12.15 ADFIFOERSCR : FIFO Error Status Clear Register

Base address: ADC_B = 0x4017_0000

Offset address: 0xCF8

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	FIFOFLC8	FIFOFLC7	FIFOFLC6	FIFOFLC5	FIFOFLC4	FIFOFLC3	FIFOFLC2	FIFOFLC1	FIFOFLC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	FIFOOVFC8	FIFOOVFC7	FIFOOVFC6	FIFOOVFC5	FIFOOVFC4	FIFOOVFC3	FIFOOVFC2	FIFOOVFC1	FIFOOVFC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	FIFOOVFC0 to FIFOOVFC8	Scan Group n FIFO Overflow Flag Clear The suffix number of each bit symbol corresponds to the scan group number n. 0: No effect 1: ADFIFOERSR.FIFOOVFn is cleared	W
15:9	—	The write value should be 0.	W
24:16	FIFOFLC0 to FIFOFLC8	Scan Group n FIFO Data Read Request Flag Clear The suffix number of each bit symbol corresponds to the scan group number n. 0: No effect 1: ADFIFOERSR.FIFOFLFn is cleared	W
31:25	—	The write value should be 0.	W

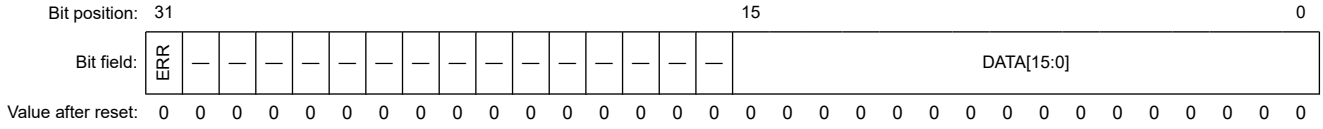
ADFIFOERSCR clears the FIFO overflow flags and the FIFO data read request flags for scan group n.

36.2.13 Data Register

36.2.13.1 ADDRn : A/D Data Register n (n = 0 to 28)

Base address: ADC_B = 0x4017_0000

Offset address: 0x1000 + 0x04 × n



Bit	Symbol	Function	R/W
15:0	DATA[15:0]	A/D conversion data	R
30:16	—	These bits are read as 0.	R
31	ERR	A/D conversion data error status 0: No error (the A/D conversion data is valid) 1: Error is detected (the A/D conversion data is not guaranteed)	R

The ADDRn registers are read-only registers to read A/D conversion results.

DATA[15:0] bits (A/D conversion data)

The DATA[15:0] bits indicate the data of the A/D conversion result for the analog input channel n.

The data format of the A/D conversion results is determined by the setting of the virtual channel m (m = 0 to 36) to which the analog input channel n is assigned.

ERR bit (A/D conversion data error status)

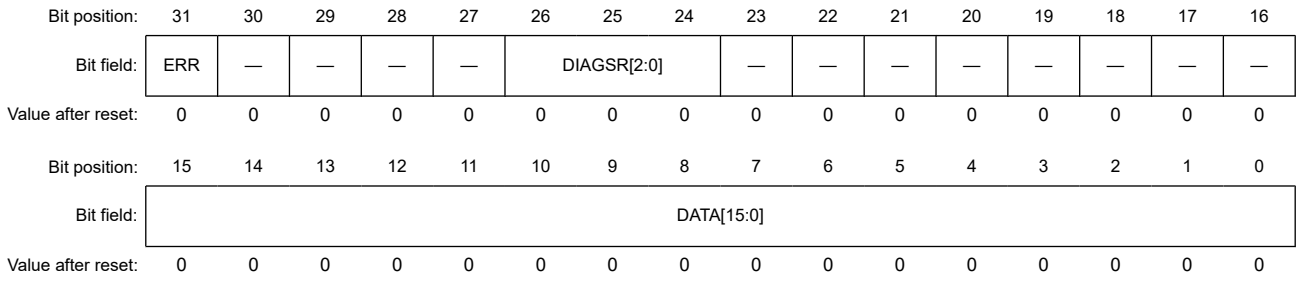
The ERR bit indicates the error status in the A/D conversion for the analog input channel n.

When ERR = 0, the A/D conversion data is valid. When ERR = 1, the A/D conversion data is invalid, and the accuracy of the A/D conversion data is not guaranteed.

36.2.13.2 ADEXDRn : A/D Extended Analog Data Register n (n = 0 to 2, 5 to 8)

Base address: ADC_B = 0x4017_0000

Offset address: 0x1180 + 0x04 × n



Bit	Symbol	Function	R/W
15:0	DATA[15:0]	A/D conversion data	R
23:16	—	These bits are read as 0.	R
26:24	DIAGSR[2:0]*1	Self-diagnosis Status*1	R
30:27	—	These bits are read as 0.	R
31	ERR	A/D Conversion Error Status 0: No error (the A/D conversion data is valid) 1: Error is detected (the A/D conversion data is not guaranteed)	R

Note 1. Only the ADEXDR0 register has the DIAGSR[2:0] bits. The DIAGSR[2:0] bits in the ADEXDRn register except for the ADEXDR0 register are reserved bits.

The ADEXDRn registers are read-only registers to read A/D conversion results of the extended analog function. The ADEXDRn registers and the extended analog functions are as follows:

- ADEXDR0 : Self-diagnosis
- ADEXDR1 : Temperature sensor
- ADEXDR2 : Internal reference voltage
- ADEXDR5 : D/A converter ch0 output
- ADEXDR6 : D/A converter ch1 output
- ADEXDR7 : D/A converter ch2 output
- ADEXDR8 : D/A converter ch3 output

DATA[15:0] bits (A/D conversion data)

The DATA[15:0] bits indicate the data of the A/D conversion result for the extended analog function channel. The data format of the A/D conversion results is determined by the setting of the virtual channel m (m = 0 to 36) to which the extended analog function channel is assigned.

DIAGSR[2:0] bits (Self-diagnosis Status)

The DIAGSR[2:0] bits indicate the voltage setting of the self-diagnosis channel. The DIAGSR[2:0] bits indicate the setting value in the ADSGDCRm.DIAGVAL[2:0] bits (m = 0 to 8). The DIAGSR[2:0] bits are only available in the ADEXDR0. The DIAGSR[2:0] bits in ADEXDRn except ADEXDR0 are reserved.

ERR bit (A/D Conversion Error Status)

The ERR bit indicates the error status in the A/D conversion for the extended analog functions.

When ERR = 0, the A/D conversion data is valid. When ERR = 1, the A/D conversion data is invalid, and the accuracy of the A/D conversion data is not guaranteed.

36.2.13.3 ADFIFODRn : FIFO Data Register n (n = 0 to 8)

Base address: ADC_B = 0x4017_0000

Offset address: 0x1200 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	ERR	CH[6:0]							—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
15:0	DATA[15:0]	A/D Conversion Data	R
23:16	—	These bits are read as 0.	R
30:24	CH[6:0]	A/D Conversion Channel Number	R
31	ERR	A/D Conversion Data Error Status 0: No error (the A/D conversion data is valid) 1: Error is detected (the A/D conversion data is not guaranteed)	R

The ADFIFODRn register is used to read data stored in the FIFO of scan group n.

DATA[15:0] bits (A/D Conversion Data)

The DATA[15:0] bits are read-only bits that read the A/D conversion data stored in the scan group n FIFO.

CH[6:0] bits (A/D Conversion Channel Number)

The CH[6:0] bits indicate the channel number of the A/D conversion data that is read from the DATA[15:0] bits.

ERR bit (A/D Conversion Data Error Status)

The ERR bit indicates the error status in the A/D conversion data of the analog channel indicated by the CH[6:0] bits.

When ERR = 0, the A/D conversion data is valid. When ERR = 1, the A/D conversion data is invalid, and the accuracy of the A/D conversion data is not guaranteed.

36.3 Operation

36.3.1 A/D Conversion Clock

A/D conversion clock (ADCLK) is the operation clock of ADC. The A/D converters (ADC0 and ADC1) are operated and controlled by ADCLK as the basic clock. Figure 36.4 shows clock structure of ADC.

ADCLK is generated from the clock source and the division ratio selected in ADCLKCR register. The frequency of ADCLK should be set so that PCLKA ≥ ADCLK. Also, the frequency of ADCLK should be set so that it is within the guaranteed operating range specified in the electrical characteristics. For details, see section 46, Electrical Characteristics.

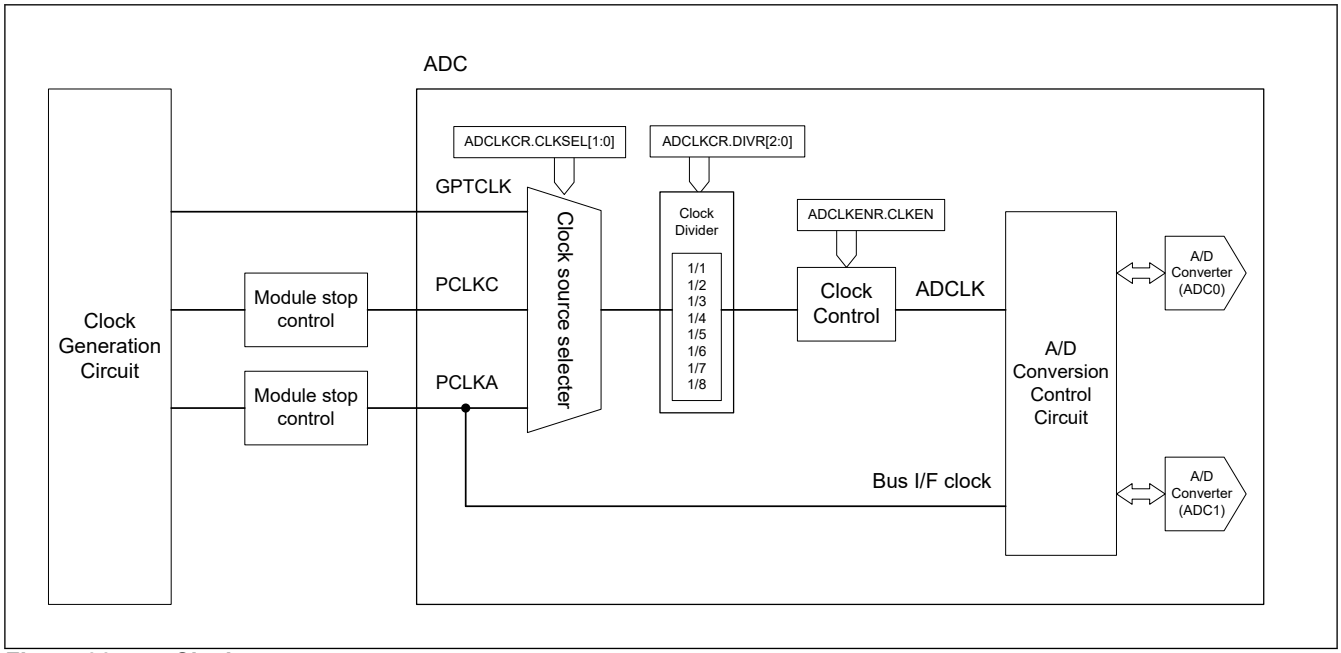


Figure 36.4 Clock structure

36.3.2 A/D Converter Operation Mode

36.3.2.1 SAR Mode

In SAR mode, A/D converter operates as a Nyquist type A/D converter. Analog channels are sampled one time and converted analog to digital by Successive Approximation Register (SAR) method. The features of SAR mode are shown in the following:

[Features of SAR mode]

- Fast conversion: A/D conversion data can be acquired with the shortest group delay.
- A/D conversion of up to 8 analog channels (virtual channels) is possible per scan group.
- Single-ended input only supported (differential input and digital filter functions not supported)*1

Note 1. In SAR mode, differential input is supported only during self-diagnosis as an exception. Other than self-diagnosis, differential input is not supported.

For details about operation of A/D conversion in SAR mode, see section 36.3.7. Scanning Operation. For restrictions on SAR mode, see section 36.10.15. Restrictions on SAR Mode.

36.3.2.2 Oversampling Mode

In Oversampling mode, A/D converter operates as Oversampling-type A/D converter. Analog channels are oversampled and converted analog to digital by Noise Shaping Successive Approximation Register method (NS-SAR). The features of Oversampling mode are shown in the following:

[Features of Oversampling mode]

- High accuracy conversion: A/D conversion data can be acquired with Noise Shaping technology and Digital Filter function.*1
- A/D conversion of up to 8 analog channels (virtual channels) is possible per scan group.
- Both of single-ended input and differential input are supported.

Note 1. Digital Filter function must be used in Oversampling mode. It is prohibited to operate without digital filter function in Oversampling mode.

For details about operation of A/D conversion in Oversampling mode, see section 36.3.7. Scanning Operation. For restrictions on Oversampling mode, see section 36.10.16. Restrictions on Oversampling Mode.

36.3.2.3 Hybrid Mode

Hybrid mode has both features of SAR mode and Oversampling mode. In Hybrid mode, A/D converter operates as Oversampling-type A/D converter. Analog channels are oversampled and converted analog to digital by Noise Shaping Successive Approximation Register (NS-SAR) method. In the hybrid mode, while switching channels for each sampling, multiple analog channels are oversampled. The features of Hybrid mode are shown in the following:

[Features of Hybrid mode]

- High accuracy conversion: A/D conversion data can be acquired with Noise Shaping technology and Digital Filter function.*1
- Fast conversion: A/D conversion data can be acquired with the shortest group delay when in the Continuous scan mode and after initial delay time.
- A/D conversion of up to 4 analog channels (virtual channels) is possible per scan group.
- Both of single-ended input and differential input are supported.

Note 1. Digital Filter function must be used in Hybrid mode. It is prohibited to operate without digital filter function in Hybrid mode.

For details about operation of A/D conversion in Hybrid mode, see section 36.3.7. Scanning Operation. For restrictions on Hybrid mode, see section 36.10.17. Restrictions on Hybrid Mode.

36.3.3 Single-ended Input and Differential Input

ADC supports Single-ended input and Differential input. The selection of Single-ended input or Differential input is specified in ADCHCRn.AINMD (n = 0 to 36) bit. For the analog channels that support differential input, see Table 36.4 and Table 36.5.

Single-ended Input

In Single-ended input mode, the difference between the voltage of the analog channel (signal source) and the analog reference ground voltage (VREFL0) is A/D-converted.

Differential Input

In Differential input mode, the even-numbered analog channel is used as the non-inverting input (+) (A_{INP}) and the odd-numbered analog channel is used as the inverting input (−) (A_{INN}), then the difference voltage between the non-inverting input (+) and the inverting input (−) ($A_{INP} - A_{INN}$) is A/D-converted.*1

For A/D conversion with Differential input mode, select the non-inverting input (+) channel*2 of the target of differential pair in ADCHCRn.CNVCS[6:0] (n = 0 to 36) bits, and select the Differential input mode in ADCHCRn.AINMD (n = 0 to 36) bit.

It is prohibited to perform A/D conversion for channels that do not support differential input with Differential input mode. In that case, A/D conversion result is not guaranteed.

Note 1. The differential input pair is a combination of channels with analog channel number 2i and 2i + 1 (i = 0, 1, 2, 3, ...), and they are named ANxxxP and ANxxxN (xxx = 000, 002, 004, ...). The combination of discontinuous analog channel numbers or the combination of analog channel numbers 2i − 1 and 2i cannot be used as a differential input pair.

Note 2. It is the even-numbered analog channel which is named as ANxxxP (xxx = 000, 002, 004, …).

36.3.4 Analog Channel

The analog channel is the source of the analog signal that is targeted for A/D conversion. The analog channels consist of the following elements:

- Analog input channels: A/D conversion channels for the analog input from MCU's I/O pin as the signal source.
- Extended analog channels: A/D conversion channels for the source of the analog signal inside the MCU.

To perform A/D conversion for the analog channels, assignments to the virtual channel and scan group are required. For each element, refer to each item.

36.3.5 Virtual Channel

The virtual channel is a group of registers that stores the A/D conversion configurations for an analog channel. The virtual channels can be specified the configurations for A/D conversion such as the selection of the analog channels, the optional settings for A/D conversion, the data processing method of the A/D conversion data, the assignment to scan groups, and so on.

To perform the A/D conversion of the analog channel, the analog channel should be assigned to any of the virtual channels. Then that virtual channel should be assigned to one of the scan groups. See Figure 36.5 for the relationship between the analog channels and the virtual channels and scan groups. Figure 36.6 shows the A/D conversion order in Figure 36.5 settings.

The virtual channel can be assigned to only one scan group. If the same analog channel is to be converted within the different scan group, or the same analog channel is to be converted several times within the same scan group, assign the analog channel to several virtual channels. When performing the A/D conversion more than once using more than one virtual channel within the same scan group for the same analog channel, the FIFO function should be used in combination.

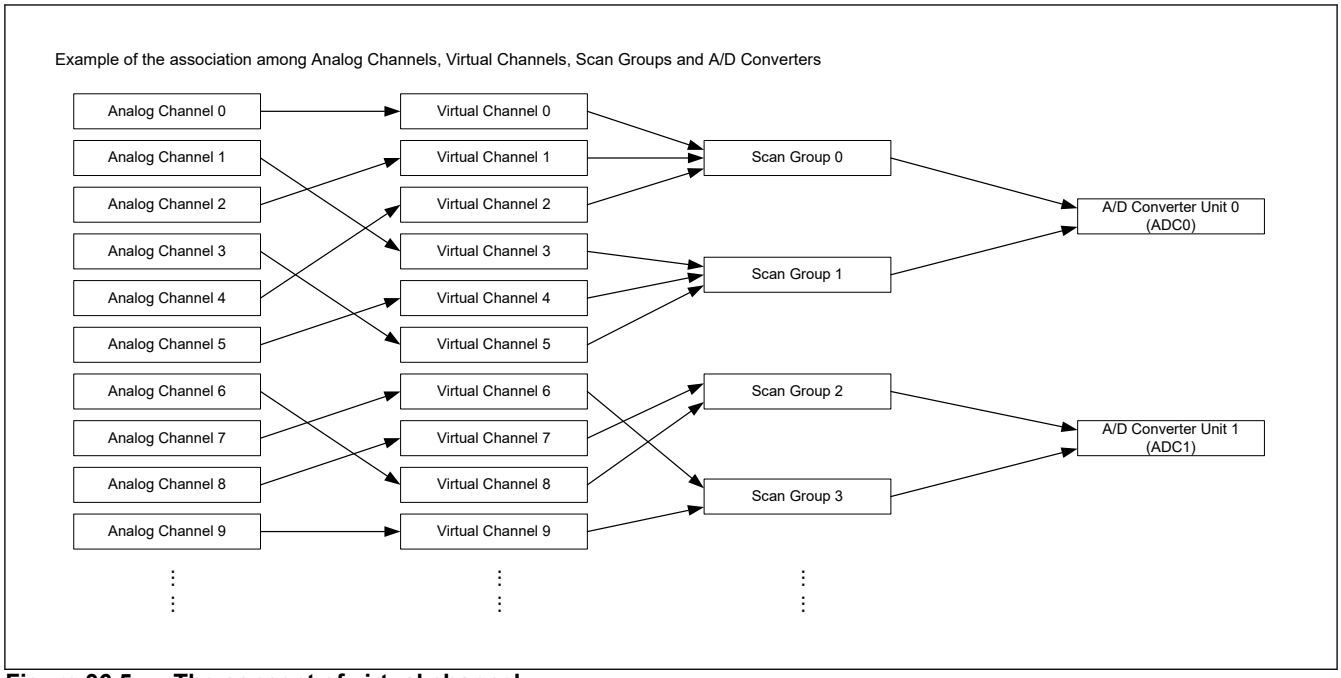


Figure 36.5 The concept of virtual channel

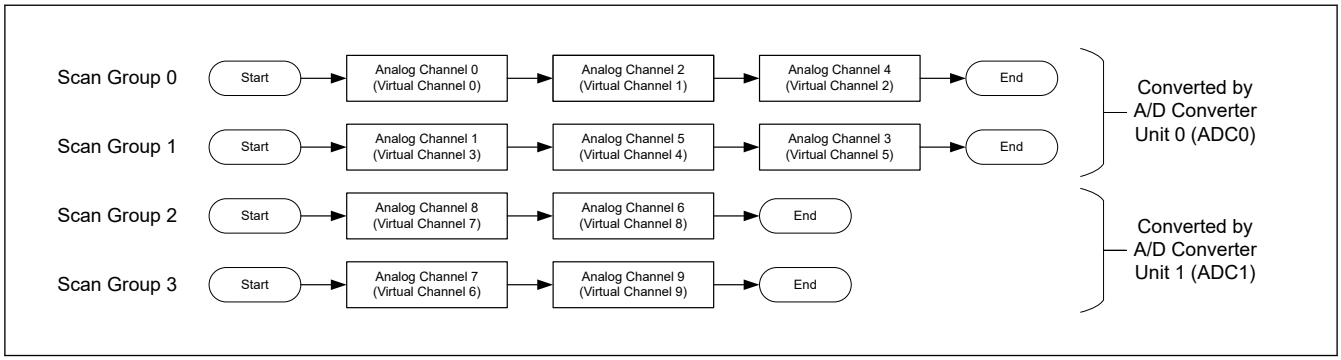


Figure 36.6 Example of A/D conversion order at the scanning operation

36.3.6 Scan Group

The scan group is a group of the analog channels that perform A/D conversion in a scanning operation. To perform the A/D conversion using the scan group, configure the following steps:

- Assign the analog channel for the A/D conversion to the virtual channel.
- Assign the virtual channel to the scan group.
- Assign the scan group to A/D converter.

See Figure 36.5 and Figure 36.6 for the relationship among the analog channels, the virtual channels, and the scan groups.

A single scan group can be assigned virtual channels in the following ranges, according to its operation mode.

- SAR mode: up to 8 virtual channels per scan group
- Oversampling mode: up to 8 virtual channels per scan group
- Hybrid mode: from 2 to 4 virtual channels per scan group

If more than the maximum number of virtual channels are allocated to one scan group, A/D conversion is performed on the 8th (SAR mode or Oversampling mode) or 4th (Hybrid mode) channels in ascending order of virtual channel numbers. Subsequent channels are not subject to A/D conversion.

Notes on the scan groups

The virtual channels assigned to one scan group should be assigned the analog channels that can be converted by the same A/D converter. If any analog channels cannot be converted by the specified A/D converter are included, the A/D conversion result of those channels are undefined.

In Hybrid mode, you must assign at least 2 virtual channels to a scan group. Operation is not guaranteed when any virtual channel is not assigned or only 1 virtual channel is assigned.

36.3.7 Scanning Operation

The scanning operation is the operation of sequential A/D conversion for the analog channels. The modes of the scanning operation are the followings:

- Single scan mode
  - In single scan mode, one scan group is scanned once for each A/D conversion start trigger input.
  - Each time an A/D conversion start trigger is input, A/D conversion is performed once for each analog channel assigned to that scan group.
- Continuous scan mode
  - The continuous scan mode repeats the scanning operation of one scan group.
  - When an A/D conversion start trigger is input, A/D conversion of each analog channel assigned to that scan group is repeated until A/D conversion stop processing is performed.
- Background continuous scan mode
  - The background continuous scan mode repeats the scanning operation of a single scanning group.
  - Scanning operation starts when the first A/D conversion start trigger is input.
  - A/D conversions of each analog channel assigned to that scan group are repeated until A/D conversion stop processing is performed. (Continuous scanning operation)
  - When an A/D conversion start trigger is input during continuous scanning operation, the A/D conversion data at that point is output. (Unless an A/D conversion start trigger is input, the A/D conversion data are not output and the values of the A/D data registers and FIFO data registers are not updated.)

The scan modes supported by each operation mode are shown in Table 36.6.

Table 36.6 Scan mode and operation mode

Scan mode	Operation mode of A/D converter		
	SAR mode	Oversampling mode	Hybrid mode
Single scan mode	✓	✓	✓
Continuous scan mode	✓	✓	✓
Background continuous scan mode	—	—	✓

Note: ✓: available, —: not available

The scanning operations are performed on a scan group basis. When the scanning operation is started, A/D conversion is performed for each analog channel based on the virtual channel setting. If the scanning operation starts for 2 or more scan group assigned to the same A/D converter at the same time, the scanning operation of the group with the lowest scan group number is performed.

The A/D conversion order of the analog channel in the scanning operation is the ascending order of the virtual channel numbers assigned to the scan group. (The channels with the lowest virtual channel numbers are performed the A/D conversion at the forward end of the scan group. The channels with larger virtual channel numbers are performed the A/D conversion at the backward of the scan group.)

See Figure 36.5 and Figure 36.6 for the relationship of the conversion order of the respective analog channel in the scanning operation.

The scanning operations in each operation mode are shown in the following.

36.3.7.1 SAR Mode – Single Scan Mode

Table 36.7 and Figure 36.7 show the operation example in SAR mode – Single scan mode.

Table 36.7 Example of the scanning operation in SAR mode – Single scan mode

Step	Operation
1	When a software trigger or a trigger from a peripheral module is input, the scanning operation of the scan group corresponding to the trigger is started.
2	Each time an A/D conversion of one analog channel is completed, A/D conversion result is stored to the data register (ADDRi (i = 0 to 28), ADEXDRj (j = 0 to 2, 5 to 8)). If FIFO is used, the A/D conversion result is also stored to the FIFO data register (ADFIFODRk (k = 0 to 8)).
3	If the scan end interrupt is enabled, the scan end interrupt corresponding to that scan group is generated when the A/D conversion of all virtual channels assigned to that scan group is completed.
4	During the scanning operation, ADGRSR.ACTGRn (n = 0 to 8) bit corresponding to that scan group is set to 1. ADSR.ADACTm (m = 0, 1) bit corresponding to the A/D converter performing the A/D conversion is also set to 1. When scanning operation is complete, each bit is cleared to 0 and A/D converter enters the idle state.

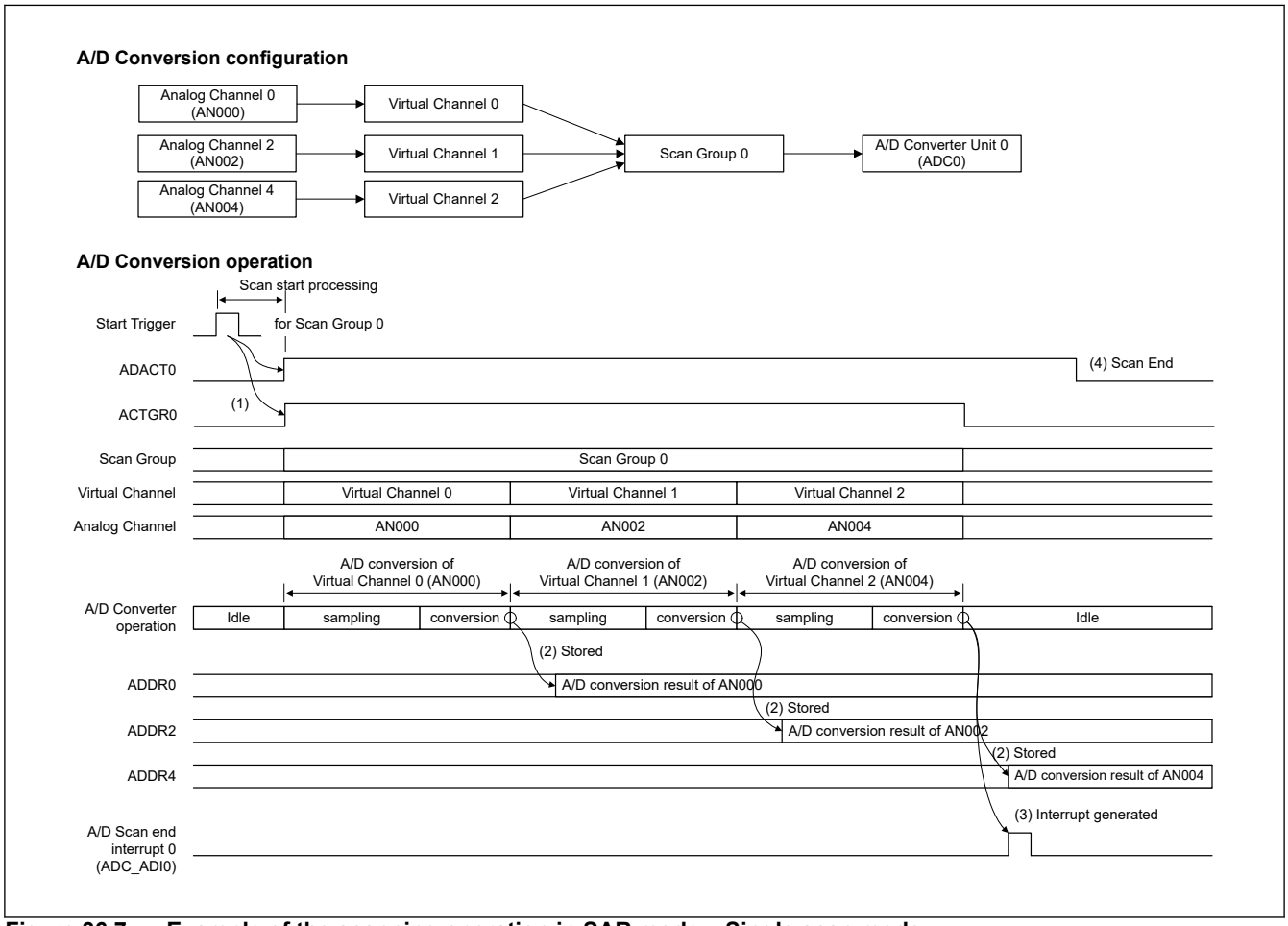


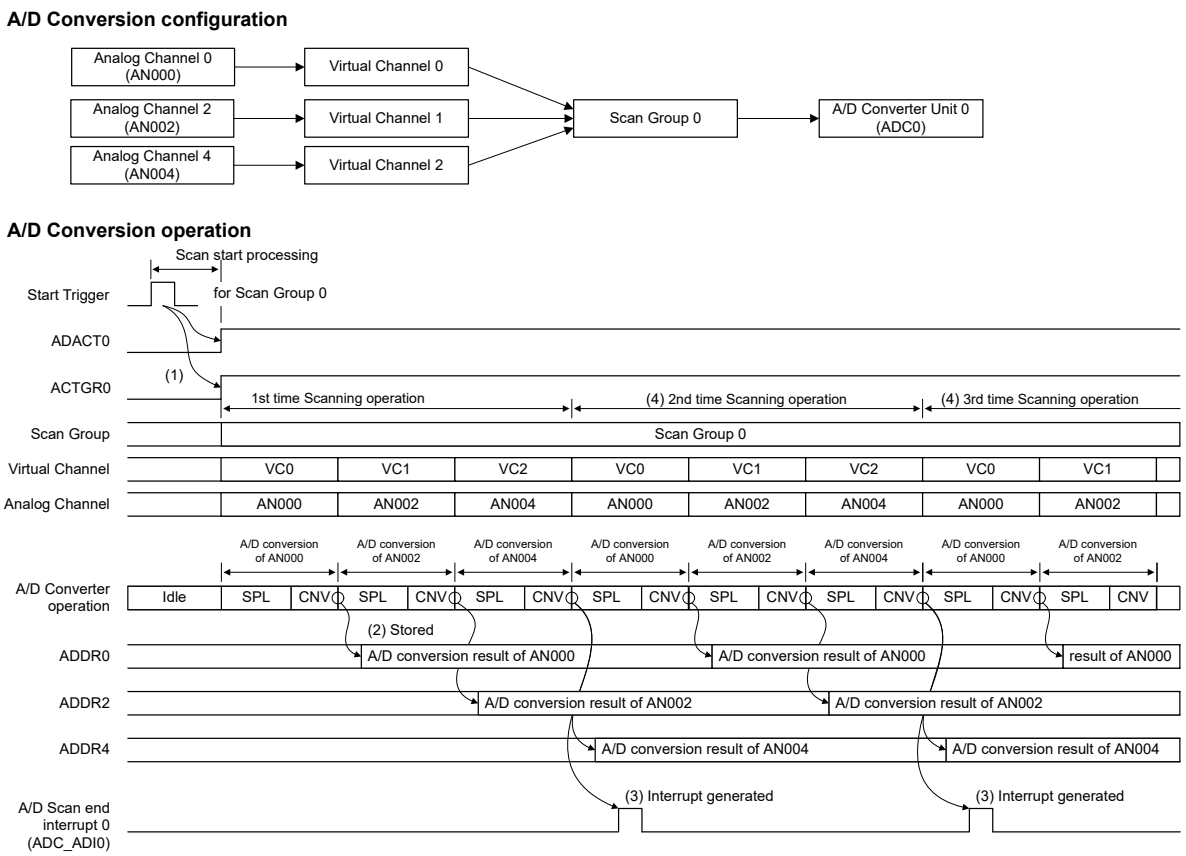
Figure 36.7 Example of the scanning operation in SAR mode – Single scan mode

36.3.7.2 SAR Mode – Continuous Scan Mode

Table 36.8 and Figure 36.8 show the operation example in SAR mode – Continuous scan mode.

Table 36.8 Example of the scanning operation in SAR mode – Continuous scan mode

Step	Operation
1	When a software trigger or a trigger from a peripheral module is input, the scanning operation of the scan group corresponding to the trigger is started. When scanning operation starts, ADGRSR.ACTGRn (n = 0 to 8) corresponding to that scan group is set to 1. ADSR.ADACTm (m = 0, 1) bit corresponding to the A/D converter performing the A/D conversion is also set to 1.
2	Each time an A/D conversion of one analog channel is completed, A/D conversion result is stored to the data register (ADDRi (i = 0 to 28), ADEXDRj (j = 0 to 2, 5 to 8)). If FIFO is used, the A/D conversion result is also stored to the FIFO data register (ADFIFODRk (k = 0 to 8)).
3	If the scan end interrupt is enabled, the scan end interrupt corresponding to that scan group is generated when the A/D conversion of all virtual channels assigned to that scan group is completed.
4	Until the A/D conversion stop process is performed, Step 2 to 3 are repeated, and the scanning operation continues. To stop the A/D conversion, you must follow to section 36.5.4. Force Stops the A/D Conversion Operation.



Note: VC: Virtual Channel SPL: Sampling state CNV: Conversion state

Figure 36.8 Example of the scanning operation in SAR mode – Continuous scan mode

36.3.7.3 Oversampling Mode – Single Scan Mode

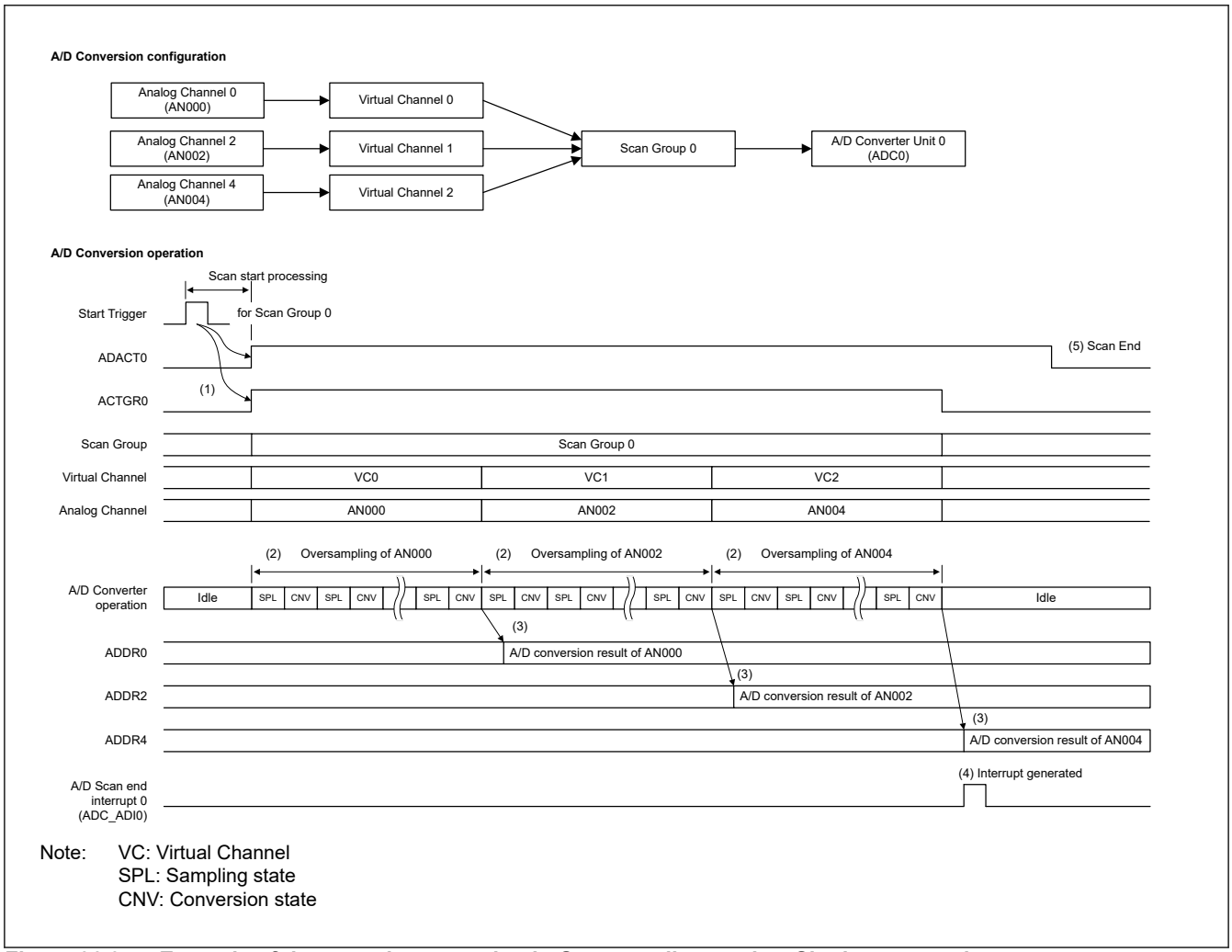
Table 36.9 and Figure 36.9 show the operation example in Oversampling mode – Single scan mode.

Table 36.9 Example of the scanning operation in Oversampling mode – Single scan mode (1 of 2)

Step	Operation
1	When a software trigger or a trigger from a peripheral module is input, the scanning operation of the scan group corresponding to the trigger is started.

Table 36.9 Example of the scanning operation in Oversampling mode – Single scan mode (2 of 2)

Step	Operation
2	Each analog channel is oversampled according to the number of TAPs in the digital filter and the number of A/D-converted value addition/averaging times. The oversampled data stored in the digital filter are discarded after the oversampling of each analog channel is completed.
3	A/D conversion data of each analog channel is output after oversampling. The A/D conversion data is stored in the data register (ADDRi (i = 0 to 28), ADEXDRj (j = 0 to 2, 5 to 8)). If FIFO is used, A/D conversion data is also stored in FIFO data register (ADFIFODRk (k = 0 to 8)).
4	If the scan end interrupt is enabled, the scan end interrupt corresponding to that scan group is generated when the A/D conversion of all virtual channels assigned to that scan group is completed.
5	During the scanning operation, ADGRSR.ACTGRn (n = 0 to 8) bit corresponding to that scan group is set to 1. ADSR.ADACTm (m = 0, 1) bit corresponding to the A/D converter performing the A/D conversion is also set to 1. When scanning operation is complete, each bit is cleared to 0 and A/D converter enters the idle state.



Note: VC: Virtual Channel
SPL: Sampling state
CNV: Conversion state

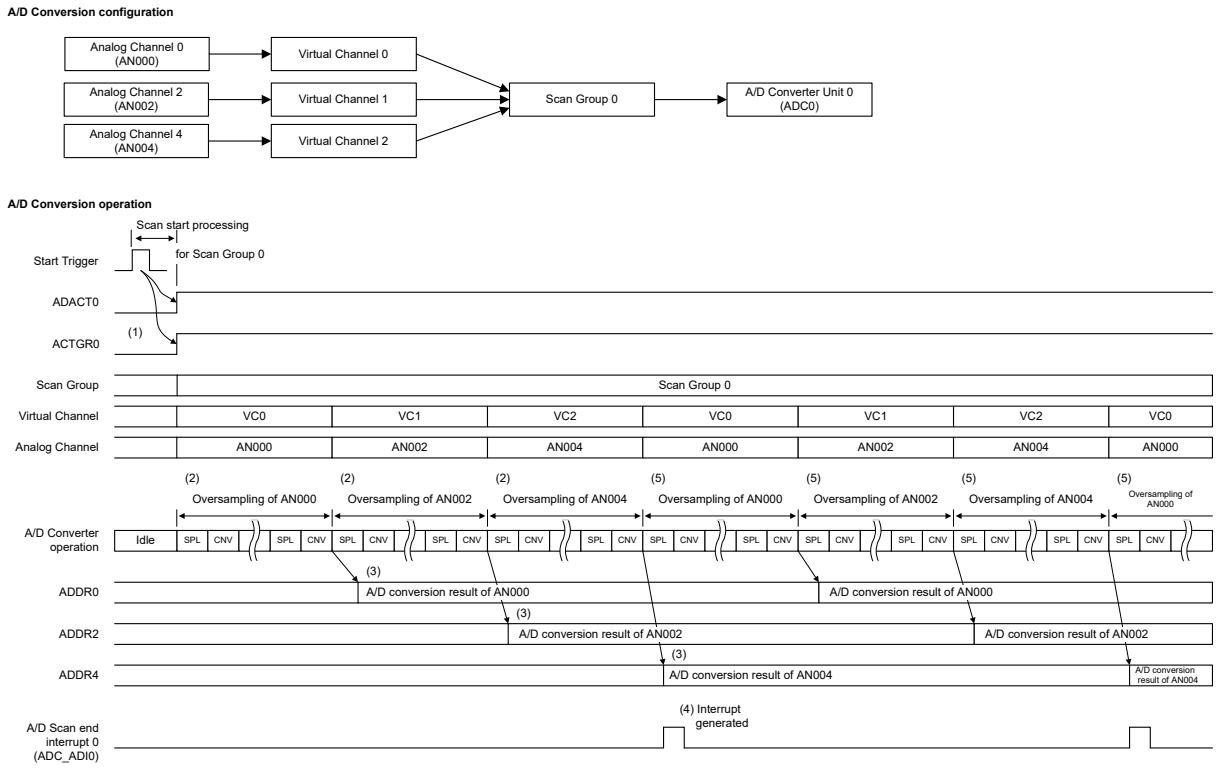
Figure 36.9 Example of the scanning operation in Oversampling mode – Single scan mode

36.3.7.4 Oversampling Mode – Continuous Scan Mode

Table 36.10 and Figure 36.10 show the operation example in Oversampling mode – Continuous scan mode.

Table 36.10 Example of the scanning operation in Oversampling mode – Continuous scan mode

Step	Operation
1	When a software trigger or a trigger from a peripheral module is input, the scanning operation of the scan group corresponding to the trigger is started. When scanning operation starts, ADGRSR.ACTGRn (n = 0 to 8) corresponding to that scan group is set to 1. ADSR.ADACTm (m = 0, 1) bit corresponding to the A/D converter performing the A/D conversion is also set to 1.
2	Each analog channel is oversampled according to the number of TAPs in the digital filter and the number of A/D-converted value addition/averaging times. The oversampled data stored in the digital filter are discarded after the oversampling of each analog channel is completed.
3	A/D conversion data of each analog channel is output after oversampling. The A/D conversion data is stored in the data register (ADDRi (i = 0 to 28), ADEXDRj (j = 0 to 2, 5 to 8)). If FIFO is used, A/D conversion data is also stored in FIFO data register (ADFIFODRk (k = 0 to 8)).
4	If the scan end interrupt is enabled, the scan end interrupt corresponding to that scan group is generated when the A/D conversion of all virtual channels assigned to that scan group is completed.
5	Until the A/D conversion stop process is performed, Step 2 to 4 are repeated, and the scanning operation continues. To stop A/D conversion, you must follow to section 36.5.4. Force Stops the A/D Conversion Operation.



Note: VC: Virtual Channel
SPL: Sampling state
CNV: Conversion state

Figure 36.10 Example of the scanning operation in Oversampling mode – Continuous scan mode

36.3.7.5 Hybrid Mode – Single Scan Mode

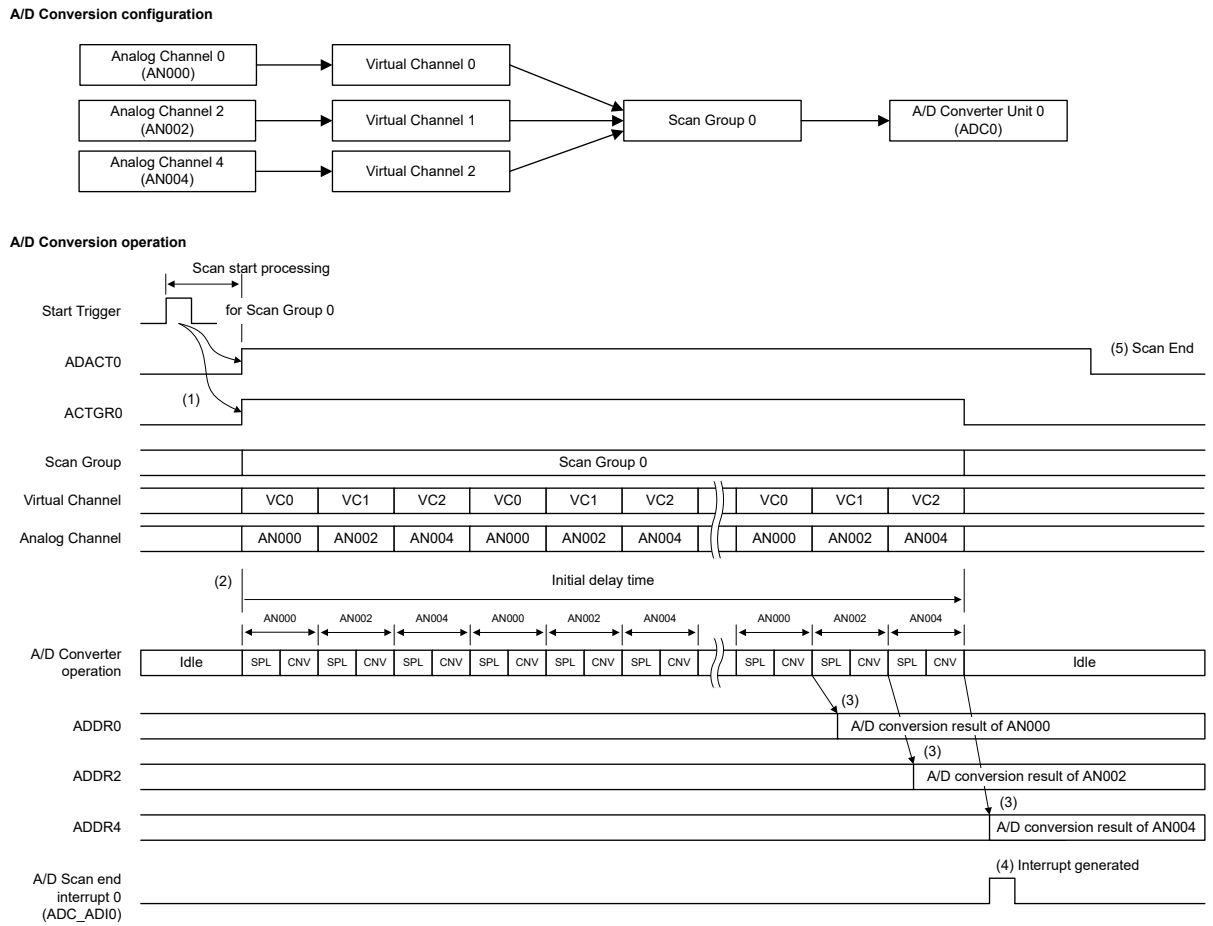
Table 36.11 and Figure 36.11 show the operation example in Hybrid mode – Single scan mode.

Table 36.11 Example of the scanning operation in Hybrid mode – Single scan mode (1 of 2)

Step	Operation
1	When a software trigger or a trigger from a peripheral module is input, the scanning operation of the scan group corresponding to the trigger is started.

Table 36.11 Example of the scanning operation in Hybrid mode – Single scan mode (2 of 2)

Step	Operation
2	In Hybrid mode, the scanning operation is performed while switching analog channels every time oversampling is performed.
3	After the oversampling corresponding to the number of TAPs of the digital filter and the number of A/D-converted value addition/averaging times is performed to each analog channel, A/D conversion data is output. (The time taken for the oversampling required to obtain the first A/D conversion data is called the initial delay time.) The A/D conversion data is stored in the data register (ADDRi (i = 0 to 28), ADEXDRj (j = 0 to 2, 5 to 8)). If FIFO is used, A/D conversion data is also stored in FIFO data register (ADFIFODRk (k = 0 to 8)).
4	If the scan end interrupt is enabled, the scan end interrupt corresponding to that scan group is generated when the A/D conversion of all virtual channels assigned to that scan group is completed.
5	During the scanning operation, ADGRSR.ACTGRn (n = 0 to 8) bit corresponding to that scan group is set to 1. ADSR.ADACTm (m = 0, 1) bit corresponding to the A/D converter performing the A/D conversion is also set to 1. When scanning operation is complete, each bit is cleared to 0 and A/D converter enters the idle state. And the oversampled data in the digital filter are also discarded at the end of the scanning operation.



Note: VC: Virtual Channel
SPL: Sampling state
CNV: Conversion state

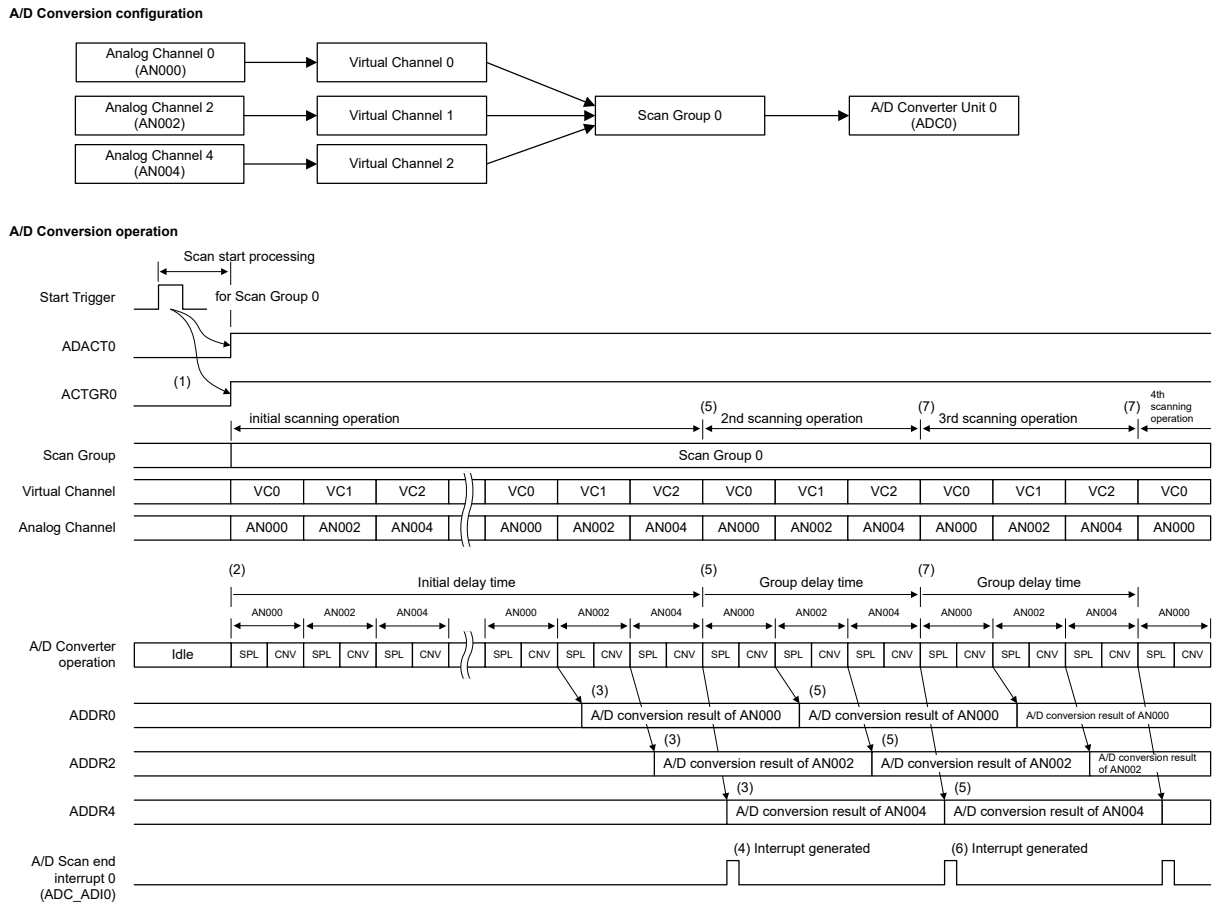
Figure 36.11 Example of the scanning operation in Hybrid mode – Single scan mode

36.3.7.6 Hybrid Mode – Continuous Scan Mode

Table 36.12 and Figure 36.12 show the operation example in Hybrid mode – Continuous scan mode.

Table 36.12 Example of the scanning operation in Hybrid mode – Continuous scan mode

Step	Operation
1	When a software trigger or a trigger from a peripheral module is input, the scanning operation of the scan group corresponding to the trigger is started. When scanning operation starts, ADGRSR.ACTGRn (n = 0 to 8) corresponding to that scan group is set to 1. ADSR.ADACTm (m = 0, 1) bit corresponding to the A/D converter performing the A/D conversion is also set to 1.
2	In Hybrid mode, the scanning operation is performed while switching analog channels every time oversampling is performed.
3	After the oversampling corresponding to the number of TAPs of the digital filter and the number of A/D-converted value addition/averaging times is performed to each analog channel, A/D conversion data is output. (The time taken for the oversampling required to obtain the first A/D conversion data is called the initial delay time.) The A/D conversion data is stored in the data register (ADDRi (i = 0 to 28), ADEXDRj (j = 0 to 2, 5 to 8)). If FIFO is used, A/D conversion data is also stored in FIFO data register (ADFIFODRk (k = 0 to 8)).
4	If the scan end interrupt is enabled, the scan end interrupt corresponding to that scan group is generated when the A/D conversion of all virtual channels assigned to that scan group is completed.
5	The second and subsequent rounds of scanning operations are performed while the oversampling data stored in the digital filter is retained. Each one time oversampling or each multiple times oversampling corresponding to the number of A/D-converted value addition/averaging times is performed for each analog channel, the next A/D conversion data is output. (The time taken for the oversampling required to obtain the second and subsequent A/D conversion data in continuous scan operation is called the group delay time.) The A/D conversion data is stored in the data register (ADDRi (i = 0 to 28), ADEXDRj (j = 0 to 2, 5 to 8)). If FIFO is used, A/D conversion data is also stored in FIFO data register (ADFIFODRk (k = 0 to 8)).
6	If the scan end interrupt is enabled, the scan end interrupt corresponding to that scan group is generated when the A/D conversion of all virtual channels assigned to that scan group is completed.
7	Thereafter, until the A/D conversion stop process is performed, Step 5 to 6 are repeated, and the scanning operation continues. To stop the A/D conversion, you must follow to section 36.5.4. Force Stops the A/D Conversion Operation.



Note: VC: Virtual Channel
SPL: Sampling state
CNV: Conversion state

Figure 36.12 Example of the scanning operation in Hybrid mode – Continuous scan mode

36.3.7.7 Hybrid Mode – Background Continuous Scan Mode

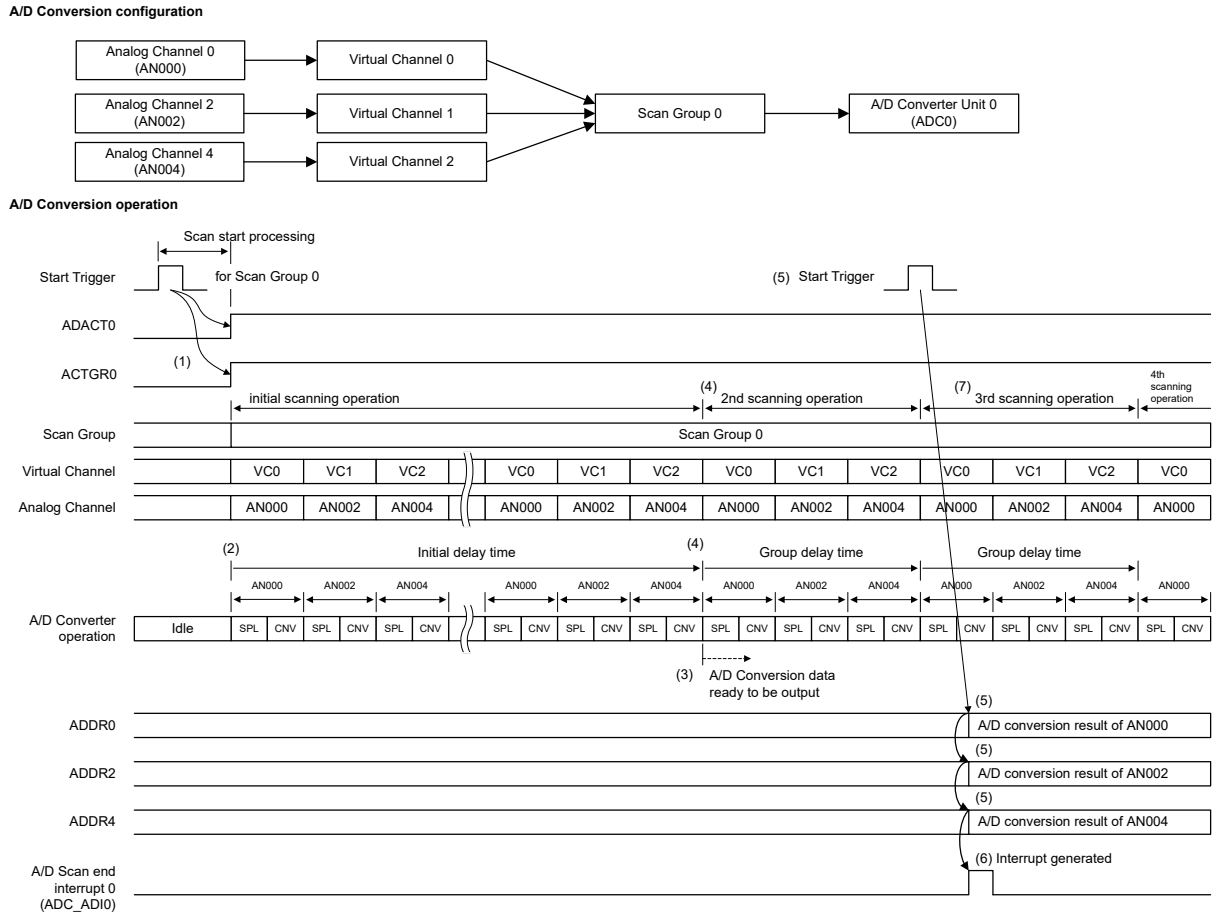
Table 36.13 and Figure 36.13 show the operation example in Hybrid mode – Background continuous scan mode.

Table 36.13 Example of the scanning operation in Hybrid mode – Background continuous scan mode (1 of 2)

Step	Operation
1	When a software trigger or a trigger from a peripheral module is input, the scanning operation of the scan group corresponding to the trigger is started. When scanning operation starts, ADGRSR.ACTGRn (n = 0 to 8) corresponding to that scan group is set to 1. ADSR.ADACTm (m = 0, 1) bit corresponding to the A/D converter performing the A/D conversion is also set to 1.
2	In Hybrid mode, the scanning operation is performed while switching analog channels every time oversampling is performed.
3	After the oversampling corresponding to the number of TAPs of the digital filter and the number of A/D-converted value addition/averaging times is performed to each analog channel, A/D conversion data becomes in ready to be output. (The time taken for the oversampling required to obtain the first A/D conversion data is called the initial delay time. In the background continuous scan operation, A/D conversion data can be obtained after the initial delay time has elapsed from the start of the scanning operation.)

Table 36.13 Example of the scanning operation in Hybrid mode – Background continuous scan mode (2 of 2)

Step	Operation
4	The second and subsequent rounds of scanning operations are performed while the oversampling data stored in the digital filter is retained. Each time oversampling is performed for each analog channel, the data in the digital filter is updated. Each one time oversampling or each multiple times oversampling corresponding to the number of A/D-converted value addition/ averaging times is performed for each analog channel, the next A/D conversion data becomes in ready to be output. (The time taken for the oversampling required to obtain the second and subsequent A/D conversion data in continuous scan operation is called the group delay time. In background continuous scan operation, after the initial delay time has elapsed from the start of the scanning operation, the updated A/D conversion data can be obtained every time the group delay time elapses.)
5	When an A/D conversion start trigger is input during background continuous scanning operation, the most recent A/D conversion data at that time is stored in the data register (ADDRi (i = 0 to 28), ADEXDRj (j = 0 to 2, 5 to 8)). If FIFO function is set to enabled, A/D conversion data is also stored in the FIFO data register (ADFIFODRk (k = 0 to 8)).
6	If scan end interrupt is set to enable, scan end interrupt is generated.
7	Thereafter, until the A/D conversion stop process is performed, Background continuous scanning operation (Step 4) is repeated. And whenever the A/D conversion start trigger is input during Background continuous scanning operation, the A/D conversion data is output (Step 5 and Step 6). To stop A/D conversion, you must follow to section 36.5.4. Force Stops the A/D Conversion Operation.



Note: VC: Virtual Channel
SPL: Sampling state
CNV: Conversion state

Figure 36.13 Example of the scanning operation in Hybrid mode – Background continuous scan mode

36.3.8 Self-calibration

ADC has a built-in self-calibration function. Self-calibration function is used to calibrate the variations of characteristics caused by the chip-by-chip variations. Self-calibration function has the following operations.

1. Internal Circuit Calibration
   - Adjusts the operation of A/D converter's internal circuitry.
2. Gain and Offset Calibration
   - Measure the A/D converter's Gain Error and Offset Error.
   - The calibration processing of the A/D conversion result based on the measured error data is performed after the A/D conversion.
     For details, see section 36.4.3.1. Gain Error and Offset Error Calibration.
   - This self-calibration should be performed after Internal-circuit self-calibration is completed.
3. Channel-dedicated sample-and-hold circuit Gain and Offset Calibration
   - Measure the Gain Error and the Offset Error when using the channel-dedicated sample-and-hold circuit.
   - The calibration processing of the A/D conversion result based on the measured error data is performed after the A/D conversion.
     For details, see section 36.4.3.1. Gain Error and Offset Error Calibration.
   - This self-calibration should be performed after A/D converter Gain/Offset self-calibration is completed.

36.3.8.1 Conditions under Which Self-calibration is Required

The conditions under which self-calibration is required are shown in Table 36.14. Under the conditions shown in Table 36.14, self-calibration should be performed before using the A/D converter. If A/D converter is operating, stop all A/D converters and perform the self-calibration again. If the required self-calibration is not performed, the A/D conversion result is not guaranteed.

Table 36.14 List of conditions under which Self-calibration should be performed

Conditions under which self-calibration is required	Internal-circuit Calibration	Gain/Offset Calibration	Channel-dedicated sample-and-hold circuit Gain and Offset Calibration *1
After reset release	✓	✓	✓
After releasing the module stop	✓	✓	✓
When returning from Software Standby mode or Deep Software Standby mode	✓	✓	✓
When changed the ADCLK setting (When clock source or frequency is changed)	✓	✓	✓
When changed the operation mode or the scan mode of the A/D converter (When changed ADMDR.ADMDm bit (m = 0, 1))	✓	✓	✓
When changed A/D successive approximation time (When changed ADCNVSTR.CSTm bit (m = 0, 1))	✓	✓	✓
When changed the operation setting of the channel-dedicated sample-and-hold circuit *1	—	—	✓

Note: ✓: Self-calibration should be performed.
—: Self-calibration is not required.

Note 1. When any of ADSHCRm.SHENn bits are set to 1, or any of ADSHCRm.SHENn bits are changed, perform the channel-dedicated sample-and-hold circuits self-calibration (m = 0, 1, n = 0 to 2, 4 to 6). Not required if the channel-dedicated sample-and-hold circuits are not used.

36.3.8.2 Self-calibration Procedure

Table 36.15 shows the self-calibration procedures.

Table 36.15 Procedure for self-calibration

No.	Step	Description
1	Disable trigger input	Disables the trigger input from the peripheral module. (Write ADTRGENR.STTRGENn = 0)
2	Waiting for all A/D converters to stop	Check that all A/D converters are stopped. If A/D conversion is operating, wait until the A/D converter stops by either of the following: ● Wait until all A/D conversions are complete ● Force the A/D converter to stop the operation using the ADSTOPR register. For details on forced stopping of A/D conversion operation, see section 36.5.4. Force Stops the A/D Conversion Operation.
3	Setting the number of states for self-calibration	Set the number of states for the self-calibration of A/D converter to ADCALSTCR register. The values in ADCALSTCR register are commonly used for ADC0 and ADC1. Set the value to be set in ADCALSTCR register to satisfy the range defined in section 46, Electrical Characteristics.
4	Clearing the Error Status Flag	If error flags are detected, clear them before performing the self-calibration. (A/D converter error flag, A/D converter overflow flag, etc.)
5	Performing the self-calibration for ADC0	Performs the self-calibration for ADC0. ● Internal-circuit Calibration ● Gain/Offset Calibration After starting the self-calibration, wait until it is completed.
6	Performing the self-calibration for ADC1	Performs the self-calibration for ADC1 in the same as Step 5.
7	Setting the self-calibration of channel-dedicated sample-and-hold circuit *1	Make settings for self-calibration of channel-dedicated sample-and-hold circuit. Enable channel-dedicated sample-and-hold circuit. *2 Sets the number of states for channel-dedicated sample-and-hold self-calibration (ADCALSTCR register *3 and ADCALSHCR register).
8	Performing channel-dedicated sample-and-hold circuit self-calibration for SH0 to SH2. *1	Performs the self-calibration for channel-dedicated sample-and-hold circuit (SH0 to SH2) connected to ADC0. ● Channel-dedicated sample-and-hold circuit Gain/Offset Calibration After starting self-calibration, wait until it is completed.
9	Performing channel-dedicated sample-and-hold circuit self-calibration for SH4 to SH6 *1	Performs the self-calibration for channel-dedicated sample-and-hold circuit (SH4 to SH6) connected to ADC1 in the same as Step 8.
10	Error Status Check *4	Check for errors in self-calibration. If no operating error occurs, self-calibration is complete.

Note: In order to increase the effectiveness of self-calibration, it is recommended to perform self-calibration for the A/D converters one by one.
Note 1. Not required if the channel-dedicated sample-and-hold circuits are not used.
Note 2. Enable all channel-dedicated sample-and-hold circuits connected to ADCm (m = 0, 1).
Note 3. When the self-calibration is performed for the channel-dedicated sample-and-hold circuitry, the number of sampling states at the self-calibration of the A/D converter (ADCALSTCR.CALADSST[9:0] bits) may also need to be changed. Set the value that satisfies the electrical characteristics.
Note 4. If the A/D converter error is detected by the self-calibration, it may be set outside the operation guaranteed range specified in the electric characteristics. Check the operation setting.

36.3.8.3 Restrictions on Self-calibration

(1) Prohibition of the scanning operation during self-calibration

Do not start the scanning operation during self-calibration. When performing the scanning operation, start the scanning operation after self-calibration is completed. Operation is not guaranteed when the scanning operation is started during self-calibration.

(2) Prohibition of additional writes to the ADCALSTR register during self-calibration

After the self-calibration is started, writing to ADCALSTR register is prohibited until self-calibration is completed. Write to ADCALSTR register after self-calibration is completed. Operation is not guaranteed if this restriction is violated.

(3) Prohibition of forcibly stop during self-calibration

Do not stop forcibly with the ADSTOPR register during self-calibration. Even if the A/D conversion operation is to be stopped forcibly due to a system error or exception handling, be sure to wait until the self-calibration is complete. Operation is not guaranteed when forced to stop during self-calibration.

(4) Restrictions on self-calibration for channel-dedicated sample-and-hold circuit

To perform the self-calibration for the channel-dedicated sample-and-hold circuit, all channel-dedicated sample-and-hold circuits connected to the A/D converter should be enabled as follows.

- For ADC0: Set 1 for ADSHCR0.SHEN0 to SHEN2 bit
- For ADC1: Set 1 for ADSHCR1.SHEN4 to SHEN6 bit

If there is a channel-dedicated sample-and-hold circuit that is not used, set it disabled (ADSHCRm.SHENn = 0 (m = 0, 1, n = 0 to 2, 4 to 6) after self-calibration completes.

Operation is not guaranteed when the self-calibration for the channel-dedicated sample-and-hold circuit is performed while any channel-dedicated sample-and-hold circuit are disabled.

(5) Restrictions on the self-calibration

Self-calibration should be performed for the A/D converters one by one. During the self-calibration operation, the other A/D converters should be in idle (not in scan operation or self-calibration operation).

If this restriction is violated, the A/D conversion accuracy will be degraded due to noise during the self-calibration operation. In this case, the A/D converter characteristics are not guaranteed.

In addition, in order to enhance the effect of self-calibration, self-calibration operation should be performed under conditions with as little noise as possible.

(6) Restrictions on setting of the number of status for self-calibration

The number of states to be set in the ADCALSTCR and ADCALSHCR registers should be set to satisfy the values specified in the Electrical Characteristics. In addition, the number of states involved in self-calibration should be set to satisfy the following restrictions.

If these restrictions are violated, A/D conversion results are not guaranteed.

[Restrictions on setting of ADCALSTCR register]

- ADCALSTCR.CALADSST[9:0] bits
  - According to the type of self-calibration, set the values to satisfy those specified in the electrical characteristics.
  - If the setting value differs according to the type of self-calibration, change the register setting value for each self-calibration.
- ADCALSTCR.CALADCST[5:0] bits
  - Set the same value as ADCNVSTR.CSTm[5:0] (m = 0, 1) bits.
    (ADCALSTCR.CALADCST[5:0] = ADCNVSTR.CSTm[5:0])

[Restrictions on setting of ADCALSHCR register] (only when channel-dedicated sample-and-hold circuit is used)

- ADCALSHCR.CALSHSST[7:0] bits
  - Set the value equal to the value of the ADSHSTRm.SHSST[7:0] bits plus 1 (m = 0, 1).
    (ADCALSHCR.CALSHSST[7:0] = ADSHSTRm.SHSST[7:0] + 1)
  - If the setting values of ADSHSTR0.SHSST[7:0] and ADSHSTR1.SHSST[7:0] are different, change the register setting values of CALSHSST[7:0] for each self-calibration of ADC0 (SH0 to SH2) and ADC1 (SH4 to SH6).
- ADCALSHCR.CALSHHST[2:0] bits
  - Set the same value as ADSHSTRm.SHHST[2:0] (m = 0, 1) bits.

36.3.9 Analog Input Channel

For the correspondence between the analog input pins and the analog channel numbers, the A/D converters, Programmable Gain Amplifier, and channel-dedicated sample-and-hold circuit, see Table 36.4.

36.3.10 Extended Analog Function

Extended Analog Function performs A/D conversion on the internal signal source of MCU. The available sources for Extended Analog Function are shown in the following:

- Self-diagnosis voltage
- Internal Reference Voltage
- Temperature Sensor
- D/A Converter Output (Internal Output Path to ACMPHS)

36.3.11 Self-diagnosis Function

This A/D converter has a built-in self-diagnosis function. The self-diagnosis function inputs the self-diagnosis voltage to the A/D converter and performs A/D conversion. The A/D conversion result of the self-diagnosis voltage can be used to confirm that the A/D converter is operating normally.

36.3.11.1 Self-diagnosis

This A/D converter has a built-in self-diagnosis function. The self-diagnosis function inputs the self-diagnosis voltage to the A/D converter and performs A/D conversion. The A/D conversion result of the self-diagnosis voltage can be used to confirm that the A/D converter is operating normally. Table 36.16 shows the operation mode in which the self-diagnostic function is available.

Table 36.16 List of operation mode for which the self-diagnostic function is available

Operation mode – Scan mode	Self-diagnosis Function
SAR mode – Single scan mode	✓
SAR mode – Continuous scan mode	—
Oversampling mode – Single scan mode	✓
Oversampling mode – Continuous scan mode	—
Hybrid mode – Single scan mode	✓
Hybrid mode – Continuous scan mode	—
Hybrid mode – Background continuous scan mode	—

Note: ✓: Available, —: Not Available

The procedure for performing self-diagnosis of A/D converter is shown in the following.

1. Assign the self-diagnosis channel to one of the virtual channels and configure the virtual channel settings.
2. Assign the virtual channel that the self-diagnosis channel is selected to one of the scan groups.
3. Set the self-diagnosis voltage in DIAGVAL[2:0] bits of ADSGDCRn (n = 0 to 8) register corresponding to the scan group assigned in 2 above.
4. Start the scanning operation by inputting the A/D conversion start trigger. The self-diagnosis operation is executed at the A/D conversion of the self-diagnosis channel.

Table 36.17 lists the operation settings when the self-diagnosis function is used. When using the self-diagnosis function, follow the setting in Table 36.17. If self-diagnosis operation is performed with settings other than those shown in Table 36.17, the self-diagnosis result (A/D conversion result) is not guaranteed.

Table 36.17 List of operation setting and expected value for self-diagnosis function

	Register setting value				Self-diagnosis operation		
	ADSGDCRx	ADCHCRy		ADDOPCRCy	Input Voltage to A/D converter		
Self-diagnosis mode	DIAGVAL[2:0]	CNVCS[6:0]	AINMD*6	SIGNSEL	Non-inverting input (+) (A_{INP})	Inverting input (−) (A_{INP})	Expected value of A/D conversion data *1 *2
Self-diagnosis mode 1	100b	0x60	1	0	VREFL0	VREFL0	0x0000 *3
Self-diagnosis mode 2	101b	0x60	1	0	VREFL0	VREFH0	0x8000 *4
Self-diagnosis mode 3	110b	0x60	1	0	VREFH0	VREFL0	0x7FFF *5

Note: x = 0 to 8, y = 0 to 36
Note 1. The expected value is the ideal A/D converter result stored in ADEXDR0.DATA[15:0] or ADFIFODRx.DATA[15:0].
Note 2. The A/D conversion data is the value when 16-bit data format (ADDOPCRCy.ADPRC[1:0] = 00b) is selected.
Note 3. The A/D conversion result becomes 0x0001 (+1) or more when a positive accuracy error occurs in A/D conversion, and becomes 0xFFFF (-1) or less when a negative accuracy error occurs.
Note 4. When an accuracy error occurs in A/D conversion, the A/D conversion result becomes greater than or equal to 0x8000.
Note 5. When an accuracy error occurs in A/D conversion, the A/D conversion result becomes less than or equal to 0x7FFF.
Note 6. In SAR mode, differential input is supported as an exception only during self-diagnostic operation.

36.3.11.2 Self-diagnosis Mode

(1) Self-diagnosis mode 1

In self-diagnosis mode 1, the non-inverting input (+) (A_{INP}) and inverting input (−) (A_{INN}) of A/D converter are input VREFH0 level. Since the differential voltage is 0, the ideal value of the self-diagnosis result is 0x0000 (±0). The A/D conversion result becomes 0x0001 (+1) or more when a positive accuracy error occurs in A/D conversion, and becomes 0xFFFF (−1) or less when a negative accuracy error occurs.

(2) Self-diagnosis mode 2

In self-diagnosis mode 2, the non-inverting input (+) (A_{INP}) of A/D converter is input VREFL0 level, and the inverting input (−) (A_{INN}) of A/D converter is input VREFH0 level. Since the differential voltage is −VREFH0 (if VREFL0 = 0 V), the ideal value of the self-diagnosis result is 0x8000 (−32768). When an A/D conversion error occurs, the A/D conversion result of self-diagnosis is greater than 0x8000.

(3) Self-diagnosis mode 3

In self-diagnosis mode 3, the non-inverting input (+) (A_{INP}) of A/D converter is input VREFH0 level, and the inverting input (−) (A_{INN}) of A/D converter is input VREFL0 level. Since the differential voltage is +VREFH0 (if VREFL0 = 0 V), the ideal value of the self-diagnosis result is 0x7FFF (+32767). When an A/D conversion error occurs, the A/D conversion result of self-diagnosis is less than 0x7FFF.

36.3.11.3 Notes on Self-diagnosis Function

(1) Notes on performing the self-diagnosis

The data register (ADEXDR0) to store the self-diagnosis result is shared by all A/D converters. When performing the self-diagnosis, ADEXDR0 register is overwritten by the result of the later self-diagnosis operation. If self-diagnosis operations of ADC0 and ADC1 are performed simultaneously, a conflict occurs when the self-diagnosis result is stored in ADEXDR0 register, and the self-diagnosis result of either ADC0 or ADC1 is lost. In addition, the value in the ADEXDR0 register cannot distinguish the data has been converted by which A/D converter.

To avoid this problem, it is recommended to perform self-diagnosis in one of the following methods:

[Recommended operation when self-diagnosis function is used]

- Perform self-diagnosis for each A/D converter one by one.
  - In order to distinguish which A/D converter's self-diagnosis result when reading the data register, schedule the self-diagnosis operation of each A/D converter to be exclusive.

- Use FIFO function for scan groups that include self-diagnosis channels.
  - Each scan group has its own FIFO. From the configuration between the scan group and the A/D converter, it is possible to distinguish which A/D converter the data read out of the FIFO was output by. Using this method, you can distinguish which A/D converter's self-diagnosis result.

(2) Notes on data format during self-diagnosis

When self-diagnosis is performed, if a data format other than 16-bit length is set, an overflow occurs due to A/D conversion data rounding, and the A/D conversion data error is detected. To perform self-diagnosis, select the 16-bit data format.

36.3.12 Internal Reference Voltage

ADC can perform A/D conversion of Internal Reference Voltage. Internal Reference Voltage outputs a constant voltage that is independent of the power supply voltage (VCC, VSS, AVCC, AVSS, VREFH0 and VREFL0). The monitoring of Internal Reference Voltage can be used to detect variations in the analog reference supply voltage (VREFH0 and VREFL0), or abnormalities or faults in the system.

When performing A/D conversion of Internal Reference Voltage, it is required to set the Internal Reference Voltage Monitor Enable Register (ADREFCR) and assign the analog channel of Internal Reference Voltage to any of the virtual channels. See Table 36.5 for the correspondence between Internal Reference Voltage function and the analog channels. For the sampling time when A/D conversion of Internal Reference Voltage is performed, see section 46, Electrical Characteristics.

36.3.13 Temperature Sensor

ADC can perform A/D conversion of Temperature Sensor. The chip temperature can be estimated from the A/D conversion result of Temperature Sensor. For more details on Temperature Sensor, see section 38, Temperature Sensor (TSN).

When A/D conversion of Temperature Sensor is performed, Temperature Sensor should be configured and the analog channel of Temperature Sensor should be assigned to any of the virtual channels. See Table 36.5 for the correspondence between Temperature Sensor and analog channels. For the sampling time when A/D conversion of Temperature Sensor is performed, see section 46, Electrical Characteristics.

36.3.14 D/A Converter

ADC can perform A/D conversion of the voltage that the D/A converter is outputting to the internal module (ACMPHS). When the output of the D/A converter is used for the reference input voltage of High-Speed Analog Comparator (ACMPHS), this function can be used to monitor the output voltage of the D/A converter. For details of the D/A converter function, see section 37, 12-Bit D/A Converter (DAC12).

When A/D conversion is performed on the output of D/A converter (DA0 to DA3) (output to ACMPHS), the D/A converter should be configured and the analog channel of the D/A converter should be assigned to one of the virtual channels. For the sampling time when A/D conversion is performed on the D/A converter output voltage, see section 46, Electrical Characteristics.

36.3.15 Programmable Gain Amplifier

ADC has built-in Programmable Gain Amplifier (PGA). The PGA amplifies an external analog input signal and outputs it to A/D converter, channel-dedicated sample-and-hold circuit, and High-Speed Analog Comparator (ACMPHS).

36.3.15.1 Configuration and Operation of PGA

The PGAs are able to select the single-ended input or the pseudo-differential input per unit. Figure 36.14 shows the internal configuration of the PGA in the Single-ended input mode and the Pseudo-differential input mode.

(1) Single-ended input

In Single-ended input mode, PGA amplifies the input from PGAIN pin with the specified gain. The gain is set with ADPGACRn.PGAGAIN[3:0] (n = 0 to 3) bits. Settable gains are ×2.000 to ×13.333. Table 36.18 shows the calculation formula of PGA output voltage.

When operating the PGA in Single-ended input mode, PGAIN should be connected to the signal source and PGAVSS should be connected to the analog ground (AVSS0). The input voltage to PGAIN must not exceed the range specified in the Electrical Characteristics.

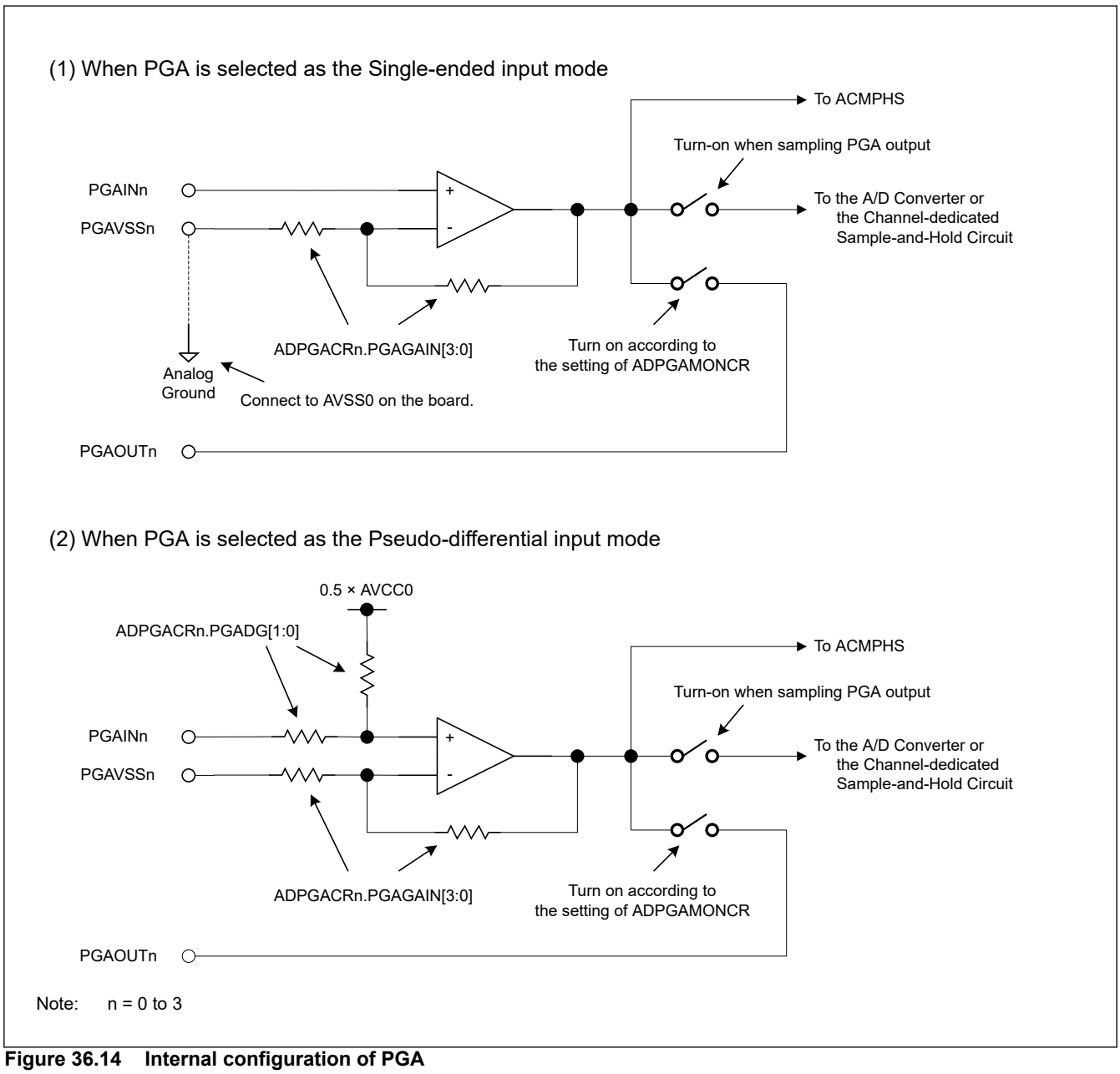
(2) Pseudo-differential input

In Pseudo-differential input mode, PGA amplifies the difference between PGAIN pin and PGAVSS pin with the specified gain and output the voltage obtained by adding the offset of AVCC×0.5 level. The gain is set by ADPGACRn.PGAGAIN[3:0] (n = 0 to 3) bits and ADPGACRn.PGADG[1:0] (n = 0 to 3) bits. Settable gains are ×1.500, ×2.333, ×4.000, and ×5.667. Set the same gain value to PGAGAIN[3:0] bits and PGADG[1:0] bits. Table 36.18 shows the calculation formula of PGA output voltage.

When operating the PGA in Pseudo-differential Input Mode, PGAIN should be connected to the signal source, and PGAVSS should be connected to the reference ground of the signal source. The inputs to PGAIN and PGAVSS pins must not exceed the range specified in the Electrical Characteristics.

(3) PGA output

The analog signal amplified by PGA can be used with A/D converter, channel-dedicated sample-and-hold circuit, and High-Speed Analog Comparator. When the PGA output is used by the A/D converter or the channel-dedicated sample-and-hold circuit, configure the virtual channel to convert the analog input channel connected to PGAIN as a signal source for single-ended input (ADCHCRn.AINMD = 0 (n = 0 to 36)).



Note: n = 0 to 3

Figure 36.14 Internal configuration of PGA

Table 36.18 PGA output voltage

PGA mode	PGA output voltage
Single-ended input	Gain × Vin
Pseudo-differential input	Gain × (Vin – Vs) + AVCC × 0.5

Note: Vin: PGAIN0 to PGAIN3, Vs: PGAVSS0 to PGAVSS3

36.3.15.2 PGA Operation Setting

Table 36.19 shows the combination of the PGA setting and the related functions that can be used.

Table 36.19 PGA settings and available related functions

Use Case	Function				Register Setting						
					PnmPFS*2	ADPGACRn (n = 0 to 3) *5					
		ACMPHS *1				b27-24	b21-20	b16	b3	b2	b1
	I/O Port	IVCMP3	IVCMP2	ADC	ASEL	PGAGAIN[3:0]	PGADG[1:0]	PGAGEN	PGAENAMP	PGASEL1	PGADEN *3
When using I/O Port	✓	—	—	—	0	0x0	00b	0	0	0	0
When using ACMPHS or ADC (not use PGA) *4	—	—	✓	✓ (ANx)	1	0x0	00b	1	0	0	0
When using PGA Single-ended input	—	✓	✓	✓ (PGA Output)	1	0x0 to 0xE	00b	1	1	1	0
When using PGA Pseudo Differential Input	—	✓	—	✓ (PGA Output)	1	0x1, 0x5, 0x9, 0xB	00b to 11b	1	1	1	1

Note: ✓: Available, —: Not available.
ANx: Analog input channel to which PGAINn or PGAVSSn (n = 0 to 3) is assigned
Note 1. ACMPHS.IVCMP2: When using ANx-input by-passing PGA
ACMPHS.IVCMP3: When PGA are used
Note 2. For details on PnmPFS register settings, see section 18, I/O Ports.
Note 3. The value after reset depends on the user-set value of section 6.2.3. OFS1, OFS1_SEC, OFS1_SEL : Option Function Select Register 1 in section 6, Option-Setting Memory.
Note 4. I/O Port and ACMPHS cannot be used at the same time.
I/O Port and ADC cannot be used at the same time.
Note 5. Only the combinations listed in the table can be set to ADPGACRn register.

36.3.15.3 PGA Output Monitor Function

To debug your system, the PGA's internal output can be output to the pin. If using this function, ADPGAMONCR register setting and the pin configuration of which PGAOUT function is assigned are required. For more details on the pin configuration, see section 18, I/O Ports.

Restrictions on the PGA Output Monitoring Function

The PGA output monitor function (hereinafter referred to as this function) is an auxiliary function for the purpose of program development and debugging of the MCU, and this function is prohibited to use for any other purpose. This function is not designed to drive any external circuit. Therefore, when this function is used, the PGA characteristics may deteriorate and unintentional operation may occur due to the effects of load or external noise caused by the terminal output, etc. For these reasons, functions, characteristics, and reliability are not guaranteed when this function is used. If using this function, be sure to give sufficient consideration to the safety of your equipment or system, and use this function at your responsibility. In addition, after the development of your equipment or system is completed, be sure to use the product with this function disabled.

36.3.15.4 Restrictions on PGA

When using a PGA, observe the following constraints:

(1) Limitations of the analog input path to which PGAIN and PGAVSS are assigned

When PGA is used, the analog input pin assigned to PGAVSS pin cannot be input to A/D conversion or channel-dedicated sample-and-hold circuit.

Regardless of the PGA is enabled or disabled, if the PGA is set to Pseudo-differential Input Mode, the analog inputs to which PGAIN and PGAVSS are assigned cannot be used by the A/D converter or channel-dedicated sample-and-hold circuit bypassing the PGA.

(2) Restrictions of the input-voltage range to pins to which PGAIN and PGAVSS are assigned

The inputs to the pins to which PGAIN and PGAVSS are connected must not exceed the range specified in the Electrical Characteristics.

Regardless of whether the PGA is enabled or disabled, when the PGA is configured in single-input mode, do not input a negative voltage to the analog pins to which PGAIN and PGAVSS are assigned.

For details of the voltage that can be applied to the pins, see section 46, Electrical Characteristics.

(3) Restrictions on PGA output monitor function

The PGA output monitor function is an auxiliary function for the purpose of program development and debugging of the MCU, and it is prohibited to use for any other purpose. When using PGA Output Monitor Function, functions, characteristics and reliability are not guaranteed. For details, see section 36.3.15.3. PGA Output Monitor Function.

36.3.16 Channel-dedicated Sample-and-hold Circuit

ADC has a built-in channel-dedicated sample-and-hold circuit. By using multiple channel-dedicated sample-and-hold circuits, it is possible to sample multiple analog inputs simultaneously.

The conditions under which channel-dedicated sample-and-hold circuit can be used are shown in Table 36.20.

Table 36.20 Operation mode in which channel-dedicated sample-and-hold circuits can be used

Operation mode	Channel-dedicated sample-and hold circuits
SAR mode – Single scan mode	✓
SAR mode – Continuous scan mode	—
Oversampling mode – Single scan mode	—
Oversampling mode – Continuous scan mode	—
Hybrid mode – Single scan mode	✓
Hybrid mode – Continuous scan mode	✓
Hybrid mode – Background continuous scan mode	✓

Note: ✓: available, —: not available

36.3.16.1 Configuration of Channel-dedicated Sample-and-hold Circuit

(1) Configuration in Single-ended input mode (without PGA)

This subsection describes the internal configuration when using channel-dedicated sample-and-hold circuit with single-ended inputs. Note that this section assumes that Programmable Gain Amplifier (PGA) is not connected or that the PGA is not used (bypassed). When using channel-dedicated sample-and-hold circuit in conjunction with PGAs, see (3)Configuration with PGA.

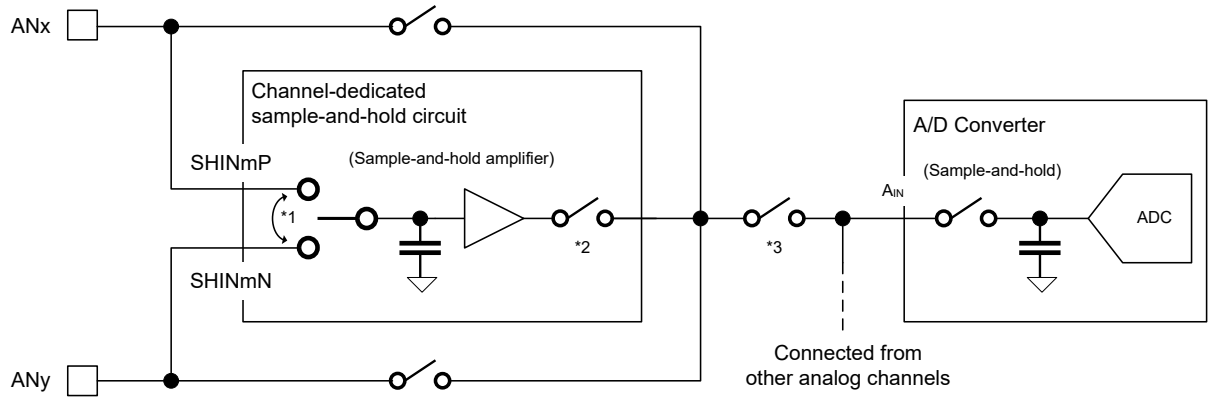
Figure 36.15 shows the configuration of channel-dedicated sample-and-hold circuit in Single-ended input mode. (no PGA or PGA bypass)

Channel-dedicated sample-and-hold circuits are connected to two analog input channels. When channel-dedicated sample-and-hold circuit is operating in Single-ended input mode, one of the two analogue inputs (ANx or ANy) is sampled and held in one scanning operation, and output to the A/D converter. The A/D conversion setting should be set to convert a single-ended input signal source. (ADCHCRn.AINMD = 0 (n = 0 to 36))

When channel-dedicated sample-and-hold circuit is operating in Single-ended input mode, while one of ANx or ANy is sampled and held, A/D conversion of the other (ANy or ANx) is prohibited. This means that when channel-dedicated

sample-and-hold circuit is enabled, both ANx and ANy cannot be A/D converted in the same scan group. (It is also impossible to convert one of ANx or ANy with channel-dedicated sample-and-hold circuit and to convert the other bypassing channel-dedicated sample-and-hold circuit.)

If you want to perform A/D conversion of both ANx and ANy with single-ended inputs while the channel-specific sample-and-hold circuit is enabled, ANx and ANy must be assigned to different scan groups for conversion.



Note m : The unit number of channel-dedicated sample-and-hold circuit
x : Even-numbered analog channel connected to the channel-dedicated sample-and-hold circuit
y : Odd-numbered analog channel connected to the channel-dedicated sample-and-hold circuit

Note 1. Turn on when the channel-dedicated sample-and-hold circuit is enabled and samples ANx or ANy channel.
Note 2. Turn on when the channel-dedicated sample-and-hold circuit is the holding state.
Note 3. Turn on when the A/D conversion (sampling state) of ANx, ANy or output of channel-dedicated sample-and-hold circuit.

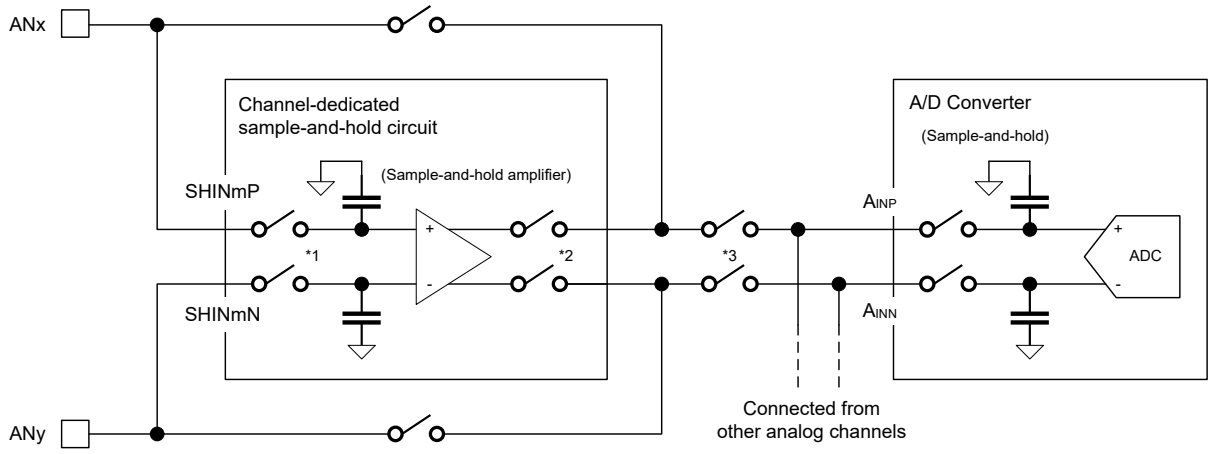
Figure 36.15 Internal configuration of channel-dedicated sample-and-hold circuit in Single-ended input mode

(2) Configuration in Differential input mode (without PGA)

This subsection describes the internal configuration when channel-dedicated sample-and-hold circuit is used with differential inputs. Note that this section assumes that Programmable Gain Amplifier (PGA) is not connected or that the PGA is not used (bypassed). When using channel-dedicated sample-and-hold circuit in conjunction with PGAs, see (3)Configuration with PGA.

Figure 36.16 shows the configuration of channel-dedicated sample-and-hold circuit in Differential input mode. (no PGA or PGA bypass)

When channel-dedicated sample-and-hold circuit is operating in Differential input mode, the even analog channel is connected to the non-inverting input (+) (SHINmP) and the odd-numbered analog channel is connected to the inverting input (−) (SHINmN). Two analog inputs (differential input) are sampled and held in one scanning operation, and output to the A/D converter. The A/D conversion setting should be set to convert a differential input signal source. (ADCHCRn.AINMD = 1 (n = 0 to 36))



Note m : The unit number of channel-dedicated sample-and-hold circuit
x : Even-numbered analog channel connected to the channel-dedicated sample-and-hold circuit
y : Odd-numbered analog channel connected to the channel-dedicated sample-and-hold circuit

Note 1. Turn on when the channel-dedicated sample-and-hold circuit is enabled and samples ANx or ANy channel.
Note 2. Turn on when the channel-dedicated sample-and-hold circuit is the holding state.
Note 3. Turn on when the A/D conversion (sampling state) of ANx, ANy or output of channel-dedicated sample-and-hold circuit.

Figure 36.16 Internal configuration of channel-dedicated sample-and-hold circuit in Differential input mode

(3) Configuration with PGA

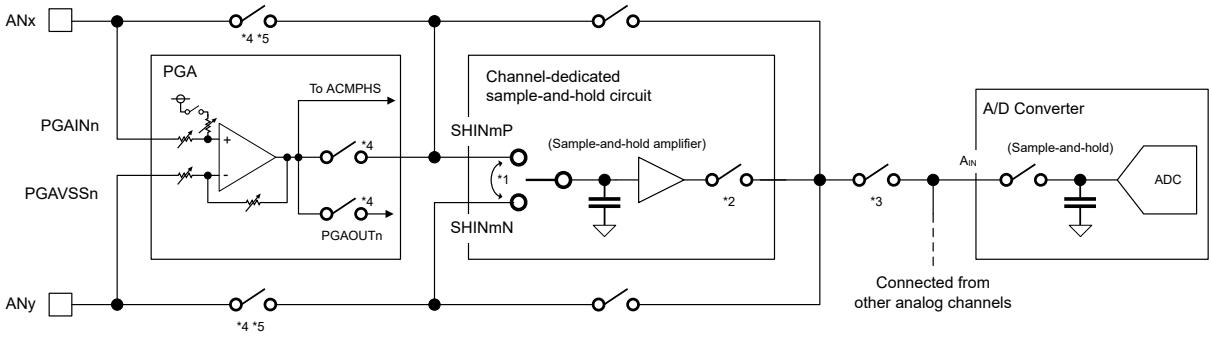
This subsection describes the internal configuration when channel-dedicated sample-and-hold circuit is used in conjunction with Programmable Gain Amplifier (PGAs). If PGA is not used (bypass), see (1)Configuration in Single-ended input mode (without PGA) or (2)Configuration in Differential input mode (without PGA).

Figure 36.17 shows the configuration that channel-dedicated sample-and-hold circuit is used in conjunction with PGA.

When channel-dedicated sample-and-hold circuit is enabled, the output of the PGA is input to the non-inverting input (+) (SHINxP) of channel-dedicated sample-and-hold circuit. Therefore, channel-dedicated sample-and-hold circuit must be operated in Single-ended input mode.

The PGA output is sampled and held at one scanning operation, and output to the A/D converter. The A/D conversion setting should be set to convert a single-ended input signal source.

When PGA is enabled, the analog path of PGAVSS side cannot be used as the inverting input (−) (SHINmN) of channel-dedicated sample-and-hold circuit.



Note m : The unit number of channel-dedicated sample-and-hold circuit
n : The unit number of Programmable Gain Amplifier
x : Even-numbered analog channel connected to the channel-dedicated sample-and-hold circuit
y : Odd-numbered analog channel connected to the channel-dedicated sample-and-hold circuit

Note 1. Turn on when the channel-dedicated sample-and-hold circuit is enabled and samples ANx or ANy channel.
Note 2. Turn on when the channel-dedicated sample-and-hold circuit is the holding state.
Note 3. Turn on when the A/D conversion (the sampling state) of ANx (PGA bypass or via PGA), ANy (PGA bypass) or channel-dedicated sample-and-hold circuit output.
Note 4. Turning on/off depends on the operating setting of the PGA.
Note 5 . Turn off whenever the PGA is selected as the pseudo-differential input mode.

Figure 36.17 Internal configuration of channel-dedicated sample-and-hold circuit with PGA

36.3.16.2 Operation in SAR Mode with Channel-dedicated Sample-and-hold Circuit

(1) Basic operation in SAR mode – Single scan mode (Single-ended input mode)

This paragraph describes the basic operation of channel-dedicated sample-and-hold circuit in SAR mode. The configuration and behavior of example operation show in Table 36.21, Table 36.22 and Figure 36.18.

Table 36.21 Configuration of example operation for channel-dedicated sample-and-hold circuit in SAR mode – Single scan mode

Item	Configuration
A/D converter	Set ADC0 to SAR mode – Single scan mode
Channel-dedicated sample-and-hold circuit	Set SH0 to SH2 to enabled and Single-ended input mode. (AN000 to AN005 are subject to A/D conversion using channel-dedicated sample-and-hold circuit.)
Virtual channel*1	Assign AN000, AN002, AN004 and AN020 to virtual channel 0 to 3, and set to Single-ended input mode.
Scan group	Assign virtual channel 0 to 3 to scan group 0. Set scan group 0 to convert with ADC0.

Note 1. When channel-dedicated sample-and-hold circuit is used, there are restrictions on the setting of virtual channels. For more details, see section 36.3.16.4. Restrictions on Channel-dedicated Sample-and-hold Circuit.

Table 36.22 Example operation of channel-dedicated sample-and-hold circuit in SAR mode – Single scan mode

Step	Detail behavior of example operation
1	The scanning operation of scan group 0 starts when a trigger for scan group 0 is input.
2	Channel-dedicated sample-and-hold circuit (SH0 to SH2) starts sampling the analog inputs (AN000, AN002, AN004) and holds them.
3	A/D conversion is performed on the voltages held by each channel-dedicated sample-and-hold circuit and on other analog input channels (AN020).
4	When A/D conversion of each channel completes, the A/D conversion result is stored in the corresponding A/D data register n (ADDRn).
5	If scan end interrupt in scan group 0 is set to enabled, a scan end interrupt occurs.

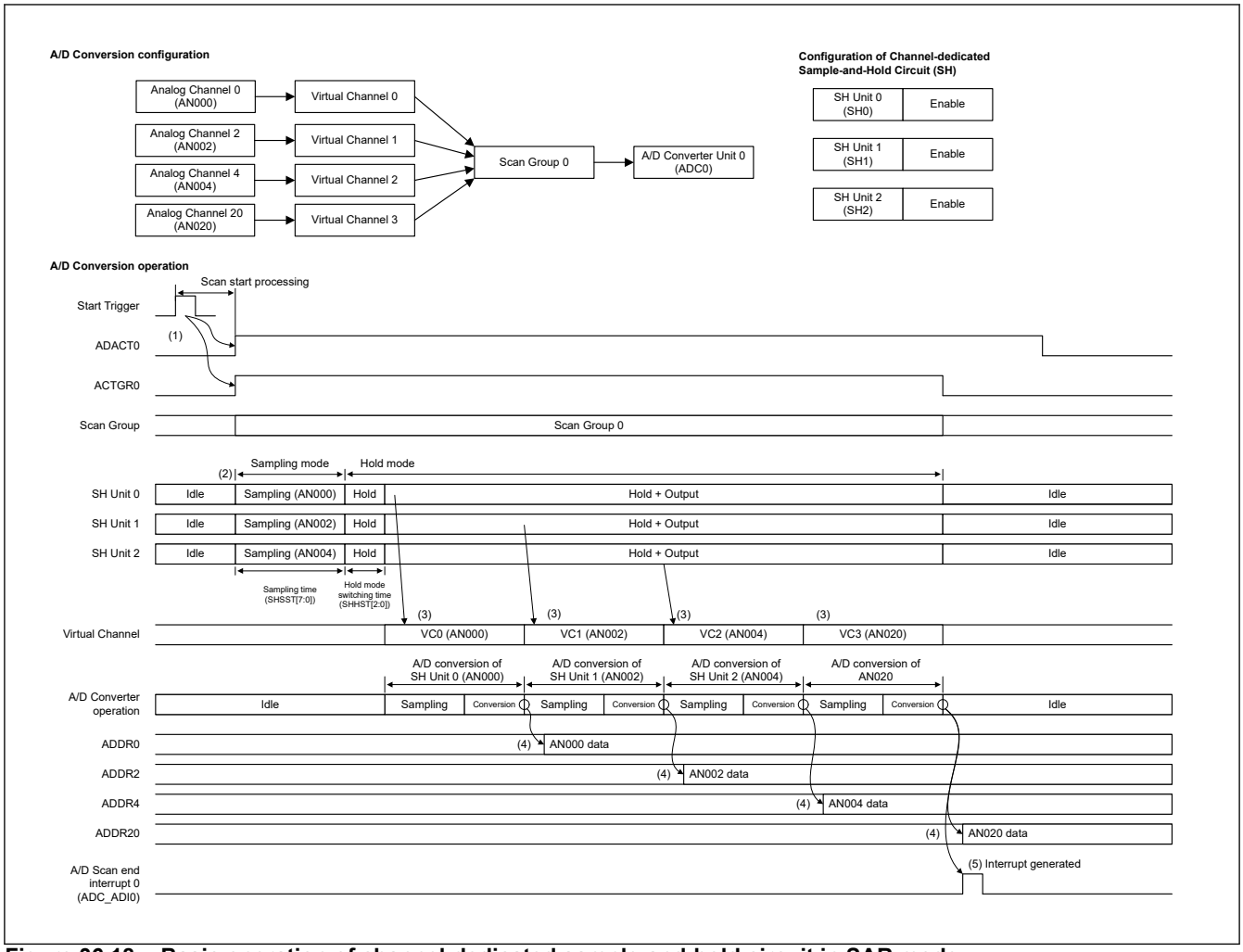


Figure 36.18 Basic operation of channel-dedicated sample-and-hold circuit in SAR mode

(2) Advanced operation in SAR mode - Single scan mode

This paragraph describes the operation of A/D conversion that sharing one channel-dedicated sample-and-hold circuit between two analog channels. The operation shown in this paragraph is possible only when all of the following conditions are met.

[Operating Conditions]

- Operation mode: SAR mode – Single scan mode
- Input method: Single-ended input mode
- Others: PGA is disabled if PGA is connected to channel-dedicated sample-and-hold circuit

The following shows an example of A/D conversions of even-numbered analog channels and odd-numbered analog channels using channel-dedicated sample-and-hold circuit in SAR mode – Single scan mode. The configuration example is shown in Table 36.23, and the operation detail is shown in Table 36.24 and Figure 36.19.

Table 36.23 Configuration example of advanced operation for channel-dedicated sample-and-hold circuit (1 of 2)

Item	Configuration
A/D converter	Set ADC0 to SAR mode – Single scan mode
Channel-dedicated sample-and-hold circuit	Set SH0 to SH2 to enabled and Single-ended input mode. (AN000 to AN005 are subject to A/D conversion using channel-dedicated sample-and-hold circuit.)

Table 36.23 Configuration example of advanced operation for channel-dedicated sample-and-hold circuit (2 of 2)

Item	Configuration
Virtual channel*1	• Assign AN000, AN002 and AN004 to virtual channel 0 to 2, and set to single-ended input mode. • Assign AN001, AN003 and AN005 to virtual channel 3 to 5, and set to single-ended input mode.
Scan group	• Scan group 0: Assign AN000, AN002, AN004 (virtual channels 0 to 2). • Scan group 1: Assign AN001, AN003, AN005 (virtual channels 3 to 5). • Set scan group 0 and scan group 1 to convert with ADC0.

Note 1. When channel-dedicated sample-and-hold circuit is used, there are restrictions on the setting of virtual channels. For more details, see section 36.3.16.4. Restrictions on Channel-dedicated Sample-and-hold Circuit.

Table 36.24 Example advanced operation of channel-dedicated sample-and-hold circuit

Step	Detail behavior of example operation
1	The scanning operation of scan group 0 starts when a trigger for scan group 0 is input.
2	Channel-dedicated sample-and-hold circuit (SH0 to SH2) starts sampling the analog inputs (AN000, AN002, AN004) and holds them.
3	A/D conversion is performed on the voltages held by channel-dedicated sample-and-hold circuit.
4	When A/D conversion of each channel in scan group 0 completes, the A/D conversion result is stored in the corresponding A/D data register n (ADDRn).
5	If scan end interrupt in scan group 0 is set to enabled, a scan end interrupt occurs.
6	After scanning of scan group 0 is completed, the scanning operation of scan group 1 starts when a trigger for scan group 1 is input.
7	Channel-dedicated sample-and-hold circuit (SH0 to SH2) starts sampling the analog inputs (AN001, AN003, AN05) and holds them.
8	A/D conversion is performed on the voltages held by channel-dedicated sample-and-hold circuit.
9	When A/D conversion of each channel in scan group 1 completes, the A/D conversion result is stored in the corresponding A/D data register n (ADDRn).
10	If scan end interrupt for scan group 1 is set to enabled, a scan end interrupt occurs.

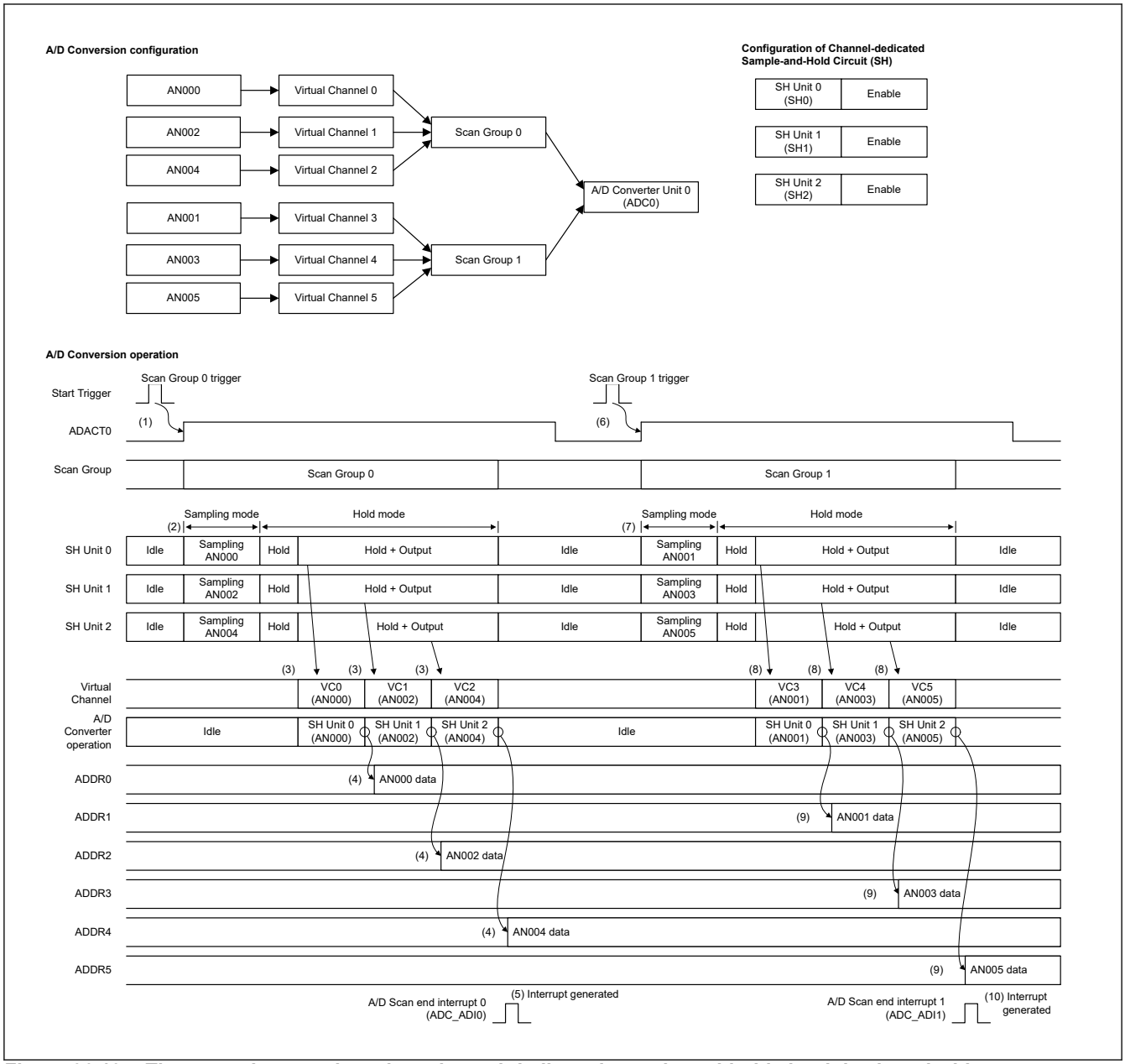


Figure 36.19 The example operation when channel-dedicated sample-and-hold circuit is shared with two analog channels in SAR mode

36.3.16.3 Operation in Hybrid Mode with Channel-dedicated Sample-and-hold Circuit

(1) Limitations on usage of channel-dedicated sample-and-hold circuit in Hybrid mode

If you use the channel-dedicated sample-and-hold circuits in Hybrid mode, the virtual channels and the scan groups are constrained to the usages shown in Table 36.25.

Table 36.25 The usage limitations on channel-dedicated sample-and-hold circuits in Hybrid mode (1 of 2)

Usage	Constraints on assignment of virtual channels to scan group
No.1	VCn: ANu (with S&H) VC (n + 1): ANv (with S&H) VC (n + 2): Dummy conversion channel
No.2	VCn: ANu (with S&H) VC (n + 1): ANv (with S&H) VC (n + 2): ANi (with S&H) VC (n + 3): Dummy conversion channel

Table 36.25 The usage limitations on channel-dedicated sample-and-hold circuits in Hybrid mode (2 of 2)

Usage	Constraints on assignment of virtual channels to scan group
No.3	VCn: ANu (with S&H) VC (n + 1): ANv (with S&H) VC (n + 2): ANi (without S&H) VC (n + 3): Dummy conversion channel

Note: VC: Virtual channel
n: Any virtual channel number
u, v, w: The channel number of any analog channel that use channel-dedicated sample-and-hold circuit.
i: The channel number of any analog channel that does not use channel-dedicated sample-and-hold circuit.
Dummy conversion channel: Any analog channel whose A/D conversion data does not used.

In Table 36.25, the dummy conversion channel means any analog channel that does not use A/D conversion data. The A/D conversion data of the analog channel that is used as the dummy conversion channels is not guaranteed (the A/D conversion accuracy will be degraded remarkably).

You must specify a self-diagnosis channel with the following settings for the dummy conversion channel:

[The example of configurations for the dummy conversion channel]

- Select the self-diagnosis channel (ADCHCRn.CNVCS[6:0] = 0x60 (n = 0 to 36))
- Differential input mode (ADCHCRn.AINMD = 1 (n = 0 to 36))
- Signed data format (ADDOPCRCn.SIGNSEL = 0 (n = 0 to 36))
- 16-bit data format (ADDOPCRCn.ADPRC[1:0] = 00b (n = 0 to 36))
- Self-diagnosis mode 1 (ADSGDCRm.DIAGVAL[2:0] = 100b (m = 0 to 8))
- Do not use User's Gain function and User's Offset function (ADDOPCRAn.GAINSEL[3:0] = 0000b, ADDOPCRAn.OFSETSEL[3:0] = 0000b)
- Other settings for the virtual channel should be selected appropriate ones.

(2) Operation in Hybrid mode

Basic Operation in Hybrid Mode - Single Scan Mode

This paragraph shows the basic operation of channel-dedicated sample-and-hold circuit in Hybrid mode – Single scan mode. Since Differential input mode is supported in Hybrid mode, this example describes a case that channel-dedicated sample-and-hold circuits are set to Differential input mode. The configuration and behavior of example operation show in Table 36.26, Table 36.27 and Figure 36.20.

Table 36.26 Configuration of example operation for channel-dedicated sample-and-hold circuit in Hybrid mode – Single scan mode

Item	Configuration
A/D converter	Set ADC0 to Hybrid mode – Single scan mode
Channel-dedicated sample-and-hold circuit	Set SH0 to SH2 to enabled and Differential input mode. (AN000 to AN005 are subject to A/D conversion using channel-dedicated sample-and-hold circuit.)
Virtual channel*1	• Assign AN000, AN002 and AN004 to virtual channel 0 to 2, and set to Differential input mode. (This will function AN000, AN002 and AN004 as the non-inverting inputs (+) of the differential input, and AN001, AN003, and AN005 as the inverting inputs (−) of the differential input.) • Assign self-diagnosis channel as the dummy conversion channel to virtual channel 3, and set to Differential input mode and Self-diagnosis mode 1.
Scan group*1	• Assign virtual channel 0 to 3 to scan group 0. • Set scan group 0 to convert with ADC0. • Set to perform Self-diagnostic mode 1 in scan group 0.

Note 1. When channel-dedicated sample-and-hold circuit is used, there are restrictions on the setting of virtual channels. For more details, see (1)Limitations on usage of channel-dedicated sample-and-hold circuit in Hybrid mode and section 36.3.16.4. Restrictions on Channel-dedicated Sample-and-hold Circuit.

Table 36.27 Example operation of channel-dedicated sample-and-hold circuit in Hybrid mode – Single scan mode

Step	Behavior
1	The scanning operation of scan group 0 starts when a trigger for scan group 0 is input.
2	Channel-dedicated sample-and-hold circuit (SH0 to SH2) starts sampling the analog inputs (AN000, AN002, AN004) and holds them.
3	The A/D converter performs oversampling once for each output of channel-dedicated sample-and-hold circuit (SH0 to SH2) and once for each of the dummy conversion channel (self-diagnosis).
4	After one oversampling for each analog channel, channel-dedicated sample-and-hold circuit releases the held voltage and enters the idle state.
5	Step 2 to 4 is repeated until oversampling corresponding to the number of TAPs of the digital filter and the number of A/D-converted value addition/average times is performed for each analog channel.
6	A/D conversion data corresponding to each analog channel is output. The A/D conversion data is stored in the data register (ADDRi (i = 0 to 28), ADEXDRj (j = 0 to 2, 5 to 8)). If FIFO is used, A/D conversion data is also stored in FIFO data register (ADFIFODRk (k = 0 to 8)). However, the A/D conversion data of the channel used as the dummy conversion channel is not guaranteed.
7	If the scan end interrupt is enabled, the scan end interrupt corresponding to that scan group is generated when the A/D conversion of all virtual channels assigned to that scan group is completed.
8	A/D converter and channel-dedicated sample-and-hold circuit enter the idle state when scanning operation is completed. The oversampling data in digital filter are discarded at the end of the scanning operation.

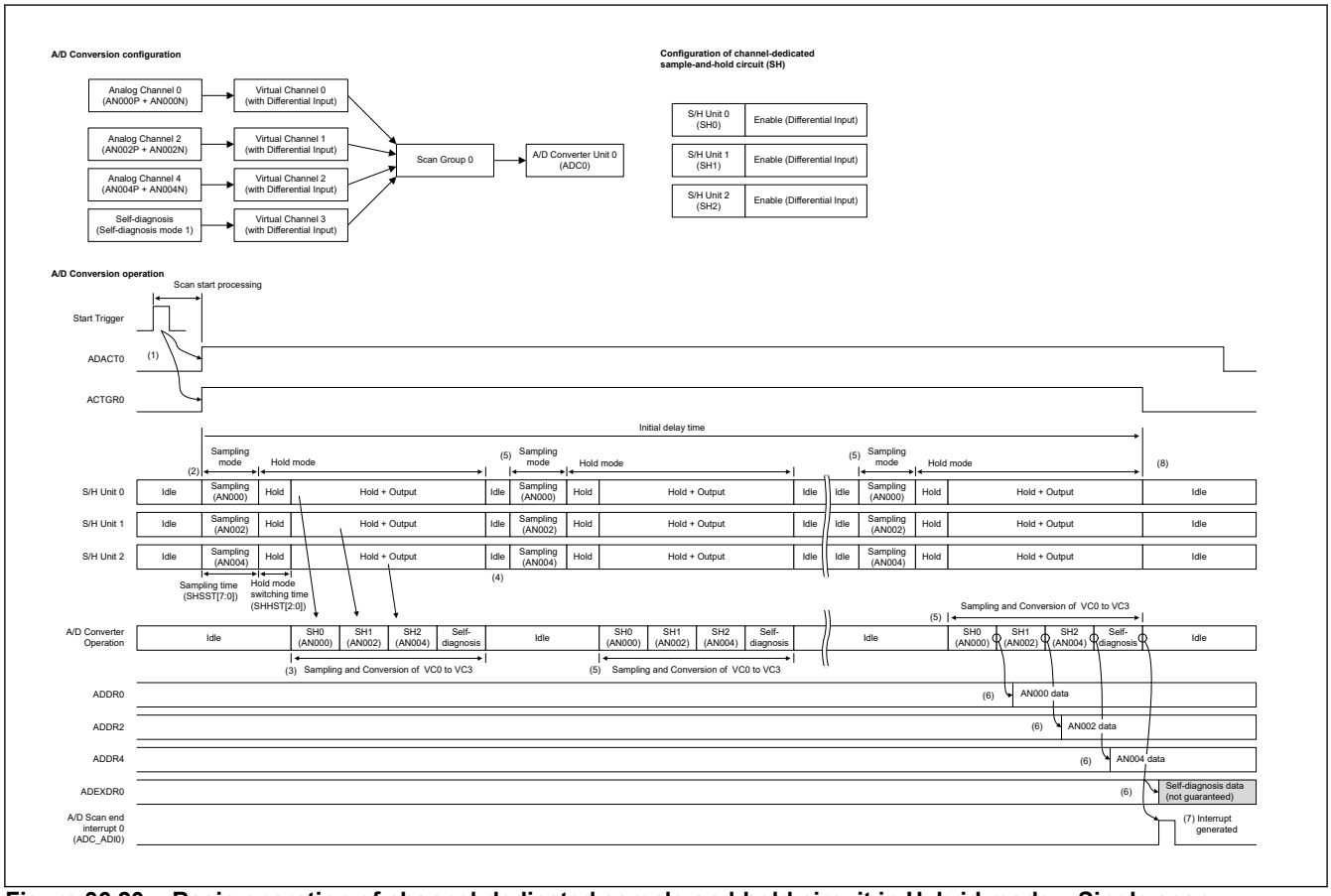


Figure 36.20 Basic operation of channel-dedicated sample-and-hold circuit in Hybrid mode – Single scan mode

Basic Operation in Hybrid Mode - Continuous Scan Mode

This paragraph shows the basic operation of channel-dedicated sample-and-hold circuit in Hybrid mode – Continuous scan mode. Since Differential input mode is supported in Hybrid mode, this example describes a case that channel-dedicated sample-and-hold circuits are set to Differential input mode. The configuration and behavior of example operation show in Table 36.28, Table 36.29 and Figure 36.21.

Table 36.28 Configuration of example operation for channel-dedicated sample-and-hold circuit in Hybrid mode – Continuous scan mode

Item	Configuration
A/D converter	Set ADC0 to Hybrid mode – Continuous scan mode
Channel-dedicated sample-and-hold circuit	Set SH0 to SH2 to enabled and Differential input mode. (AN000 to AN005 are subject to A/D conversion using channel-dedicated sample-and-hold circuit.)
Virtual channel*1	• Assign AN000, AN002 and AN004 to virtual channel 0 to 2, and set to Differential input mode. (This will function AN000, AN002 and AN004 as the non-inverting inputs (+) of the differential input, and AN001, AN003, and AN005 as the inverting inputs (−) of the differential input.) • Assign self-diagnosis channel as the dummy conversion channel to virtual channel 3, and set to Differential input mode and Self-diagnosis mode 1.
Scan group*1	• Assign virtual channel 0 to 3 to scan group 0. • Set scan group 0 to convert with ADC0. • Set to perform Self-diagnostic mode 1 in scan group 0.

Note 1. When channel-dedicated sample-and-hold circuit is used, there are restrictions on the setting of virtual channels. For more details, see (1)Limitations on usage of channel-dedicated sample-and-hold circuit in Hybrid mode and section 36.3.16.4. Restrictions on Channel-dedicated Sample-and-hold Circuit.

Table 36.29 Example operation of channel-dedicated sample-and-hold circuit in Hybrid mode – Continuous scan mode

Step	Behavior
1	The scanning operation of scan group 0 starts when a trigger for scan group 0 is input.
2	Channel-dedicated sample-and-hold circuit (SH0 to SH2) starts sampling the analog inputs (AN000, AN002, AN004) and holds them.
3	The A/D converter performs oversampling once for each output of channel-dedicated sample-and-hold circuit (SH0 to SH2) and once for each of the dummy conversion channel (self-diagnosis).
4	After one oversampling for each analog channel, channel-dedicated sample-and-hold circuit releases the held voltage and enters the idle state.
5	Step 2 to 4 is repeated until oversampling corresponding to the number of TAPs of the digital filter and the number of A/D-converted value addition/average times is performed for each analog channel.
6	A/D conversion data corresponding to each analog channel is output. The A/D conversion data is stored in the data register (ADDRi (i = 0 to 28), ADEXDRj (j = 0 to 2, 5 to 8)). If FIFO is used, A/D conversion data is also stored in FIFO data register (ADFIFODRk (k = 0 to 8)). However, the A/D conversion data of the channel used as the dummy conversion channel is not guaranteed.
7	If the scan end interrupt is enabled, the scan end interrupt corresponding to that scan group is generated when the A/D conversion of all virtual channels assigned to that scan group is completed.
8	The second and subsequent rounds of scanning operations repeat the operation of Step 2 to 4, retaining the oversampling data stored in the digital filter. Each one time oversampling or each multiple times oversampling corresponding to the number of A/D-converted value addition/averaging times is performed for each analog channel, the next A/D conversion data is output. The A/D conversion data is stored in the data register (ADDRi (i = 0 to 28), ADEXDRj (j = 0 to 2, 5 to 8)). If FIFO is used, A/D conversion data is also stored in FIFO data register (ADFIFODRk (k = 0 to 8)). However, the A/D conversion data of the channel used as the dummy conversion channel is not guaranteed.
9	If the scan end interrupt is enabled, the scan end interrupt corresponding to that scan group is generated when the A/D conversion of all virtual channels assigned to that scan group is completed.
10	Thereafter, until the forcibly stop process is performed, Step 8 to 9 are repeated, and the scanning operation continues. To perform the forcibly stop, follow to section 36.5.4. Force Stops the A/D Conversion Operation. If scanning operation is aborted by the forcibly stop process, A/D converter and channel-dedicated sample-and-hold circuit enter the idle state. The oversampling data in digital filter are discarded at the end of the scanning operation.

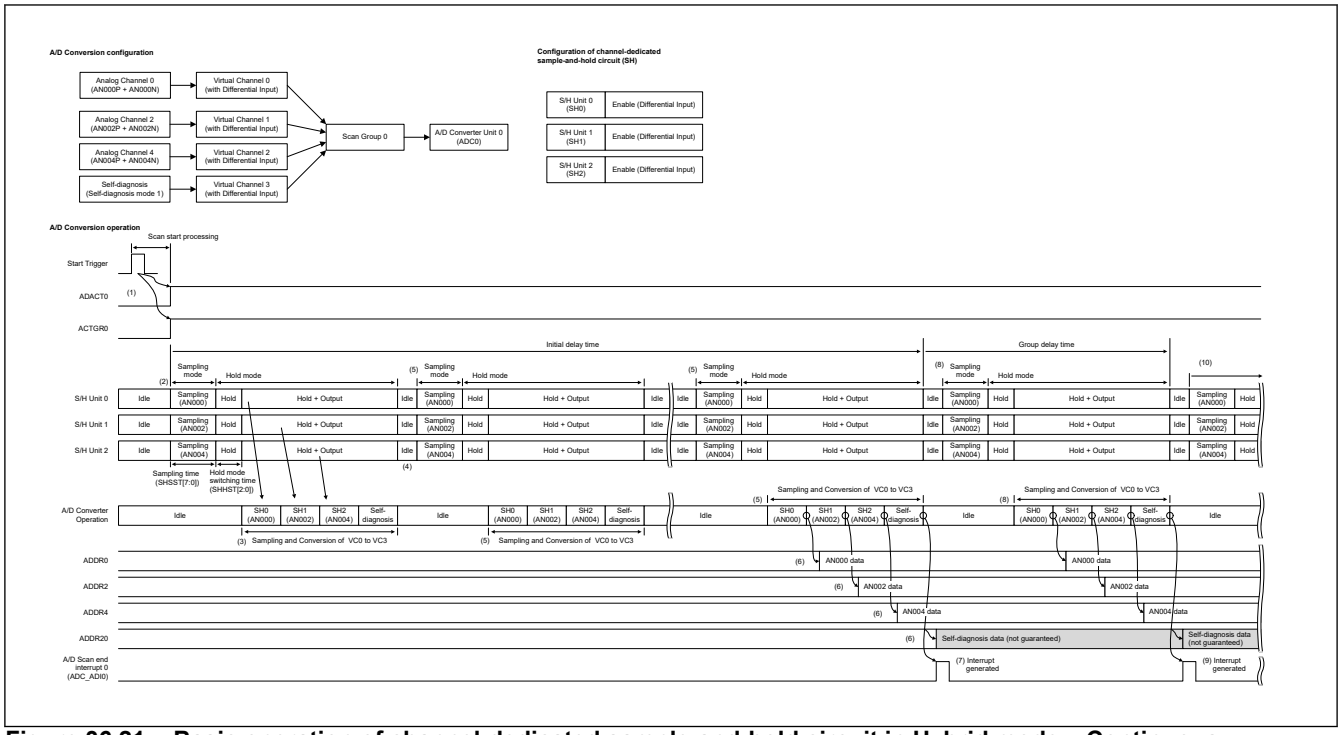


Figure 36.21 Basic operation of channel-dedicated sample-and-hold circuit in Hybrid mode – Continuous scan mode

Basic Operation in Hybrid Mode - Background Continuous Scan Mode

This paragraph shows the basic operation of channel-dedicated sample-and-hold circuit in Hybrid mode – Background continuous scan mode. Since Differential input mode is supported in Hybrid mode, this example describes a case that channel-dedicated sample-and-hold circuits are set to Differential input mode. The configuration and behavior of example operation show in Table 36.30, Table 36.31 and Figure 36.22.

Table 36.30 Configuration of example operation for channel-dedicated sample-and-hold circuit in Hybrid mode – Background continuous scan mode

Item	Configuration
A/D converter	Set ADC0 to Hybrid mode – Continuous scan mode
Channel-dedicated sample-and-hold circuit	Set SH0 to SH2 to enabled and Differential input mode. (AN000 to AN005 are subject to A/D conversion using channel-dedicated sample-and-hold circuit.)
Virtual channel*1	• Assign AN000, AN002 and AN004 to virtual channel 0 to 2, and set to Differential input mode. (This will function AN000, AN002 and AN004 as the non-inverting inputs (+) of the differential input, and AN001, AN003, and AN005 as the inverting inputs (−) of the differential input.) • Assign self-diagnosis channel as the dummy conversion channel to virtual channel 3, and set to Differential input mode and Self-diagnosis mode 1.
Scan group*1	• Assign virtual channel 0 to 3 to scan group 0. • Set scan group 0 to convert with ADC0. • Set to perform Self-diagnostic mode 1 in scan group 0.

Note 1. When channel-dedicated sample-and-hold circuit is used, there are restrictions on the setting of virtual channels. For more details, see (1)Limitations on usage of channel-dedicated sample-and-hold circuit in Hybrid mode and section 36.3.16.4. Restrictions on Channel-dedicated Sample-and-hold Circuit.

Table 36.31 Example operation of channel-dedicated sample-and-hold circuit in Hybrid mode – Background continuous scan mode (1 of 2)

Step	Behavior
1	The scanning operation of scan group 0 starts when a trigger for scan group 0 is input.
2	Channel-dedicated sample-and-hold circuit (SH0 to SH2) starts sampling the analog inputs (AN000, AN002, AN004) and holds them.

Table 36.31 Example operation of channel-dedicated sample-and-hold circuit in Hybrid mode – Background continuous scan mode (2 of 2)

Step	Behavior
3	The A/D converter performs oversampling once for each output of channel-dedicated sample-and-hold circuit (SH0 to SH2) and once for each of the dummy conversion channel (self-diagnosis).
4	After one oversampling for each analog channel, channel-dedicated sample-and-hold circuit releases the held voltage and enters the idle state.
5	Step 2 to 4 is repeated. After the oversampling corresponding to the number of TAPs of the digital filter and the number of A/D-converted value addition/averaging times is performed to each analog channel, first A/D conversion data becomes in ready to be output. Step 2 to 4 is repeated until oversampling corresponding to the number of TAPs of the digital filter and the number of A/D-converted value addition/average times is performed for each analog channel.
6	The second and subsequent rounds of scanning operations repeat the operation of Step 2 to 4, retaining the oversampling data stored in the digital filter. Each time oversampling is performed for each analog channel, the data in the digital filter is updated. Each one time oversampling or each multiple times oversampling corresponding to the number of A/D-converted value addition/averaging times is performed for each analog channel, the next A/D conversion data becomes in ready to be output.
7	When an A/D conversion start trigger is input during background continuous scanning operation, the most recent A/D conversion data at that time is stored in the data register (ADDRi (i = 0 to 28), ADEXDRj (j = 0 to 2, 5 to 8)). When FIFO function is set to enabled, A/D conversion data is also stored in the FIFO data register (ADFIFODRk (k = 0 to 8)). However, the A/D conversion data of the channel used as the dummy conversion channel is not guaranteed. Also, if scan end interrupt is set to enable, scan end interrupt is generated. If FIFO interrupt is set to enable, FIFO data read request is generated when the condition for generating it is met.
8	Thereafter, until the forcibly stop process is performed, background continuous scanning operation (Step 6) is repeated. And whenever the A/D conversion start trigger is input during background continuous scanning operation, the A/D conversion data is output (Step 7). To perform the forcibly stop, follow to section 36.5.4. Force Stops the A/D Conversion Operation. If scanning operation is aborted by the forcibly stop process, A/D converter and channel-dedicated sample-and-hold circuit enter the idle state. The oversampling data in digital filter are discarded at the end of the scanning operation.

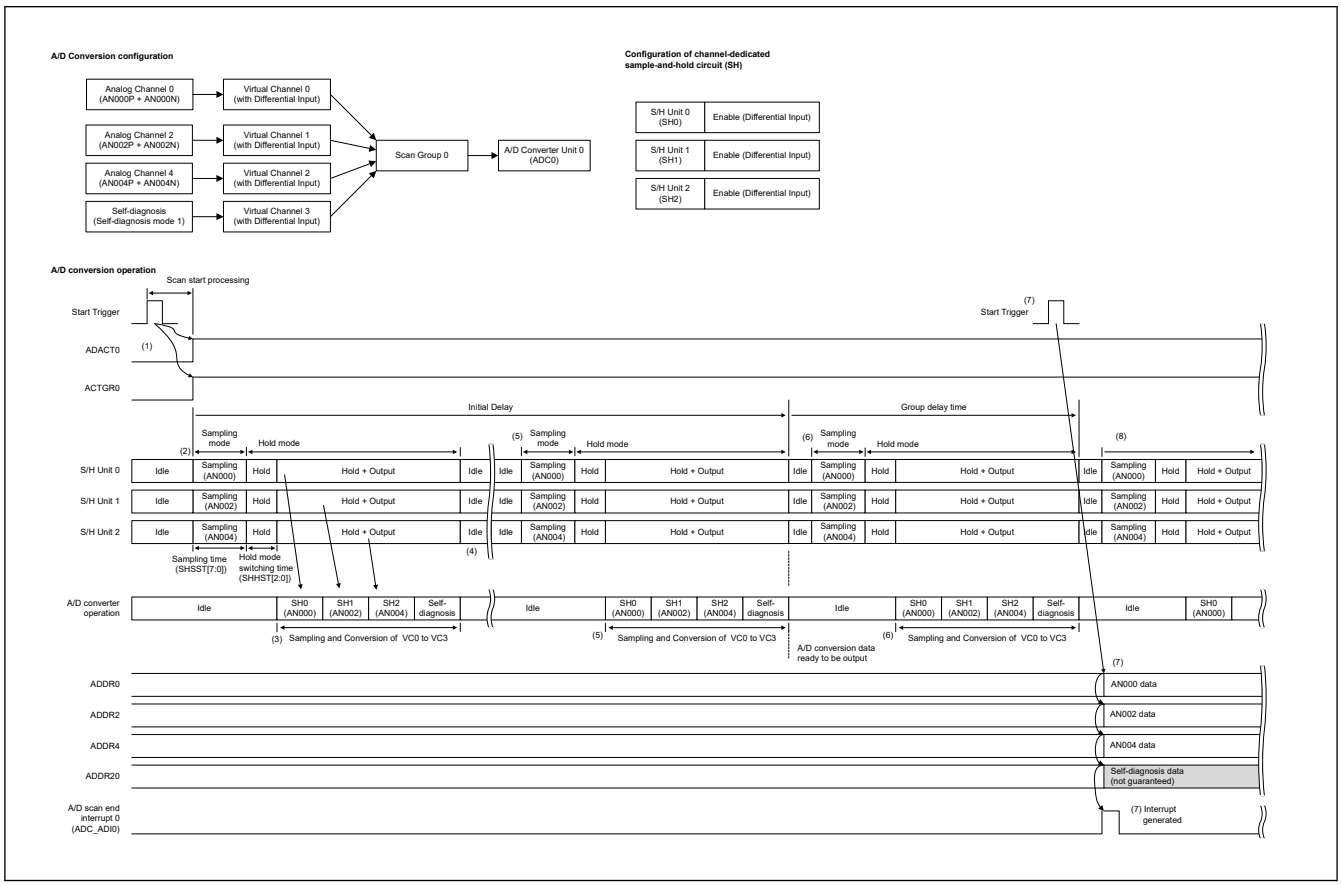


Figure 36.22 Basic operation of channel-dedicated sample-and-hold circuit in Hybrid mode – Background continuous scan mode

36.3.16.4 Restrictions on Channel-dedicated Sample-and-hold Circuit

When using channel-dedicated sample-and-hold circuit, observe the following restrictions:

(1) Limitations on usage of channel-dedicated sample-and-hold circuit

Channel-dedicated sample-and-hold circuit is only available in the following:

- SAR mode – Single scan mode
- Hybrid mode – Single scan mode
- Hybrid mode – Continuous scan mode
- Hybrid mode – Background continuous scan mode

In the operation mode other than above, channel-dedicated sample-and-hold circuit is prohibited.

In Hybrid mode, there are the usage limitations of channel-dedicated sample-and-hold circuit (the configuration constraints on scan group and virtual channel). For details, see (1)Limitations on usage of channel-dedicated sample-and-hold circuit in Hybrid mode.

(2) Restrictions on self-calibration operation of channel-dedicated sample-and-hold circuit

Self-calibration operation is required when channel-dedicated sample-and-hold circuit is used. For more details on self-calibration procedures for channel-dedicated sample-and-hold circuit and the restrictions on it, see section 36.3.8. Self-calibration.

(3) Restrictions on assigning virtual channels and scan groups

For analog channels using channel-dedicated sample-and-hold circuit, assign them to virtual channels so that A/D conversion is performed at the front of the scan group. Specifically, assign an analog channel that uses channel-dedicated sample-and-hold circuit from the virtual channel that has the smallest virtual channel number among the virtual channels assigned to that scan group. (Assign the analog channels of channel-dedicated sample-and-hold circuit to the front virtual channels in ascending order of the virtual channel number.)

If this restriction is violated, the operation of channel-dedicated sample-and-hold circuit and A/D converter is not guaranteed.

(4) Restrictions on analog paths when using channel-dedicated sample-and-hold circuit

When channel-dedicated sample-and-hold circuit is enabled, both the even-numbered analog channels and the odd-numbered analog channels connected to channel-dedicated sample-and-hold circuit are performed the A/D conversion with channel-dedicated sample-and-hold circuit. A/D conversion cannot be performed by bypassing channel-dedicated sample-and-hold circuit for only one of the analog channels.

See also the next restriction related to this restriction.

(5) Prohibition of multiple assignments of channels using the same channel-dedicated sample-and-hold circuit in the same scan group

It is prohibited to assign more than one virtual channel that uses the same channel-dedicated sample-and-hold circuit to a single scan group. Specific restrictions are the followings:

[Restrictions]

- The virtual channel whose signal source is selected as the even-numbered analog channel connected to channel-dedicated sample-and-hold circuit and the virtual channel whose signal source is selected as the odd-numbered analog channel must not be assigned to the same scan group.
- It is prohibited to assign the even-numbered analog channels or odd-numbered analog channels connected to channel-dedicated sample-and-hold circuit to multiple virtual channels, and assign them to the same scan group.

Operation is not guaranteed if this restriction is violated.

To avoid this restriction, assign virtual channels that use the same channel-dedicated sample-and-hold circuit to separate scan groups. This restriction can be avoided by preventing A/D conversion from occurring more than once for the same

channel-dedicated sample-and-hold circuit during one scanning operation. For the example workaround for this restriction, see (2)Advanced operation in SAR mode - Single scan mode.

(6) Restrictions on group priority operation

Channel-dedicated sample-and-hold circuit is not available in group priority operation.

(7) Restrictions on Hybrid mode

In Hybrid mode, the way to use of the channel-dedicated sample-and-hold circuit is limited. For details, see section 36.3.16.3. Operation in Hybrid Mode with Channel-dedicated Sample-and-hold Circuit.

36.3.17 Disconnection Detection Assist Function

Disconnection detection assist function is the function that fixes the charges of the sampling capacitance of the A/D converter to a specified condition prior to starting A/D conversion. This function can be used to detect the wiring disconnection connected to analog input.

Figure 36.23 shows an example operation of A/D conversion when disconnection detection assist function is used. Figure 36.24 shows an example of the disconnection detection when precharge is selected, and Figure 36.25 shows an example of the disconnection detection when discharge is selected.

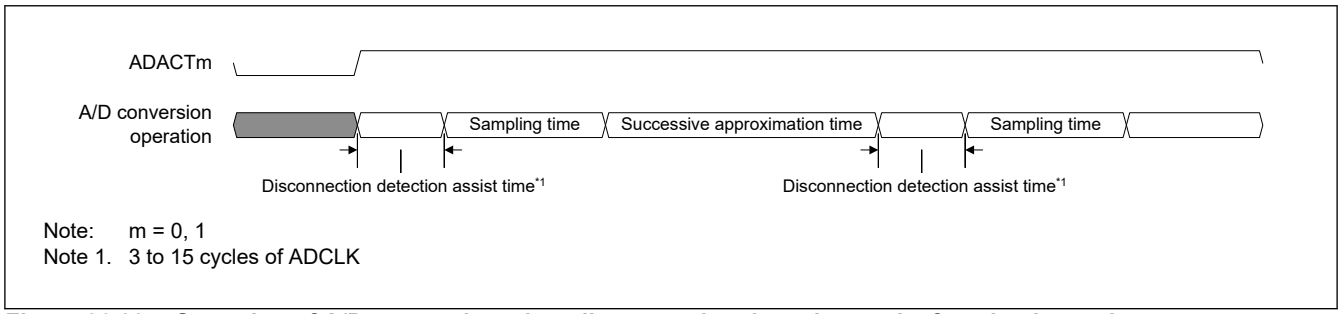
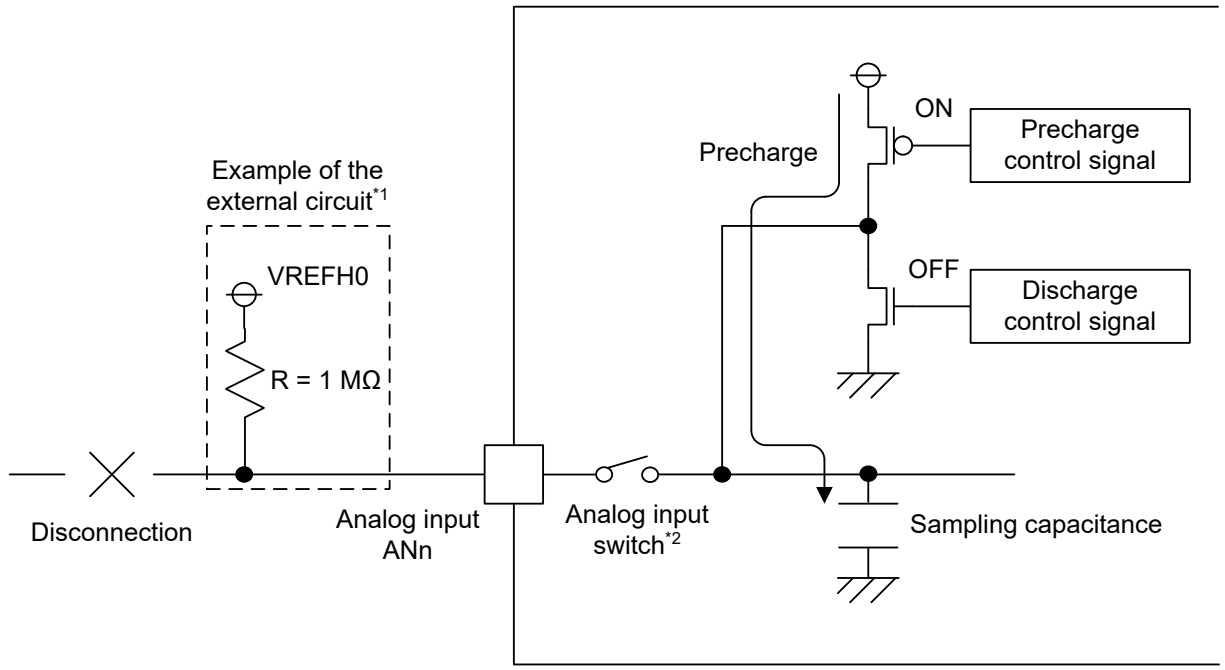
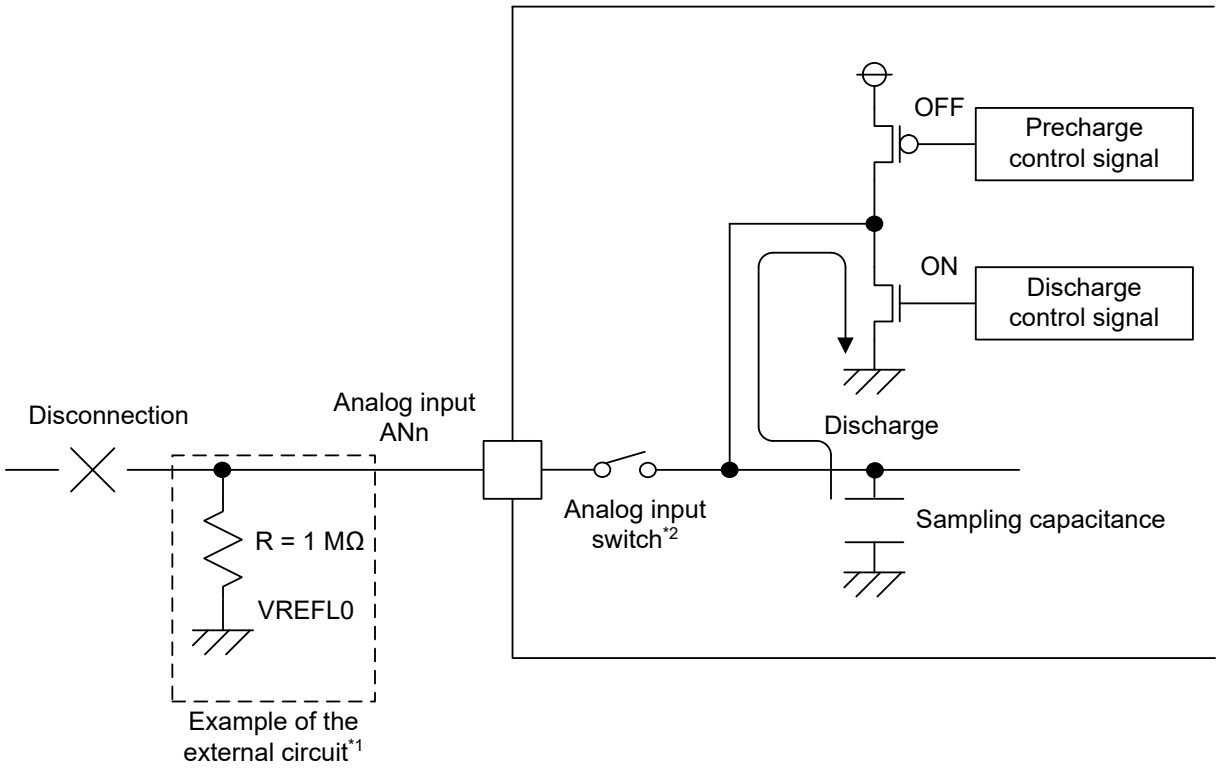


Figure 36.23 Operation of A/D conversion when disconnection detection assist function is used



Note 1. The converted result should be used after it is fully evaluated because the result data when disconnection occurs varies depending on the external circuit.
Note 2. Turn off when precharge, and turn on when sampling the Analog input ANn.

Figure 36.24 Example of disconnection detection when precharge is selected



Note 1. The converted result should be used after it is fully evaluated because the result data when disconnection occurs varies depending on the external circuit.
Note 2. Turn off when discharge and turn on when sampling the Analog input ANn.

Figure 36.25 Example of disconnection detection when discharge is selected

36.3.18 Group Priority Operation

Group priority operation function is the function that performs the scanning operation (A/D conversion) for a scan group with a higher priority based on the priority of scan group.

The priority of scan groups is group 0 > group 1 > … > group n. (n: the number of maximum scan group - 1)

Table 36.32 shows the conditions when group priority operation can be used and the corresponding register settings. Set the operation according to Table 36.32.

Table 36.32 Group priority operation usable conditions and corresponding register settings

Operating conditions			Group priority operation				
Operation mode of A/D converter		Number of scan groups *1	Operation	Setting for ADGSPCR register *2			
				PGSm	RSCNm	LGRRSm	GRPm
SAR mode	Single scan mode	2 Group	✓	1	1	1	0
		3 Group or more	—	0	0	0	0
	Continuous scan mode	2 Group	✓	1	1	1	1
		3 Group or more	✓	1	1	1	1
Oversampling mode		—	—	0	0	0	0
Hybrid mode		—	—	0	0	0	0

Note: ✓: Available, —: Not Available (Prohibited)
m = 0, 1
Note 1. The number of scan groups is the total number of scan groups used for one A/D converter.
Note 2. Settings other than the specified value are prohibited.

36.3.18.1 Group Priority Operation in SAR Mode – Single Scan Mode

In group priority operation in SAR mode – Single scan mode, the single scanning operation is performed according to the priority of scan group.

If an A/D conversion start trigger for the high priority group is input during a scanning operation of the low priority group, the scanning operation of the low priority group is interrupted, and the scanning operation of the high priority group is started. After the scanning operation of the high priority group is completed, the scanning operation of the low priority group is resumed.

When an A/D conversion start trigger of the low priority group during a scanning operation of the high priority group is input, the scanning operation of the low priority group is performed after the scanning operation of the high priority group is completed.

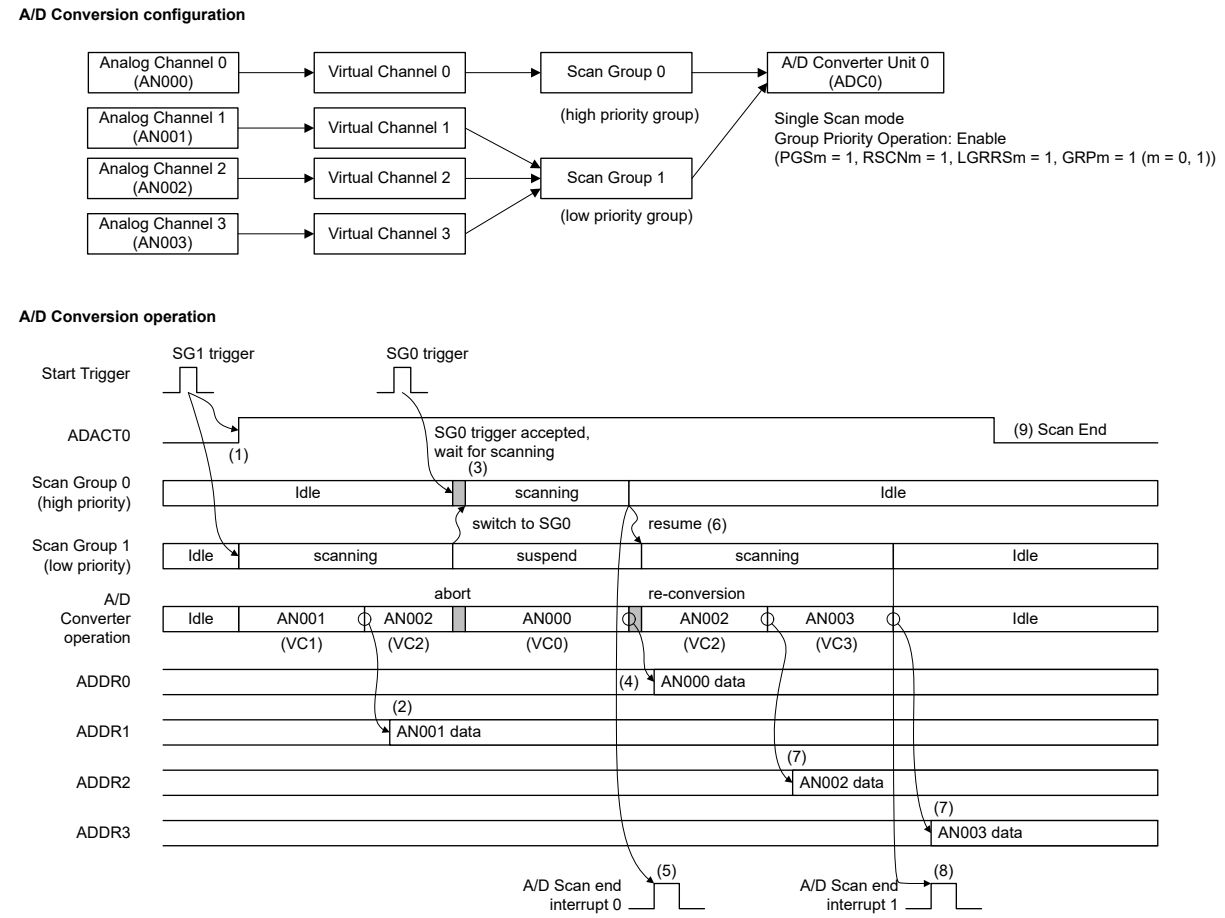
Example of the Group Priority Operation in SAR Mode – Single Scan Mode

Table 36.33 and Figure 36.26 show the example of group priority operation in SAR mode – Single scan mode when analog channel 0 (AN000) is assigned to scan group 0 (high priority) and analog channel 1 to 3 (AN001 to AN003) are assigned to scan group 1 (low priority).

Table 36.33 Example of the group priority operation in SAR mode – Single scan mode (2 groups)

Step	Operation
1	The scanning operation of scan group 1 starts when a trigger for scan group 1 is input.
2	When A/D conversion of each channel in scan group 1 completes, the A/D conversion result is stored in the corresponding A/D data register n (ADDRn).
3	If a trigger for scan group 0 is input during A/D conversion for scan group 1, the scanning operation of scan group 1 is suspended and the scanning operation of scan group 0 is started. *1
4	When A/D conversion of each channel in scan group 0 completes, the A/D conversion result is stored in the corresponding A/D data register n (ADDRn).
5	If a scan end interrupt for scan group 0 is enabled, a scan end interrupt occurs.
6	The scanning operation of scan group 1 is resumed after scanning of scan group 0 is completed. The scanning operation is resumed from the A/D conversion incomplete channel. *1
7	When A/D conversion of each channel in scan group 1 completes, the A/D conversion result is stored in the corresponding A/D data register n (ADDRn).
8	If a scan end interrupt for scan group 1 is enabled, a scan end interrupt occurs.
9	When all scanning operations are completed, ADSR.ADACTm (m = 0, 1) bits are cleared and the A/D converter enters the idle status.

Note 1. The timing of interruption and restart of scanning operation depends on the operating conditions. For details, see section 36.3.18.3. Restrictions on Group Priority Operation.



Note: SG: Scan Group VC: Virtual channel

Figure 36.26 Example of group priority operation in SAR mode – Single scan mode (2 groups)

36.3.18.2 Group Priority Operation in SAR Mode – Continuous Scan Mode

In group priority operation in SAR mode – Continuous scan mode, the continuous scanning operation is performed for the scan group (continuous-scan group) in which scanning operation was first started. If an A/D conversion start trigger of another scan group (interrupt-scan group) is input during the continuous scanning operation of the continuous-scan group, the following operations are performed according to the priority of the interrupt-scan group.

1. Priority: When Interrupt-scan group (high) > Continuous-scan group (low)
   - The scanning operation of the continuous-scan group is interrupted, and the single scanning operation of the interrupt-scan group is started.
   - The continuous scanning operation of the continuous-scan group is restarted after the single scanning operation of the interrupt-scan group is completed.
2. Priority: When Continuous-scan group (high) > Interrupt-scan group (low)
   - This operation is prohibited. Operation is not guaranteed.

   In case 1. above, if an A/D conversion start trigger for another scan group (interrupt-scan group B) is input during the single scanning operation of the interrupt-scan group (interrupt-scan group A), the following operation is performed according to the priority of interrupt-scan group B.
3. Priority: When Interrupt-scan group B (high) > Interrupt-scan group A (middle) > Continuous-scan group (low)
   - The single scanning operation of interrupt-scan group A is suspended, and the single scanning operation of interrupt-scan group B is started.

- After the single scanning operation of interrupt-scan group B is completed, the single scanning operation of interrupt-scan group A is resumed.
- After the scanning operation of interrupt-scan group A is completed, the continuous scanning operation of the continuous-scan group is resumed.

4. Priority: When interrupt-scan group A (high) > interrupt-scan group B (medium) > continuous-scan group (low)
   - After the single scanning operation of interrupt-scan group A is completed, the single scanning operation of interrupt-scan group B is started.
   - After the single scanning operation of interrupt-scan group B is completed, the continuous scanning operation of the continuous-scan group is resumed.
5. Priority: When interrupt-scan group A (high) > continuous-scan group (medium) > interrupt-scan group B (low)
   - This operation is prohibited. Operation is not guaranteed.

Example operations are shown in the followings:

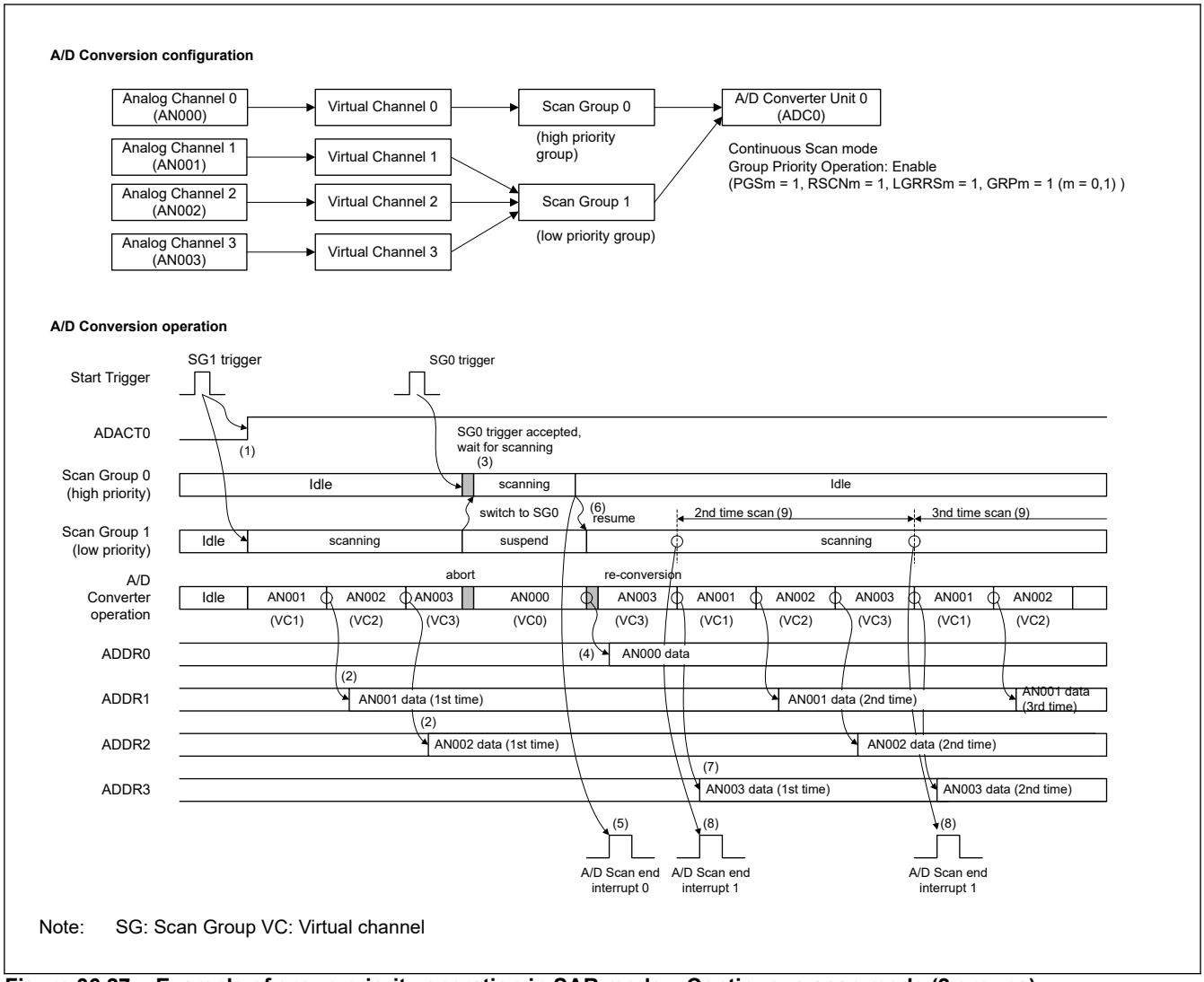
(1) Example of two-group priority operation in SAR mode – Continuous scan mode

Table 36.34 and Figure 36.27 show the example of group priority operation in SAR mode – Continuous scan mode when analog channel 0 (AN000) is assigned to scan group 0 (high-priority group) and analog channel 1 to 3 (AN001 to AN003) is assigned to scan group 1 (low-priority group).

Table 36.34 Example of the group priority operation in SAR mode – Continuous scan mode (2 groups)

Step	Operation
1	The continuous scanning operation of scan group 1 (low priority group) is started by the trigger input of scan group 1.
2	When A/D conversion of each channel in scan group 1 completes, the A/D conversion result is stored in the corresponding A/D data register n (ADDRn).
3	If a trigger for scan group 0 is input during A/D conversion for scan group 1, the scanning operation of scan group 1 is suspended and scanning operation of scan group 0 is started. *1
4	When A/D conversion of each channel in scan group 0 completes, the A/D conversion result is stored in the corresponding A/D data register n (ADDRn).
5	If a scan end interrupt for scan group 0 is enabled, a scan end interrupt occurs.
6	The scanning operation of scan group 1 is resumed after scanning of scan group 0 is completed. The scanning operation is resumed from the A/D conversion incomplete channel. *1
7	When A/D conversion of each channel in scan group 1 completes, the A/D conversion result is stored in the corresponding A/D data register n (ADDRn).
8	If a scan end interrupt for scan group 1 is enabled, a scan end interrupt occurs.
9	Scan group 1 continues the continuous scanning operation. *2

Note 1. The timing of interruption and restart of scanning operation depends on the operating conditions. For details, see section 36.3.18.3. Restrictions on Group Priority Operation.
Note 2. To stop the continuous scanning operation, follow the procedures described in section 36.5.4. Force Stops the A/D Conversion Operation.



Note: SG: Scan Group VC: Virtual channel

Figure 36.27 Example of group priority operation in SAR mode – Continuous scan mode (2 groups)

(2) Example of three-groups priority operation in SAR mode – Continuous scan mode

Table 36.35 and Figure 36.28 show the example of group priority operation when analog channel 0 (AN000) is assigned to scan group 0 (high priority), analog channels 1 and 2 (AN001 and AN002) are assigned to scan group 1 (middle priority), and analog channel 3 to 5 (AN003 to AN005) are assigned to scan group 2 (low priority).

Table 36.35 Example of the group priority operation in SAR mode – Continuous scan mode (3 groups) (1 of 2)

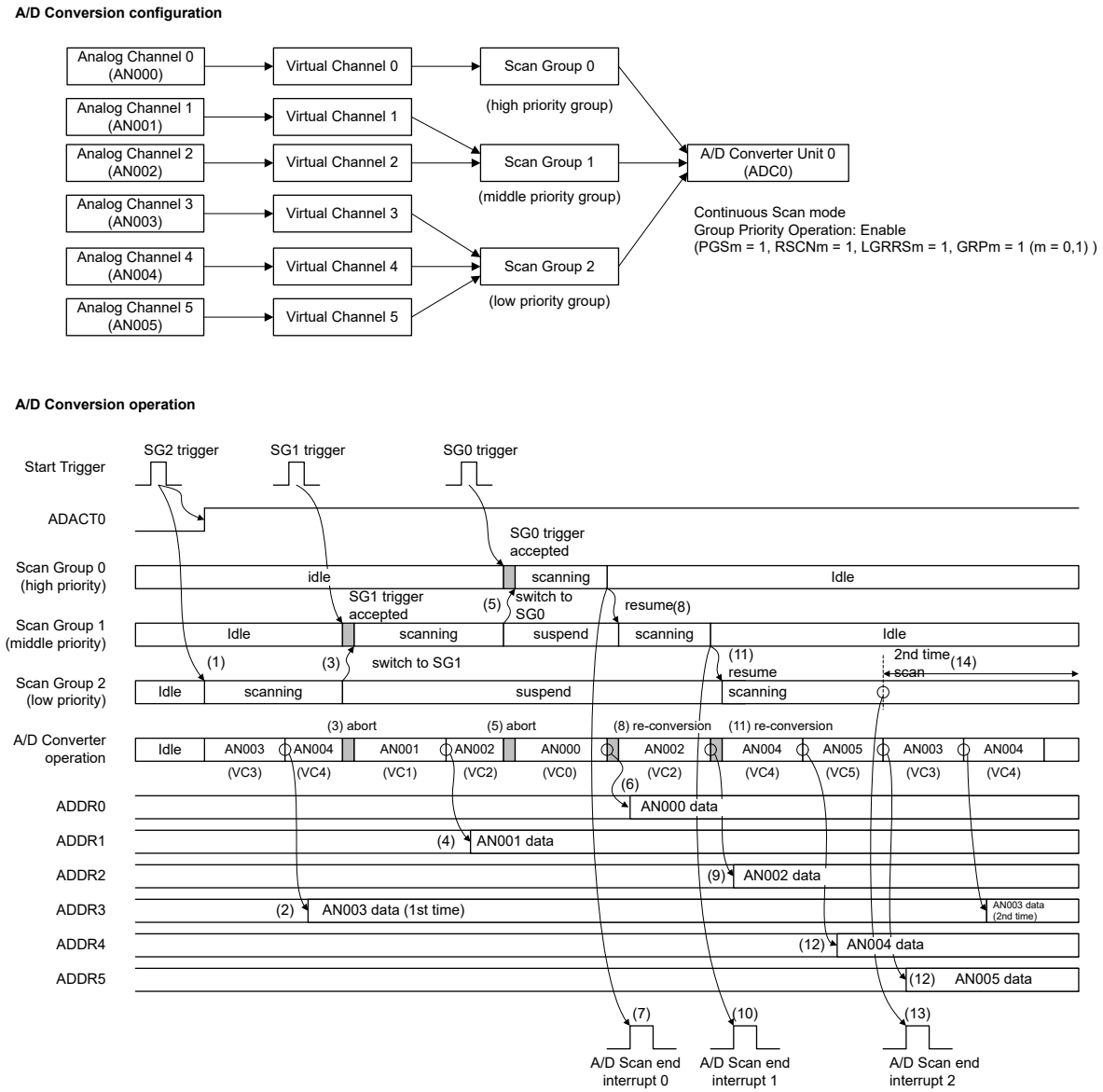
Step	Operation
1	The continuous scanning operation of scan group 2 is started by the trigger input of scan group 2.
2	When A/D conversion of each channel in scan group 2 completes, the A/D conversion result is stored in the corresponding A/D data register n (ADDRn).
3	If a trigger for scan group 1 is input during A/D conversion on scan group 2, the scanning operation of scan group 2 is suspended and the scanning operation of scan group 1 is started. *1
4	When A/D conversion of each channel in scan group 1 completes, the A/D conversion result is stored in the corresponding A/D data register n (ADDRn).
5	If a trigger for scan group 0 is input during A/D conversion for scan group 1, the scanning operation of scan group 1 is suspended and the scanning operation of scan group 0 is started. *1
6	When A/D conversion of each channel in scan group 0 completes, the A/D conversion result is stored in the corresponding A/D data register n (ADDRn).
7	If a scan end interrupt for scan group 0 is enabled, a scan end interrupt occurs.

Table 36.35 Example of the group priority operation in SAR mode – Continuous scan mode (3 groups) (2 of 2)

Step	Operation
8	The scanning operation of scan group 1 is resumed after scanning of scan group 0 is completed. The scanning operation is restarted from the A/D conversion incomplete channel. *1
9	When A/D conversion of each channel in scan group 1 completes, the A/D conversion result is stored in the corresponding A/D data register n (ADDRn).
10	If a scan end interrupt for scan group 1 is enabled, a scan end interrupt occurs.
11	Scan group 2 is resumed after scanning of scan group 1 is completed. The scanning operation is resumed from the A/D conversion incomplete channel. *1
12	When A/D conversion of each channel in scan group 2 completes, the A/D conversion result is stored in the corresponding A/D data register n (ADDRn).
13	If a scan end interrupt for scan group 2 is enabled, a scan end interrupt occurs.
14	Scan group 1 continues the continuous scanning operation. *2

Note 1. The timing of interruption and restart of scanning operation depends on the operating conditions. For details, see section 36.3.18.3. Restrictions on Group Priority Operation.

Note 2. To stop the continuous scanning operation, follow the procedures described in section 36.5.4. Force Stops the A/D Conversion Operation.



Note: SG: Scan Group VC: Virtual channel

Figure 36.28 Example of group priority operation in SAR mode – Continuous scan mode (3 groups)

36.3.18.3 Restrictions on Group Priority Operation

(1) Channel-dedicated sample-and-hold circuit restrictions

The use of channel-dedicated sample-and-hold circuit is prohibited in group priority operation.

(2) Restrictions on the operation mode and operation setting of the A/D converter

Group priority operation is only available in SAR mode; Group priority operation is prohibited in Oversampling mode or Hybrid mode.

Group priority operation in SAR mode – Single scan mode prohibits the use of three or more scan groups for a single A/D converter. If three or more scan groups are to be used for one A/D converter in group priority operation, SAR mode – Continuous scan mode must be used. For details, see Table 36.32.

If these restrictions are violated, operation is not guaranteed.

(3) Restrictions on group priority operation in SAR mode – Continuous scan mode

In group priority operation in SAR mode – Continuous scan mode, do not input A/D conversion start triggers for scan groups that have a lower priority than the scan group that started continuous scanning operation. If this restriction is violated, the operation is not guaranteed.

(4) Restrictions on trigger input for the same scan group

Input the A/D conversion start triggers for the same scan group in group priority operation for the following periods.

[When synchronous operation is disabled (ADSYCR.ADSYDISm = 1)]

- 2 ADCLK + 2 PCLK cycles after the scan end flag is set *1

[When synchronous operation is enabled (ADSYCR.ADSYDISm = 0)]

- Synchronous operation period cycle × 2 after the scan end flag is set *1

Note: m = 0, 1

Note 1. ADSCANENDSR.SCENDFn = 1 (n = 0 to 8)

If A/D conversion start triggers are input without waiting for the above period, the A/D conversion start triggers may be lost. To accept A/D conversion start triggers reliably, input the trigger after the above period has elapsed.

(5) Restrictions on suspend and resume for scanning operation of low-priority groups

When an A/D conversion start trigger for the high priority group is input during scanning of the low priority group in group priority operation, the scanning operation of the low priority group is interrupted at the timing shown in the following:

[Scan interruption timing of low priority group]

- When synchronous operation is enabled
  - After the A/D conversion start trigger of the high-priority group is accepted, the scanning operation of the low-priority group is suspended after waiting until the next synchronous operation period timing.
- When synchronous operation is disabled
  - The scanning operation of the low-priority group is immediately suspended after the A/D conversion start trigger of the high-priority group is accepted.

If A/D conversion is being performed on channels for which A/D-converted value addition/averaging function is to be used at the timing of scan interruption of a low-priority group, the midway result of A/D-converted value addition/averaging value at that time is discarded. When the scanning operation of the low-priority group is resumed, A/D conversion starts over from the beginning of the number of times specified for A/D-converted value addition/averaging function.

If the scanning operation is suspended by group priority operation, ADGRSR.ACTGRn (n = 0 to 8) bit of the suspended scan group remains set to 1.

36.3.19 Synchronous Operation

Synchronous operation is the function that controls the operations of multiple A/D converters so that they are synchronized.

36.3.19.1 Synchronous Operation Examples

(1) Basic synchronous operation

The basic operation of synchronous operation is shown in the following. The configuration example is shown in Table 36.36, and the detail of operation is shown in Table 36.37 and Figure 36.29.

Table 36.36 Configuration example of basic synchronous operation (1 of 2)

Item	Configuration
A/D converter	• ADC0: Set to SAR mode – Single scan mode • ADC1: Set to SAR mode – Single scan mode

Table 36.36 Configuration example of basic synchronous operation (2 of 2)

Item	Configuration
Virtual channel	● Virtual channel 0 to 2: Assign AN000 to AN002 and set to Single-ended input mode. ● Virtual channel 6 to 8: Assign AN006 to AN008 and set to Single-ended input mode.
Scan group	[Scan Group 0] ● Assign AN000 to AN002 (Virtual channel 0 to 2) ● Set to convert with ADC0 [Scan Group 1] ● Assign AN006 to AN008 (Virtual channel 6 to 8) ● Set to convert with ADC1

Table 36.37 Example of basic synchronous operation

Step	Detail behavior of example operation
1	When a trigger of scan group 0 is input, a wait is inserted until the A/D conversion synchronization timing.
2	At the timing of synchronous operation period, the trigger input of scan group 0 is accepted, and the scanning operation of scan group 0 (for ADC0) starts.
3	When a trigger of scan group 1 is input, a wait is inserted until the A/D conversion synchronization timing.
4	At the timing of synchronous operation period, the trigger input of scan group 1 is accepted, and the scanning operation of scan group 1 (for ADC1) starts.
5	During synchronous operation, the A/D converter starts the sampling operation in accordance with synchronous operation period.

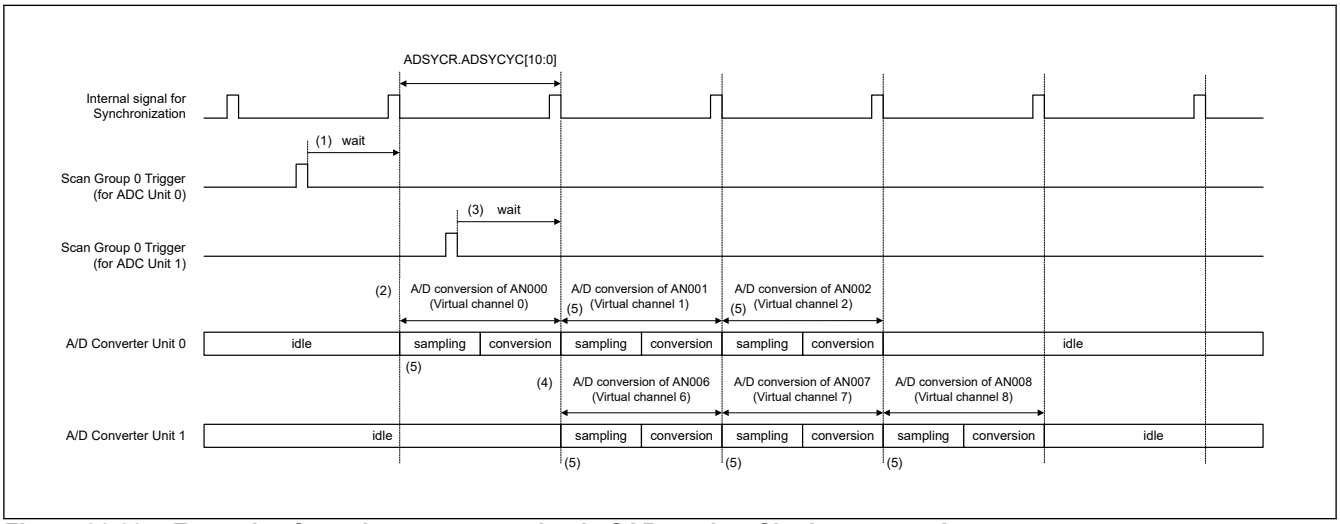


Figure 36.29 Example of synchronous operation in SAR mode – Single scan mode

(2) Synchronous operation using with channel-dedicated sample-and-hold circuit

An example of synchronous operation using with channel-dedicated sample-and-hold circuit is shown in the following. The configuration example is shown in Table 36.38, and the detail of operation is shown in Table 36.39 and Figure 36.30.

Table 36.38 Configuration example of basic synchronous operation (1 of 2)

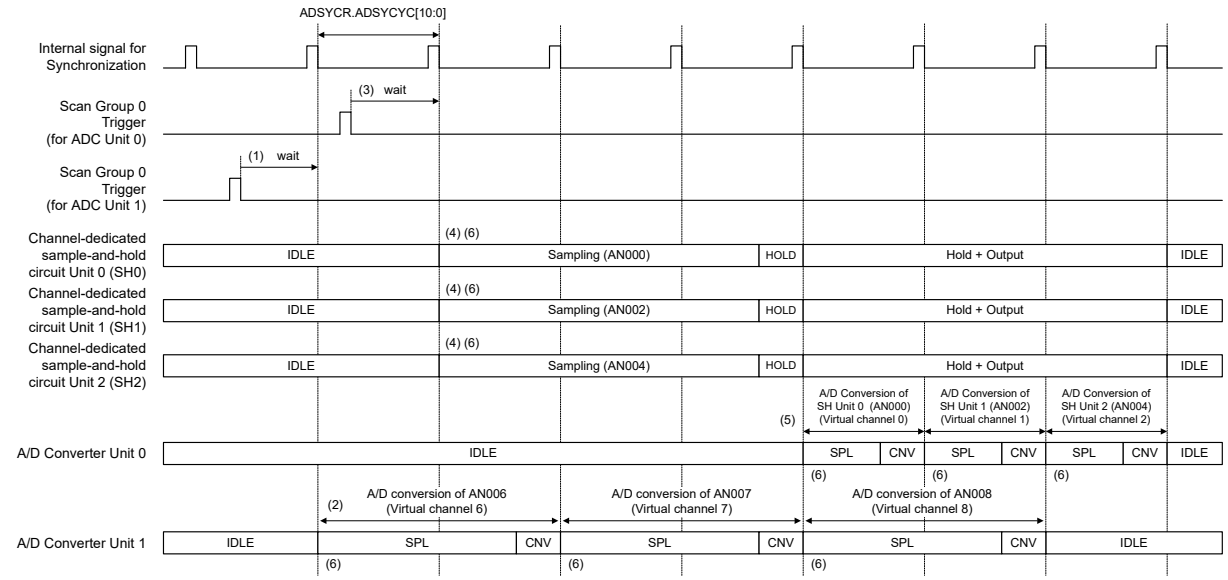
Item	Configuration
A/D converter	● ADC0: Set to SAR mode – Single scan mode ● ADC1: Set to SAR mode – Single scan mode
Channel-dedicated sample-and-hold circuit	● Set SH0 to SH2 to enabled and Single-ended input mode. ● Set SH4 to SH6 to enabled and Single-ended input mode.
Virtual channel	● Virtual channel 0 to 2: Assign AN000, AN002 and AN004, and set to Single-ended input mode. ● Virtual channel 6 to 8: Assign AN006, AN008 and AN010, and set to Single-ended input mode.

Table 36.38 Configuration example of basic synchronous operation (2 of 2)

Item	Configuration
Scan group	[Scan Group 0] ● Assign AN000, AN002 and AN004 (Virtual channel 0 to 2) ● Set to convert with ADC0 [Scan Group 1] ● Assign AN006, AN008 and AN010 (Virtual channel 6 to 8) ● Set to convert with ADC1

Table 36.39 Example of synchronous operation with channel-dedicated sample-and-hold circuit

Step	Detail behavior of example operation
1	When a trigger of scan group 1 is input, a wait is inserted until the A/D conversion synchronization timing.
2	At the timing of synchronous operation period, the trigger input of scan group 1 is accepted, and the scanning operation of scan group 1 (for ADC1) starts.
3	When a trigger of scan group 0 is input, a wait is inserted until the A/D conversion synchronization timing.
4	At the timing of synchronous operation period, the trigger input of scan group 0 is accepted, and the scanning operation of scan group 0 (for ADC0) starts. Channel-dedicated sample-and-hold circuit (SH0 to SH2) samples and holds the analog-channels of AN000/AN002/AN004 at the beginning of the scanning operation in scan group 0.
5	After the holding by channel-dedicated sample-and-hold circuit, A/D converter starts the samplings and the conversions.
6	During synchronous operation, channel-dedicated sample-and-hold circuit and A/D converter starts the sampling operation in accordance with synchronous operation period.



Note: SPL: Sampling state CNV: Conversion state

Figure 36.30 Example of synchronous operation using with channel-dedicated sample-and-hold circuit

36.3.19.2 Restrictions on Synchronous Operation

When synchronous operation is used, observe the following restrictions. If the following restrictions are violated, operations are not guaranteed.

1. Basic restrictions
   - Set the synchronous operation period so that it is an even number of cycles.
     - ADSYCR.ADSYCYC[10:0] = 2 × i
       (i: Any integer greater than or equal to 1 (i = 1, 2, 3…))
   - Set the synchronous operation period to a value larger than successive approximation time of ADCm (m = 0, 1).

  - ADSYCR.ADSYCYC[10:0] ≥ ADCNVSTR.CSTm[5:0] + 1
    (m = 0, 1)
- Set so that the sum of the sampling times of the analog channels and successive approximation time of ADCm (m = 0, 1) is an integral multiple of the synchronous operation period.
  - ADSSTRx.SSTy[9:0] + ADCNVSTR.CSTm[5:0] = ADSYCR.ADSYCYC[10:0] × i
    (x = 0 to 7, y = 0 to 15, m = 0, 1, i: Any integer greater than or equal to 1 (i = 1, 2, 3…))
- Set the synchronous operation period to a value larger than successive approximation time at self-calibration.*1
  - ADSYCR.ADSYCYC[10:0] ≥ ADCALSTCR.CALADCST[5:0] + 1
- Set the sampling time and successive approximation time at self-calibration to be integral multiples of the synchronous operation period.*1
  - ADCALSTCR.CALADSST[9:0] + ADCALSTCR.CALADCST[5:0] = ADSYCR.ADSYCYC[10:0] × i
    (i: Any integer greater than or equal to 1 (i = 1, 2, 3…))

2. Restrictions when using channel-dedicated sample-and-hold circuit
   - Set the synchronous operation period to a value greater than the hold mode switching time of channel-dedicated sample-and-hold circuit.
     - ADSYCR.ADSYCYC[10:0] ≥ ADSHSTRm.SHHST[2:0] + 1
       (m = 0, 1)
   - Set the sum of channel-dedicated sample-and-hold circuit sampling time and the hold mode switching time to be an integral multiple of the synchronous operation period.
     - ADSHSTRm.SHSST[7:0] + ADSHSTRm.SHHST[2:0] = ADSYCR.ADSYCYC[10:0] × i
       (m = 0, 1, i: Any integer greater than or equal to 1 (i = 1, 2, 3…))
   - Set the hold mode switching time of channel-dedicated sample-and-hold circuit so that it is the same as successive approximation time of ADCm (m = 0, 1).
     - ADSHSTRm.SHHST[2:0] = ADCNVSTR.CSTm[5:0]
       (m = 0, 1)
   - Set the synchronous operation period to a value greater than the hold mode switching time in channel-dedicated sample-and-hold circuit's self-calibration operation.*1
     - ADSYCR.ADSYCYC[10:0] ≥ ADCALSHCR.CALSHHST[2:0] + 1
   - Set so that the sum of channel-dedicated sample-and-hold circuit sampling time and the hold mode switching time in self-calibration operation is an integral multiple of the synchronous operation period.*1
     - ADCALSHCR.CALSHSST[7:0] + ADCALSHCR.CALSHHST[2:0] = ADSYCR.ADSYCYC[10:0] × i
       (i: Any integer greater than or equal to 1 (i = 1, 2, 3…))
3. Restrictions when using disconnection detection assist function
   - Set Disconnection Detection Assist period so that it is the same as the synchronous operation period.
     - ADSGDCRn.ADNDIS[3:0] = ADSYCR.ADSYCYC[10:0]
       (n = 0 to 8)

Note 1. In relation to this restriction, the restrictions on the settings of self-calibration should be satisfied at the same time. For restrictions on the setting of self-calibration, see section 36.3.8.3. Restrictions on Self-calibration.

36.4 A/D Conversion Data

This section describes the internal processing for A/D conversion data. For details about the operation of A/D conversion, see section 36.3. Operation.

36.4.1 Internal Data Processing Flow

Figure 36.31 shows the internal processing-flow of A/D conversion data.

The A/D conversion data output from the A/D converter is temporarily stored in the data buffer. And then, various data processing is performed based on the register setting. If the function that corresponds to each data processing is not used, its data processing is skipped.

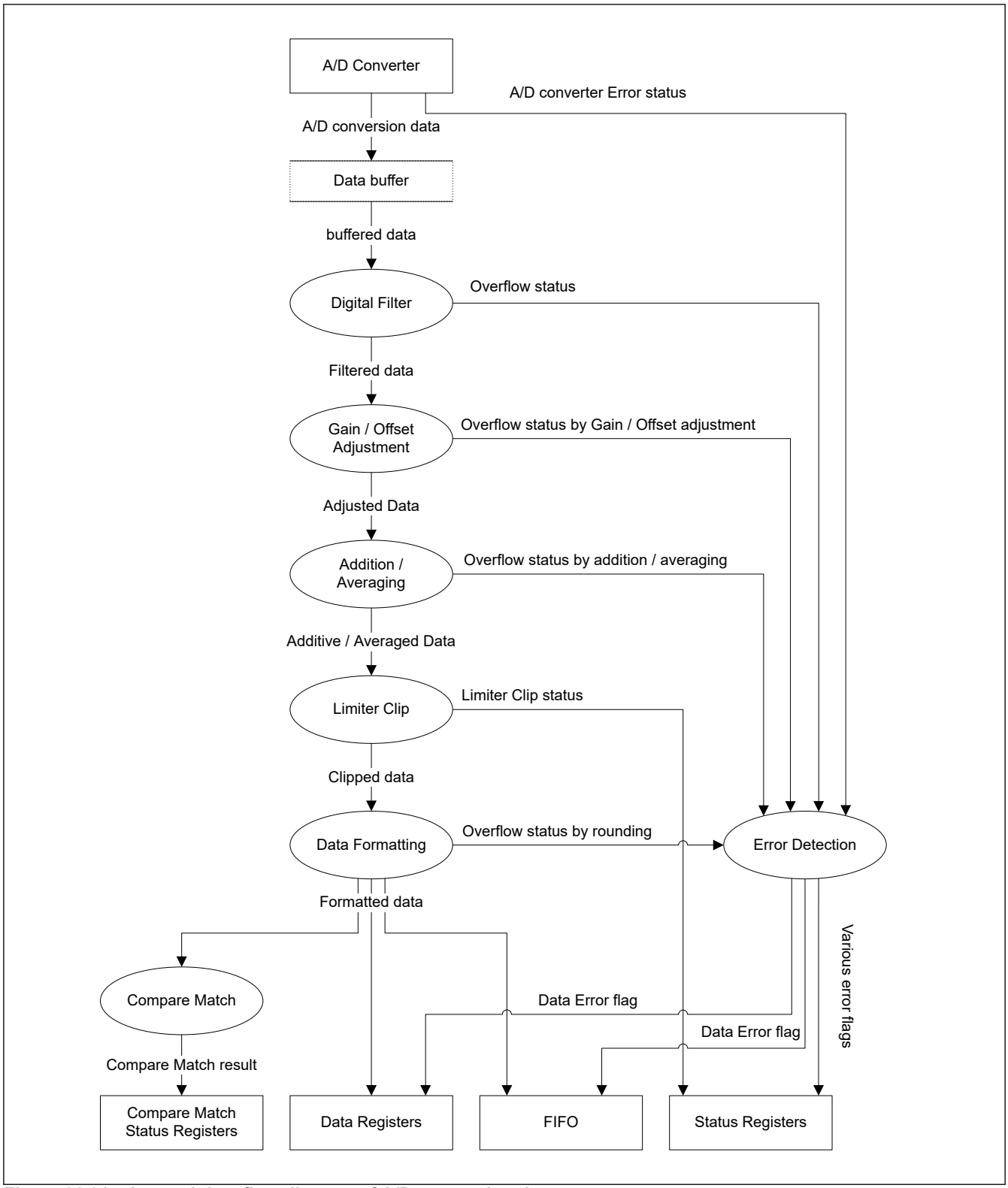


Figure 36.31 Internal data flow diagram of A/D conversion data

36.4.2 Digital Filter Function

ADC has 4 digital filters per 1 unit of A/D converter. High-precision A/D conversion results can be obtained by using a digital filter. The digital filter function must be used in Oversampling mode or Hybrid mode. The digital filter function in SAR mode is prohibited.

36.4.2.1 Configuration and Characteristics

The digital filter consists of a FIR type filter with 22 taps. The digital filter is set by ADDOPCRAn.DFSEL[2:0] (n = 0 to 36) and ADDFSRm (m = 0, 1) registers. The configuration of the digital filter is shown in Figure 36.32.

The type of the digital filter (FIR filter) can be selected from the following. For the characteristics of the digital filters, see section 46, Electrical Characteristics.

[List of Digital filter characteristics]

- Sinc filter
- Minimum phase filter

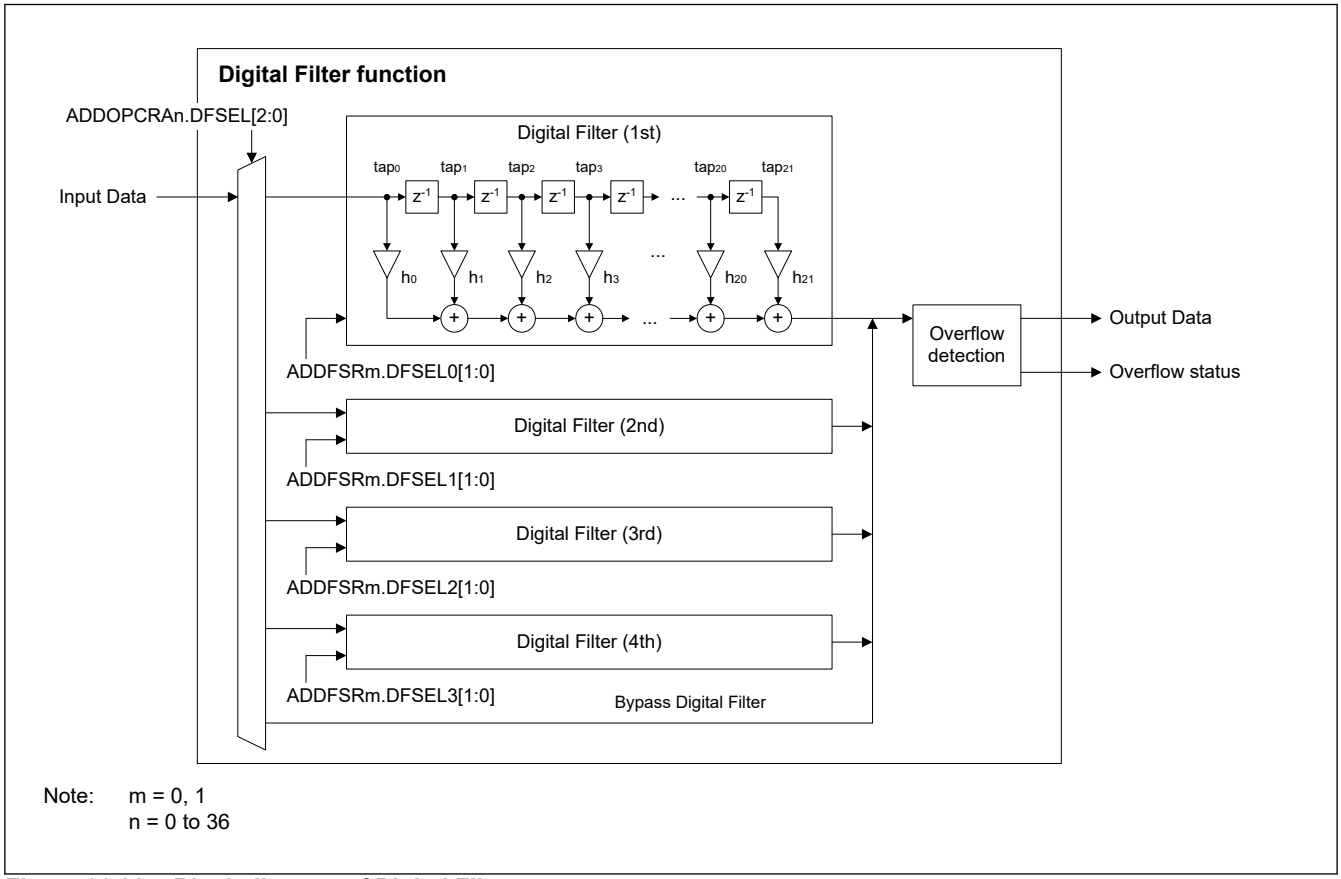


Figure 36.32 Block diagram of Digital Filter

36.4.2.2 Operation of Digital Filter Function

When the digital filter function is enabled, A/D conversion data is sequentially input to the digital filter. When all the taps in the digital filter are filled with data, the calculation result is output and sent to the next data processing.

The operation of the digital filters in Oversampling mode and Hybrid mode is shown in the followings.

(1) Digital filter operation in Oversampling mode

In Oversampling mode, oversampling (A/D conversion) is performed continuously for one analog channel. Each time oversampling is performed, A/D conversion data is sequentially input to the digital filter. When all the taps in the digital filter are filled with data, the calculation result is output and sent to the next data processing.

In Oversampling mode, the data of the taps in the digital filter is discarded when the A/D conversion data is output to the next data processing. However, when A/D-converted value addition/averaging function is used, the data of taps in the digital filter is kept until the data required to calculate the addition value or average value of the A/D converted value is collected. When the A/D-converted value addition/averaging value is calculated, the tap data in the digital filter is discarded.

(2) Digital filter operation in Hybrid mode

Hybrid mode can process oversampled data (A/D converted data) of up to four analog channels in parallel by using several digital filter circuits at the same time. Each time oversampling is performed, A/D conversion data is sequentially input to the digital filter. When all the taps in the digital filter are filled with data, the calculation result is output and sent to the next data processing.

In Hybrid mode – Single scan mode, taps in the digital filter are discarded at the end of scanning operation.

In Hybrid mode – Continuous scan mode, the data of the taps in the digital filter continues to be updated as long as the continuous scanning operation is continued. Therefore, after the data is filled in all taps in the digital filter, a new calculation result is output for each oversampling. When scanning operation is aborted due to a forced stop of A/D conversion, the data of taps in the filter is discarded.

The operation of the digital filter in Hybrid mode – Background continuous scan mode is the same as in Hybrid mode – Continuous scan mode. During the background continuous scanning operation, the digital filters and other data processing continue to take place in the background. When scanning operation is aborted due to a forced stop of A/D conversion, the data of taps in the filter is discarded.

36.4.3 Calibration and Adjustment

In this process, the following processing is performed for the results output from the A/D converter or for the data output from Digital Filter.

1. Calibrate Gain Error and Offset Error
2. Adjust User's Gain
3. Adjust User's Offset

For details on each process, refer to each item.

36.4.3.1 Gain Error and Offset Error Calibration

This process calibrates the A/D converter's Gain Error and Offset Error due to by chip-by-chip characteristic variations.

Calibration for the A/D converter's Gain Error and Offset Error is performed by calculation based on error data measured inside the ADC by self-calibration operation.

36.4.3.2 User's Gain Adjustment

User's Gain adjusting function multiplies the A/D conversion data by an arbitrary coefficient value.

The coefficient value of User's Gain is set in ADUGTRn.UGAIN[23:0] (n = 0 to 7). User's Gain can be set for each virtual channel. The User's Gain Table to be used is selected in ADDOPCRAm.GAINSEL[3:0] (m = 0 to 36). The coefficient value of User's Gain is the sum of the gain corresponding to the bits that are set to 1 in ADUGTRn.UGAIN[23:0]. Table 36.40 shows the gain factors corresponding to the respective bit in ADUGTRn.UGAIN[23:0]. An example of setting User's Gain is shown in Table 36.41. Figure 36.33 and Figure 36.34 show the relationship between before adjustment (input) and after adjustment (output) to the A/D conversion result when User's Gain adjustment function is used.

The overflow of A/D conversion data may occur when User's Gain adjusting function is used.

Table 36.40 List of gains corresponding to each gain setting bit in User's Gain setting table register (1 of 2)

UGAIN[23:16]	Gain value	UGAIN[15:8]	Gain value	UGAIN[7:0]	Gain value
b23	2^{1} = 2.0	b15	2^{-7} = 7.813E-03	b7	2^{-15} = 3.052E-05
b22	2^{0} = 1.0	b14	2^{-8} = 3.906E-03	b6	2^{-16} = 1.526E-05
b21	2^{-1} = 0.5	b13	2^{-9} = 1.953E-03	b5	2^{-17} = 7.629E-06

Table 36.40 List of gains corresponding to each gain setting bit in User's Gain setting table register (2 of 2)

UGAIN[23:16]	Gain value	UGAIN[15:8]	Gain value	UGAIN[7:0]	Gain value
b20	2^{-2} = 0.25	b12	2^{-10} = 9.766E-04	b4	2^{-18} = 3.815E-06
b19	2^{-3} = 0.125	b11	2^{-11} = 4.883E-04	b3	2^{-19} = 1.907E-06
b18	2^{-4} = 0.0625	b10	2^{-12} = 2.441E-04	b2	2^{-20} = 9.537E-07
b17	2^{-5} = 0.03125	b9	2^{-13} = 1.221E-04	b1	2^{-21} = 4.768E-07
b16	2^{-6} = 1.563E-02	b8	2^{-14} = 6.104E-04	b0	2^{-22} = 2.384E-07

Table 36.41 Example of User's Gain Settings

ADUGTRn.UGAIN[23:0] (n = 0 to 7)	Gain value
0x000000	x0.0000
⋮	⋮
0x040000	x0.0625
⋮	⋮
0x080000	x0.1250
⋮	⋮
0x100000	x0.2500
⋮	⋮
0x200000	x0.5000
⋮	⋮
0x400000 (initial value)	x1.0000
⋮	⋮
0x800000	x2.0000
⋮	⋮
0xFFFFFF	x3.9999

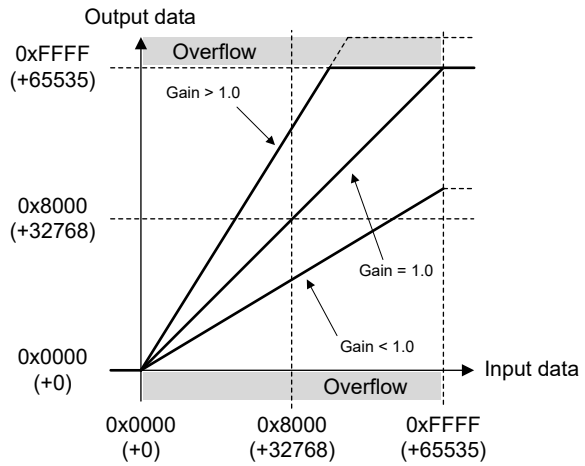


Figure 36.33 User's Gain Adjustment (16-bit data length format and Unsigned data format)

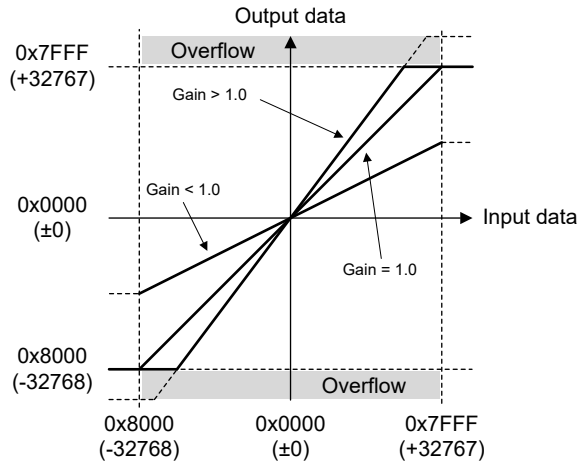


Figure 36.34 User's Gain Adjustment (16-bit data length format and Signed data format)

36.4.3.3 User's Offset Adjustment

User's Offset adjustment function adds or subtracts any constant value to or from the A/D conversion data.

The constant value of User's Offset is set in ADUOFTRn.UOFSET[15:0] (n = 0 to 7). User's Offset can be set for each virtual channel. User's Offset Table to be used is selected in ADDOPCRAm.OFSETSEL[3:0] (m = 0 to 36). Table 36.42 shows the relationship between the register setting value of User's Offset and the offset value. Figure 36.35 and Figure 36.36 show the relationship between before adjustment (input) and after adjustment (output) to the A/D conversion result when User's Offset adjustment function is used.

When User's Offset adjusting function is used, overflow of A/D conversion data may occur.

Table 36.42 Relationship between User's Offset register value and offset value

ADUOFTRn.UOFSET[15:0] (n = 0 to 7)	Offset value (16bit data length format)
0x7FFF	+32767
0x7FFE	+32766
0x7FFD	+32765
⋮	⋮
0x0003	+3
0x0002	+2
0x0001	+1
0x0000 (initial value)	0
0xFFFF	-1
0xFFFE	-2
0xFFFD	-3
⋮	⋮
0x8002	-32766
0x8001	-32767
0x8000	-32768

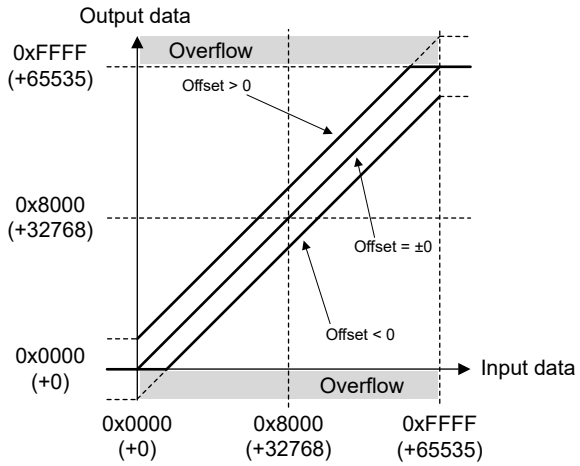


Figure 36.35 User's Offset Adjustment (16-bit data length format and Unsigned data format)

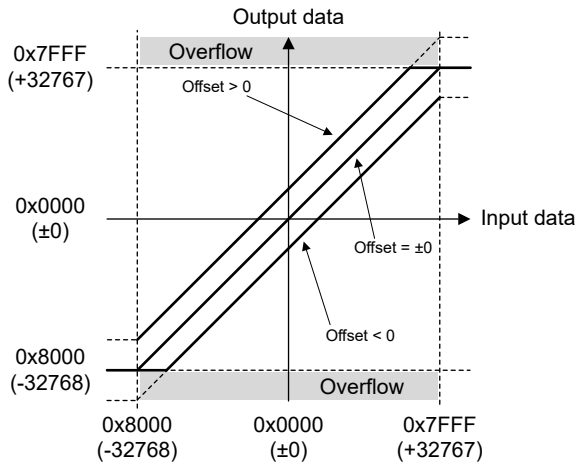


Figure 36.36 User's Offset Adjustment (16-bit data length format and Signed data format)

36.4.4 A/D-converted Value Addition/Averaging Function

A/D-converted value addition/averaging function is a function to calculate the sum value or average value of the result of A/D conversion of analog channels continuously for the specified number of times. A/D-converted value addition/averaging function can be configured for each virtual channel in ADDOPCRBn.AVEMD[1:0] bits and ADDOPCRBn.ADC[3:0] bits (n = 0 to 36).

A/D-converted value addition/averaging function inputs the output of the previous data processing. When the specified number of times of data is input, the added value or average value is calculated, and the result is output to the next data processing. When A/D-converted value addition/averaging function is used, the A/D converter repeats the A/D conversion until the data is collected for the number of times specified by the ADDOPCRBn.ADC[3:0] (n = 0 to 36) bits. One calculation result (total value or average value) is output every time the specified number of times of data is input to A/D-converted value addition/averaging. Figure 36.37 shows the operation of A/D-converted value addition/averaging function in SAR mode or Oversampling mode. Figure 36.38 shows the operation of A/D-converted value addition/averaging function in Hybrid mode.

When the A/D-converted value addition/averaging function is used, overflow of the A/D-converted data may occur. However, A/D conversion overflow may not be detected under certain conditions. For more information on A/D conversion overflow, see section 36.6.2. A/D Conversion Overflow.

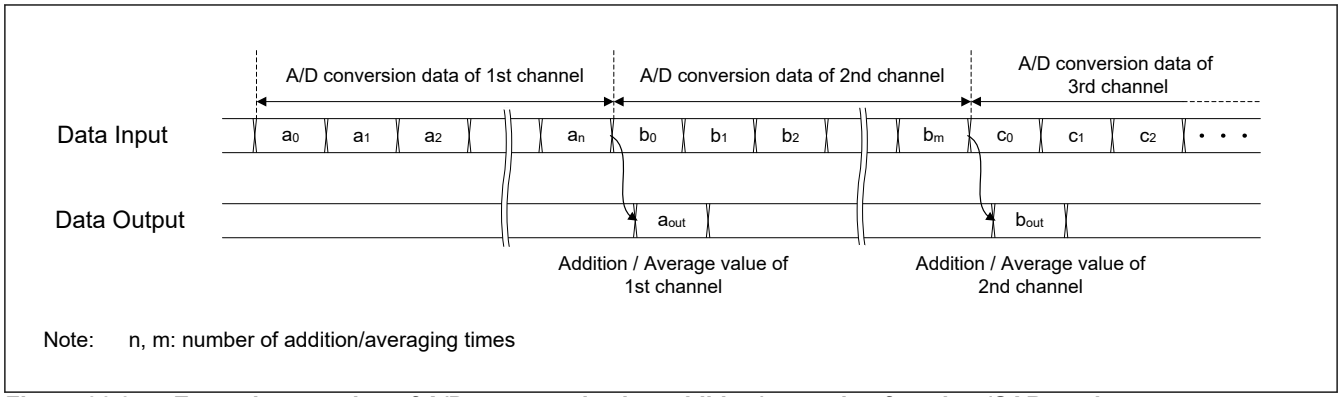


Figure 36.37 Example operation of A/D-converted value addition/averaging function (SAR mode or Oversampling mode)

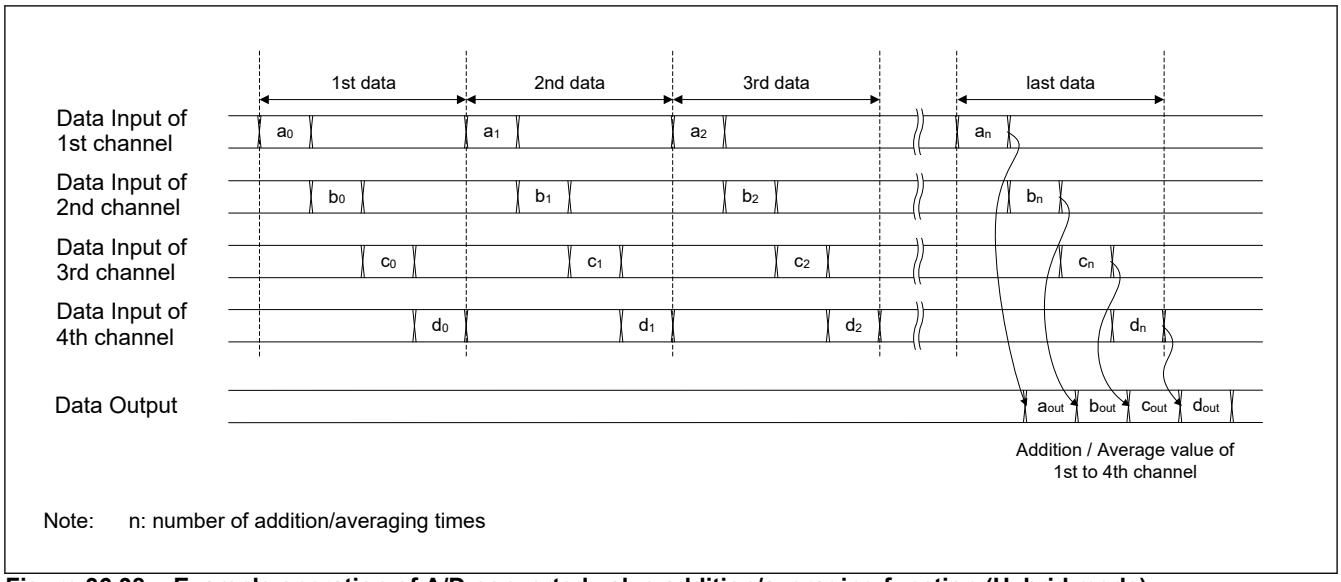


Figure 36.38 Example operation of A/D-converted value addition/averaging function (Hybrid mode)

36.4.5 Limiter Clip Function

Limiter Clip Function is the function to set the upper and lower limits of A/D conversion data. When the A/D conversion data exceeds the specified upper limit value, it is clipped to the upper limit value. If the A/D conversion data falls below the specified lower limit, it is clipped to the lower limit value. Figure 36.39 shows an example of Limiter Clip Function operation.

The upper and lower limits of Limiter Clip Function can be specified in ADLIMTRn.LIMU[15:0] (n = 0 to 7) and ADLIMTRn.LIML[15:0] (n = 0 to 7). The Limiter Clip Function can be selected use or not (enable or disable) for each virtual channel in ADDOPCRCm.LIMTBLS[3:0] (m = 0 to 36).

Limiter Clip Function is processed by 16-bit length. When 14-/12-/10-bit is selected as the data length of the A/D conversion data (when ADDOPCRCm.ADPRC[1:0] = 01b, 10b, or 11b (m = 0 to 36)), the rounding to the specified data length is performed after the clipping by Limiter Clip Function. For details, see section 36.4.6. Data Formatting process.

The upper and lower limits of the limiter clip are treated as signed or unsigned depending on the setting of the sign selection bit (ADDOPCRCm.SIGNSEL (m = 0 to 36)) in the A/D conversion data.

For Limiter Clip Function, the upper limit value should be greater than the lower limit value (ADLIMTRn.LIMU[15:0] > ADLIMTRn.LIML[15:0] (n = 0 to 7)). If the upper limit value is less than or equal to the lower limit value, the A/D conversion data always becomes 0x0000.

If a limiter clip occurs, the flags are set in the following status registers:

- ADLIMGRSR: The flag is set to the bit corresponding to the scan group where the limiter clip occurred.
- ADLIMCHSR0: If a limiter clip occurs during A/D conversion of an analog input channel, the flag is set to the corresponding bit.

- ADLIMEXSR: If a limiter clip occurs during A/D conversion of Extended Analog Function, the flag is set to the corresponding bit.

To clear the flags in the status registers, write 1 to the corresponding bit in ADLIMGRSCR, ADLIMCHSCR0, ADLIMEXSCR.

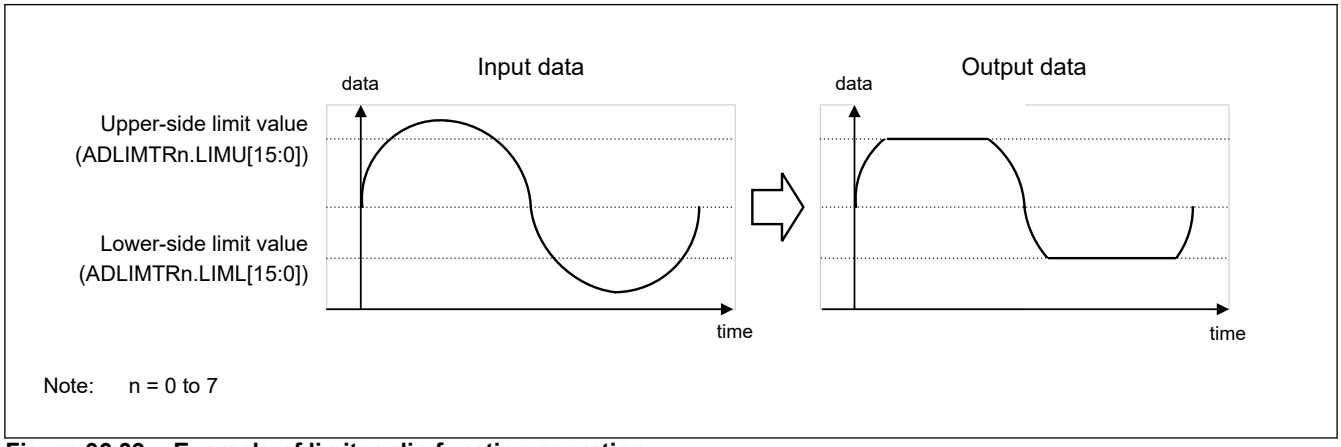


Figure 36.39 Example of limiter clip function operation

36.4.6 Data Formatting process

In this process, the following formatting is performed on the A/D conversion data.

- Signed/Unsigned data processing
- Data Rounding process

The data formatted in this process is stored to DATA[15:0] bits of the A/D data register (ADDRi (i = 0 to 28)), the extended A/D data register (ADEXDRj (j = 0 to 2, 5 to 8)), and the FIFO data register (ADFIFODRk (k = 0 to 8)) as the result of A/D conversion.

For details on the data format, see section 36.4.7. Data Format.

36.4.6.1 Signed/Unsigned Data Process

The A/D conversion data is formatted into Signed or Unsigned data format based on the setting of the ADDOPCRCn.SIGNSEL (n = 0 to 36) bits. Select the signed data format (SIGNSEL = 0) for Differential input mode and the unsigned format (SIGNSEL = 1) for Single-ended input mode.

36.4.6.2 Data Rounding Process

The A/D conversion data is processed as a fraction of the data length based on the setting of the ADDOPCRCn.ADPRC[1:0] (n = 0 to 36) bits. If 14-/12-/10-bit data format is selected (ADPRC[1:0] = 01b, 10b, 11b), the lower bits of the A/D conversion data will be rounding. (If the most significant bit of the digit to be cut off is 0, the digit is rounded down; if it is 1, the digit is rounded up.) If the 16-bit data format is selected (ADPRC[1:0] = 00b), the A/D conversion data is not rounded.

36.4.7 Data Format

This subsection describes the data format of the A/D conversion result stored to DATA[15:0] bits in A/D data register (ADDRi (i = 0 to 28)), Extended A/D data register (ADEXDRj (j = 0 to 2, 5 to 8)), and FIFO data register (ADFIFODRk (k = 0 to 8)).

Note: The data format does not guarantee the resolution and accuracy of the A/D converter itself. See section 46, Electrical Characteristics for the characteristics of the A/D converter.

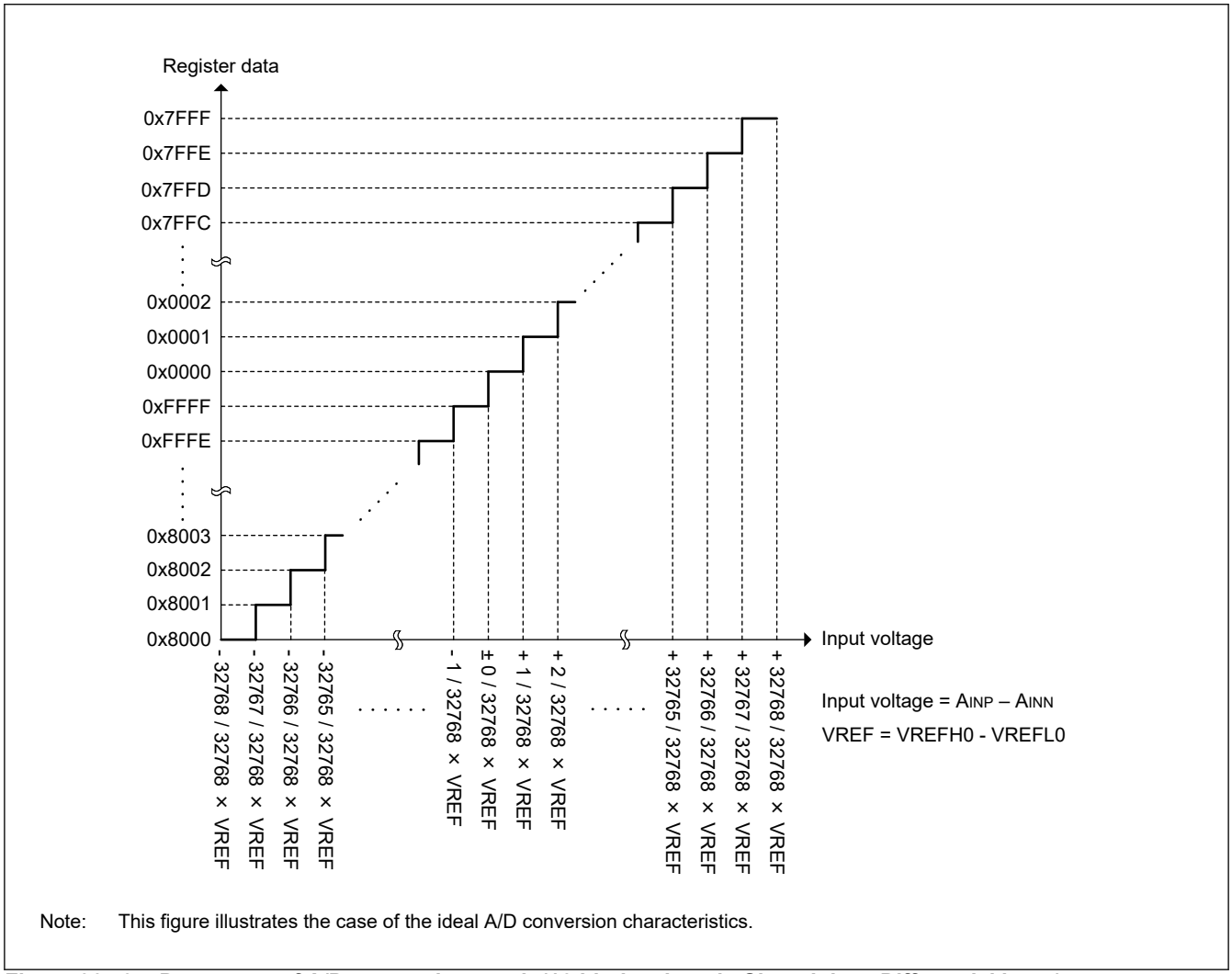
(1) 16-bits data length format

Table 36.43 shows the alignment of A/D conversion data on 16-bit data format. Figure 36.40 and Figure 36.41 show the data range for 16-bit data format.

For 16-bit data length signed data format, the differential input voltage is A/D-converted in the range of 0x8000 (-VREFH0) to 0x7FFF (+VREFH0). In 16-bit data length unsigned data format, the single-ended input voltage is A/D-converted in the range of 0x0000 (VREFL0) to 0xFFFF (VREFH0).

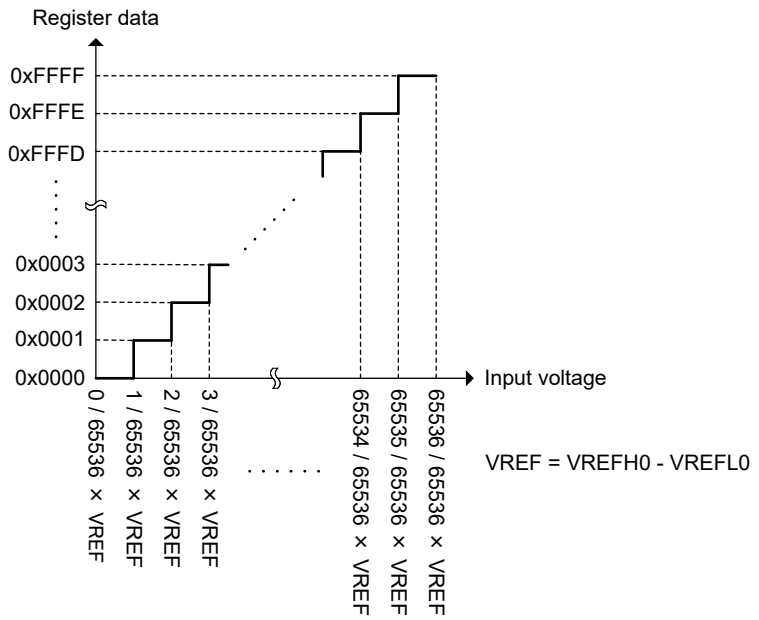
Table 36.43 Alignment of A/D conversion result data (16-bit data length)

Bit position:	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Bit field:	DATA [15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R



Note: This figure illustrates the case of the ideal A/D conversion characteristics.

Figure 36.40 Data range of A/D conversion result (16-bit data length, Signed data, Differential input)



Note: This figure illustrates the case of the ideal A/D conversion characteristics.

Figure 36.41 Data range of A/D conversion result (16-bit data length, Unsigned data, Single-ended input)

(2) 14-bits data length format

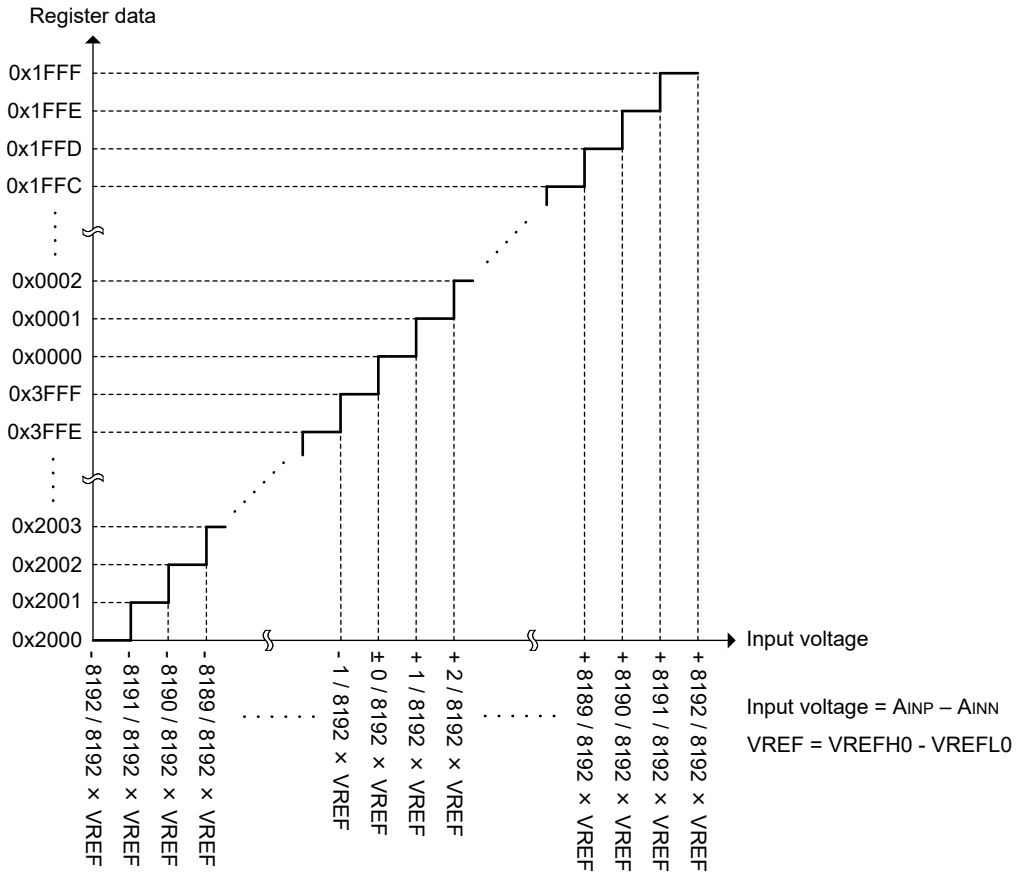
Table 36.44 shows the alignment of A/D conversion data on 14-bit data format. Figure 36.42 and Figure 36.43 show the data range for 14-bit data format.

For 14-bit data length signed data format, the differential input voltage is A/D-converted in the range of 0x2000 (-VREFH0) to 0x1FFF (+VREFH0). In 14-bit data length unsigned data format, the single-ended input voltage is A/D-converted in the range of 0x0000 (VREFL0) to 0x3FFF (VREFH0).

For 14-bit data length, the upper 2 bits (bit15 to bit14) are always 0.

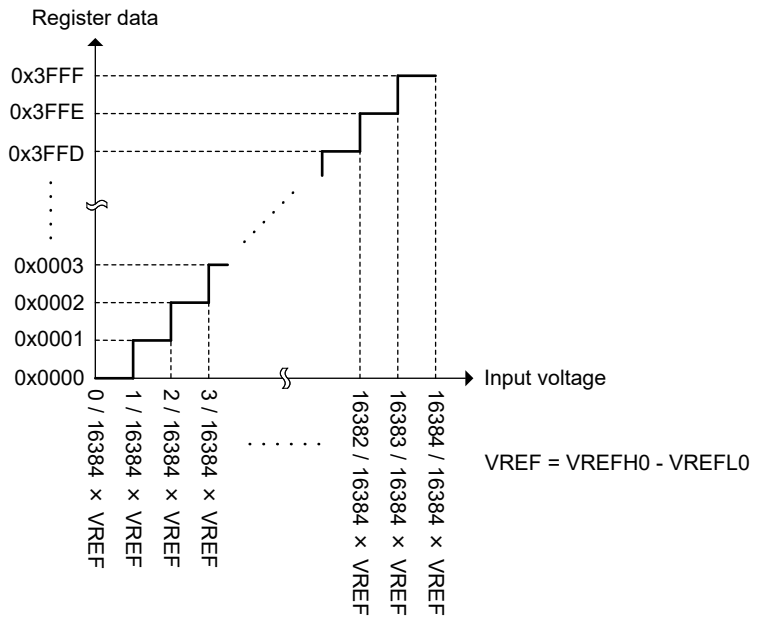
Table 36.44 Alignment of A/D conversion result data (14-bit data length)

Bit position:	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Bit field:	0	0	DATA[13:0]													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R



Note: This figure illustrates the case of the ideal A/D conversion characteristics.

Figure 36.42 Data range of A/D conversion result (14-bit data length, Signed data, Differential input)



Note: This figure illustrates the case of the ideal A/D conversion characteristics.

Figure 36.43 Data range of A/D conversion result (14-bit data length, Unsigned data, Single-ended input)

(3) 12-bits data length format

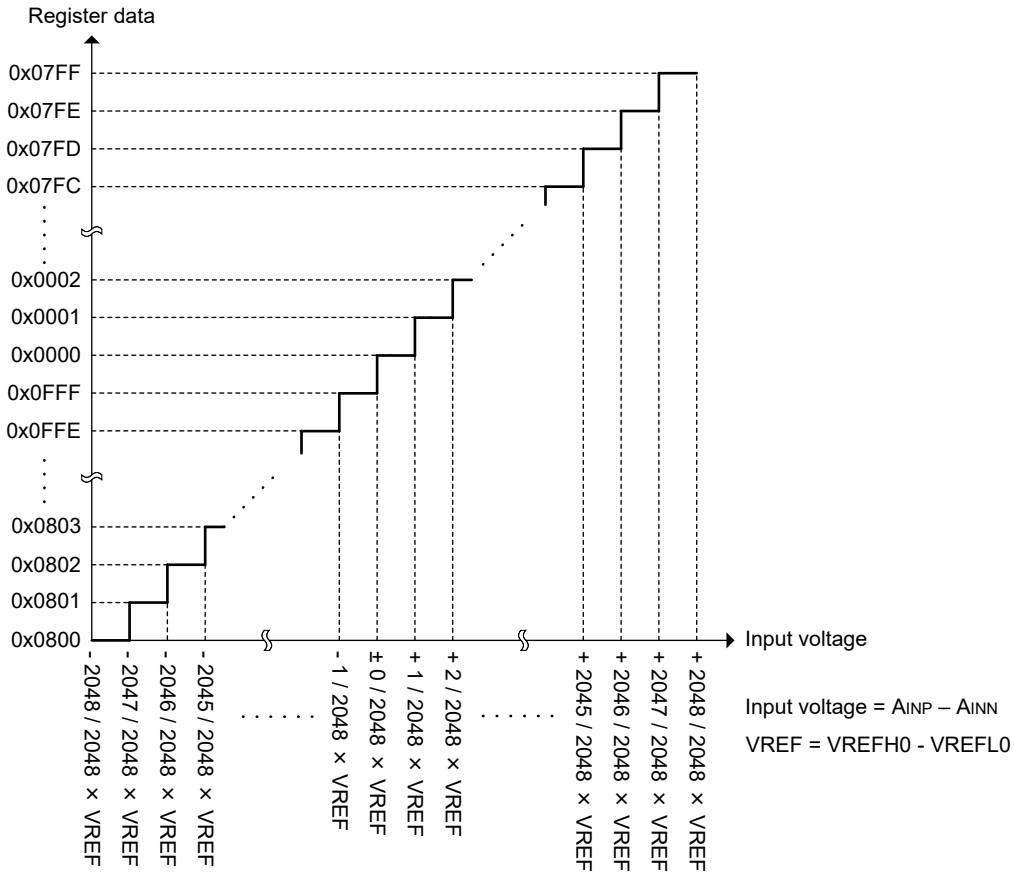
Table 36.45 shows the alignment of A/D conversion data on 12-bit data format. Figure 36.44 and Figure 36.45 show the data range for 12-bit data format.

For 12-bit data length signed data format, the differential input voltage is A/D-converted in the range of 0x0800 (-VREFH0) to 0x07FF (+VREFH0). In 12-bit data length unsigned data format, the single-ended input voltage is A/D-converted in the range of 0x0000 (VREFL0) to 0x0FFF (VREFH0).

For 12-bit data length, the upper 4 bits (bit15 to bit12) are always 0.

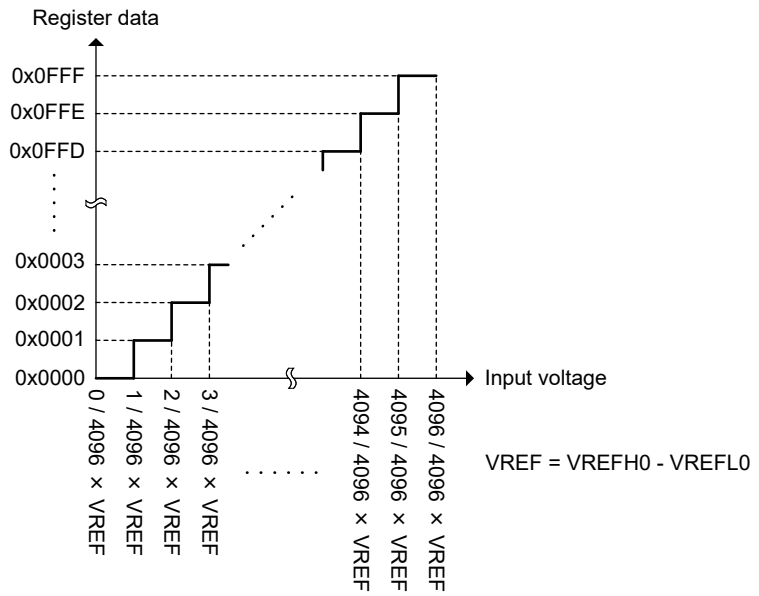
Table 36.45 Alignment of A/D conversion result data (12-bit data length)

Bit position:	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Bit field:	0	0	0	0	DATA[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R



Note: This figure illustrates the case of the ideal A/D conversion characteristics.

Figure 36.44 Data range of A/D conversion result (12-bit data length, Signed data, Differential input)



Note: This figure illustrates the case of the ideal A/D conversion characteristics.

Figure 36.45 Data range of A/D conversion result (12-bit data length, Unsigned data, Single-ended input)

(4) 10-bits data length format

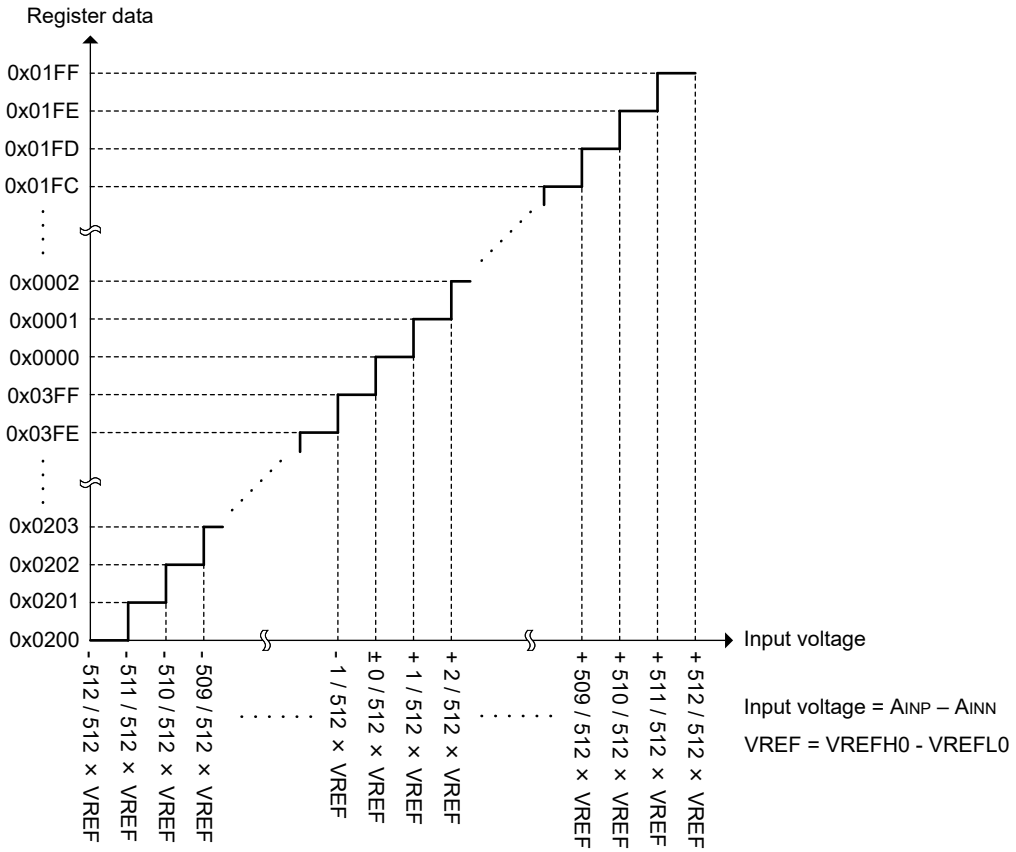
Table 36.46 shows the alignment of A/D conversion data on 10-bit data format. Figure 36.46 and Figure 36.47 show the data range for 10-bit data format.

For 10-bit data length signed data format, the differential input voltage is A/D-converted in the range of 0x0200 (-VREFH0) to 0x03FF (+VREFH0). In 10-bit data length unsigned data format, the single-ended input voltage is A/D-converted in the range of 0x0000 (VREFL0) to 0x03FF (VREFH0).

For 10-bit data length, the upper 6 bits (bit15 to bit10) are always 0.

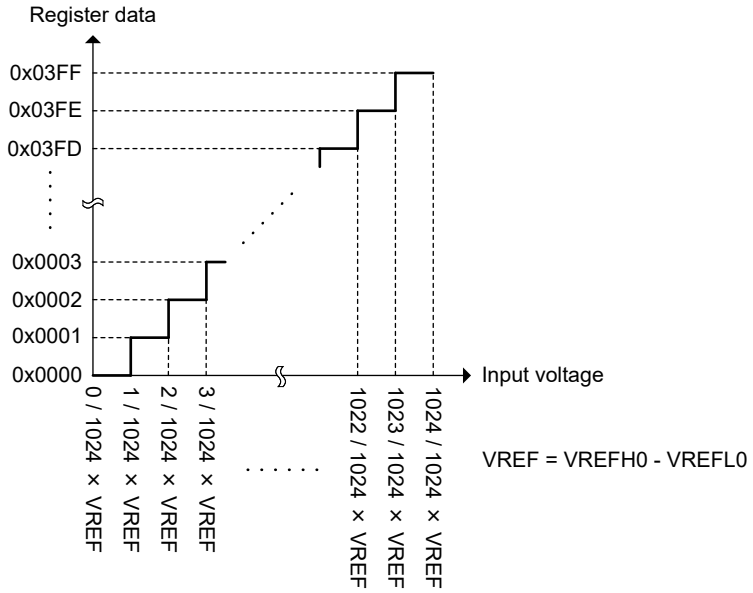
Table 36.46 Alignment of A/D conversion result data (10-bit data length)

Bit position:	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Bit field:	0	0	0	0	0	0	DATA [9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R



Note: This figure illustrates the case of the ideal A/D conversion characteristics.

Figure 36.46 Data range of A/D conversion result (10-bit data length, Signed data, Differential input)



Note: This figure illustrates the case of the ideal A/D conversion characteristics.

Figure 36.47 Data range of A/D conversion result (10-bit data length, Unsigned data, Single-ended input)

36.4.8 Compare Match Function

36.4.8.1 Compare Match

The compare match function compares the A/D conversion result with the reference value set in the compare match table register (ADCMPTBRn (n = 0 to 7). The compare match function compares the A/D conversion data after the data formatting process.

(1) Compare match mode

The comparison mode for detecting the compare match is selected in ADCMPMDRm.CMPMDn[1:0] (m = 0, 1. n = 0 to 7). The comparison mode can be selected from the following four modes.

1. Compare match is detected when the A/D conversion value is greater than or equal to the specified upper limit value.
2. Compare match is detected when the A/D conversion value is less than or equal to the specified lower limit.
3. Compare match is detected when the A/D conversion value is greater than or equal to the specified upper limit value, or less than or equal to the lower limit value.
4. Compare match is detected when the A/D conversion value is within the specified upper and lower limit values (when the value is lower than or equal to the upper limit value and greater than or equal to the lower limit value).

Figure 36.48 shows an example of compare match detection.

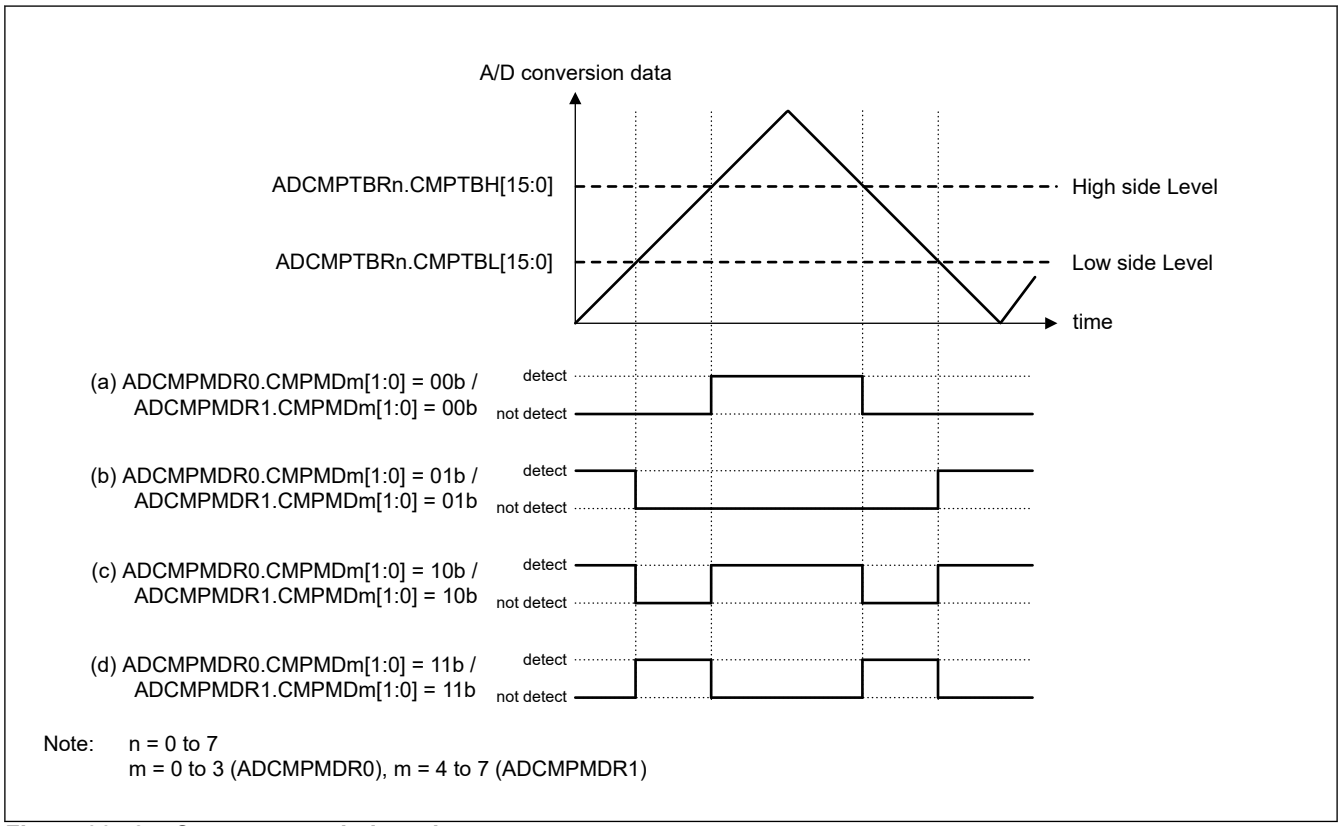


Figure 36.48 Compare match detection

(2) Compare match detection

To use the compare match function, enable the compare match table used in ADCMPENR register, and select the compare match table in ADDOPCRBn.CMPTBLEm (n = 0 to 36, m = 0 to 7) for each virtual channel.

When A/D conversion is performed on a virtual channel for which the compare match function is enabled, a compare match is judged based on the A/D conversion result of its channel.

When a compare match is detected, the flags are set in the following status registers:

- ADCMPTBSR: The flag is set to the bit corresponding to the compare match table used when a compare match is detected.
- ADCMPCHSR0: The flag is set to the bit corresponding to the analog input channel that detected the compare match.
- ADCMPEXSR: The flag is set to the bit corresponding to Extended Analog Function where a compare match is detected.

To clear the flag of each status register, write 1 to the corresponding bit of ADCMPTBSCR, ADCMPCHSCR0, ADCMPEXSCR.

If compare match interrupt is enabled in ADCMPINTCR register and a compare match using compare match table 0 to 3 is detected, the corresponding interrupt is generated. Check the compare match of compare match table 4 to 7 in ADCMPTBSR register.

36.4.8.2 Composite Compare Match

The composite compare match function generates interrupts and ELC events by combining the comparison results of multiple compare match tables. The combination and condition of the compare match table for the composite compare match function is set in ADCCMPCR0 or ADCCMPCR1.

Composite compare match interrupt is generated when a compare match is detected that matches the specified condition.

Figure 36.49 shows the relationship between composite compare match and compare match function. Table 36.47 shows the relationship between the composite compare match function and the control register.

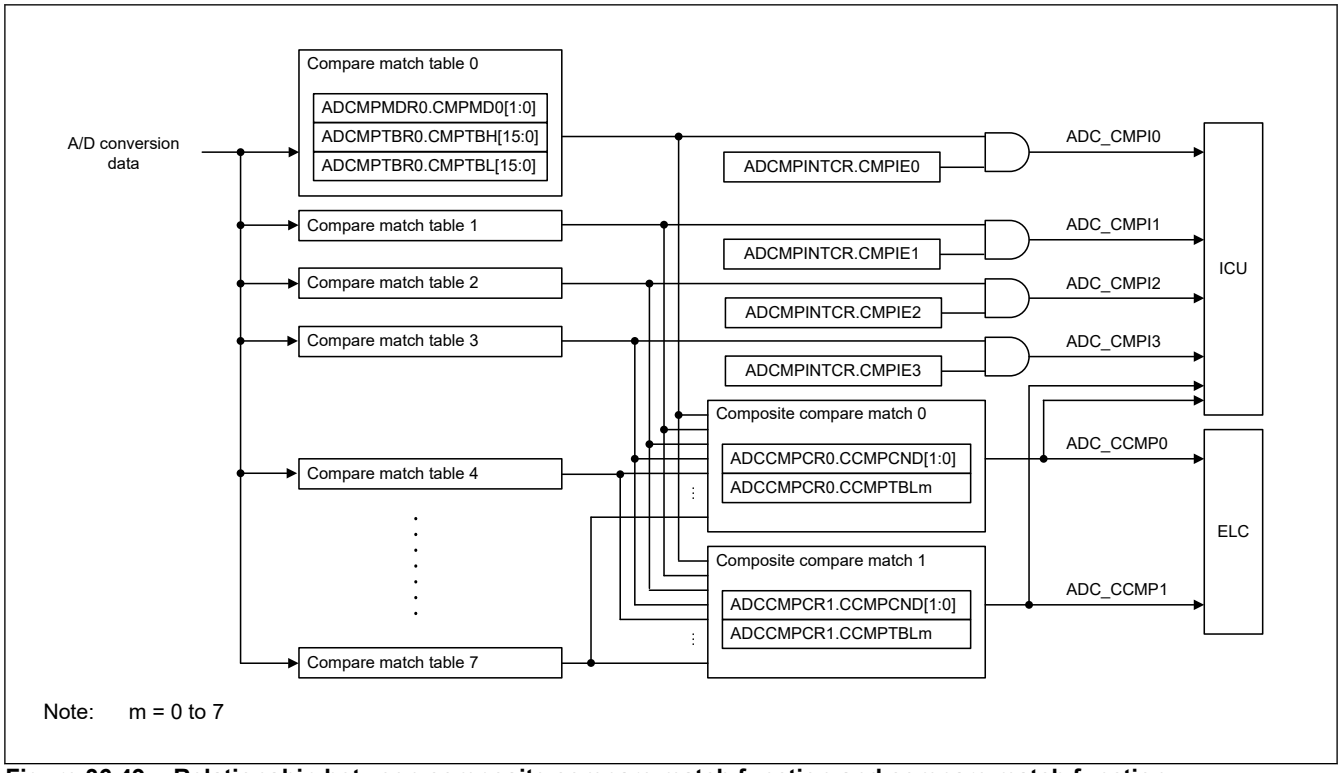


Figure 36.49 Relationship between composite compare match function and compare match function

Table 36.47 Composite compare match and control register

Function name	Enable composite compare function	Composite compare mode selection	Composite compare match interrupt
Composite compare match 0	ADCCMPCR0.CCMPTBLm (m = 0 to 7)	ADCCMPCR0.CCMPCND[1:0]	ADC_CCMPM0
Composite compare match 1	ADCCMPCR1.CCMPTBLm (m = 0 to 7)	ADCCMPCR1.CCMPCND[1:0]	ADC_CCMPM1

36.4.9 Data Registers

A/D data register (ADDRi (i = 0 to 28)) and Extended A/D data register (ADEXDRj (j = 0 to 2, 5 to 8)) store the A/D conversion data of the analog channels corresponding to each register. These registers are updated (overwritten) when A/D conversion for the analog channel corresponding to each register is completed.

36.4.10 FIFO Function

FIFO consists of 8-stage registers and can hold up to 8 A/D conversion data. One FIFO is implemented for each scan group. The A/D conversion data stored in FIFO can be read from ADFIFODRn register (n = 0 to 8).

FIFO acts as a ring buffer. Writing and reading A/D conversion data to and from FIFO is controlled by the write pointer and read pointer inside FIFO. The block diagram of FIFO is shown in Figure 36.50.

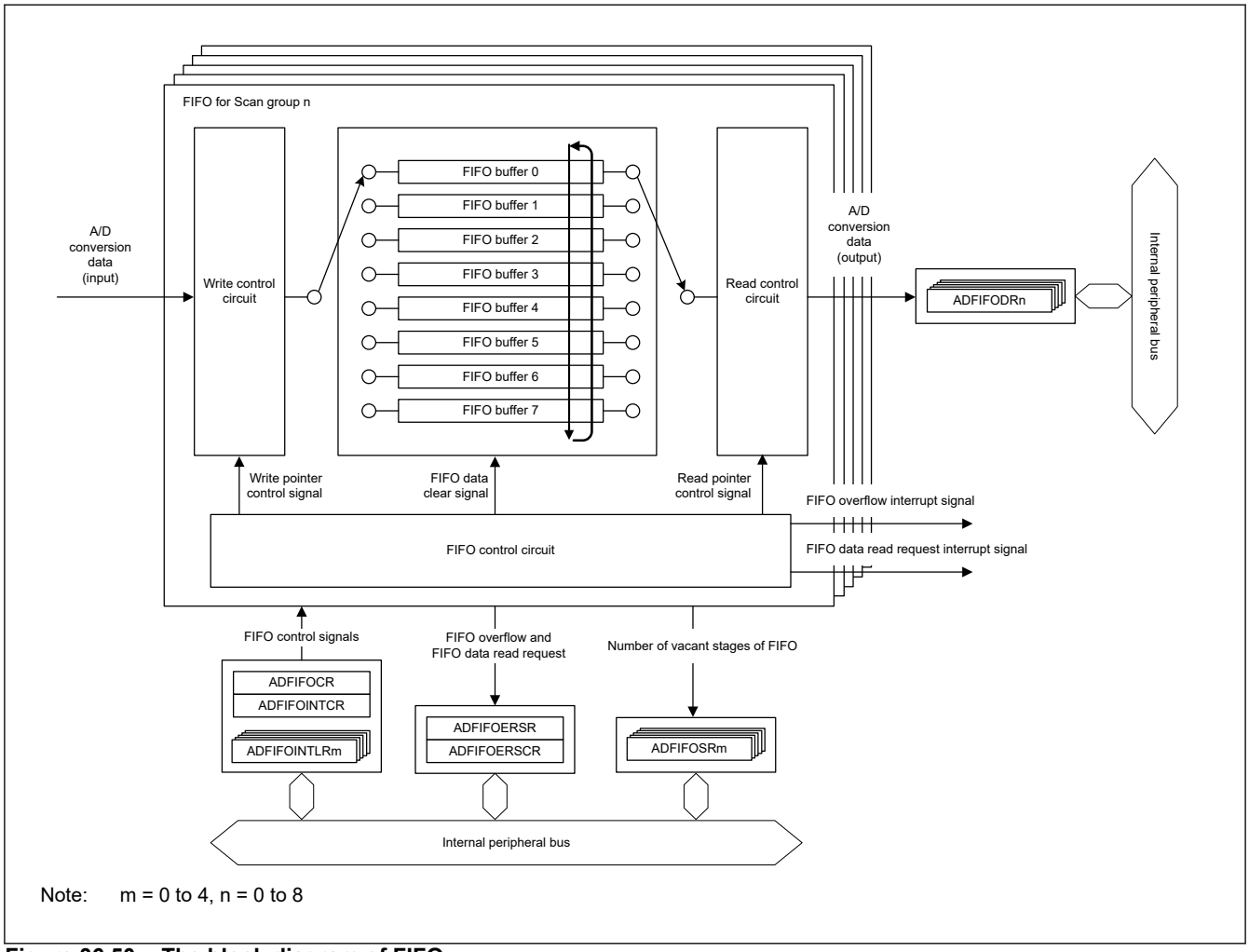


Figure 36.50 The block diagram of FIFO

36.4.10.1 Operation When Writing A/D Conversion Data to FIFO

The A/D conversion data is stored sequentially to the register pointed by the write pointer. When A/D conversion data is written to a register in FIFO, the write pointer is switched to the next register. At this time, the number of vacant stages (ADFIFOSRm.FIFOSTn[3:0] (m = 0 to 4, n = 0 to 8) bits) is decreased 1.

A FIFO overflow occurs when FIFO is full (ADFIFOSRm.FIFOSTn[3:0] bits are 0000b) and an additional A/D conversion data is written to FIFO. When a FIFO overflow occurs, the A/D conversion data is not written and the write pointer does not change.

36.4.10.2 Operation When Reading A/D Conversion Data from FIFO

When A/D conversion data is read from FIFO data register (ADFIFODRn (n = 0 to 8)), the data in the register pointed to by FIFO read pointer is read. When A/D conversion data is read from FIFO, the read pointer is switched to the next register. At this time, the number of vacant stages (ADFIFOSRm.FIFOSTn[3:0] (m = 0 to 4, n = 0 to 8) bits) is increased 1.

If FIFO is empty (ADFIFOSRm.FIFOSTn[3:0] bits are 1111b) and the reading from the FIFO is occurred, an invalid data is read. (If FIFO is read after a reset or after FIFO data is cleared, 0x00000000 is read. If data had been written to register in FIFO, the previous data is read.) The read pointer does not change in the read operation when FIFO is empty.

Read A/D conversion data from FIFO data register (ADFIFODRn) by 32-bit access. For details, see section 36.4.10.5. Usage Notes on FIFO.

36.4.10.3 FIFO Data Read Request and Overflow

When A/D conversion data is written to or read from FIFO, the write pointer or read pointer changes. Depending on the state of the write pointer and the read pointer, the number of vacant stages in FIFO is reflected in ADFIFOSRm.FIFOSTn[3:0] (m = 0 to 4, n = 0 to 8).

FIFO data read request flag is set (ADFIFOERSR.FIFOFLFn = 1 (n = 0 to 8)) when the number of vacant stages become less than or equal to the value specified in ADFIFOINTLRm.FIFOILVn[3:0] (m = 0 to 4, n = 0 to 8) bit. If a FIFO data read request occurs, FIFO data read request interrupt can be generated.

When FIFO is full (ADFIFOSRm.FIFOSTn[3:0] bits are 0000b) and additional A/D conversion data are written, FIFO overflow flag is set (ADFIFOERSR.FIFOOVFn = 1 (n = 0 to 8)). If a FIFO overflow occurs, FIFO data overflow interrupt can be generated.

36.4.10.4 Clearing FIFO Data

The data stored in the registers in FIFO can be cleared by writing to ADFIFODCR.FIFODCn (n = 0 to 8) bit. When writing 1 to ADFIFODCR.FIFODCn bit, the registers, read pointer, and write pointer in FIFO of the scan group corresponding to FIFODCn bit are initialized. Clearing FIFO data should be performed while A/D conversion is not being performed.

36.4.10.5 Usage Notes on FIFO

Reading the A/D conversion data from the FIFO data register (ADFIFODRn (n = 0 to 8)) should be performed with a single 32-bit access; access to each bit of ADFIFODRn, 16-bit read access, and 8-bit read access are prohibited. If these restrictions are violated, the operation of FIFO is not guaranteed. (If these restrictions are violated, the read pointer may not change normally, and correct data may not be able to read. Alternatively, the read pointer may advance unintentionally, and A/D conversion data may be lost.)

36.4.11 A/D Conversion Data Error Detection

The ERR bit in A/D data register (ADDRi (i = 0 to 28)), Extended A/D data register (ADEXDRj (j = 0 to 2, 5 to 8)), and FIFO data register (ADFIFODRk (k = 0 to 8)) is a flag that indicates an A/D conversion data error. An A/D conversion data error indicates that an invalid A/D conversion data has been detected. If an A/D conversion data error occurs (ERR = 1), the A/D conversion data cannot be guaranteed.

A/D conversion data error is generated by the following factors.

- A/D converter error
- A/D conversion overflow

For more details about these errors, see section 36.6. Error Detection.

36.5 Start and Stop Control of A/D Conversion

36.5.1 Software Trigger

A/D conversion of scan group n can be started by writing 1 to ADSTRn.ADST (n = 0 to 8) bit or ADSYSTR.ADSYSTn (n = 0 to 8) bit.

ADSTRn.ADST bit is used to start A/D conversion individually for each scan group. ADSYSTR.ADSYSTn bit is used to start A/D conversion on the scan group assigned to ADC0 or ADC1 at the same time.

Except when group priority operation, writing to ADSTRn.ADST bit or ADSYSTR.ADSYSTn bit for the scan groups, which use A/D converter that is in A/D conversion, is ignored.

36.5.2 Peripheral Module Triggers

A/D conversion can be started by triggers from the peripheral modules listed in the followings.

- ELC Trigger
- GPT Trigger
- External Trigger (I/O port)

To perform A/D conversion by a trigger from a peripheral module, configure the trigger for each scan group and enable the trigger input from the peripheral module in ADTRGENR register.

36.5.2.1 ELC Trigger

A/D conversion can be started by an event (ELC event) from Event Link Controller. To start A/D conversion using an ELC event, configure the scan group using the ELC event in ADTRGELCn (n = 0 to 8) register and enable the A/D conversion start trigger in ADTRGENR register. Table 36.48 shows the correspondence between ADTRGELCn register and ELC event.

Table 36.48 Correspondence between ELC Trigger Enable Register and ELC Event

Resister bit	Event Name
ADTRGELCn.TRGELC0	ELC_AD00
ADTRGELCn.TRGELC1	ELC_AD01
ADTRGELCn.TRGELC2	ELC_AD02
ADTRGELCn.TRGELC3	ELC_AD10
ADTRGELCn.TRGELC4	ELC_AD11
ADTRGELCn.TRGELC5	ELC_AD12

Note: n = 0 to 8

36.5.2.2 GPT Trigger

A/D conversion can be started by an interrupt source from the general-purpose PWM Timer (GPT). To start A/D conversion using an interrupt source from the GPT, configure the scan group using the GPT interrupt source in ADTRGGPTn (n = 0 to 8) register and enable the A/D conversion start trigger in ADTRGENR register. Table 36.49 shows the correspondence between ADTRGGPTn register and GPT interrupt sources.

Table 36.49 Correspondence between GPT Trigger Enable Registers and GPT Interrupt Sources (1 of 2)

Resister bit	Event Name
ADTRGGPTn.TRGGPTA0	GPT0_ADTRGA
ADTRGGPTn.TRGGPTA1	GPT1_ADTRGA
ADTRGGPTn.TRGGPTA2	GPT2_ADTRGA
ADTRGGPTn.TRGGPTA3	GPT3_ADTRGA
ADTRGGPTn.TRGGPTA4	GPT4_ADTRGA
ADTRGGPTn.TRGGPTA5	GPT5_ADTRGA
ADTRGGPTn.TRGGPTA6	GPT6_ADTRGA
ADTRGGPTn.TRGGPTA7	GPT7_ADTRGA
ADTRGGPTn.TRGGPTA8	GPT8_ADTRGA
ADTRGGPTn.TRGGPTA9	GPT9_ADTRGA
ADTRGGPTn.TRGGPTB0	GPT0_ADTRGB
ADTRGGPTn.TRGGPTB1	GPT1_ADTRGB

Table 36.49 Correspondence between GPT Trigger Enable Registers and GPT Interrupt Sources (2 of 2)

Resister bit	Event Name
ADTRGGPTn.TRGGPTB2	GPT2_ADTRGB
ADTRGGPTn.TRGGPTB3	GPT3_ADTRGB
ADTRGGPTn.TRGGPTB4	GPT4_ADTRGB
ADTRGGPTn.TRGGPTB5	GPT5_ADTRGB
ADTRGGPTn.TRGGPTB6	GPT6_ADTRGB
ADTRGGPTn.TRGGPTB7	GPT7_ADTRGB
ADTRGGPTn.TRGGPTB8	GPT8_ADTRGB
ADTRGGPTn.TRGGPTB9	GPT9_ADTRGB

Note: n = 0 to 8

36.5.2.3 External Trigger

A/D conversion can be started by the input from the external trigger pin (ADTRG0 and ADTRG1). To start A/D conversion using an external trigger, configure the scan group using the external trigger in ADTRGEXTn (n = 0 to 8) register and enable the A/D conversion start trigger in ADTRGENR register.

External trigger is active low. Before setting to enable the external trigger (ADTRG0 and ADTRG1), you should input the High level to the external trigger pin.

36.5.3 Trigger Delay

The trigger delay function adds a delay to the A/D conversion start trigger in order to adjust the A/D conversion start timing for each scan group. This function adds a delay to the internal trigger for each scan group generated by accepting an external trigger, ELC trigger or GPT trigger. This function cannot be used to add a delay to a software trigger.

The delay value to be added to the internal trigger is set for each scan group in ADTRGDLRi (i = 0 to 4) register. The delay value to be added is the register setting value × A/D conversion clock (ADCLK) cycle.

36.5.4 Force Stops the A/D Conversion Operation

The scanning operation of the A/D converter can be forcibly stopped by writing 1 to ADSTOPR.ADSTOPm (m = 0, 1) bit during scanning operation. If the scanning operation is forcibly stopped, the A/D conversion data is not guaranteed.

36.5.4.1 Forced Stop Procedure

Table 36.50 shows the procedure for forced stop of A/D conversion operation. Observe Table 36.50 when performing the forced stop. If this procedure is violated, the A/D converter may not be able to stop and then it may not operate normally. In that case, a reset is required for recovery.

Table 36.50 Procedure for forced stop of A/D conversion operation (1 of 2)

No.	Step Name	Description
1	Disable trigger input	Disables the trigger input from the peripheral module. (Write ADTRGENR.STTRGENn = 0)
2	Waiting time	After setting the above Step 1, a wait time is required to safely stop the A/D converter. Proceed to the next processing after the specified waiting time has elapsed. For details on the wait time, see section 36.5.4.2. Waiting Time after Disabling Trigger Input for Forced Stop Processing.
3	Checking the A/D converter status	Check if the A/D converter is operating after the waiting time of the above Step 2 has elapsed. If the A/D converter is running (ADSR.ADACTm = 1), proceed to Step 4. If the A/D converter is stopped (ADSR.ADACTm = 0), no further processing is required (proceed to Step 6).
4	Forced stop of A/D converter	Forcibly stop the A/D converter by ADSTOPR register. *1 (Write ADSTOPR.ADSTOPm = 1)
5	Waiting for A/D converter to stop	Wait until the A/D converter stops. (ADSR.ADACTm = 0)

Table 36.50 Procedure for forced stop of A/D conversion operation (2 of 2)

No.	Step Name	Description
6	End	Processing of forced stop is completed.

Note: m = 0, 1, n = 0 to 8
Do not write to ADSYSTR register or ADSTRn register while this forced stop procedure is in progress.
Note 1. Forced stopping during self-calibration operation is prohibited. For details, see section 36.3.8. Self-calibration.

36.5.4.2 Waiting Time after Disabling Trigger Input for Forced Stop Processing

In the forced stop procedure shown in Table 36.50, after disabling the trigger input from the peripheral module, a wait time is required to perform the forced stop (Step 2 of Table 36.50). This wait time is the time that must be secured to perform a forced stop safely.

The wait time after disabling the trigger input for forced stop processing is calculated as follows.

[Waiting time after disabling trigger input in forced stop processing]

- When synchronous operation is enabled (ADSYCR.ADSYDISm = 0)
  - (ADTRGDLRi.TRGDLYn[7:0] setting value + ADSYCR.ADCSYCYC[10:0] setting value × 2) × t_{ADCLK}
- When synchronous operation is disabled (ADSYCR.ADSYDISm = 1)
  - (4 + ADTRGDLRi.TRGDLYn[7:0] setting value) × t_{ADCLK}

Note: i = 0 to 4, n = 0 to 8, m = 0, 1 t_{ADCLK}: ADCLK period

After the waiting time calculated by the above expressions has elapsed, proceed to the Step 3 in Table 36.50. There are two ways to generate the waiting time: (a) Use the timer function built into MCU. (b) Read the status register multiple times. If you choose (b), the number of register reads to generate the waiting time required is calculated by the following equation:

[Number of register reads to generate the waiting time]

N_{RD} = (Wait time after disabling trigger input ÷ (t_{PCLK} × N_{RDCYC})) + 1

N_{RD}: Number of register reads required to generate the trigger input disable wait time (truncate after the decimal point)

t_{PCLK}: The period of the Bus I/F clock (PCLKA)

N_{RDCYC}: Minimum number of register read access cycles (see section Appendix 3, I/O Registers.)

36.6 Error Detection

36.6.1 A/D Converter Error

An A/D converter error is detected when an abnormal operation of the A/D converter occurs. If an A/D converter error is detected, the A/D conversion result is not guaranteed.

An A/D converter error occurs in the following cases:

[A/D Converter Error Generation Conditions]

- When A/D conversion is performed without self-calibration.
- When the operating frequency of ADCLK exceeds the guaranteed operating frequency range specified in the electric characteristics.
- When the successive approximation time of the A/D converter exceeds the guaranteed operating range specified in the electric characteristics.
- When an accidental abnormal operation occurs in the A/D converter due to external factors.

If an A/D converter error is detected, the flag is set in ADERSR.ADERFn (n = 0, 1) bit. To clear the A/D converter error flag, write 1 to the corresponding bit in ADERSCR register.

Notes on A/D Converter Error

The A/D converter error is an auxiliary function used to detect abnormal operation of the A/D converter. It does not guarantee the reliable detection of abnormal operation. Even under the A/D converter error occurrence conditions described above, an A/D converter error may not be detected depending on the individual chip differences and chip operating conditions.

36.6.2 A/D Conversion Overflow

A/D conversion overflow is detected when A/D conversion data exceeds the range that can be handled in the specified data format. If A/D conversion overflow occurs, the A/D conversion data becomes the upper or lower limit value of the specified data format. For the range that can be handled as A/D conversion data, see section 36.4.7. Data Format.

A/D conversion overflow is detected in the following cases:

- When the input to the A/D converter exceeds the range between VREFH0 to VREFL0
  - Single-ended input: When the input to the A/D converter exceeds VREFH0 or falls below VREFL0
  - Differential input: When the differential input to the A/D converter exceeds +VREFH0 or falls below -VREFH0
- When overflow occurs by the internal processing (calculation) for the A/D conversion data shown in the following:
  - Gain error and Offset error calibration
  - User's Gain/User's Offset adjustment function
  - When using A/D-converted value addition function
  - Data formatting process

When an A/D conversion overflow is detected, the flag is set in the following status register:

- ADOVFERSR: The flag is set to the bit corresponding to the A/D converter that detected the A/D conversion overflow.
- ADOVFCHSR0: The flag is set to the bit corresponding to the analog input channel on which A/D conversion overflow is detected.
- ADOVFEXSR: The flag is set to the bit corresponding to the extended analog channel on which A/D conversion overflow is detected.

To clear the flags in the status registers, write 1 to the corresponding bit in ADOVFERSCR, ADOVFCHSCR0, ADOVFEXSCR.

Restrictions on A/D conversion overflow

When A/D-converted value addition/averaging function is used, A/D conversion overflow is not detected under the following conditions.

- Leading channel of scan group
- Restart channel of the interrupted group in group priority operation

To detect that the A/D conversion data is illegal under these conditions, use limiter clip function or compare match function.

36.6.3 FIFO Overflow

If A/D-converted data is written (added) to FIFO while FIFO is full, a FIFO overflow is detected. For more details about FIFO overflow, see section 36.4.10. FIFO Function.

36.7 Procedure for Setting up and Changing

36.7.1 Initial Setup Procedure

Table 36.51 shows the initial setup procedure.

Table 36.51 Initial setup procedure

No.	Step	Description
1	Release module-stop	Release the module-stop bit for ADC in MSTPCR register.
2	I/O port configuration	Set ASEL bit of the pin used as analog input to 1.
3	Synchronous operation configuration	Set the synchronous operation function. The synchronous operation function is enabled at the initial value of the register after reset release. If the synchronous operation function is not used, be sure to disable the synchronous operation function (ADSYCR.ADSYDISm = 1 (m = 0, 1)).
4	ADCLK configuration	Set the clock source and division ratio for ADCLK. Then, set ADCLKENR.CKEN bit to 1, and wait for ADCLK to supply (ADCLKSR.CLKSR = 1).
5	A/D conversion configuration	Configure the A/D conversion settings.
6	Wait for operation stabilization	Wait until the operating stabilization times specified in the Electrical Characteristics have elapsed.
7	Self-calibration	Self-calibration must be executed prior to starting A/D conversion. Set up for self-calibration and execute it. For details, see section 36.3.8. Self-calibration.
8	Trigger configuration	To start A/D conversion by a trigger from peripheral modules, configure the triggers for each scan group.
9	Start of A/D conversion	When a software trigger or a trigger from peripheral modules is input, A/D conversion (scanning operation) starts.

36.7.2 Procedure for Changing ADCLK Settings

Table 36.52 shows the procedure for changing ADCLK setting.

Table 36.52 Procedure for changing ADCLK setting (1 of 2)

No.	Step	Description
1	Disable trigger input	Disables the trigger input from the peripheral module. (Write ADTRGENR.STTRGENn = 0 (n = 0 to 8))
2	Stop A/D conversion	Check that all A/D converter is stopped. When A/D conversion is in progress, wait until all A/D conversion is completed or forcibly stop A/D conversion operation. For details about forcibly stop A/D conversion operation, see section 36.5.4. Force Stops the A/D Conversion Operation.
3	Stop ADCLK supplies	Set ADCLKENR.CKEN bit to 0. Then, wait for ADCLK to stop (ADCLKSR.CLKSR = 0).
4	Change ADCLK setting	Change the clock-source and the division ratio for ADCLK.
5	Started supplying ADCLK	Set ADCLKENR.CKEN bit to 1, and wait for ADCLK to supply (ADCLKSR.CLKSR = 1).
6	Change the A/D conversion configuration	Change the following settings according to the changed ADCLK frequency. • Successive approximation time for A/D converter • Number of sampling states for A/D conversion • Number of sampling states and hold mode switching states for channel-dedicated sample-and-hold circuit *1 • Number of states for self-calibration operation (A/D converter and channel-dedicated sample-and-hold circuit *1) • Synchronous operation period *2 • Disconnection detection assist period *3 If other settings related to A/D conversion are also to be changed, change them in this step.
7	Wait for operation stabilization	Wait until the operating stabilization times specified in the section 46, Electrical Characteristics have elapsed.
8	Self-calibration	Execute self-calibration operation prior to starting A/D conversion. For details, see section 36.3.8. Self-calibration.

Table 36.52 Procedure for changing ADCLK setting (2 of 2)

No.	Step	Description
9	Trigger configuration	To start A/D conversion by a trigger from peripheral modules, configure the triggers for each scan group.
10	Start of A/D conversion	When a software trigger or a trigger from peripheral modules is input, A/D conversion (scanning operation) starts.

Note 1. Setting is not required when channel-dedicated sample-and-hold circuit is not used.
Note 2. No change is required when synchronous operation setting is disabled (ADSYCR, ADSYDISm = 1 (m = 0, 1))
Note 3. Setting is not required when disconnection detection assist function is not used.

36.7.3 Procedure for Changing the Settings of the A/D Converter

Table 36.53 shows the procedure for changing the A/D conversion configuration. If there is a change in ADCLK setting, follow the procedures in section 36.7.2. Procedure for Changing ADCLK Settings.

Table 36.53 Procedure for changing the settings of the A/D converter

No.	Step	Description
1	Disable trigger input	Disables the trigger input from the peripheral module. (Write ADTRGENR.STTRGENn = 0 (n = 0 to 8))
2	Stop A/D conversion	Check that all A/D converter is stopped. When A/D conversion is in progress, wait until all A/D conversion is completed or forcibly stop A/D conversion operation. For details about forcibly stop A/D conversion operation, see section 36.5.4. Force Stops the A/D Conversion Operation.
3	Change the A/D conversion configuration	Change the A/D conversion settings. (except the settings for ADCLK)
4	Wait for operation stabilization	Wait until the various operating stabilization times specified in the Electrical Characteristics have elapsed.
5	Self-calibration	Execute self-calibration operation prior to starting A/D conversion. For details, see section 36.3.8. Self-calibration.
6	Trigger configuration	To start A/D conversion by a trigger from peripheral modules, configure the triggers for each scan group.
7	Start of A/D conversion	When a software trigger or a trigger from peripheral modules is input, A/D conversion (scanning operation) starts.

36.8 Interrupt Sources and ELC Events

Table 36.54 lists ADC interrupt sources or ELC event requests.

For more details about interrupts, see section 12, Interrupt Controller Unit (ICU).

For more details about ELC events, see section 17, Event Link Controller (ELC).

Table 36.54 Interrupt sources (1 of 3)

Interrupt request or ELC event	Symbol	Description	Status flag
A/D converter error interrupt	ADC_ERR0	Generated when abnormal operation of ADC0 is detected.	ADERSR.ADERF0
	ADC_ERR1	Generated when abnormal operation of ADC1 is detected.	ADERSR.ADERF1
A/D converter self-calibration end interrupt	ADC_CALEND0	Generated when self-calibration operation of ADC0 is finished.	ADCALENDSR.CALENDF0
	ADC_CALEND1	Generated when self-calibration operation of ADC1 is finished.	ADCALENDSR.CALENDF1

Table 36.54 Interrupt sources (2 of 3)

Interrupt request or ELC event	Symbol	Description	Status flag
A/D scan end interrupt	ADC_ADI0	Generated at the scan end of scan group 0	ADSCANENDSR.SCENDF0
	ADC_ADI1	Generated at the scan end of scan group 1	ADSCANENDSR.SCENDF1
	ADC_ADI2	Generated at the scan end of scan group 2	ADSCANENDSR.SCENDF2
	ADC_ADI3	Generated at the scan end of scan group 3	ADSCANENDSR.SCENDF3
	ADC_ADI4	Generated at the scan end of scan group 4	ADSCANENDSR.SCENDF4
	ADC_ADI5678	Generated at the scan end of scan group 5 to 8	ADSCANENDSR.SCENDF5 ADSCANENDSR.SCENDF6 ADSCANENDSR.SCENDF7 ADSCANENDSR.SCENDF8
Limitter clip interrupt	ADC_LIMCLPI	Generated when a limiter clip using limiter table 0 to 7 is detected for A/D conversion.	ADLIMGRSR.LIMGRFn ADLIMCHSR0.LIMCHFi ADLIMEXSR.LIMEXFj
A/D conversion overflow interrupt	ADC_RESOVF0	Generated when the overflow is detected in A/D conversion result with ADC0.	ADOVFERSR.ADOVFEF0 ADOVFCHSR0.OVFCHFi ADOVFEXSR.OVFEXFj
	ADC_RESOVF1	Generated when the overflow is detected in A/D conversion result with ADC1.	ADOVFERSR.ADOVFEF1 ADOVFCHSR0.OVFCHFi ADOVFEXSR.OVFEXFj
Compare match interrupt	ADC_CMPI0	Generated when a compare match with compare match table 0 is detected.	ADCMPTBSR.CMPTBF0
	ADC_CMPI1	Generated when a compare match with compare match table 1 is detected.	ADCMPTBSR.CMPTBF1
	ADC_CMPI2	Generated when a compare match with compare match table 2 is detected.	ADCMPTBSR.CMPTBF2
	ADC_CMPI3	Generated when a compare match with compare match table 3 is detected.	ADCMPTBSR.CMPTBF3
Composite compare match interrupt	ADC_CCMPM0 ADC_CCMPM1	Generated when a composite compare match with combined conditions using compare match table 0 to 7 is detected.	ADCMPTBSR.CMPTBF0 to ADCMPTBSR.CMPTBF8

Table 36.54 Interrupt sources (3 of 3)

Interrupt request or ELC event	Symbol	Description	Status flag
FIFO data read request interrupt	ADC_FIFOREQ0	Generated when the number of vacant stages in FIFO for scan group 0 become less than or equal to the specified value.	ADFIFOSR0.FIFOST0[3:0]
	ADC_FIFOREQ1	Generated when the number of vacant stages in FIFO for scan group 1 become less than or equal to the specified value.	ADFIFOSR0.FIFOST1[3:0]
	ADC_FIFOREQ2	Generated when the number of vacant stages in FIFO for scan group 2 become less than or equal to the specified value.	ADFIFOSR1.FIFOST2[3:0]
	ADC_FIFOREQ3	Generated when the number of vacant stages in FIFO for scan group 3 become less than or equal to the specified value.	ADFIFOSR1.FIFOST3[3:0]
	ADC_FIFOREQ4	Generated when the number of vacant stages in FIFO for scan group 4 become less than or equal to the specified value.	ADFIFOSR2.FIFOST4[3:0]
	ADC_FIFOREQ5678	Generated when the number of vacant stages in any of FIFO for scan groups 5 to 8 become less than or equal to the specified value.	ADFIFOSR2.FIFOST5[3:0] ADFIFOSR3.FIFOST6[3:0] ADFIFOSR3.FIFOST7[3:0] ADFIFOSR4.FIFOST8[3:0]
FIFO data overflow interrupt	ADC_FIFOOVF	Generated when FIFO overflow is detected in any of FIFO for scan group 0 to 8.	ADFIFOERSR.FIFOOVF0 to ADFIFOERSR.FIFOOVF8

Note: n = 0 to 8.
i = 0 to 28.
j = 0 to 2, 5 to 8.

(1) A/D converter error interrupt

A/D converter error interrupt can be generated when an A/D converter error is detected.

A/D converter error interrupt (ADC_ERRm (m = 0, 1)) is generated when ADERINTCR.ADEIEm (m = 0, 1) bit is 1 and ADERSR.ADERFm (m = 0, 1) bit is 1.

(2) A/D converter self-calibration end interrupt

A/D converter self-calibration end interrupt can be generated at the end of A/D converter self-calibration operation.

A/D converter self-calibration end interrupt (ADC_CALENDm (m = 0, 1)) is generated when ADCALINTCR.CALENDIEm (m = 0, 1) bit is 1 and ADCALENDSR.CALENDFm (m = 0, 1) bit is 1.

(3) A/D scan end interrupt

A/D scan end interrupt can be generated at the end of scanning operation of scan group n (n = 0 to 8).

A/D scan end interrupt for scan group 0 to 4 (ADC_ADI0 to ADC_ADI4) are generated when ADINTCR.ADIEn (n = 0 to 4) bit is set to 1 and ADSCANENDSR.SCENDFn (n = 0 to 4) bit is set to 1.

A/D scan end interrupt for scan group 5 to 8 (ADC_ADI5678) is generated when ADINTCR.ADIEn (n = 5 to 8) bit is set to 1 and ADSCANENDSR.SCENDFn (n = 5 to 8) bit is set to 1 is satisfied in any of the scan group 5 to 8.

However, A/D scan end interrupt is not generated when A/D conversion operation (scanning operation) is forcibly stopped by ADSTOPR register.

(4) Limitter clip interrupt

Limiter clip interrupt (ADC_LIMCLPI) can be generated when a limiter clip with limiter table i (i = 0 to 7) is detected.

Limiter clip interrupt is generated when either ADLIMINTCR.LIMIEi (i = 0 to 8) bit is 1 and ADLIMGRSR.LIMGRFi (i = 0 to 8) bit is 1 is satisfied.

(5) A/D conversion overflow interrupt

A/D conversion overflow interrupt can be generated when A/D conversion overflow in the A/D conversion result using ADCm (m = 0, 1) is detected. For details about A/D conversion overflow, see section 36.6.2. A/D Conversion Overflow.

A/D conversion overflow interrupt (ADC_RESOVFm (m = 0, 1)) is generated when ADOVFINTCR.ADOVFIEm (m = 0, 1) bit is 1 and ADOVFERSR.ADOVFEFm (m = 0, 1) bit is 1.

(6) Compare match interrupt

Compare match interrupt can be generated when a compare match is detected.

Compare match interrupt (ADC_CMPIj (j = 0 to 3)) is generated when ADCMPINTCR.CMPIEj (j = 0 to 3) bit is 1 and ADCMPTBSR.CMPTBFj (j = 0 to 3) bit is 1.

There are no interrupts corresponding to the compare match tables 4 to 7.

(7) Composite compare match interrupt

Composite compare match interrupt (ADC_CCMPMk (k = 0, 1)) can be generated by combining the comparison results of two or more compare match tables. For details about composite compare match, see section 36.4.8.2. Composite Compare Match.

(8) FIFO data read request interrupt

FIFO data read request interrupt can be generated when the number of vacant stages in FIFO become less than or equal to the specified value.

FIFO data read request interrupt for scan group 0 to 4 (ADC_FIFOREQ0 to ADC_FIFOREQ4) are generated when ADFIFOINTCR.FIFOIEn (n = 0 to 4) is set to 1 and ADFIFOSRm.FIFOSTn[3:0] ≤ ADFIFOINTLRm.FIFOILVn[3:0] (m = 0 to 2, n = 0 to 4).

FIFO data read request interrupt for scan group 5 to 8 (ADC_FIFOREQ5678) are generated when the following condition is satisfied in any FIFO for scan group 5 to 8: ADFIFOINTCR.FIFOIEn (n = 5 to 8) is set to 1 and ADFIFOSRm.FIFOSTn[3:0] ≤ ADFIFOINTLRm.FIFOILVn[3:0] (m = 3 to 4, n = 5 to 8)

However, FIFO read request interrupt is not generated while the ADFIFOERSR.FIFOFLFn (n = 0 to 8) bit corresponding to its interrupt source is set to 1.

(9) FIFO data overflow interrupt

FIFO data overflow interrupt (ADC _FIFOOVF) can be generated when a FIFO overflow is detected in any of FIFO in scan groups 0 to 8.

FIFO data overflow interrupt is generated when ADFIFOINTCR.FIFOIEn (n = 0 to 8) bit is 1 and ADFIFOERSR.FIFOOVFn (n = 0 to 8) bit is 1.

36.9 Scan Conversion Time

36.9.1 Scan Start Processing Time

The processing time from the input of the A/D conversion start trigger to the start of the scanning operation is shown in Table 36.55 and Figure 36.51.

Table 36.55 Scan start processing time (1 of 2)

Item	Symbol	Processing time
Peripheral module trigger input processing time*1	t_{D_TRG}	1 ADSRCCLK + 3 ADCLK + (ADTRGDLRr.TRGDLYn[7:0] + 1) × ADCLK

Table 36.55 Scan start processing time (2 of 2)

Item		Symbol	Processing time
Software trigger input processing time		t_{D_SW}	[When ADCLK = PCLKA/1][*2] ● Number of access cycles to I/O registers[*3] + 1 PCLKA [Other than above] ● (Number of access cycles to I/O registers[*3]) + 1 PCLKA + (3 to 4 ADCLK)
Internal Trigger processing time		t_{D_ITRG}	3 ADCLK
Wait Time for Synchronous Operation		t_{D_SYOP}	[When synchronous operation is disabled] (ADSYCR.ADSYDISm = 1) ● 0 [When synchronous operation is enabled] (ADSYCR.ADSYDISm = 0) ● 0 to ADSYCR.ADSYCYC[10:0] × ADCLK
Total scan start processing time	Started by Peripheral Module Trigger	t_{D_ADST}	$t_{D\_TRG} + t_{D\_ITRG} + t_{D\_SYOP}$
	Started by Software Trigger		$t_{D\_SW} + t_{D\_ITRG} + t_{D\_SYOP}$

Note: ADSRCCLK: Clock source of ADCLK
n = 0 to 8, m = 0, 1, r = 0 to 4
Note 1. It does not include the input delay time from the source of the peripheral module trigger to ADC.
Note 2. When ADCLKCR.CLKSEL[1:0] = 10b and ADCLKCR.DIVR[2:0] = 000b are set.
Note 3. It is the number of access cycles to ADSTRn register (n = 0 to 8) or ADSYSTR register. For more details about the number of access cycles for I/O registers, see section Appendix 3, I/O Registers.

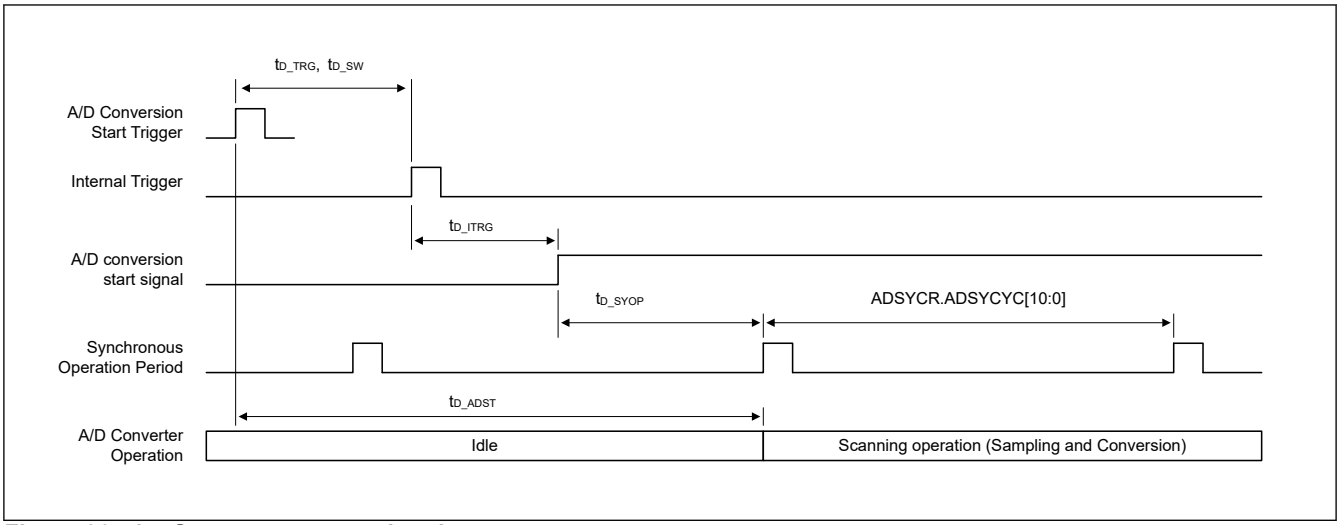


Figure 36.51 Scan start processing time

36.9.2 Conversion Processing Time

The various processing times in the conversion operation are shown in Table 36.56 and Figure 36.52 to Figure 36.56.

Table 36.56 A/D conversion processing time (1 of 2)

Item		Symbol	Processing Time
Channel-dedicated sample-and-hold processing time	Sampling time	t_{SH_SPL}	ADSHSTRm.SHSST[7:0] × ADCLK
	Hold mode switching time	t_{SH_HLD}	ADSHSTRm.SHHST[2:0] × ADCLK
	Sampling mode switching time (only when Hybrid mode)	t_{SH_D}	(ADSYCR.ADSYCYC[10:0] - 1) × ADCLK
Disconnection detect assist processing time		t_{DDA}	[When disconnection detect assist function is disabled] ● 0 [When disconnection detect assist function is enabled] ● ADSGDCRn.ADNDIS[3:0] × ADCLK

Table 36.56 A/D conversion processing time (2 of 2)

Item			Symbol	Processing Time
A/D conversion time	Sampling time		t_{AD_SPL}	ADSSTRp.SSTq[9:0] × ADCLK
	Successive approximation time		t_{AD_CNV}	ADCNVSTR.CSTm[5:0] × ADCLK
A/D conversion data processing time	SAR mode (Not using digital filter function)	When ADCLK = PCLKA/1 is set*1	t_{ADDP}	[When A/D-converted value addition/averaging function is not used] • 6 ADCLK + 2 PCLKA [When A/D-converted value addition/averaging function is used] • 7 ADCLK + 2 PCLKA
		Other than above		[When A/D-converted value addition/averaging function is not used] • 7 ADCLK + (5 to 6 PCLKA) [When A/D-converted value addition/averaging function is used] • 8 ADCLK + (5 to 6 PCLKA)
	Oversampling mode or Hybrid mode (Using with digital filter function)	When ADCLK = PCLKA/1 is set*1		[When A/D-converted value addition/averaging function is not used] • 8 ADCLK + 2 PCLKA [When A/D-converted value addition/averaging function is used] • 9 ADCLK + 2 PCLKA
		Other than above		[When A/D-converted value addition/averaging function is not used] • 9 ADCLK + 5 to 6 PCLKA [When A/D-converted value addition/averaging function is used] • 10 ADCLK + 5 to 6 PCLKA
Total A/D conversion time (SAR mode)*3	Channel conversion time*2		t_{ADCH_S}	$(t_{DDA} + t_{AD\_SPL} + t_{AD\_CNV}) \times N_{ADC} \times$ ADCLK
	Scan conversion time*4		t_{SCAN_S}	[When channel-dedicated sample-and-hold circuit is not used] • Σt_{ADCH_S} [When channel-dedicated sample-and-hold circuit is used] • $t_{SH\_SPL} + t_{SH\_HLD} + \Sigma t_{ADCH\_S}$
Total A/D conversion time (Oversampling mode)*3	Oversampling period		t_{OV_OS}	$(t_{DDA} + t_{AD\_SPL} + t_{AD\_CNV}) \times$ ADCLK
	Channel conversion time		t_{ADCH_O}	$(t_{DDA} + t_{AD\_SPL} + t_{AD\_CNV}) \times (N_{TAP} + N_{ADC}) \times$ ADCLK
	Scan conversion time*5		t_{SCAN_O}	Σt_{ADCH_O}
Total A/D conversion time (Hybrid mode)*3	Channel-dedicated sample-and-hold processing time in Hybrid mode		t_{HY_SH}	$t_{SH\_SPL} + t_{SH\_HLD} + t_{SH\_D}$
	Oversampling period		t_{HY_OS}	$(t_{DDA} + t_{AD\_SPL} + t_{AD\_CNV}) \times$ ADCLK
	Scan conversion time	Initial delay	t_{HY_ID}	[When channel-dedicated sample-and-hold circuit is not used] • $(N_{TAP} + N_{ADC}) \times \Sigma t_{HY\_OS}$*6 [When channel-dedicated sample-and-hold circuit is used] • $(t_{HY\_SH} \times (N_{ADC} + N_{TAP}) - t_{SH\_D}) + (N_{TAP} + N_{ADC}) \times \Sigma t_{HY\_OS}$*6
		Group delay	t_{HY_GD}	[When channel-dedicated sample-and-hold circuit is not used] • Σt_{HY_OS}*6 [When channel-dedicated sample-and-hold circuit is used] • $t_{HY\_SH} + \Sigma t_{HY\_OS}$*6

Note: n = 0 to 8, m = 0, 1, p = 0 to 7, q = 0 to 15
N_{ADC}: This value is the number of the times of the addition/averaging according to the setting value of ADDOPCRBx.ADC[3:0] (x = 0 to 36). If A/D-converted value addition/averaging function is not used, this value is 1.
N_{TAP}: This value is the number of TAP of the digital filter that selected in ADDOPCRAx.DFSEL[2:0] (x = 0 to 36) and ADDFSRm.DFSELy[1:0] (m = 0, 1, y = 0 to 4).
N_{SGCH}: This value is the number of the channels in the scan group.

Note 1. When ADCLKCR.CLKSEL[1:0] = 10b and ADCLKCR.DIVR[2:0] = 000b are set.

Note 2. It does not include the channel-dedicated sample-and-hold processing time.
Note 3. It does not include the A/D conversion data processing time.
Note 4. This is the sum of the channel conversion times (t_{ADCH_N}) that calculated from the conversion settings for each analog channel assigned to the scan group. If channel-dedicated sample-and-hold circuits are used, channel-dedicated sample-and-hold processing times (t_{SH_SPL} and t_{SH_HLD}) are also added.
Note 5. This is the sum of the channel conversion times (t_{ADCH_O}) that calculated from the conversion settings for each analog channel assigned to the scan group.
Note 6. This is the sum of the oversampling period (t_{HY_OS}) that calculated from the conversion settings for each analog channel assigned to the scan group.

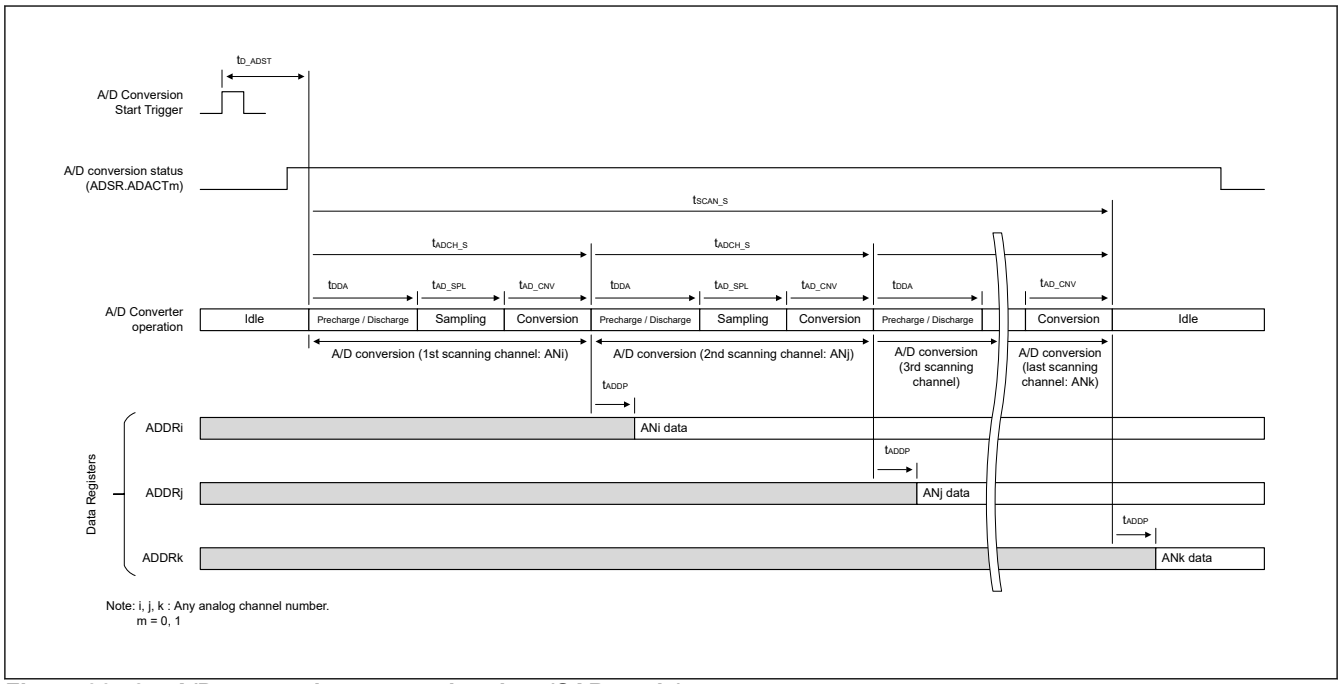


Figure 36.52 A/D conversion processing time (SAR mode)

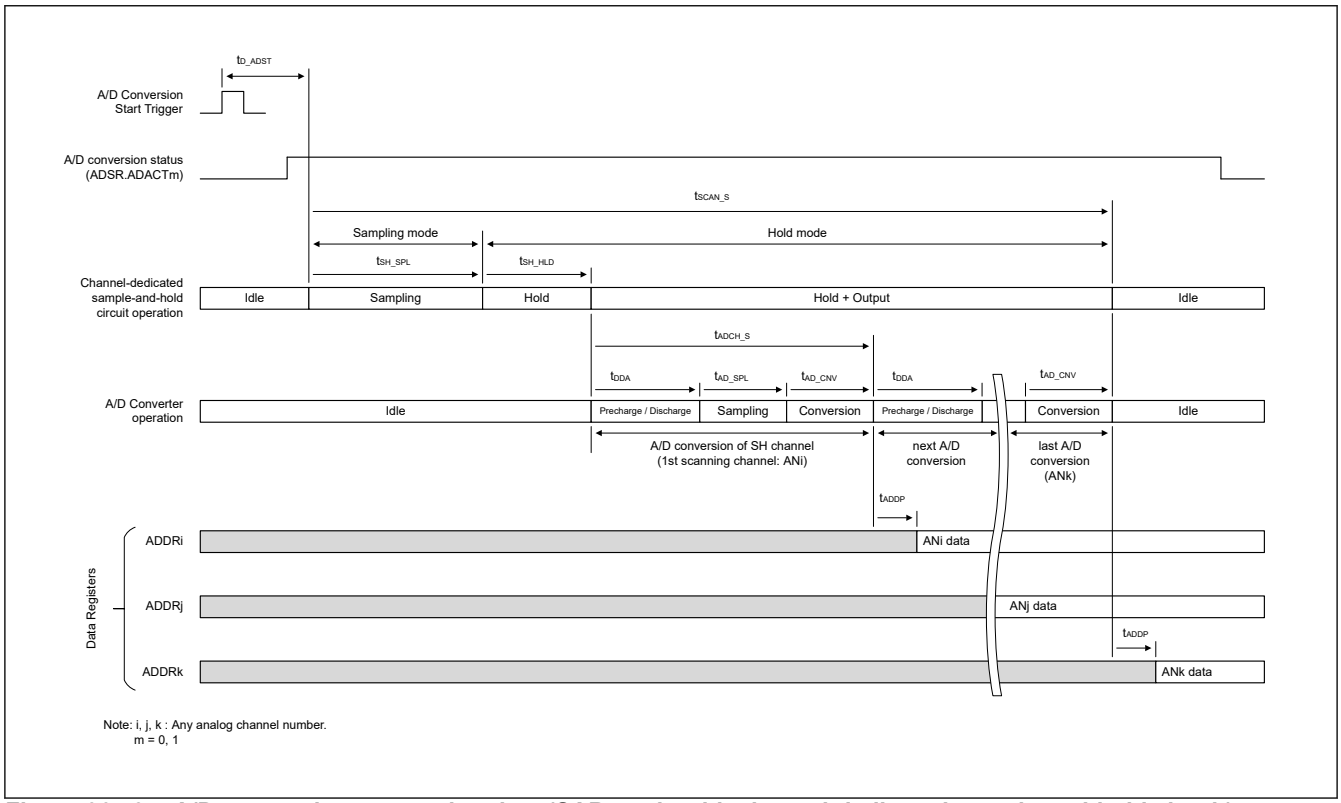


Figure 36.53 A/D conversion processing time (SAR mode with channel-dedicated sample-and-hold circuit)

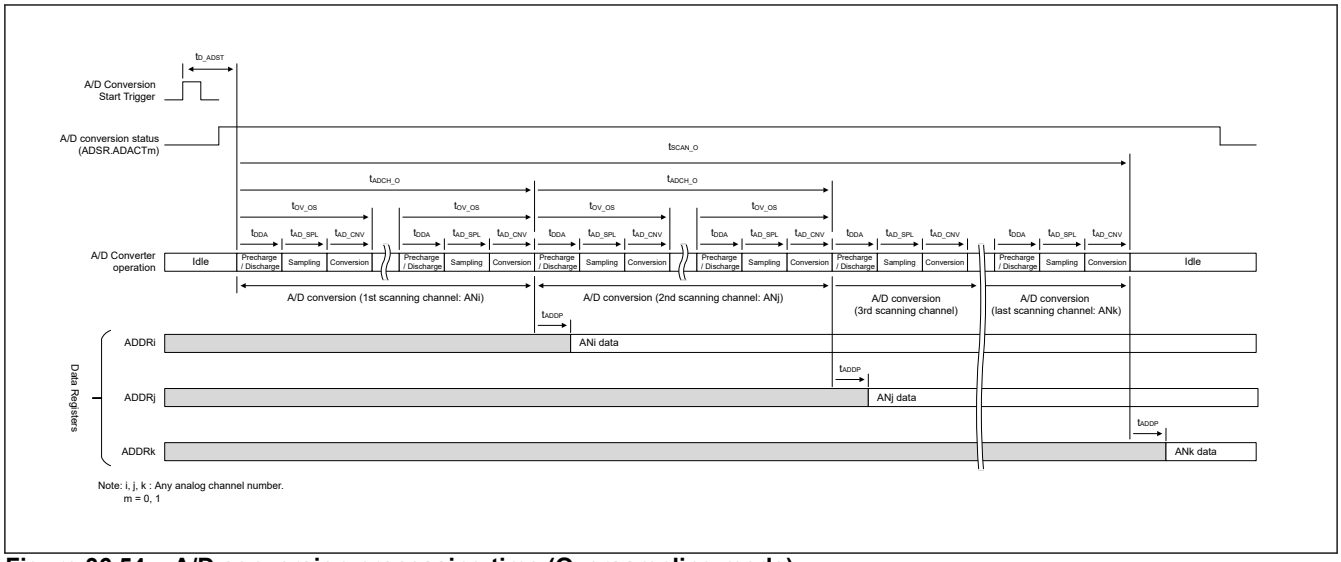


Figure 36.54 A/D conversion processing time (Oversampling mode)

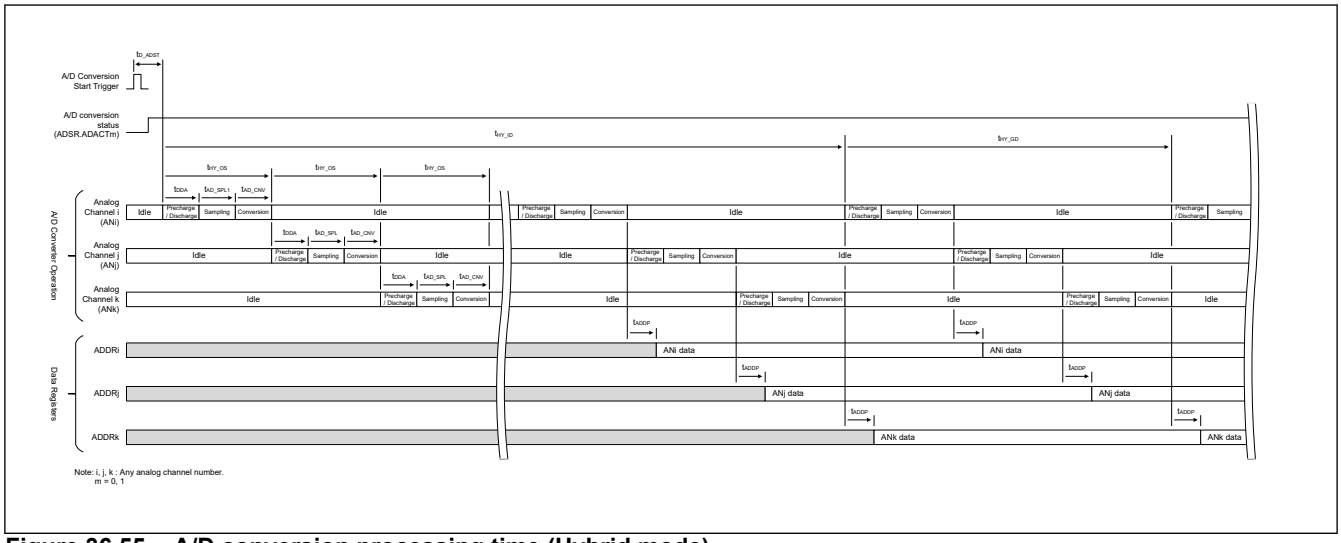


Figure 36.55 A/D conversion processing time (Hybrid mode)

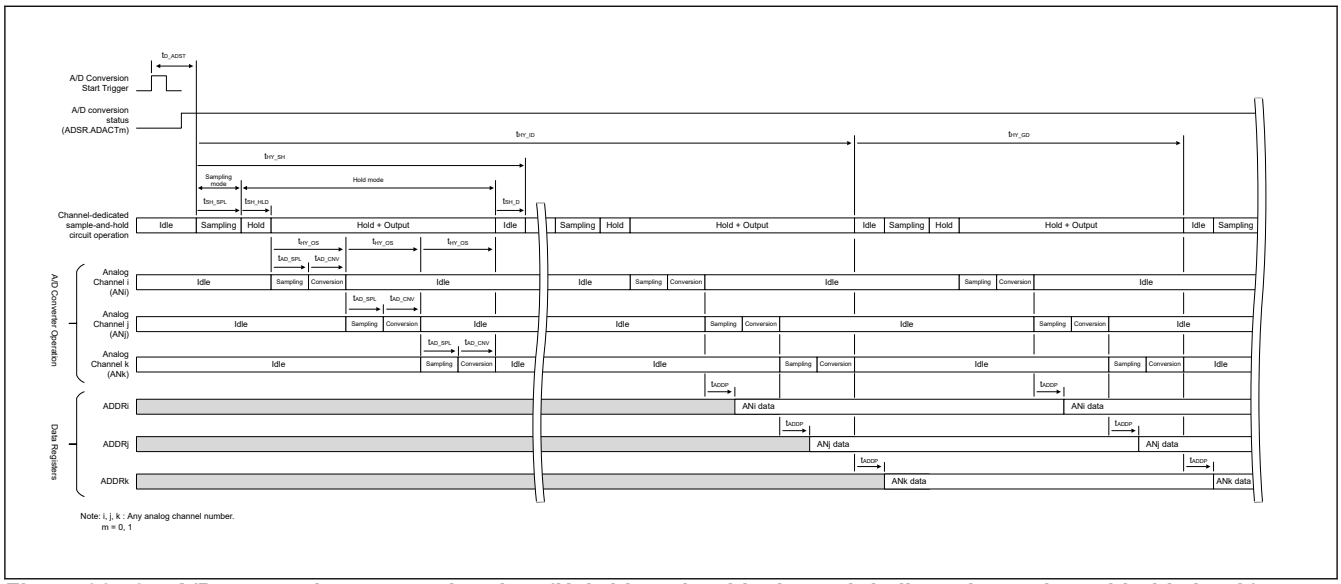


Figure 36.56 A/D conversion processing time (Hybrid mode with channel-dedicated sample-and-hold circuit)

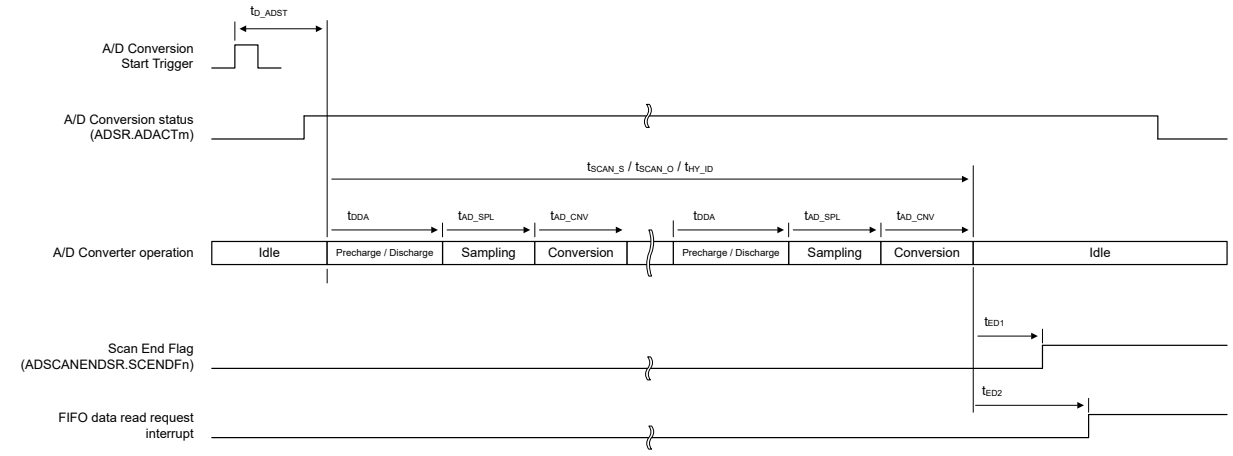
36.9.3 Scan End Processing Time

The scan end processing time and the forcibly stop processing time are shown in Table 36.57, Figure 36.57, Figure 36.58.

Table 36.57 Scan end processing time and forcibly stop processing time

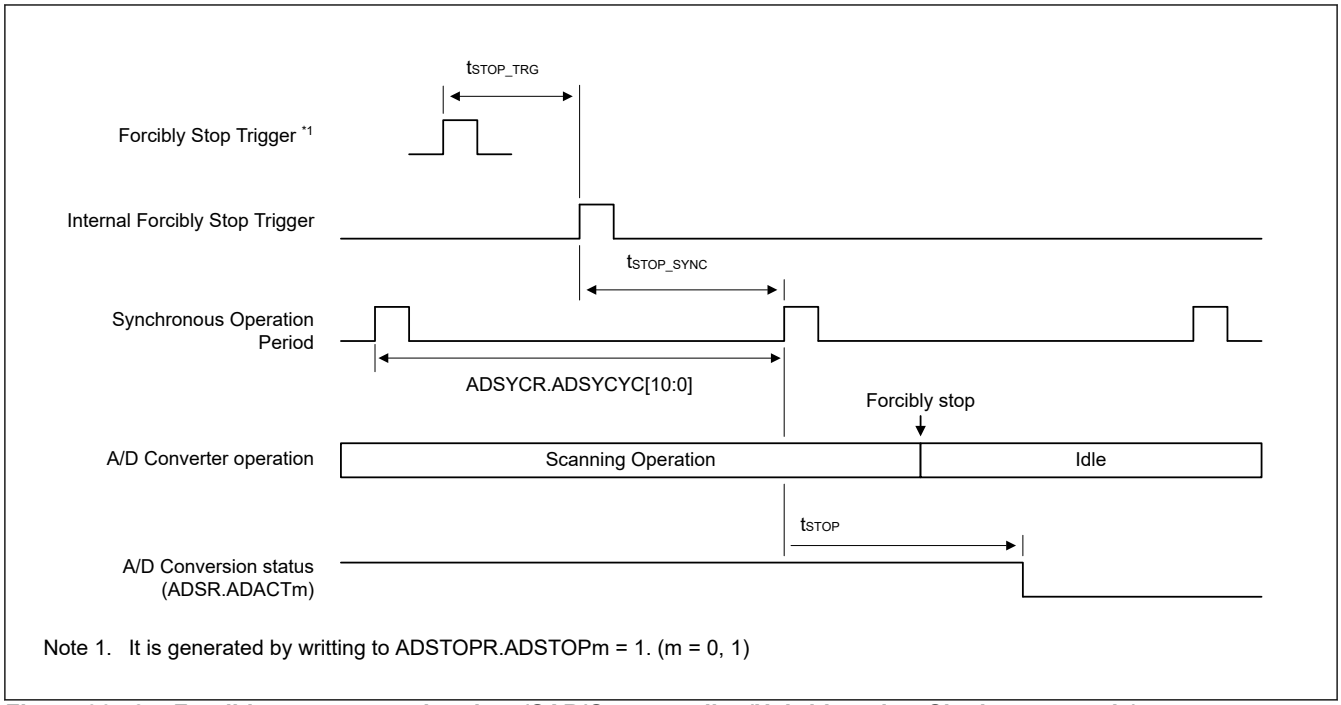
Item		Symbol	Processing Time
Scan end processing time	Until reflected in the status register*1	t_{ED1}	t_{ADDP}
	Until output the FIFO data read request interrupt	t_{ED2}	t_{ED1} + 2 PCLKA
Forcibly Stop processing time	Forcibly stop trigger input processing time	t_{STOP_TRG}	[When ADCLK = PCLKA/1 is set]*2 ● Number of access cycles to I/O registers*3 + 1 PCLKA [Other than above] ● (Number of access cycles to I/O registers*3) + 1 PCLKA + (3 to 4 ADCLK)
	Wait time for synchronous operation	t_{STOP_SYNC}	[When synchronous operation is disabled] (ADSYCR.ADSYDISm = 1) ● 0 [When synchronous operation is enabled] (ADSYCR.ADSYDISm = 0) ● 0 to ADSYCR.ADSYCYC[10:0] × ADCLK
	Forcibly stop processing time	t_{STOP}	[When ADCLK = PCLKA/1 is set]*2 (ADSYCR.ADSYDISm = 1) ● 4 PCLKA (ADSYCR.ADSYDISm = 0) ● 3 PCLKA [Other than above] (ADSYCR.ADSYDISm = 1) ● 4 PCLKA + (2 to 3 PCLKA) (ADSYCR.ADSYDISm = 0) ● 3 PCLKA + (2 to 3 PCLKA)

Note: m = 0, 1
Note 1. This is the time it takes for ADSCANENDSR.SCENDFn = 1 (n = 0 to 8) or until an end-of-scan interrupt is generated.
Note 2. When ADCLKCR.CLKSEL[1:0] = 10b and ADCLKCR.DIVR[2:0] = 000b are set.
Note 3. It is the number of access cycles to ADSTOPR register. For more details about the number of access cycles for I/O registers, see section Appendix 3, I/O Registers.



Note: n = 0 to 8, m = 0, 1

Figure 36.57 Scan end processing time (SAR/Oversampling/Hybrid mode – Single scan mode)



Note 1. It is generated by writting to ADSTOPR.ADSTOPm = 1. (m = 0, 1)

Figure 36.58 Forcibly stop processing time (SAR/Oversampling/Hybrid mode – Single scan mode)

36.10 Usage Notes

36.10.1 Prohibition of Changing the Operation Settings During A/D Conversion Operation

The registers related to the operation setting of the A/D converter should be set while all A/D converters are stopped (ADSR.ADACTm = 0 and ADSR.CALACTm = 0 (m = 0, 1)). Changing (writing) of registers except for those listed below is prohibited during A/D conversion. If the operation setting is changed during A/D conversion, operation is not guaranteed.

[Registers that can be written while the A/D converter is operating]

- Status clear registers
  - Status clear registers related to A/D converter operation (ADERSCR, ADCALSCR, ADCALENDSCR, ADSCANENDSCR)
  - A/D conversion overflow status clear registers (ADOVFERSCR, ADOVFCHSCR0, ADOVFEXSCR)
  - Limiter clip status clear register (ADLIMGRSCR, ADLIMCHSCR0, ADLIMEXSCR)
  - Compare match status clear register (ADCMPTBSCR, ADCMPCHSCR0, ADCMPEXSCR)
  - FIFO error status clear register (ADFIFOERSCR)
- Software trigger register (ADSYSTR, ADSTRn (n = 0 to 8))
- A/D converter stop register (ADSTOPR)
- A/D converter start trigger enable register (ADTRGENR) *1

Note 1. Writing during operation is only permitted for disabling the trigger input (ADTRGENR.STTRGENn = 0 (n = 0 to 8)) in order to stop the A/D conversion. To avoid unintended operation, do not change the trigger input to Enable (ADTRGENR.STTRGENn = 1) during operation.

36.10.2 Usage Notes on Forced Stop of A/D Conversion

If you want to forcibly stop the A/D conversion operation, follow the procedure in section 36.5.4. Force Stops the A/D Conversion Operation.

36.10.3 Usage Notes on A/D Data Registers

When A/D conversion is performed multiple times on the same analog channel, A/D data register ADDRn (n = 0 to 28) or A/D extended analog data register ADEXDRm (m = 0 to 2, 5 to 8) corresponding to that analog channel is overwritten at the later A/D conversion.

If you intend to keep the data for each A/D conversion for the same analog channel, perform one of the following methods.

- Read out data from registers at each scanning operation
  - At the end of scanning each scan group, read out the A/D conversion data of the analog channel that was subject to A/D conversion from ADDRn or ADEXDRm.
  - This method is effective if A/D conversion of the same analog channel is not performed more than once in the same scan group, and A/D conversion data can be read from the registers before the next scanning operation.
- Use FIFO function to hold data for each A/D conversion
  - By using FIFO, multiple A/D conversion data for the same analog channel can be kept.
  - Read out the A/D conversion data before FIFO overflow.

36.10.4 Settings for the Module-stop Function

The Module Stop Control Register can enable or disable ADC operation. The ADC is initially stopped after a reset. The registers become accessible on release from the module-stop state.

See section 36.10.5. Restrictions on Entering and Releasing the Low-Power States in connection with this usage note.

36.10.5 Restrictions on Entering and Releasing the Low-Power States

Before entering the module-stop state or Software Standby mode, be sure to stop A/D conversion. In Addition, configure the A/D conversion not to start during the entering to the low power state.

To operate the A/D converter after the module-stop state or software standby mode is released, wait for the operation stabilization time specified in the Electrical Characteristics, execute self-calibration operation, and then start A/D conversion.

Also, when releasing the module-stop state again after entering to the module-stop state, make sure the shutdown time specified in the Electrical Characteristics has elapsed before releasing the module stop state.

Operations are not guaranteed if these restriction are violated.

36.10.6 Notes on Board Design

(1) Protection Circuit

To prevent the analog input pins from being destroyed by abnormal voltage such as excessive surge, take the following measures.

- Insert a capacitor between AVCC0 and AVSS0 and between VREFH0 and VREFL0.
- Connect a protection circuit to protect the analog input pins.

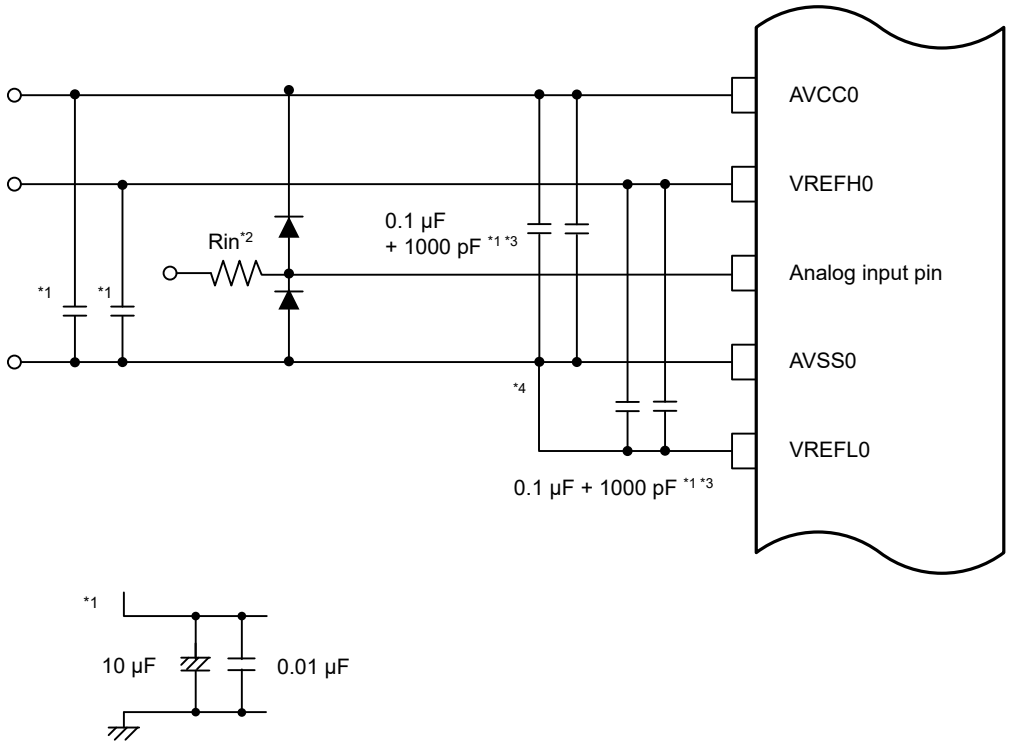
An example of a protection circuit is shown in Figure 36.59.

(2) Board design to ensure A/D conversion accuracy

In order to ensure the accuracy of A/D conversion, design the board considering the following:

- Analog circuits and digital circuits should be separated from each other as far as possible.
- Analog circuit signal lines and digital circuit signal lines should not intersect or be placed near each other.
- The analog input, analog reference power supply (VREFH0), analog reference ground (VREFL0), and analog power supply (AVCC0) should be separated from digital circuits using the analog ground (AVSS0).

- The analog ground (AVSS0) should be connected to a stable digital ground (VSS) on the board (single-point ground plane connection).
- Place a capacitor for noise filters between AVCC0 pin and AVSS0 pin and between VREFH0 pin and VREFL0 pin close to the pin and connect them. Also, connect AVSS0 and VREFL0 pins to the analog ground on the board as close to the pins as possible. An example of connection is shown in Figure 36.59.



Note 1. The values shown here are for reference only.
The appropriate values depend on your board design, operating conditions, etc.
Note 2. Rin: Signal source impedance
Note 3. To ensure the better A/D conversion accuracy, these capacitors must be place as close as possible to the appropriate pins.
Note 4. Connect to the analog ground of the board at the position as close as possible to the capacitor for noise filtering that ensure the A/D conversion accuracy.

Figure 36.59 Example of protection circuit and noise prevention

36.10.7 Notes on Using Analog Channels to Which the PGA is Connected

The analog input pin, which is connected to PGA, is limited by the port function and the analog path depending on the PGA setting. The initial status of PGA setting after reset released depends on the setting on PGADEN[3:0] bits in Option Function Select Register 1 (OFS1, OFS1_SEC, OFS1_SEL).

For the setting when using analog input pin which is connected to PGA, see section 36.3.15.2. PGA Operation Setting.

36.10.8 Notes on Synchronous Operation

Synchronous operation function is enabled in the initial state after reset released. If using synchronous operation function, follow the restrictions described in section 36.3.19. Synchronous Operation. If synchronous operation function is not used, disable synchronous operation in ADSYCR register.

36.10.9 Notes on Channel-dedicated Sample-and-hold Circuit

When using channel-dedicated sample-and-hold circuit, follow the restrictions. For details, see section 36.3.16. Channel-dedicated Sample-and-hold Circuit.

36.10.10 Restrictions on Analog Channel Shared Among Multiple A/D Converters

To avoid degradation of the accuracy of the A/D conversion result, A/D conversion of the same analog channel (same analog signal source) from both ADC0 and ADC1 is prohibited, except for the self-diagnosis channel.

If this restriction is violated, the A/D conversion results are not guaranteed because the A/D conversion accuracy of the target analog channel may deteriorate significantly.

36.10.11 Notes on A/D Conversion Start Trigger

A/D conversion start triggers for the same scan group are not accepted until scanning operation of the scan group is completed (in this case, A/D converter start triggers are ignored).

An A/D conversion trigger for the same scan group should be input after the scanning operation of its scan group is completed (ADSCANENDSR.SCENDFn = 1 (n = 0 to 8)) and after a 6 PCLKA clock cycles or more has elapsed.

36.10.12 Notes on Self-calibration

Observe the restrictions on self-calibration function. For details, see section 36.3.8. Self-calibration.

36.10.13 Notes on Group Priority Operation

Observe the restrictions when using group priority operation. For details, see section 36.3.18. Group Priority Operation.

36.10.14 Notes on PGA Output Monitor Function

There are restrictions when using the PGA output monitor function. For details, see section 36.3.15. Programmable Gain Amplifier.

36.10.15 Restrictions on SAR Mode

(1) Notes on resolution and accuracy

In SAR mode, the A/D converter functions as a 12-bit A/D converter. Although the data output from the A/D converter is internally extended to 16-bit data length for calculation processing, even if 14-bit or 16-bit length data format is selected, the resolution and accuracy exceeding 12 bits are not guaranteed for that A/D conversion data. Select 14-bit or 16-bit length data format in SAR mode if you want to increase the computational resolution, such as when using A/D-converted value addition/averaging function or User's Gain/User's Offset function.

(2) Restrictions on differential input

In SAR mode, differential input setting is prohibited except during self-diagnosis operation. If the differential input is set except for self-diagnosis, the A/D conversion result is not guaranteed.

(3) Restrictions on digital filter function

The digital filter is prohibited in SAR mode. Operation is not guaranteed when the digital filter function is enabled.

36.10.16 Restrictions on Oversampling Mode

In Oversampling mode, the use of the digital filter function is mandatory. If A/D conversion is performed without using the digital filter function, the A/D conversion characteristics are not guaranteed.

36.10.17 Restrictions on Hybrid Mode

In Hybrid mode, observe the following restrictions.

(1) Prohibition of scanning operation with only one channel

In Hybrid mode, be sure to assign two or more virtual channels to one scan group (up to 4 virtual channels). Scanning operation with only one virtual channel assigned to one scan group is prohibited. If this restriction is violated, operation is not guaranteed.

(2) Restrictions on virtual channel configuration

In Hybrid mode, do not assign the same analog channel to multiple virtual channels that are assigned to the same scan group. If this restriction is violated, A/D conversion result is not guaranteed.

(3) Restrictions on digital filter configuration

In Hybrid mode, the use of the digital filter function is mandatory. If A/D conversion in Hybrid mode is performed without using the digital filter function, the A/D conversion characteristics are not guaranteed.

In addition, since multiple digital filters are used simultaneously in Hybrid mode, the digital filters used for virtual channels in the same scan group must be set to be exclusive. The digital filters selected by the ADDOPCRAn.DFSEL[2:0] (n = 0 to 36) bits must be set to be exclusive among virtual channels in the same scan group. If the same digital filter is selected from multiple virtual channels in the same scan group, operation is not guaranteed.

(4) Restrictions on A/D-converted value addition/averaging function

When using A/D-converted value addition/averaging function in Hybrid mode, set the number of addition/averaging times for all virtual channels in the same scan group to be the same. If this restriction is violated, operation is not guaranteed.

(5) Restrictions on trigger interval in Background continuous scan mode

In Hybrid mode – Background continuous scan mode, A/D conversion start trigger should be input after the following interval or more.

[Required Interval for A/D Conversion Start Trigger Input]

- When trigger delay function is not used, or software trigger: 8 ADCLK, or more
- When trigger delay function is used: (8 + (the setting value of ADTRGDLRm.TRGDLYn[7:0])) × ADCLK, or more

Note: m = 0 to 4, n = 0 to 8

If this restriction is violated, the A/D conversion start trigger is not accepted and is ignored.

(6) Restrictions on channel-dedicated sample-and-hold circuit

In Hybrid mode, the way to use of the channel-dedicated sample-and-hold circuit is limited. For details, see section 36.3.16.3. Operation in Hybrid Mode with Channel-dedicated Sample-and-hold Circuit.

36.10.18 Sampling Time Estimation

This subsection describes how to estimate the sampling time.

The sampling time required in A/D conversion for externally input analog signal is determined by the charging time to the sampling capacitance in A/D converter. The simplified model of the circuit is shown in Figure 36.60. The sampling time can be roughly estimated by the following equation.

[Estimation Equation for Sampling Time]

$$t_{SPL} = \left(R_{EXT} + R_{AD}\right) \times \left(C_{EXT} + kC_{AD}\right) \times \ln\left(kC_{AD} / \left(C_{EXT} + kC_{AD}\right) \times \left(2^N / M\right)\right)$$

t_{SPL}: Estimated sampling time

C_{EXT}: External capacitance (pin capacitance + PCB parasitic capacitance)

C_{AD}: Internal sampling capacitance

R_{EXT}: External input signal source Impedance

R_{AD}: Internal resistance

k: Correction coefficients according to operation mode

N: Target conversion resolution (16, 14, 12 or 10)

M: Sampling error based on 1 LSB in N-bits A/D converter (1/4, 1/2, 1, 2, or 4 LSB etc.)

Note: The typical values for each parameter are as follows.

These are reference values:

- Pin capacitance at analog input pin: 5 pF

- Internal sampling capacitance (C_{AD}): 5 pF
- R_{AD} at High-speed channels: 0.7 kΩ
- R_{AD} at High-precision channels: 1.2 kΩ
- R_{AD} at Normal-precision channels: 3.0 kΩ
- k at SAR mode: 1.2
- k at Oversampling mode: 1.0
- k at Hybrid mode: 1.2

In above equation, you can estimate the time, which is taken for the difference between the analog input voltage (V_{IN}) and the sampling capacitance voltage (V_{AD}) to be less than or equal to sampling error based on N-bits A/D converter, as the sampling time.

For example, if R_{EXT} is 1 kΩ, C_{EXT} is 10 pF, N is 12-bits and M is 1/4 LSB, the sampling time (t_{SPL}) for High-speed channel in SAR mode can be estimated as 237 ns.

[Usage Notes]

The equation described in this subsection is simplified in view of general use case. It does not guarantee the accurate sampling time. You should only use it to estimate the roughly sampling time. Especially for Normal-precision channels, the accuracy of the estimate for sampling time deteriorates if ($2^N/M > 16384$).

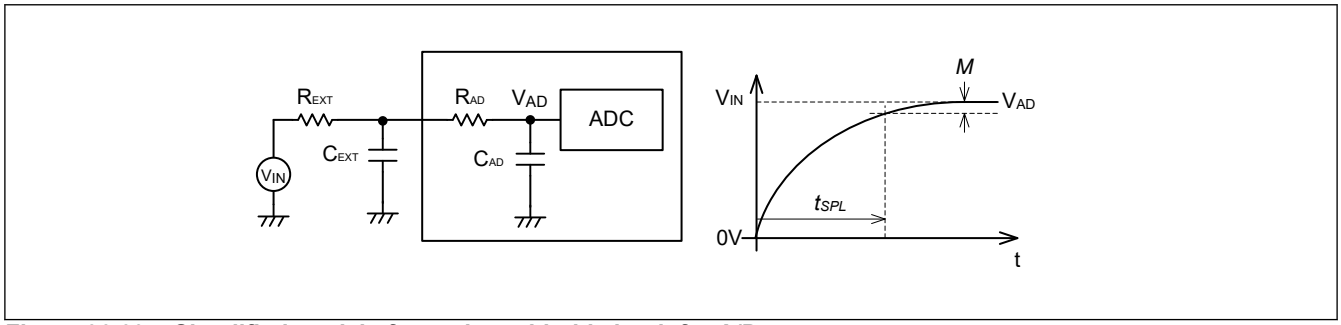


Figure 36.60 Simplified model of sample-and-hold circuit for A/D converter

37. 12-Bit D/A Converter (DAC12)

37.1 Overview

The MCU provides a 12-bit D/A Converter (DAC12) with an output amplifier. Table 37.1 lists the DAC12 specifications, Figure 37.1 shows a block diagram, and Table 37.2 lists the I/O pins.

Table 37.1 DAC12 specifications

Parameter	Specifications
Resolution	12 bits
Output channels	4 channels
Module-stop function	Module-stop state can be set to reduce power consumption
Event link function (input)	The DA0, DA1, DA2 and DA3 conversion can be started on input of an event signal
D/A output amplifier control function	Controls whether the output amplifier (for both amplifier-through and amplifier-bias controls) is used
Destination of D/A output control function	Controls whether the output to the external pin or to the internal modules (ACMPHS) is used
TrustZone Filter	Security attribution can be set

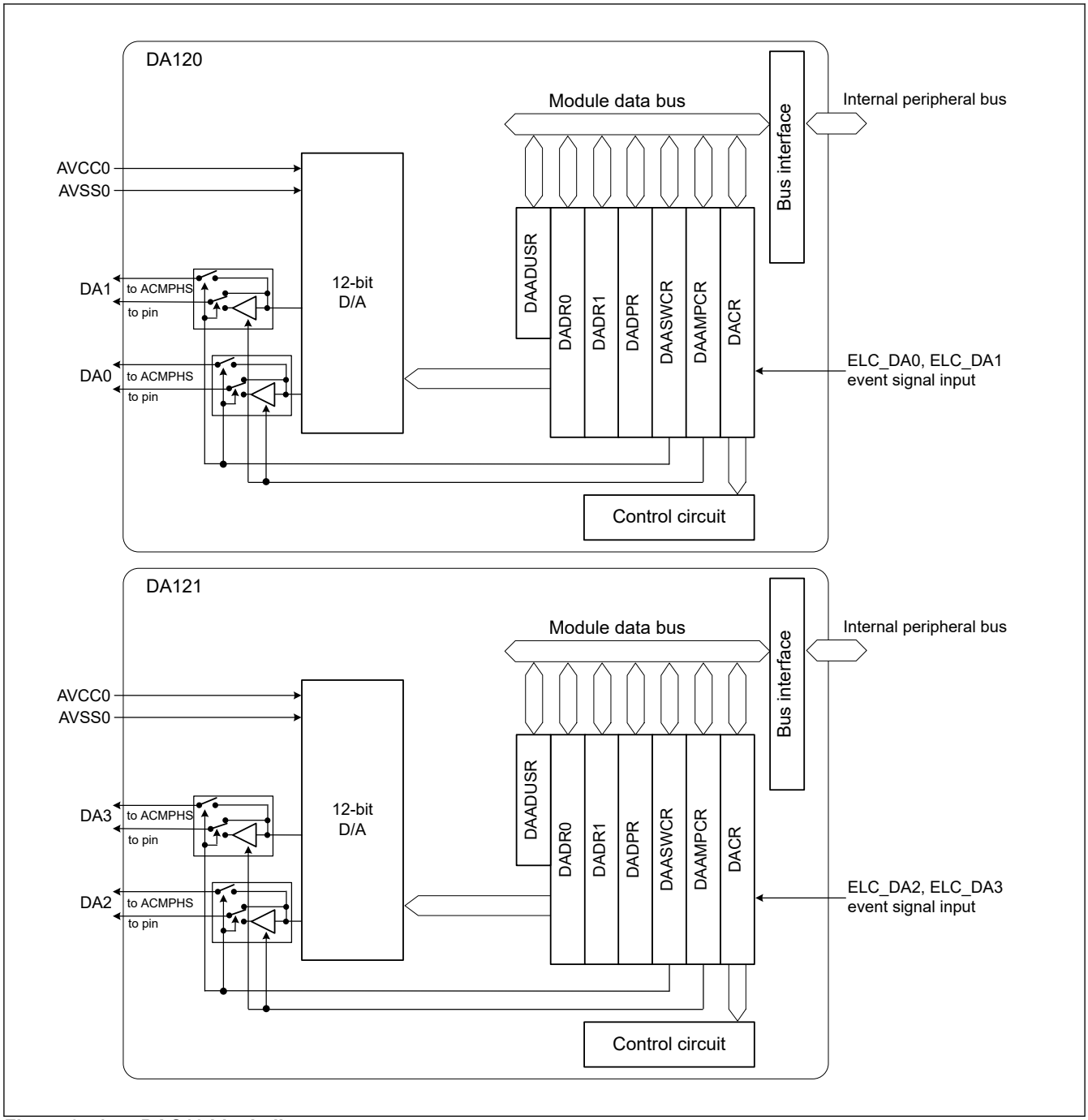


Figure 37.1 DAC12 block diagram

Table 37.2 lists the pin configuration of the DAC12.

Table 37.2 DAC12 I/O pins (1 of 2)

Pin name	I/O	Function
AVCC0	Input	• Analog power and analog reference top voltage supply pin for ADC and DAC12. • Connect to VCC when these modules are not used.
AVSS0	Input	• Analog ground and analog reference ground supply pin for ADC and DAC12. • Connect to VSS when these modules are not used.
DA0	Output	Channel 0 output pin for the analog signals processed by the DAC12
DA1	Output	Channel 1 output pin for the analog signals processed by the DAC12
DA2	Output	Channel 2 output pin for the analog signals processed by the DAC12

Table 37.2 DAC12 I/O pins (2 of 2)

Pin name	I/O	Function
DA3	Output	Channel 3 output pin for the analog signals processed by the DAC12

37.2 Register Descriptions

37.2.1 DADRn : D/A Data Register n (n = 0, 1)

Base address: DAC12m = 0x4017_2000 + 0x0100 × m (m = 0, 1)

Offset address: 0x00 + 0x02 × n

Bit position:	15															0
Bit field:																
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DADRn register is 16-bit read/write registers that store data for D/A conversion. When an analog output is enabled, the values in DADRn are converted and output to the analog output pins.

12-bit data can be formatted as left- or right-justified in the DADPR.DPSEL bit setting. In right-justified format (DADPR.DPSEL = 0), the lower 12 bits, [11:0], are valid. In left-justified format (DADPR.DPSEL = 1), the upper 12 bits, [15:4], are valid.

37.2.2 DACR : D/A Control Register

Base address: DAC12m = 0x4017_2000 + 0x0100 × m (m = 0, 1)

Offset address: 0x04

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DAOE1	DAOE0	DAE	—	—	—	—	—
Value after reset:	0	0	0	1	1	1	1	1

Bit	Symbol	Function	R/W
4:0	—	These bits are read as 1. The write value should be 1.	R/W
5	DAE*1	D/A Enable 0: Control D/A conversion of channels 0 and 1 individually Control D/A conversion of channels 2 and 3 individually 1: Control D/A conversion of channels 0 and 1 collectively Control D/A conversion of channels 2 and 3 collectively	R/W
6	DAOE0	D/A Output Enable 0 0: Disable analog output of channel 0 (DA0) Disable analog output of channel 2 (DA2) 1: Enable D/A conversion of channel 0 (DA0) Enable D/A conversion of channel 2 (DA2)	R/W
7	DAOE1	D/A Output Enable 1 0: Disable analog output of channel 1 (DA1) Disable analog output of channel 3 (DA3) 1: Enable D/A conversion of channel 1 (DA1) Enable D/A conversion of channel 3 (DA3)	R/W

Note 1. This bit controls D/A conversion in combination with the DAOEi bit (i = 0, 1), which controls the output of the conversion results. For details, see Table 37.3.

Table 37.3 D/A conversion controls (m = 0)

DAE	DAOE1	DAOE0	Description
0	0	0	Disable D/A conversion and analog output pins (DA0, DA1)*1
		1	• Enable D/A conversion of channel 0 and disable D/A conversion of channel 1 • Enable analog output of channel 0 (DA0) and disable analog output of channel 1 (DA1)*1
	1	0	• Disable D/A conversion of channel 0 and enable D/A conversion of channel 1 • Disable analog output of channel 0 (DA0)*1 and enable analog output of channel 1 (DA1)
		1	• Enable D/A conversion of channels 0 and 1 • Enable analog output of channels 0 and 1 (DA0, DA1)
1	x	x	• Enable D/A conversion of channels 0 and 1 • Collective enable analog output of channels 0 and 1 (DA0, DA1)

Note: x: Don't care
Note 1. When analog output is disabled, the analog output signal is placed in the Hi-Z state.

Table 37.4 D/A conversion controls (m = 1)

DAE	DAOE1	DAOE0	Description
0	0	0	Disable D/A conversion and analog output pins (DA2, DA3)*1
		1	• Enable D/A conversion of channel 2 and disable D/A conversion of channel 3 • Enable analog output of channel 2 (DA2) and disable analog output of channel 3 (DA3)*1
	1	0	• Disable D/A conversion of channel 2 and enable D/A conversion of channel 3 • Disable analog output of channel 2 (DA2)*1 and enable analog output of channel 3 (DA3)
		1	• Enable D/A conversion of channels 2 and 3 • Enable analog output of channels 2 and 3 (DA2, DA3)
1	x	x	• Enable D/A conversion of channels 2 and 3 • Collective enable analog output of channels 2 and 3 (DA2, DA3)

Note: x: Don't care
Note 1. When analog output is disabled, the analog output signal is placed in the Hi-Z state.

DAE bit (D/A Enable)

The DAE bit controls D/A conversion, amplifier operation, and analog output in combination with the DAOEi bit (i = 0, 1) and the DAAMPCR.DAAMPi bit (i = 0, 1). See Table 37.5.

DAOEi bit (D/A Output Enable i)

The DAOEi bit (i = 0, 1) controls D/A conversion, amplifier operation, and analog output in combination with the DAE bit and DAAMPCR.DAAMPi bit (i = 0, 1). See Table 37.5.

When both the DAOEi bit (i = 0, 1) and DAE bit are 0, D/A conversion of channel i (i = 0, 1) is not processed, and no conversion result is output.

The event link function can be used to set the DAOEi bit to 1. The DAOE0 bit is set to 1 when the event specified in the ELSR12 register of the ELC (ELC_DA0 event) occurs, and output of the D/A conversion results starts. The DAOE1 bit is set to 1 when the event specified in the ELSR13 register of the ELC (ELC_DA1 event) occurs, and output of the D/A conversion results starts.

D/A conversions for channel 2 and 3 are controled by same bits in the register for the address when m = 1.

The setting of the ELC functions for channel 2 and 3 is possible in same manner too.

The event specified in the ELSR28 register of the ELC triggers the set of DAOE0 bit for address of m = 1 and the output of the D/A conversion for channel 2. The event specified in the ELSR29 triggers the set of DAOE1 bit and the output of the D/A conversion for channel 3.

Table 37.5 D/A conversion and analog output control

DACR		DAAMPCR	DAASWCR	Channel i operation	Amplifier operation of channel i	Analog external output of channel i*1	Analog internal output of channel i*2
DAE	DAOEi	DAAMPi	DAASWi				
0	0	x	x	Stop	Stop	Hi-Z	Hi-Z
0	1	0	0	Run	Stop	Amplifier-through	Hi-Z
0	1	0	1	Run	Stop	Hi-Z	Amplifier-through
0	1	1	0	Run	Run	Amplifier output	Hi-Z
0	1	1	1	Run	Run	Hi-Z	Hi-Z
1	x	0	0	Run	Stop	Amplifier-through	Hi-Z
1	x	0	1	Run	Stop	Hi-Z	Amplifier-through
1	x	1	0	Run	Run	Amplifier output	Hi-Z
1	x	1	1	Run	Run	Hi-Z	Hi-Z

Note: x : Don't care
Note 1. output to pin
Note 2. output to ACMPHS

37.2.3 DADPR : DADRn Format Select Register

Base address: DAC12m = 0x4017_2000 + 0x0100 × m (m = 0, 1)

Offset address: 0x05

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

Bit	Symbol	Function	R/W
6:0	—	These bits are read as 0. The write value should be 0.	R/W
7	DPSEL	DADRn Format Select 0: Right-justified format 1: Left-justified format	R/W

37.2.4 DAAMPCR : D/A Output Amplifier Control Register

Base address: DAC12m = 0x4017_2000 + 0x0100 × m (m = 0, 1)

Offset address: 0x08

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DAAMP1	DAAMP0	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	—	These bits are read as 0. The write value should be 0.	R/W
6	DAAMP0	Amplifier Control 0 0: Do not use channel 0 output amplifier (m = 0) Do not use channel 2 output amplifier (m = 1) 1: Use channel 0 output amplifier (m = 0) Use channel 2 output amplifier (m = 1)	R/W

Bit	Symbol	Function	R/W
7	DAAMP1	Amplifier Control 1 0: Do not use channel 1 output amplifier (m = 0) Do not use channel 3 output amplifier (m = 1) 1: Use channel 1 output amplifier (m = 0) Use channel 3 output amplifier (m = 1)	R/W

The DAAMPCR register selects D/A output with or without using the amplifier.

DAAMP0 bit (Amplifier Control 0)

When the DAAMP0 bit is 0, analog values are output for D/A output of channel 0 (m = 0) and channel 2 (m = 1) without using the amplifier. When the DAAMP0 bit is 1, analog values are output for D/A output of channel 0 (m = 0) and channel 2 (m = 1) through the amplifier.

When both the DACR.DAE and DACR.DAOE0 bits are 0, the amplifier is not used regardless of the setting of the DAAMP0 bit. See Table 37.5 for details.

DAAMP1 bit (Amplifier Control 1)

When the DAAMP1 bit is 0, analog values are output for D/A output of channel 1 (m = 0) and channel 3 (m = 1) without using the amplifier. When the DAAMP1 bit is 1, analog values are output for D/A output of channel 1 (m = 0) and channel 3 (m = 1) through the amplifier.

When both the DACR.DAE and DACR.DAOE1 bits are 0, the amplifier is not used regardless of the setting of the DAAMP1 bit. See Table 37.5 for details.

37.2.5 DAASWCR : D/A Amplifier Stabilization Wait Control Register

Base address: DAC12m = 0x4017_2000 + 0x0100 × m (m = 0, 1)

Offset address: 0x1C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DAASW1	DAASW0	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	—	These bits are read as 0. The write value should be 0.	R/W
6	DAASW0	D/A Amplifier Stabilization Wait 0 and D/A internal output control 0: For output to external pin: Amplifier stabilization wait off (output) for channel 0 For output to internal module: Disable output for channel 0 1: For output to external pin: Amplifier stabilization wait on (high-Z) for channel 0 For output to internal module: Enable output for channel 0	R/W
7	DAASW1	D/A Amplifier Stabilization Wait 1 and D/A internal output control 0: For output to external pin: Amplifier stabilization wait off (output) for channel 1 For output to internal module: Disable output for channel 1 1: For output to external pin: Amplifier stabilization wait on (high-Z) for channel 1 For output to internal module: Enable output for channel 1	R/W

The DAASWCR register controls D/A output with the output amplifier or D/A output for internal modules. This register is used in the initialization procedure to wait for stabilization of the D/A output amplifier. Each bit in DAASWCR should be set to 1 when both the DACR.DAE bit and the DACR.DAOEi (i = 0, 1) bit are 0 . See section 37.6.5. Initialization Procedure with the Output Amplifier.

DAASW0 bit (D/A Amplifier Stabilization Wait 0)

Set the DAASW0 bit to 1 in the initialization procedure to wait for the stabilization of the D/A channel 0 and 2 output amplifier. When DAASW0 is set to 1, D/A conversion operates, but the conversion result of D/A is not output from channel 0 to the DA0 pin (m = 0) and channel 2 to the DA2 pin (m = 1). When the DAASW0 bit is 0, the stabilization wait time stops, and the D/A conversion result of channel 0 is the output amplifier to the DA0 pin. When the amplifier is not used (DAAMPCR.DAAMP0 bit is 0) and DAASW0 is set to 1, D/A conversion result of channel 0 is output to the internal modules.

DAASW1 bit (D/A Amplifier Stabilization Wait 1)

Set the DAASW1 bit to 1 in the initialization procedure to wait for the stabilization of the D/A channel 1 and 3 output amplifier. When DAASW1 is set to 1, D/A conversion operates, but the conversion result of D/A is not output from channel 1 to the DA1 pin (m = 0) and channel 3 to the DA3 pin (m = 1). When the DAASW1 bit is 0, the stabilization wait time stops, and the D/A conversion result of channel 1 is the output amplifier to the DA1 pin. When the amplifier is not used (DAAMPCR.DAAMP1 bit is 0) and DAASW1 is set to 1, D/A conversion result of channel 1 is output to the internal modules.

37.3 Operation

The DAC12 includes D/A conversion circuits for four channels, each of which can operate independently. When the DAOEn bit (n = 0, 1) in DACR is set to 1, DAC12 is enabled and the conversion result is output.

This following example shows D/A conversion on channel 0. Figure 37.2 shows the timing of this operation.

To process D/A conversion on channel 0:

1. Set the data for D/A conversion in the DADR0 register and the data format in the DADPR.DPSEL bit.
2. Set the DACR.DAOE0 bit to 1 to start D/A conversion. The conversion result is output from the analog output pin DA0 after the conversion time t_{DCONV} elapses. The conversion result continues to be output until DADR0 is written to again or the DAOE0 bit is set to 0. The output value (reference) is expressed by the following formula:

$$\frac{\text{Setting in DADR0}}{4096} \times \text{AVCC0}$$

3. To start conversion again, write another value to DADR0. The conversion result is output after the conversion time t_{DCONV} elapses.
4. To disable analog output, set the DAOE0 bit to 0.

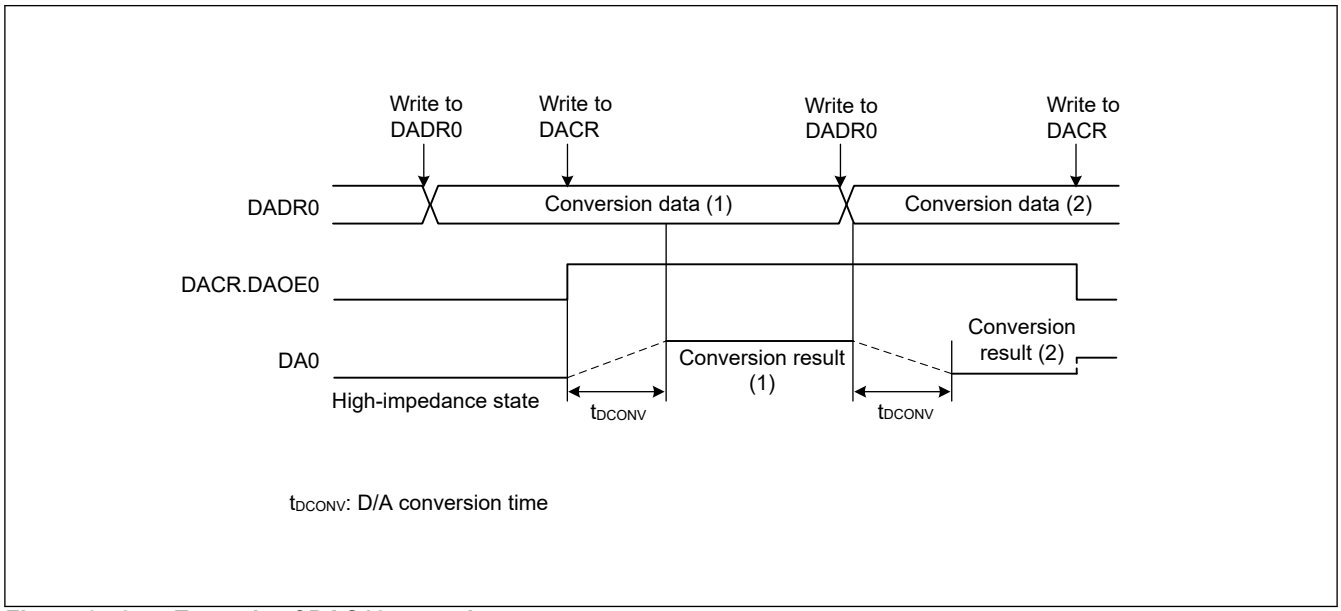


Figure 37.2 Example of DAC12 operation

37.4 Event Link Operation Setting Procedure

This section describes the procedures used in event link operation.

37.4.1 DA0 Event Link Operation Setting Procedure

To set up DA0 event link operation:

1. Set the DADPR.DPSEL bit and the data for D/A conversion in the DADR0 register.
2. Set the ELC_DA0 event signal to be linked to each peripheral module in the ELSR12 register.
3. Set the ELCR.ELCON bit to 1. This enables event link operation for all modules with the event link function selected.
4. Set the event output source module to activate the event link. After the event is output from the module, the DACR.DAOE0 bit becomes 1, and D/A conversion starts on channel 0.
5. Set the ELSR12 register to 0x0000 to stop event link operation of DAC12 channel 0. All event link operation is stopped when the ELCR.ELCON bit is set to 0.

37.4.2 DA1 Event Link Operation Setting Procedure

To set up DA1 event link operation:

1. Set the DADPR.DPSEL bit and set the data for D/A conversion in the DADR1 register.
2. Set the ELC_DA1 event signal to be linked to each peripheral module in the ELSR13 register.
3. Set the ELCR.ELCON bit to 1. This enables the event link operation for all modules with the event link function selected.
4. Set the event output source module to activate the event link. After the event is output from the module, the DACR.DAOE1 bit is becomes 1, and D/A conversion starts on channel 1.
5. Set the ELSR13 register to 0x0000 to stop event link operation on DAC12 channel 1. All event link operation is stopped when the ELCR.ELCON bit is set to 0.

37.4.3 DA2 Event Link Operation Setting Procedure

To set up DA2 event link operation:

1. Set the DADPR.DPSEL bit and the data for D/A conversion in the DADR0 register.
2. Set the ELC_DA2 event signal to be linked to each peripheral module in the ELSR28 register.
3. Set the ELCR.ELCON bit to 1.
   This enables event link operation for all modules with the event link function selected.
4. Set the event output source module to activate the event link.
   After the event is output from the module, the DACR.DAOE0 bit becomes 1, and D/A conversion starts on channel 2.
5. Set the ELSR28 register to 0x0000 to stop event link operation of DAC12 channel 2.
   All event link operation is stopped when the ELCR.ELCON bit is set to 0.

37.4.4 DA3 Event Link Operation Setting Procedure

To set up DA3 event link operation:

1. Set the DADPR.DPSEL bit and set the data for D/A conversion in the DADR1 register.
2. Set the ELC_DA3 event signal to be linked to each peripheral module in the ELSR29 register.
3. Set the ELCR.ELCON bit to 1.
   This enables the event link operation for all modules with the event link function selected.
4. Set the event output source module to activate the event link.
   After the event is output from the module, the DACR.DAOE1 bit becomes 1, and D/A conversion starts on channel 3.
5. Set the ELSR29 register to 0x0000 to stop event link operation on DAC12 channel 3.
   All event link operation is stopped when the ELCR.ELCON bit is set to 0.

37.5 Usage Notes on Event Link Operation

- When the event link function is used, do not use the amplifier output function.
- When the event link function is used, set the DACR.DAE bit to 0.
- When the event specified for the ELC_DA0 event signal is generated while a write to the DACR.DAOE0 bit is performed, the write cycle is stopped, and the generated event takes precedence in setting the bit to 1.
- When the event specified for the ELC_DA1 event signal is generated while a write to the DACR.DAOE1 bit is performed, the write cycle is stopped, and the generated event takes precedence in setting the bit to 1.
- When the event specified for the ELC_DA2 event signal is generated while a write to the DACR.DAOE0 (m = 1) bit is performed, the write cycle is stopped, and the generated event takes precedence in setting the bit to 1.
- When the event specified for the ELC_DA3 event signal is generated while a write to the DACR.DAOE1 (m = 1) bit is performed, the write cycle is stopped, and the generated event takes precedence in setting the bit to 1.

37.6 Usage Notes

37.6.1 Settings for the Module-Stop Function

DAC12 operation can be disabled or enabled using the Module Stop Control Register. The DAC12 is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 10, Low Power Modes.

37.6.2 DAC12 Operation in the Module-Stop State

When the MCU enters the module-stop state with D/A conversion enabled, the D/A outputs are retained, and the analog power supply current is the same as during D/A conversion. If the analog power supply current must be reduced in the module-stop state, disable D/A conversion by setting the DACR.DAOE1, DAOE0, and DAE bits to 0.

37.6.3 DAC12 Operation in Software Standby Mode

When the MCU enters Software Standby mode with D/A conversion enabled, the D/A outputs are retained, and the analog power supply current is the same as during D/A conversion. If the analog power supply current must be reduced in Software Standby mode, disable D/A conversion by setting the DACR.DAOE1, DAOE0, and DAE bits to 0.

37.6.4 Constraint on Entering Deep Software Standby Mode

When the MCU enters Deep Software Standby mode with D/A conversion enabled, the outputs of the DAC12 are placed in a high impedance state.

37.6.5 Initialization Procedure with the Output Amplifier

Use the following initialization procedures with the output amplifier. The example shows the case for channel 0.

To initialize the DAC12 with the output amplifier:

1. Write 0x0000 to the DADR0 register.
2. Set the DAASWCR.DAASW0 bit to 1.
3. Set the DAAMPCR.DAAMP0 bit to 1.
4. Set the DACR.DAE bit or the DACR.DAOE0 bit to 1 to start operation of the amplifier.
5. Clear the DAASWCR.DAASW0 bit to 0 after waiting for the duration of D/A conversion time t_{DCONV}.
6. Write the value to be converted in the DADR0 register.

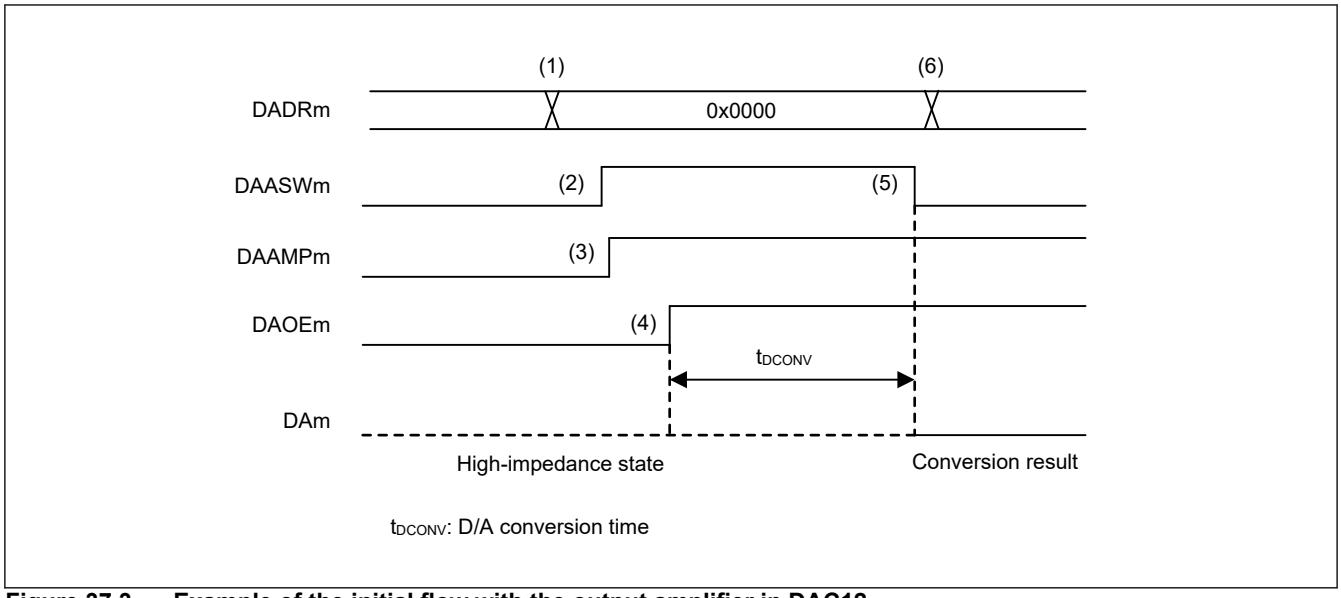


Figure 37.3 Example of the initial flow with the output amplifier in DAC12

While the amplifier is running, clearing the DACR.DAE and DACR.DAOE0 bits to 0 allows the amplifier to stop operation. To use the amplifier again, repeat steps 1 to 6.

37.6.6 Initialization Procedure of the Output to internal modules

Use the following initialization procedures for the output to internal modules.

The example shows the case for channel 0.

1. Set the DAASWCR.DAASW0 bit to 1.
2. Set the DACR.DAE bit or the DACR.DAOE0 bit to 1.
3. Write the value to be converted in the DADR0 register.

38. Temperature Sensor (TSN)

38.1 Overview

The on-chip Temperature Sensor (TSN) determines and monitors the die temperature for reliable operation of the device. The sensor outputs a voltage directly proportional to the die temperature, and the relationship between the die temperature and the output voltage is fairly linear. The output voltage is provided to the ADC for conversion and can be further used by the end application.

Table 38.1 lists the TSN specifications, and Figure 38.1 shows a block diagram.

Table 38.1 TSN specifications

Item	Description
Temperature sensor voltage output	Temperature sensor outputs a voltage to the A/D converter
Module-stop function	Module-stop state can be set to reduce power consumption
Temperature sensor calibration data	Reference data measured for each chip at factory shipment is stored in a register
TrustZone Filter	Security attribution can be set

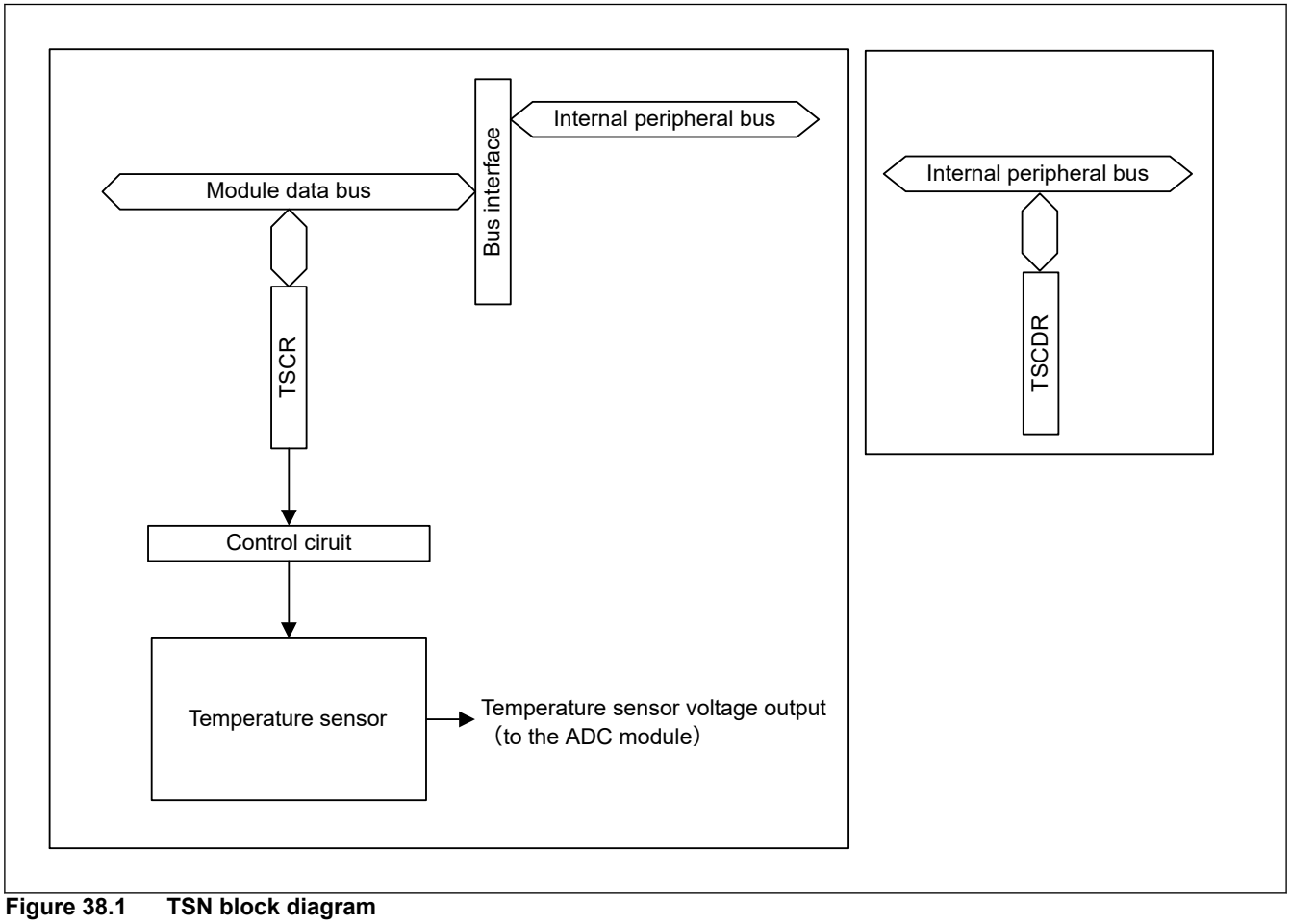


Figure 38.1 TSN block diagram

38.2 Register Descriptions

38.2.1 TSCR : Temperature Sensor Control Register

Base address: TSN = 0x400F_3000

Offset address: 0x00

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

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
4	TSOE	Temperature Sensor Output Enable 0: Disable output from the temperature sensor to the ADC 1: Enable output from the temperature sensor to the ADC	R/W
6:5	—	These bits are read as 0. The write value should be 0.	R/W
7	TSEN	Temperature Sensor Enable 0: Stop the temperature sensor 1: Start the temperature sensor.	R/W

The TSCR is a register which controls the temperature sensor. The timing constraints shown in Figure 38.3 apply to the settings of the TSCR register.

TSOE bit (Temperature Sensor Output Enable)

The TSOE bit enables or disables the temperature sensor output to ADC.

TSEN bit (Temperature Sensor Enable)

The TSEN bit starts or stops the temperature sensor.

38.2.2 TSCDR : Temperature Sensor Calibration Data Register

Base address: TSD = 0x407F_B000

Offset Address: 0x017C

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

Bit	Symbol	Function	R/W
15:0	TSCDR[15:0]	Temperature Sensor Calibration Data Chip-specific value	R
31:16	—	These bits are read as 0.	R

The TSCDR register stores temperature sensor calibration data measured for each chip at factory shipment.

Temperature sensor calibration data is the output voltage of the temperature sensor under the conditions Tj = 127°C and AVCC0 = VREFH0 = 3.3 V converted to a digital value by the A/D converter.

The TSCDR register is a read-only 32-bit register. Read from this register in 32-bit units.

Temperature sensor calibration data is stored in the lower 12 bits of the TSCDR register.

38.3 Using the Temperature Sensor

The temperature sensor outputs a voltage that varies with the temperature. This voltage is converted to a digital value by the A/D converter. To obtain the die temperature, convert this value into the temperature.

38.3.1 Preparation for Using the Temperature Sensor

The ambient temperature (T) is proportional to the temperature sensor voltage output (Vs), so ambient temperature is calculated with the following formula:

T = (Vs - V1) / slope + T1

- T: Ambient temperature of MCU as calculation result (°C)
- Vs: Voltage output by the temperature sensor on temperature measurement (V)
- T1: Temperature experimentally measured at one point (°C)
- V1: Voltage output by the temperature sensor on measurement of T1 (V)
- T2: Temperature experimentally measured at a second point (°C)
- V2: Voltage output by the temperature sensor on measurement of T2 (V)
- Slope: Temperature gradient of the temperature sensor (V / °C), slope = (V2 - V1) / (T2 -T1)

Characteristics vary between sensors, so Renesas recommends measuring two different sample temperatures as follows:

1. Use the A/D converter to measure the voltage V1 output by the temperature sensor at temperature T1.
2. Again use the A/D converter to measure the voltage V2 output by the temperature sensor at a different temperature T2.
3. Obtain the temperature gradient (slope = (V2 - V1) / (T2 - T1)) from these results.
4. Subsequently, obtain temperatures by substituting the slope into the formula for the temperature characteristic (T = (Vs - V1) / slope + T1).

If you are using the temperature gradient given in section 46, Electrical Characteristics, use the A/D converter to measure the voltage V1 output by the temperature sensor at temperature T1, then calculate the temperature characteristic using the following formula:

T = (Vs - V1) / slope + T1

Note: This method produces less accurate temperatures than measurement at two points.

In this MCU, the TSCDR register stores the temperature value (CAL127) of the temperature sensor measured under the condition Ta = Tj = 127°C and AVCC0 = VREFH0 = 3.3 V. If you use this value as the sample measurement result at the first point, you can omit the preparation before using the temperature sensor.

V1 is calculated from CAL127:

V1 = 3.3 × CAL127 / 4096 [V] (In case of 12 bit accuracy)

Using this value, the measured temperature can be calculated according to the following formula:

T = (Vs - V1) / slope + 127 [°C]

- T: Ambient temperature of MCU as calculation result (°C)
- Vs: Voltage output by the temperature sensor when the temperature is measured (V)
- V1: Voltage output by the temperature sensor when Ta = Tj = 127°C and AVCC0 = VREFH0 = 3.3 V (V)
- Slope: Temperature gradient of the temperature sensor*1 / 1000 (V/°C)

Note 1. See section 46, Electrical Characteristics

38.3.2 Procedures for Using the Temperature Sensor

Figure 38.2 shows the procedure for using the TSN.

For details, see section 36, A/D Converter.

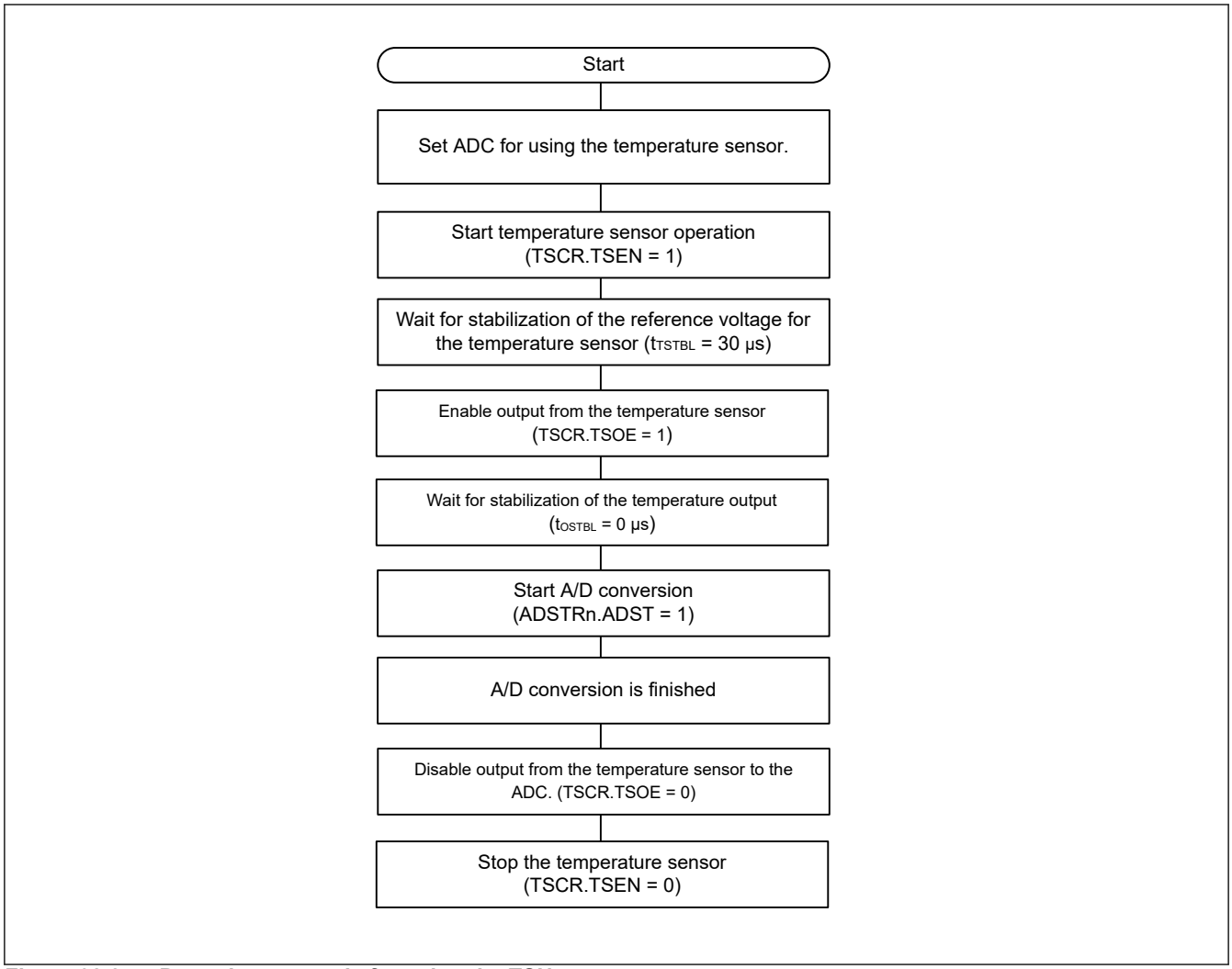


Figure 38.2 Procedure example for using the TSN

Figure 38.3 shows the timing from the start of temperature sensor operation until the completion of A/D conversion when the ADC is in single scan mode (the conversion target is the temperature sensor output only). See section 46, Electrical Characteristics for each time. The sampling time tSPL should be set longer than the value described in section 46, Electrical Characteristics.

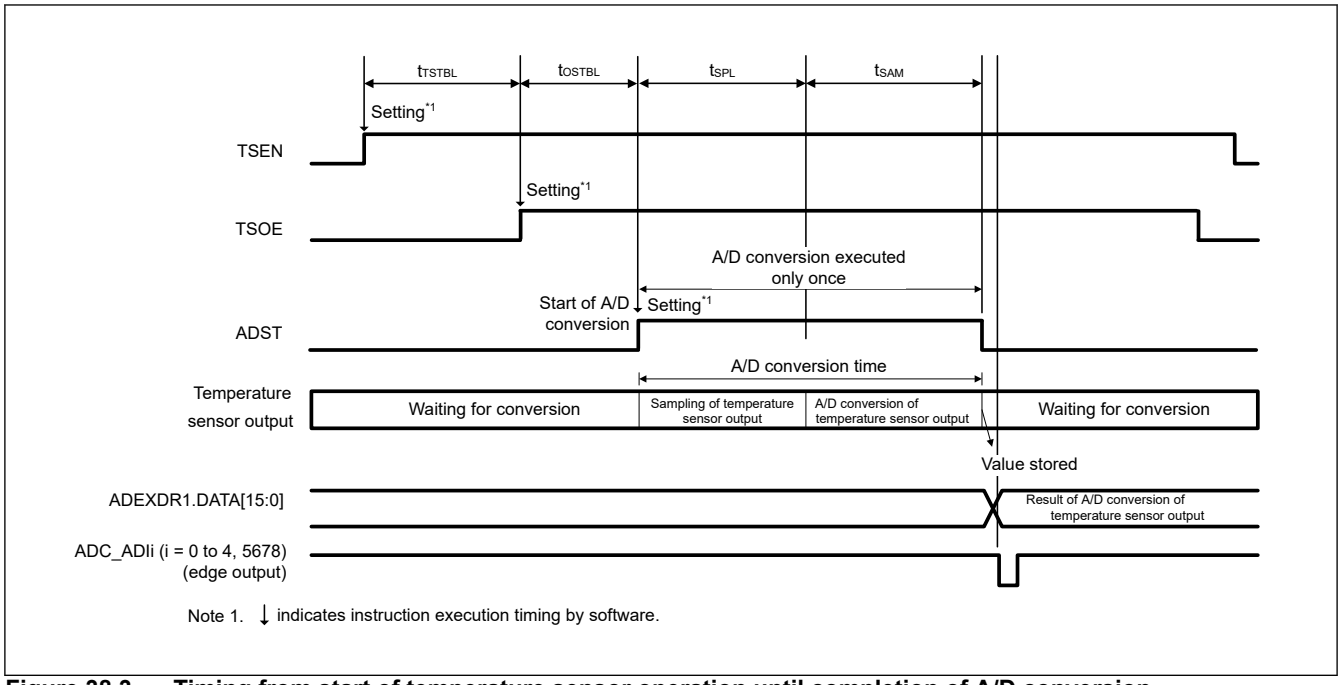


Figure 38.3 Timing from start of temperature sensor operation until completion of A/D conversion

38.4 Usage Notes

38.4.1 Settings for the Module-Stop Function

TSN operation can be disabled or enabled using the associated bit in Module Stop Control Register D (MSTPCRD). The TSN is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 10, Low Power Modes.

39. High-Speed Analog Comparator (ACMPHS)

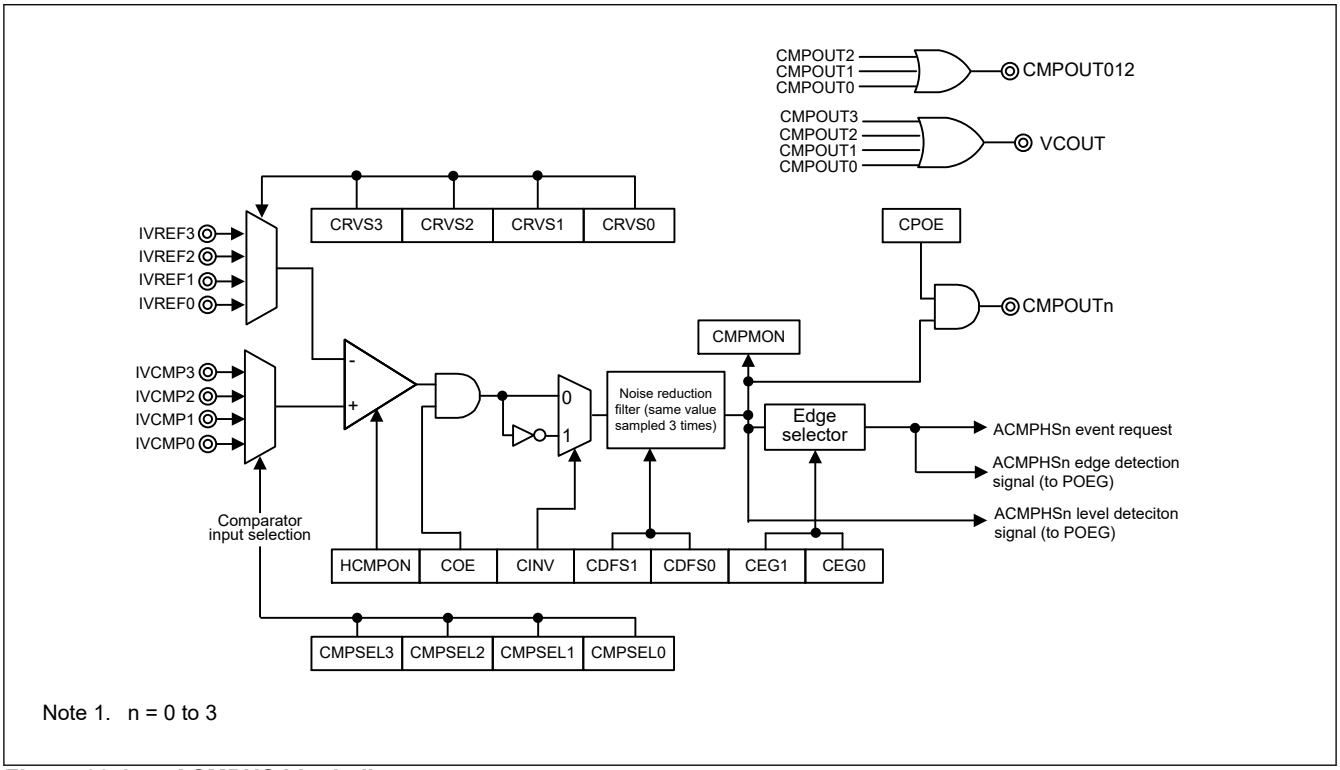
39.1 Overview

The High-Speed Analog Comparator (ACMPHS) can be used to compare a test voltage with a reference voltage and to provide a digital output based on the result of conversion. Both the test voltage and the reference voltage can be provided to the ACMPHS from internal sources (D/A converter output, Programable Gain Amplifire (PGA) output) and an external source. Such flexibility is useful in applications that require go/no-go comparisons to be performed between analog signals without necessarily requiring A/D conversion.

Table 39.1 lists the ACMPHS specifications, Figure 39.1 shows a block diagram, and Table 39.2 shows the input source configurations.

Table 39.1 ACMPHS specifications

Parameter	Specifications
Number of channels	4 channels: ACMPHSn (n = 0 to 3)
Analog input voltage	• Output from internal PGA • Input from internal A/D converter input pin (one selectable)
Reference voltage	• Output from internal D/A converter • Input from internal A/D converter input pin (one selectable)
ACMPHS output	• Comparison result • Generation of ELC event output • Monitor output from register
Interrupt request signal	• Interrupt request generated on valid edge detection from comparison result • Selectable to rising edge, falling edge, or both edges
Digital filter function	• Selectable to one of three sampling frequencies • Not using the filter function is selectable



Note 1. n = 0 to 3

Figure 39.1 ACMPHS block diagram

Table 39.2 Input source configuration of the ACMPHS

Comparator	Reference voltage input source				Analog voltage input source				Output pin
	IVREF3	IVREF2	IVREF1	IVREF0	IVCMP3	IVCMP2	IVCMP1	IVCMP0	
ACMPHS0	DA0	DA3	AN017	AN016	PGA0 output	AN000	—	AN012	VCOUT*1, CMPOUT012*2, CMPOUT0
ACMPHS1	DA1	DA3	AN017	AN016	PGA1 output	AN002	—	AN013	VCOUT*1, CMPOUT012*2, CMPOUT1
ACMPHS2	DA2	DA3	AN017	AN016	PGA2 output	AN004	—	AN014	VCOUT*1, CMPOUT012*2, CMPOUT2
ACMPHS3	DA3	DA2	AN017	AN016	PGA3 output	AN018	—	AN015	VCOUT*1, CMPOUT3

Note 1. ACMPHS0 to ACMPHS3 compare outputs are bundled with the VCOUT pin.
Note 2. ACMPHS0 to ACMPHS2 compare outputs are bundled with the CMPOUT012 pin.

39.2 Register Descriptions

39.2.1 CMPCTL : Comparator Control Register

Base address: ACMPHSn = 0x400F_4000 + 0x0100 × n (n = 0 to 3)

Offset address: 0x000

Bit position:	7	6	5	4	3	2	1	0
Bit field:	HCMPON	CDFS[1:0]		CEG[1:0]		—	COE	CINV
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CINV	Comparator Output Polarity Selection*1 *2 0: Do not invert comparator output 1: Invert comparator output	R/W
1	COE	Comparator Output Enable 0: Disable comparator output (output signal is low level) 1: Enable comparator output	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W
4:3	CEG[1:0]	Selection of Valid Edge (Edge Selector) 0 0: Do not detect edge 0 1: Detect rising edge 1 0: Detect falling edge 1 1: Detect both edges	R/W
6:5	CDFS[1:0]	Noise Filter Selection *1 *2 *3 0 0: Do not use noise filter 0 1: Use noise filter sampling frequency of PCLKB/2^3 1 0: Use noise filter sampling frequency of PCLKB/2^4 1 1: Use noise filter sampling frequency of PCLKB/2^5	R/W
7	HCMPON	Comparator Operation Control*4 0: Stop operation (comparator outputs a low-level signal) 1: Enable operation (enables input to the comparator pins)	R/W

Note 1. Disable the ACMPHS output (COE= 0) before changing the CDFS[1:0] and CINV bits.
Note 2. If the CDFS[1:0] and CINV bits are changed, an ACMPHS interrupt request and an ELC event might be generated. Before changing these bits, set the ELSRn register to 0 (the ACMPHS output is not linked). After changing these bits, clear the IR flag in the IELSRn register to 0 to clear the interrupt status.
Note 3. If the CDFS[1:0] bits are changed from 00b (noise filter not used) to a value other than 00b (noise filter used), perform sampling four times and update the filter output, and then use the ACMPHS interrupt request or the ELC event.
Note 4. A stabilization wait time is required to permit ACMPHS operation after enabling it (HCMPON = 1). The operation stabilization wait time for ACMPHS is 300 ns.

Note: Set this register before setting registers in the POEG when using comparator output as a POEG source.

The CMPCTL register controls the ACMPHS operation, enables or disables the ACMPHS output, selects the noise filter, selects the valid edge of the interrupt signal, and selects the interrupt.

39.2.2 CMPSEL0 : Comparator Input Select Register

Base address: ACMPHSn = 0x400F_4000 + 0x0100 × n (n = 0 to 3)

Offset address: 0x004

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

Bit	Symbol	Function	R/W
3:0	CMPSEL[3:0]	Comparator Input Selection*1 0x0: Do not input 0x1: Select IVCMP0*2 0x2: Setting prohibited*2 0x4: Select IVCMP2*2 0x8: Select IVCMP3*2 Others: Setting prohibited	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Use the following procedure to change the CMPSEL[3:0] bits. Writing a value other than 0x00 while the value of the CMPSEL0 register is not 0x00 is invalid. Writing 1 to two or more bits is also invalid. In both cases, the previous value is retained.
To change the CMPSEL[3:0] bits:
1. Set the CMPCTL.COE bit to 0.
2. Set the CMPSEL0 register to 0x00.
3. Set a new value in the CMPSEL[3:0] bits, with 1 set in only one of the bits.
4. Wait for the input switching stabilization wait time (200 ns).
5. Set the CMPCTL.COE bit to 1.
6. Clear the IR flag in the IELSRn register to clear the interrupt status.

Note 2. For details, see Table 39.2.

39.2.3 CMPSEL1 : Comparator Reference Voltage Select Register

Base address: ACMPHSn = 0x400F_4000 + 0x0100 × n (n = 0 to 3)

Offset address: 0x008

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

Bit	Symbol	Function	R/W
3:0	CRVS[3:0]	Reference Voltage Selection*1 0x0: Do not input 0x1: Select IVREF0*2 0x2: Select IVREF1*2 0x4: Select IVREF2*2 0x8: Select IVREF3*2 Others: Setting prohibited	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Use the following procedure to change the CRVS[3:0] bits. Writing a value other than 0x00 while the value of the CMPSEL1 register is not 0x00 is invalid. Writing 1 to two or more bits is also invalid. In both cases, the previous value is retained.
To change the CRVS[3:0] bits:
1. Set the CMPCTL.COE bit to 0.
2. Set the CMPSEL1 register to 0x00.
3. Set a new value to the CRVS[3:0] bits, with 1 set in only one of the bits.
4. Wait for the input switching stabilization wait time (200 ns)

5. Set the CMPCTL.COE bit to 1.
6. Clear the IR flag in the IELSRn register to clear the interrupt status.

Note 2. For details, see Table 39.2.

39.2.4 CMPMON : Comparator Output Monitor Register

Base address: ACMPHSn = 0x400F_4000 + 0x0100 × n (n = 0 to 3)

Offset address: 0x00C

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

Bit	Symbol	Function	R/W
0	CMPMON	Comparator Output Monitor*1 0: Comparator output is low 1: Comparator output is high	R
7:1	—	These bits are read as 0. The write value should be 0.	R

Note 1. When ACMPHS operation is enabled (HCMPON = COE = 1) but the noise filter is not in use (CDFS[1:0] = 00b), design the software so that the CMPMON bit is read twice and the values are only used after the two consecutive values match.

39.2.5 CPIOC : Comparator Output Control Register

Base address: ACMPHSn = 0x400F_4000 + 0x0100 × n (n = 0 to 3)

Offset address: 0x010

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

Bit	Symbol	Function	R/W
0	CPOE	Comparator Output Selection 0: Disable CMPOUTn pin output of the comparator (output signal is low fixed) 1: Enable CMPOUTn pin output of the comparator	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

39.3 Operation

The ACMPHS compares a reference voltage to an analog input voltage. Operation is not guaranteed when the values of registers are changed during ACMPHS operation. Table 39.3 shows the procedures for setting the registers associated with ACMPHS.

Table 39.3 Procedure for setting registers associated with ACMPHSn (n = 0 to 3) (1 of 2)

Step	Register	Bit	Setting
1	Associated MSTPCRD register	MSTPD28 to MSTPD25	0: Input clock supply.
2	Associated pin function control register (PFS)	ASEL	1: Select the function of pins IVREF and IVCMP.
3	Associated D/A convertor		When using the D/A convertor, select in the register.
4	CMPSEL0, CMPSEL1	CMPSEL0 to CMPSEL3, CRVS0 to CRVS3	Select the ACMPHSn input, with 1 set in only one of the bits.
5	CMPCTL	CDFS[1:0], CEG1, CEG0, and CINV	Set up ACMPHSn control.
		HCMPON	1: Enable ACMPHSn operation.

Table 39.3 Procedure for setting registers associated with ACMPHSn (n = 0 to 3) (2 of 2)

Step	Register	Bit	Setting
6	Waiting for the ACMPHS stabilization time (minimum 300 ns).		
7	CMPCTL	COE	1: Enable ACMPHSn output.
8	CPIOC	CPOE	Set the CMPOUTn output
	Associated pin function control register (PFS)	PSEL, PMR	Select the ACMPHS port function.
9	IELSRn	IR, IELS[8:0]	When using an interrupt, select the interrupt status flag and the ICU event link.*1
10	ELSRn	ELS[8:0]	When using an ELC, select the event link.*2
11	Operation started		
12	CMPCTL	COE	0: When changing IVREF or IVCMP, to disable ACMPHSn output.
13	CMPSEL1	CRVS0 to CRVS3	Change the CMPSEL1 bits as follows: 1. Set bits CMPSEL1 to 0000 0000b. 2. Set a new value to the CMPSEL1 bits, with 1 set in only one of the bits.
	CMPSEL0	CMPSEL0 to CMPSEL3	Change the CMPSEL0 bits as follows: 1. Set bits CMPSEL0 to 0000 0000b. 2. Set a new value to the CMPSEL0 bits, with 1 set in only one of the bits.
14	Waiting for the ACMPHS switching stabilization time (minimum 200 ns).		
15	CMPCTL	COE	1: Enable ACMPHSn output.
16	Operation restarted		

Note 1. After ACMPHSn is set, an unnecessary interrupt might occur until operation becomes stable, so initialize the interrupt flag.
Note 2. After ACMPHSn is set, an unnecessary interrupt might occur until operation becomes stable, so initialize the event link select.

Figure 39.2 shows an example of ACMPHS operation. The VCOUT output becomes 1 when the analog input voltage is higher than the ACMPHS reference input voltage, and the VCOUT output becomes 0 when the analog input voltage is lower than the reference voltage. When the ACMPHS output changes, an interrupt request and an ELC event are output.

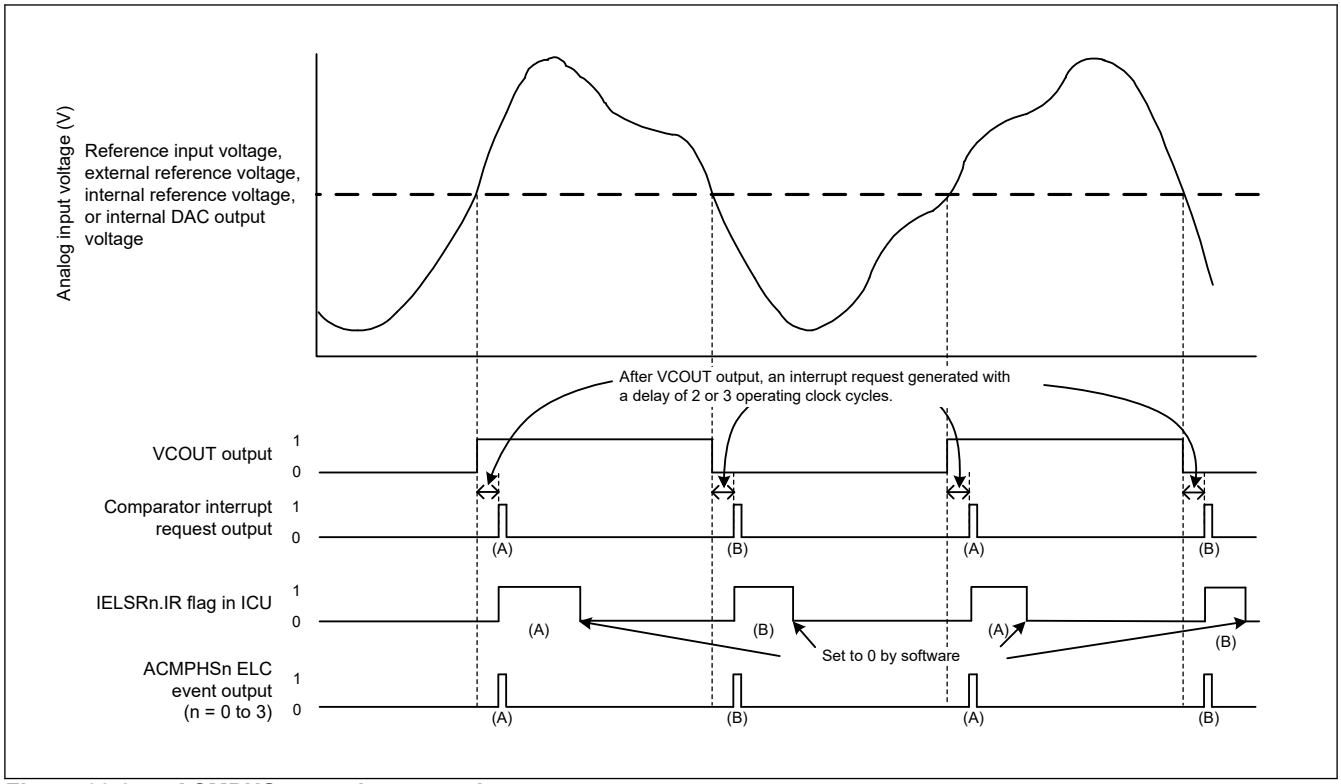


Figure 39.2 ACMPHS operation example

Figure 39.2 applies when CPOE = 1 (pin output enabled), CDFS[1:0] = 00b (filter not used), and CEG1 = CEG0 = 1 (both-edge detection selected). When CINV = 0, CEG0 = 1, and CEG1 = 0 (rising-edge detection selected for non-inversion output signal from the ACMPHS), the IELSR.IR flag changes as shown by (A) only. When CINV = 0, CEG0 = 0, and CEG1 = 1 (falling-edge detection selected for non-inversion output signal from the ACMPHS), the IR flag changes as shown by (B) only.

39.4 Noise Filter

The ACMPHS contains a noise filter. The sampling clock can be selected in the CMPCTL.CDFS[1:0] bits. The comparator output signal is sampled every sampling clock, and if the same value is sampled three times, the noise filter output at the next sampling clock cycle is used as the ACMPHS output.

Figure 39.3 shows the configuration of the noise filter and edge detector, and Figure 39.4 shows an example of noise filter and interrupt operation.

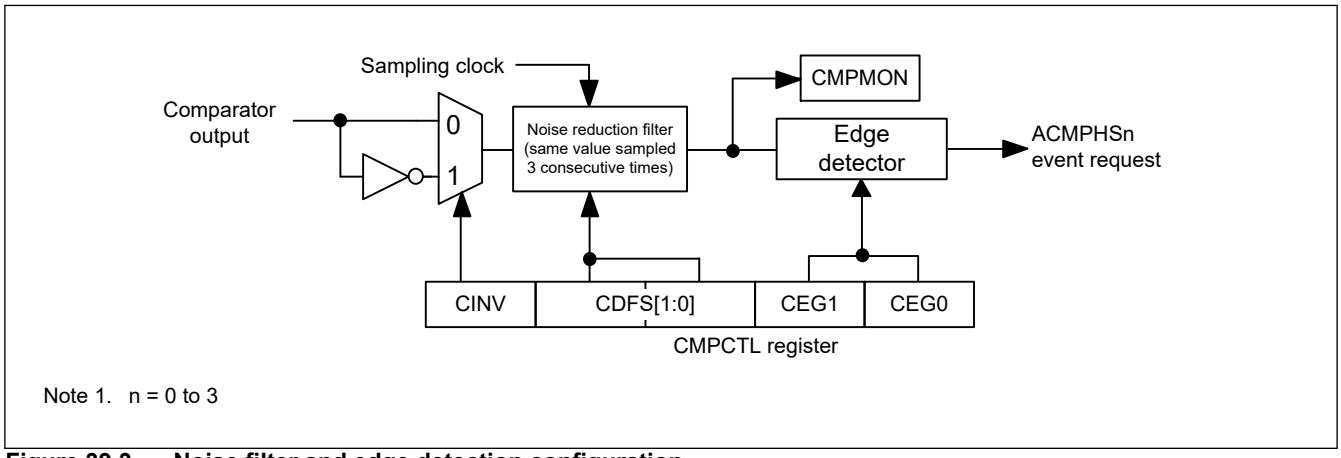


Figure 39.3 Noise filter and edge detection configuration

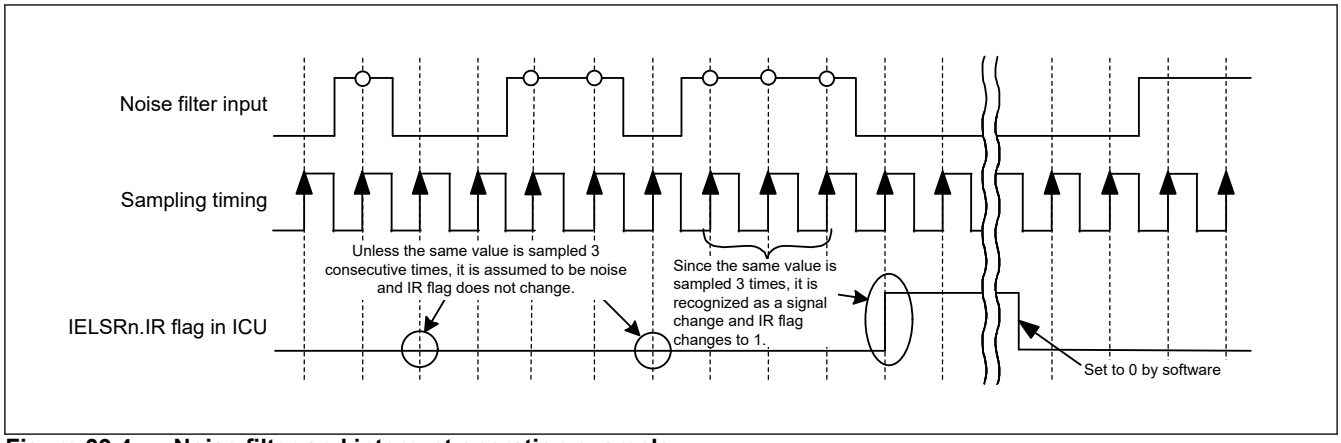


Figure 39.4 Noise filter and interrupt operation example

The operation example in Figure 39.4 applies when the CMPCTL.CDFS[1:0] bits are 01b, 10b, or 11b (noise filter used).

39.5 ACMPHS Interrupts

The ACMPHS generates four interrupt requests from sources ACMPHSn (n = 0 to 3). To use an ACMPHS interrupt, select it in the IELSR register in the Interrupt Controller Unit (ICU).

When using the ACMPHS interrupt through the edge selector, set at least one of the CMPCTL.CEG0 and CMPCTL.CEG1 bits to 1 (to a value other than 00b for no edge selection).

For details on the register settings related to ACMPHS interrupt requests, see section 39.2.1. CMPCTL : Comparator Control Register.

39.6 ACMPHS Output to the Event Link Controller (ELC)

The ELC uses the ACMPHS interrupt request signal as an ELC event signal, enabling link operation for the preset module. To use the ACMPHS ELC event, select them in the ELSR register in the ELC. When using the ELC event request, set at least one of the CMPCTL.CEG0 and CMPCTL.CEG1 bits to 1 (to a value other than 00b for no edge selection).

39.7 ACMPHS Pin Output

The comparison result from the ACMPHS can be output to external pins. Use the CMPCTL.CINV and CPIOC.CPOE bits to set the output polarity (non-inverted or inverted output) and enable or disable output. To output the ACMPHS comparison result to the CMPOUTn (n = 0 to 3), VCOUT and CMPOUT012 output pins, set the associated port mn pin function control register (PmnPFS) in the I/O register. The CMPOUT0 to CMPOUT3 compare outputs are bundled with the VCOUT pin. The CMPOUT0 to CMPOUT2 compare outputs are bundled with the CMPOUT012 pin too.

39.8 Usage Notes

39.8.1 Settings for the Module-Stop Function

ACMPHS operation can be disabled or enabled using the Module Stop Control Register. The ACMPHS is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 10, Low Power Modes.

40. Data Operation Circuit (DOC)

This is the DOC_B version of the DOC peripheral module.

DOC_B is referred to as DOC in this chapter.

40.1 Overview

The data operation circuit (DOC) is used to compare, add, and subtract 16 or 32-bit data. An interrupt can be generated when the following conditions apply.

- When the 16 or 32-bit compared values match the detection condition
- When the result of 16 or 32-bit data addition overflows
- When the result of 16 or 32-bit data subtraction underflows

Table 40.1 lists the data operation circuit specifications and Figure 40.1 shows a block diagram of the data operation circuit.

Table 40.1 DOC specifications

Item	Description
Data operation function	• 16 or 32-bit data comparison, comparison to detect data above or below thresholds, and window comparison • 16 or 32-bit data addition, and subtraction
Module-stop function	The module-stop state can be set to reduce power consumption.
Interrupts	• The compared values match the detection condition • The result of data addition is greater than 0xFFFF (DOCR.DOBW = 0) or 0xFFFF_FFFF (DOCR.DOBW = 1) • The result of data subtraction is less than 0x0000 (DOCR.DOBW = 0) or 0x0000_0000h (DOCR.DOBW = 1)
Event link function (output)	• The result of data comparison is consistent with detection condition • The result of data addition is greater than 0xFFFF (DOCR.DOBW = 0) or 0xFFFF_FFFF (DOCR.DOBW = 1) • The result of data subtraction is less than 0x0000 (DOCR.DOBW = 0) or 0x0000_0000 (DOCR.DOBW = 1)
TrustZone Filter	Security attribution can be set

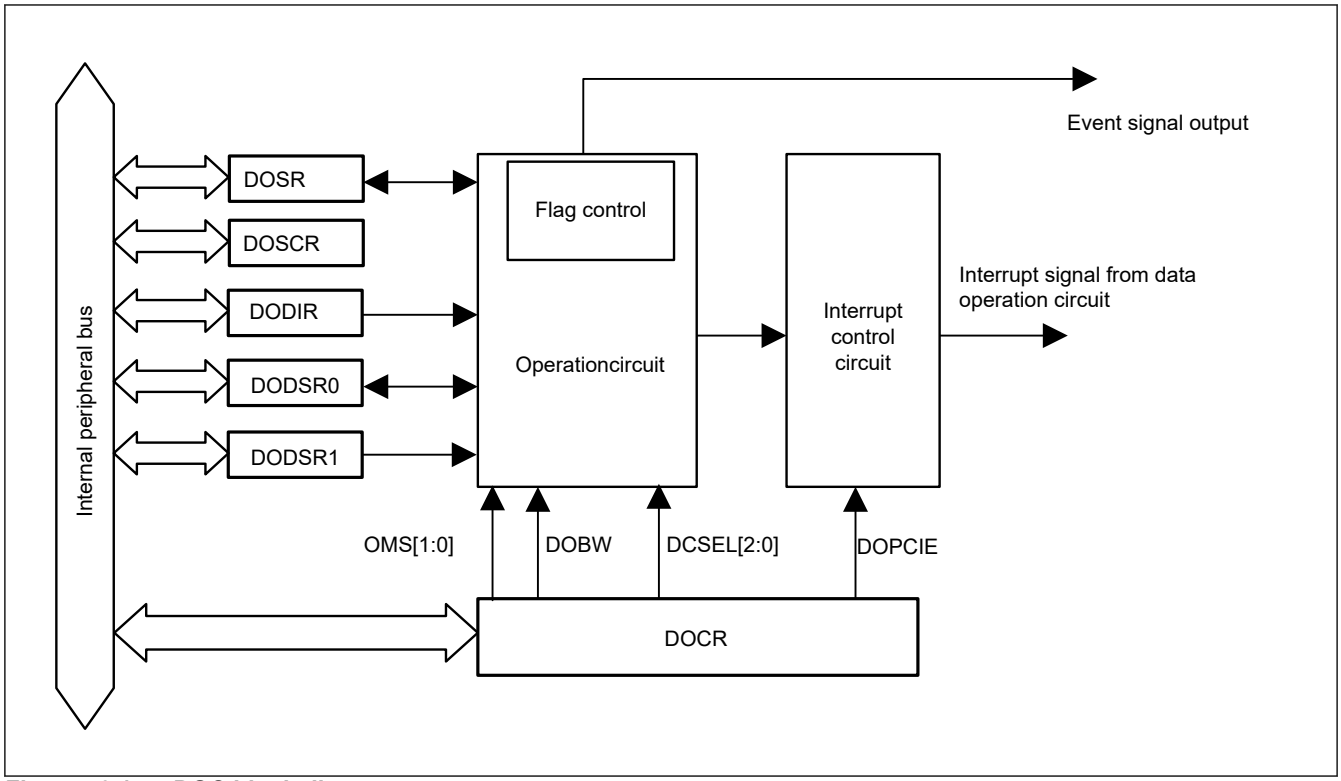


Figure 40.1 DOC block diagram

40.2 DOC Register Descriptions

40.2.1 DOCR : DOC Control Register

Base address: DOC_B = 0x4010_9000

Offset address: 0x00

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DOPCIE	DCSEL[2:0]			DOBW	—	OMS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	OMS[1:0]	Operating Mode Select 0 0: Data comparison mode 0 1: Data addition mode 1 0: Data subtraction mode 1 1: Setting prohibited	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W
3	DOBW	Data Operation Bit Width Select 0: 16-bit 1: 32-bit	R/W
6:4	DCSEL[2:0]*1	Detection Condition Select 0 0 0: Mismatch (DODSR0 ≠ DODIR) 0 0 1: Match (DODSR0 = DODIR) 0 1 0: Lower (DODSR0 > DODIR) 0 1 1: Upper (DODSR0 < DODIR) 1 0 0: Inside window (DODSR0 < DODIR < DODSR1) 1 0 1: Outside window (DODIR < DODSR0, DODSR1 < DODIR) Others: Setting prohibited	R/W

Bit	Symbol	Function	R/W
7	DOPCIE	Data Operation Circuit Interrupt Enable 0: Disables interrupts from the data operation circuit. 1: Enables interrupts from the data operation circuit.	R/W

Note 1. Valid only when data comparison mode is selected.

The DOCR is a register which can set the operation mode of data operation circuit and interrupt enable/disable.

OMS[1:0] bits (Operating Mode Select)

These bits select the operating mode of the data operation circuit.

DOBW bit (Data Operation Bit Width Select)

This bit selects the bit width of data operation.

DCSEL[2:0] bits (Detection Condition Select)

These bits are valid only when data comparison mode is selected.

These bits select the condition for detection in data comparison mode.

DOPCIE bit (Data Operation Circuit Interrupt Enable)

Setting this bit to 1 enables interrupts from the data operation circuit.

40.2.2 DOSR : DOC Flag Status Register

Base address: DOC_B = 0x4010_9000

Offset address: 0x04

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

Bit	Symbol	Function	R/W
0	DOPCF	Data Operation Circuit Flag Indicates the result of an operation.	R
7:1	—	These bits are read as 0.	R

The DOSR register indicates the status of the data operation.

DOPCF flag (Data Operation Circuit Flag)

[Setting conditions]

- DOCR.OMS[1:0] bits = 00b (Data comparison mode): The compared value matches the detection condition selected by DOCR.DCSEL[2:0] bits
- DOCR.OMS[1:0] bits = 01b (Data addition mode): The result of data addition is greater than FFFFh (DOCR.DOBW = 0) or FFFF_FFFFh (DOCR.DOBW = 1)
- DOCR.OMS[1:0] bits = 10b (Data subtraction mode): The result of data subtraction is less than 0000h (DOCR.DOBW = 0) or 0000_0000h (DOCR.DOBW = 1)

[Clearing condition]

- Writing 1 to the DOSCR.DOPCFCL bit

40.2.3 DOSCR : DOC Flag Status Clear Register

Base address: DOC_B = 0x4010_9000

Offset address: 0x08

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

Bit	Symbol	Function	R/W
0	DOPCFCL	DOPCF Clear 0: Maintains the DOPCF flag state. 1: Clears the DOPCF flag.	W
7:1	—	The write value should be 0.	W

The DOSCR is a register which can clear the status of data operation. This register is read as 0x00.

DOPCFCL bit (DOPCF Clear)

Setting this bit to 1 clears the DOPCF flag.

40.2.4 DODIR : DOC Data Input Register

Base address: DOC_B = 0x4010_9000

Offset address: 0x0C

Bit position:	31	0
Bit field:		
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
31:0	n/a	It stores data used in the operations. Access the DODIR with the bit width of data operation selected by the DOCR.DOBW bit.	R/W

40.2.5 DODSR0 : DOC Data Setting Register 0

Base address: DOC_B = 0x4010_9000

Offset address: 0x10

Bit position:	31	0
Bit field:		
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
31:0	n/a	Access the DODSR0 with the bit width of data operation selected by the DOCR.DOBW bit. This register stores data for use as a reference in data comparison mode. When selecting window comparison (DOCR.DCSEL[2:0] = 100b, 101b), set a value less than DODSR1 (DODSR1 > DODSR0). This register also stores the results of operations in data addition and data subtraction modes.	R/W

40.2.6 DODSR1 : DOC Data Setting Register 1

Base address: DOC_B = 0x4010_9000

Offset address: 0x14

Bit position:	31																												0
Bit field:																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	n/a	Access the DODSR1 with the bit width of data operation selected by the DOCR.DOBW bit. This register stores data for use as a reference in data comparison mode. When selecting window comparison (DOCR.DCSEL[2:0] = 100b, 101b), set a value greater than DODSR0 (DODSR1 > DODSR0). This register is only used for window comparisons.	R/W

40.3 Operation

40.3.1 Data Comparison Mode

Figure 40.2 to Figure 40.7 shows an example of the steps involved in data comparison mode operation by the data operation circuit.

The following is an example of operation when the bit width of data operation is 32-bit.

1. Writing 00b to the DOCR.OMS[1:0] bits selects data comparison mode, and setting the DOCR.DCSEL[2:0] to selects detection condition.
2. The 32-bit reference data is set in DODSR0 and DODSR1.*1
3. 32-bit data for comparison is written to DODIR.
4. If a value written to DODIR match the detection condition set by DOCR.DCSEL[2:0], the DOCR.DOPCF flag is set to 1 and an ELC event is generated. When the DOCR.DOPCIE bit is 1, a data operation circuit interrupt is also generated.

Note: The comparison operation is executed only by writing to the DODIR

Note 1. The DODSR1 register setting is required only when window comparison is selected. Set a value greater than DODSR0 (DODSR1 > DODSR0).

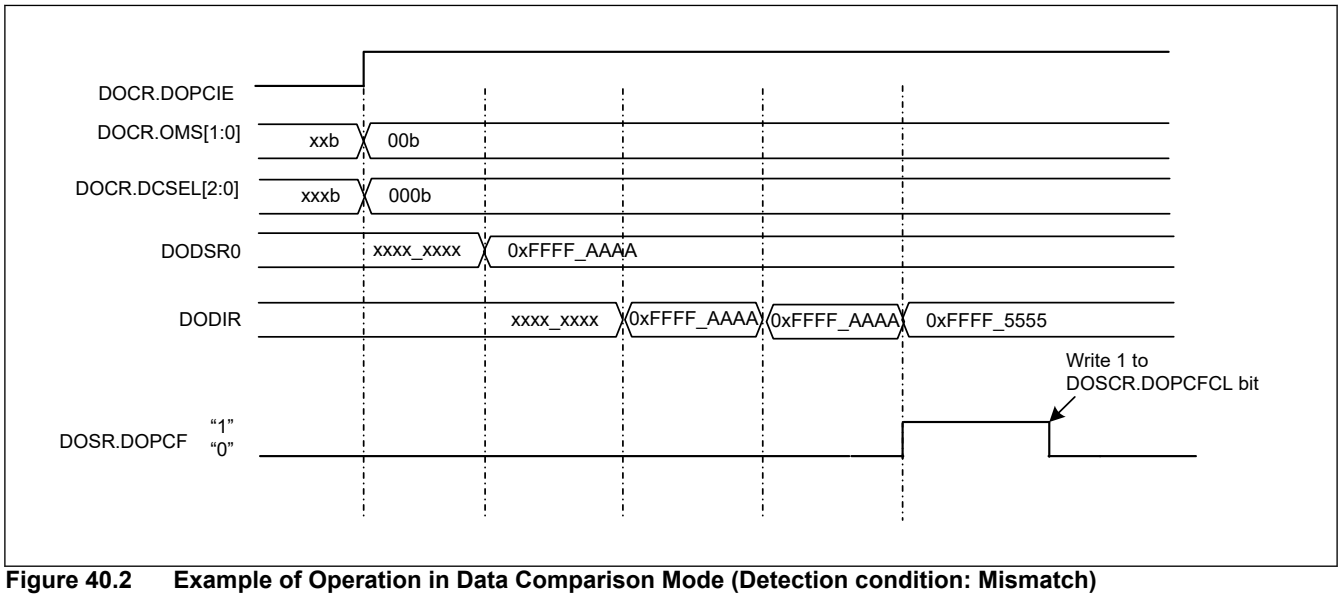


Figure 40.2 Example of Operation in Data Comparison Mode (Detection condition: Mismatch)

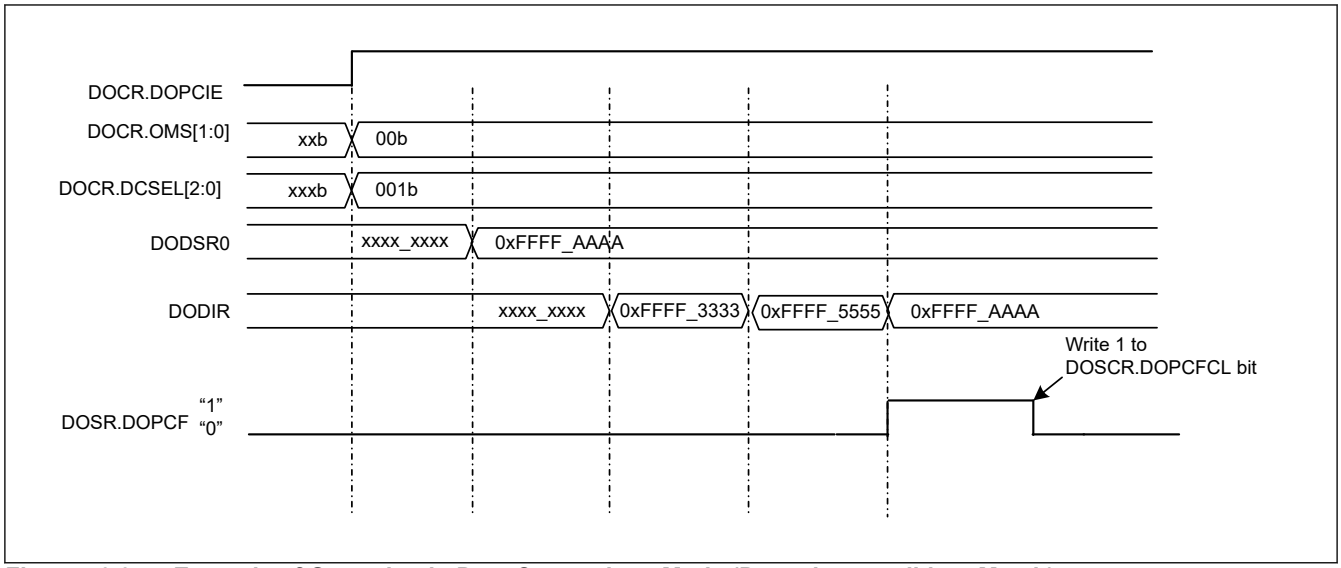


Figure 40.3 Example of Operation in Data Comparison Mode (Detection condition: Match)

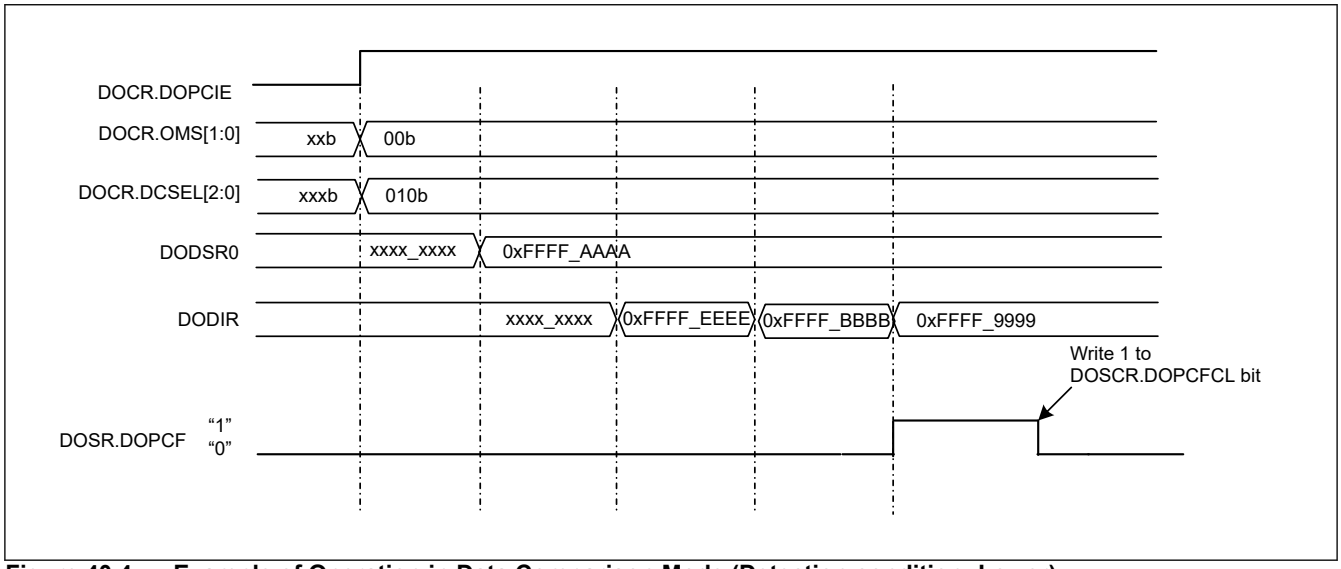


Figure 40.4 Example of Operation in Data Comparison Mode (Detection condition: Lower)

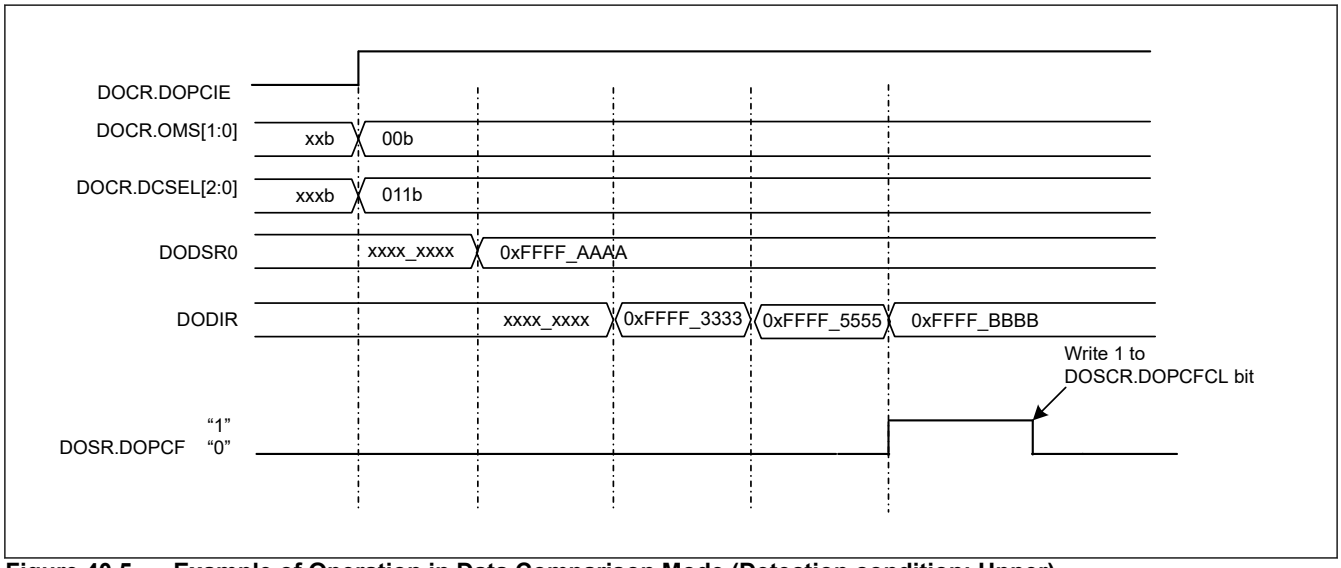


Figure 40.5 Example of Operation in Data Comparison Mode (Detection condition: Upper)

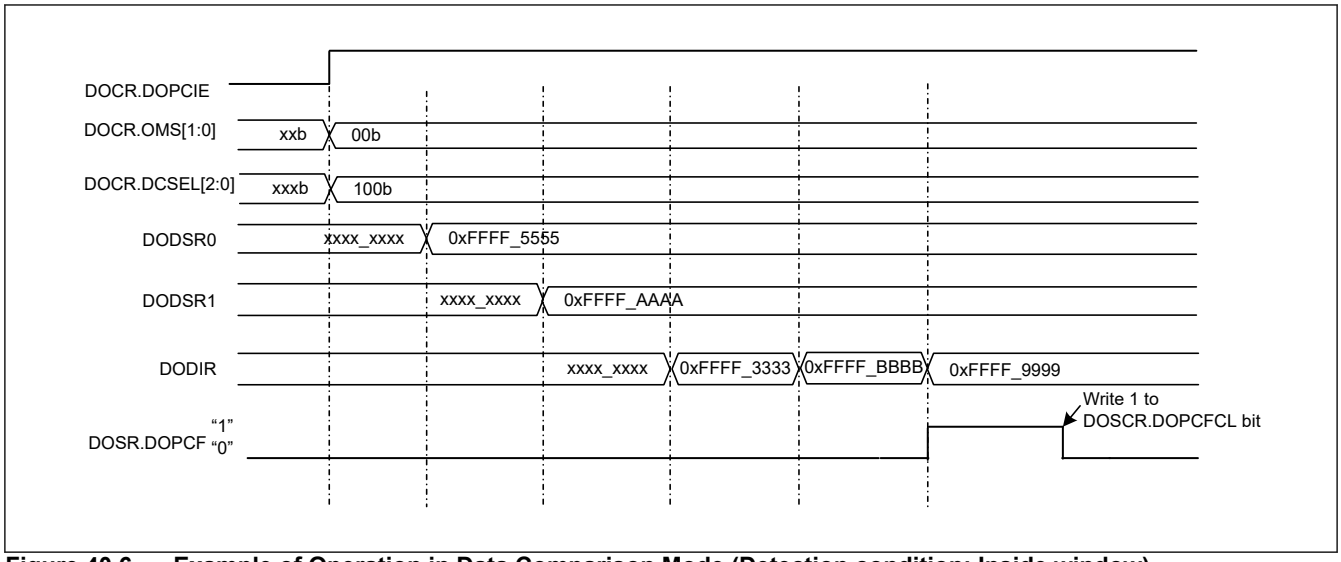


Figure 40.6 Example of Operation in Data Comparison Mode (Detection condition: Inside window)

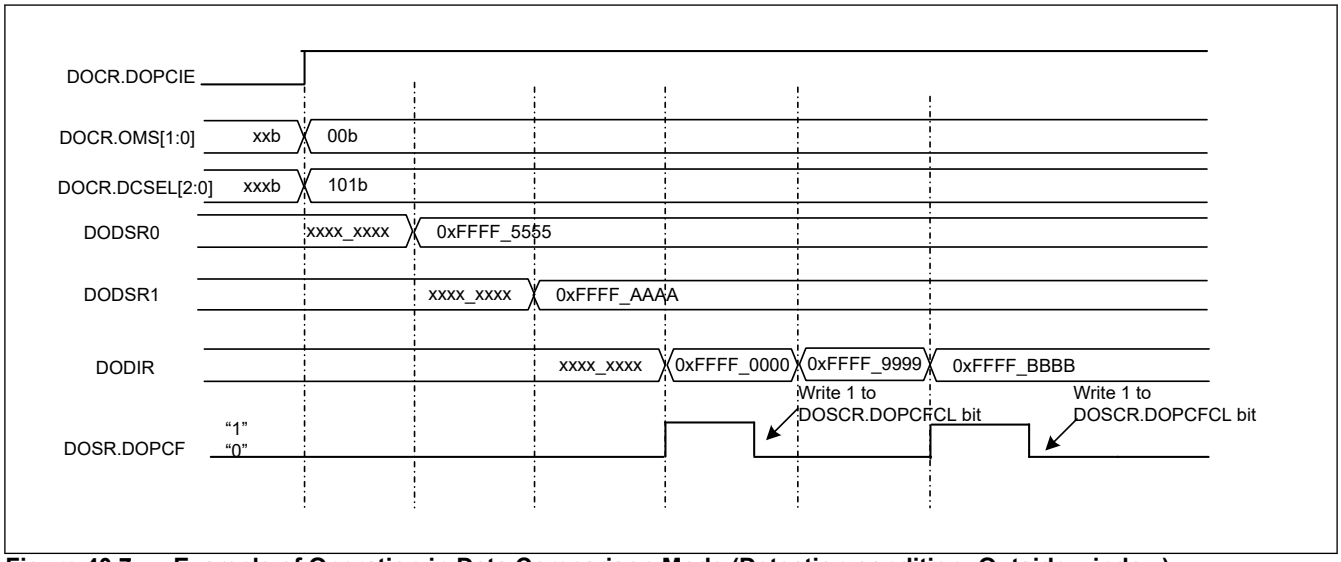


Figure 40.7 Example of Operation in Data Comparison Mode (Detection condition: Outside window)

40.3.2 Data Addition Mode

Figure 40.8 shows an example of the steps involved in data addition mode [*1] operation by the data operation circuit.

The following is an example of operation when the bit width of data operation is 32-bit.

1. Writing 01b to the DOCR.OMS[1:0] bits selects data addition mode.
2. 32-bit data is set in the DODSR0 register as the initial value.
3. 32-bit data to be added is written to DODIR. The result of the operation is stored in DODSR0.
4. Writing of 32-bit data continues until all data for addition have been written to DODIR.
5. If the result of an operation is greater than 0xFFFF_FFFF, the DOSR.DOPCF flag is set to 1 and an ELC event is generated. When the DOCR.DOPCIE bit is 1, a data operation circuit interrupt is also generated.

Note 1. Addition is executed only by writing to the DODIR.

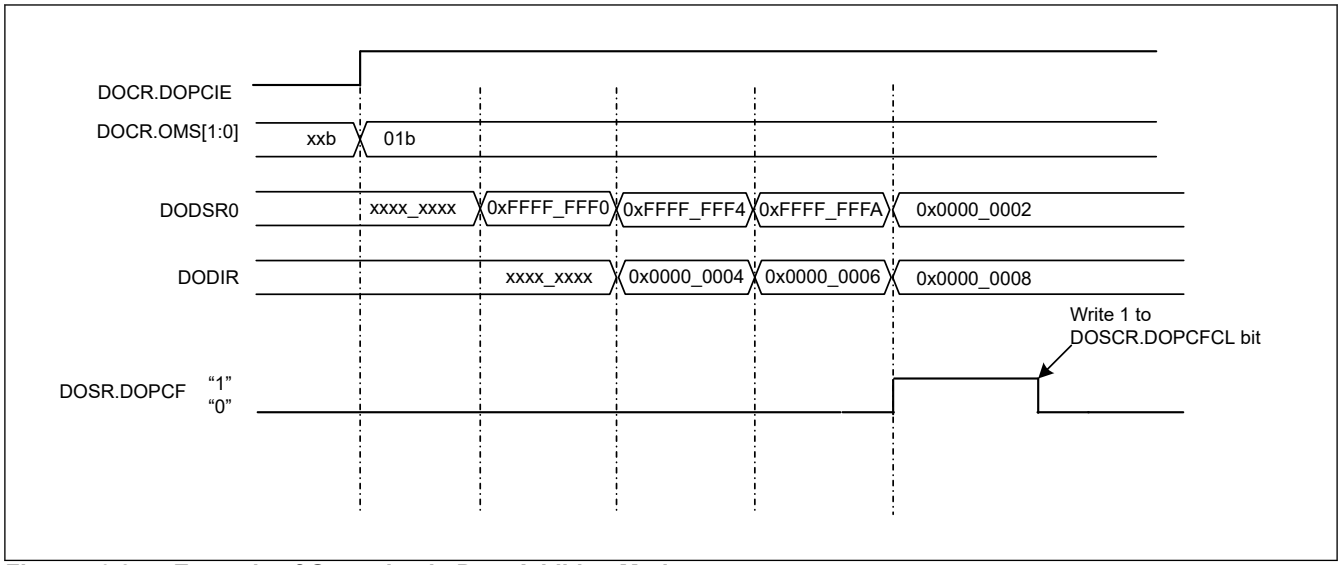


Figure 40.8 Example of Operation in Data Addition Mode

40.3.3 Data Subtraction Mode

Figure 40.9 shows an example of the steps involved in data subtraction mode [*1] operation by the data operation circuit.

The following is an example of operation when the bit width of data operation is 32-bit.

1. Writing 10b to the DOCR.OMS[1:0] bits selects data subtraction mode.
2. 32-bit data is set in the DODSR0 register as the initial value.
3. 32-bit data to be subtracted is written to DODIR. The result of the operation is stored in DODSR0.
4. Writing of 32-bit data continues until all data for subtraction have been written to DODIR.
5. If the result of an operation is less than 0x0000_0000, the DOSR.DOPCF flag is set to 1 and an ELC event is generated. When the DOCR.DOPCIE bit is 1, a data operation circuit interrupt is also generated.

Note 1. Subtraction is executed only by writing to the DODIR.

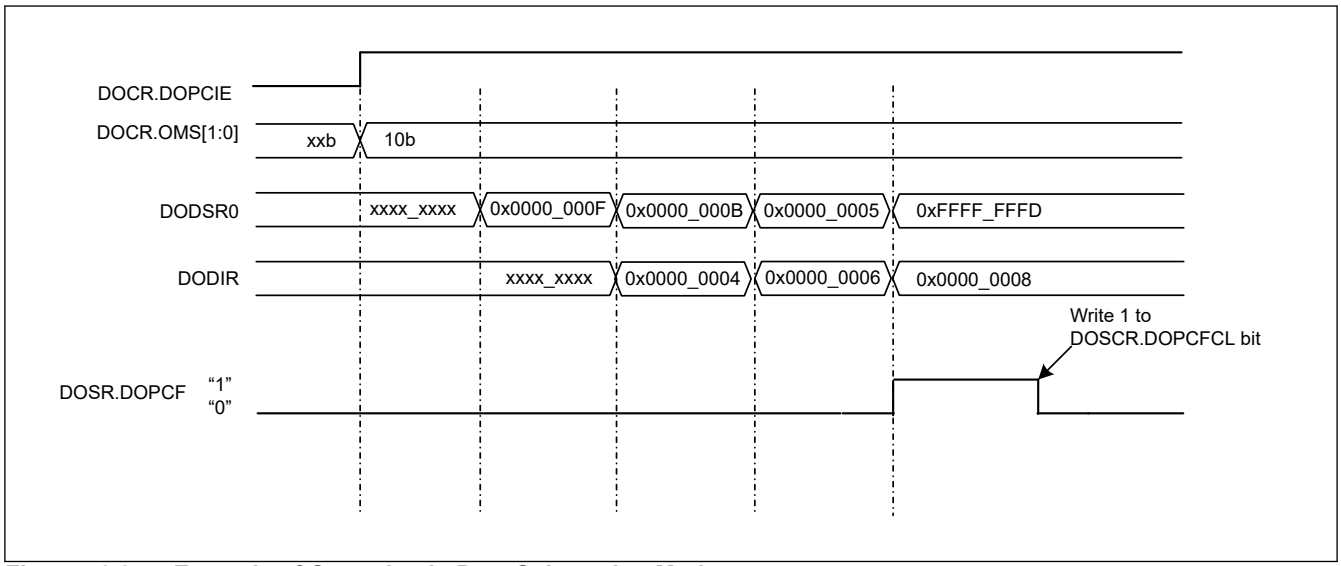


Figure 40.9 Example of Operation in Data Subtraction Mode

40.4 Interrupt Source

The data operation circuit generates the data operation circuit interrupt (DOC_DOPCI) as an interrupt request. When an interrupt source is generated, the data operation circuit flag corresponding to the interrupt is set to 1, when the data

operation circuit interrupt enable bit is enabled, interrupt request signal is generated. Table 40.2 describes the interrupt request.

Table 40.2 Interrupt request from DOC

Interrupt request	Status flag	Interrupt source
DOC interrupt	DOPCF	• The compared values match the detection condition. • The result of data addition is greater than 0xFFFF (DOCR.DOBW = 0) or 0xFFFF_FFFF (DOCR.DOBW = 1). • The result of data subtraction is less than 0x0000 (DOCR.DOBW = 0) or 0x0000_0000 (DOCR.DOBW = 1).

40.5 Event Link Output

The DOC outputs event signals for the event link controller (ELC) under the following conditions, and these can be used to initiate operations by other modules selected in advance.

- The compared values match the detection condition
- The data addition result is greater than 0xFFFF (DOCR.DOBW = 0) or 0xFFFF_FFFF (DOCR.DOBW = 1)
- The data subtraction result is less than 0x0000 (DOCR.DOBW = 0) or 0x0000_0000 (DOCR.DOBW = 1)

40.6 Interrupt Handling and Event Linking

The DOC has a bit to enable or disable interrupts. An interrupt request signal is output for the CPU when an interrupt source is generated while the corresponding enable bit is enabled.

In contrast, an event link output signal is sent to other modules as an event signal via the ELC when an interrupt source is generated, regardless of the setting of the corresponding interrupt enable bit.

40.7 Usage Notes

40.7.1 Settings for the Module-Stop State

The module Stop Control Register C (MSTPCRC) can enable or disable DOC operation. The DOC is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 10, Low Power Modes.

41. SRAM

41.1 Overview

The MCU provides an on-chip, high-density SRAM module with Error Correction Code (ECC).

Table 41.1 lists the SRAM specifications.

Table 41.1 SRAM specifications

Item	Specification
SRAM capacity	SRAM0: 64 KB
SRAM address	SRAM0: 0x2000_0000 to 0x2000_FFFF
Access	No wait states are inserted into the read cycle.
Data retention function	Not available in Deep Software Standby mode
Module-stop function	Module-stop state can be set to reduce power consumption
Error checking	SEC-DED (Single-Error Correction and Double-Error Detection Code)
Security	TrustZone Filter is integrated for memory access and SFR access. Access to the memory space is controlled by setting the memory Security Attribution (SA). And, access to I/O space (SFR) space is controlled by setting the register SA. See section 41.3.5. TrustZone Filter function.

41.2 Register Descriptions

41.2.1 SRAMSAR : SRAM Security Attribution Register

Base address: CPSCU = 0x4000_8000

Offset address: 0x10

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	SRAMSA2	—	SRAMSA0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	SRAMSA0	Security attributes of registers for SRAM Protection 0: Secure 1: Non-Secure	R/W
1	—	This bit is read as 1. The write value should be 1.	R/W
2	SRAMSA2	Security attributes of registers for ECC Relation 0: Secure 1: Non-Secure	R/W
31:3	—	These bits are read as 1. The write value should be 1.	R/W

Note: Only secure access can write to this register. Both secure access and Non-secure read access are allowed but Non-secure write access is not allowed and TrustZone access error is not generated.

Note: This register is write-protected by PRCR register.

SRAMSA0 bit (Security attributes of registers for SRAM Protection)

Security attributes of registers for Standby SRAM Protection. The target registers are as follow.

- PARIOAD
- SRAMPRCR

SRAMSA2 bit (Security attributes of registers for ECC Relation)

Security attributes of registers for ECC Relation. The target registers are as follow.

- ECCMODE
- ECC2STS
- ECC1STSEN
- ECC1STS
- ECCPRCR
- ECCPRCR2
- ECCETST
- ECCOAD

41.2.2 PARIOAD : SRAM Parity Error Operation After Detection Register

Base address: SRAM = 0x4000_2000

Offset address: 0x00

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

Bit	Symbol	Function	R/W
0	OAD	Operation After Detection 0: Non-maskable interrupt 1: Reset	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

The PARIOAD register controls the operation on detection of a parity error. The SRAM Protection Register (SRAMPRCR) protects this register against writes. Always set the SRAMPRCR bit in SRAMPRCR to 1 before writing to this bit. Do not write to the PARIOAD register while accessing the SRAM.

OAD bit (Operation After Detection)

The OAD bit specifies the generation of either a reset or non-maskable interrupt when a parity error is detected. The OAD bit is commonly used for Standby SRAM.

41.2.3 SRAMPRCR : SRAM Protection Register

Base address: SRAM = 0x4000_2000

Offset address: 0x04

Bit position:	7	6	5	4	3	2	1	0
Bit field:	KW[6:0]							SRAM PRCR
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SRAMPRCR	Register Write Control 0: Disable writes to protected registers 1: Enable writes to protected registers	R/W

Bit	Symbol	Function	R/W
7:1	KW[6:0]	Write Key Code These bits enable or disable writes to the SRAMPRCR bit	W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

SRAMPRCR bit (Register Write Control)

The SRAMPRCR bit controls the write mode of the PARIOAD register. Setting the bit to 1 enables writes to the PARIOAD register. When you write to this bit, always write 0x78 to KW[6:0] bits simultaneously.

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

The KW[6:0] bits enable or disable writes to the SRAMPRCR bit. When you write to the SRAMPRCR bit, always write 0x78 to these bits simultaneously. When a value other than 0x78 is written to KW[6:0], the SRAMPRCR bit is not updated. The KW[6:0] bits are always read as 0x00.

41.2.4 ECCMODE : ECC Operating Mode Control Register

Base address: SRAM = 0x4000_2000

Offset address: 0xC0

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

Bit	Symbol	Function	R/W
1:0	ECCMOD[1:0]	ECC Operating Mode Select 0 0: Disable ECC function 0 1: Setting prohibited 1 0: Enable ECC function without error checking 1 1: Enable ECC function with error checking	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

The ECCMODE register specifies the ECC operating mode. The ECC Protection Register (ECCPRCR) protects this register against writes. Before writing to this register, set the ECCPRCR bit in the ECCPRCR register to 1 (write protection disabled). Do not write to the ECCMODE register while accessing the SRAM.

ECCMOD[1:0] bits (ECC Operating Mode Select)

The ECCMOD[1:0] bits set the SRAM0 ECC operating mode.

41.2.5 ECC2STS : ECC 2-Bit Error Status Register

Base address: SRAM = 0x4000_2000

Offset address: 0xC1

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	ECC2 ERR
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ECC2ERR	ECC 2-Bit Error Status 0: No 2-bit ECC error occurred 1: 2-bit ECC error occurred	R/W*1
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note 1. Only 0 can be written to clear the bit.

ECC2ERR bit (ECC 2-Bit Error Status)

The ECC2ERR bit indicates whether a 2-bit ECC error occurred in SRAM0. When a 2-bit error is detected while ECC operations are enabled and error checking is selected, theECC2ERR bit is set to 1. The SRAM error signal is also asserted at this time. The 2-bit ECC error can be cleared by writing 0 to the ECC2ERR bit.

The SRAM error can be specified as a non-maskable interrupt or a reset in the ECCOAD register. Do not access the SRAM0 while writing 0 to this register.

41.2.6 ECC1STSEN : ECC 1-Bit Error Information Update Enable Register

Base address: SRAM = 0x4000_2000

Offset address: 0xC2

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	E1STSEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	E1STSEN	ECC 1-Bit Error Information Update Enable 0: Disable updating of 1-bit ECC error information 1: Enable updating of 1-bit ECC error information	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

The ECC1STSEN register enables or disables updating of the ECC 1-bit Error Status Register (ECC1STS) in response to a 1-bit error ECC error in the SRAM0.

The ECC Protection Register (ECCPRCR) protects this register against writes. Before writing to this bit, set the ECCPRCR bit in the ECCPRCR register to 1 (write protection disabled).

E1STSEN bit (ECC 1-Bit Error Information Update Enable)

The E1STSEN bit enables or disables updating of the SRAM0 1-Bit Error Status Register (ECC1STS) in response to a 1-bit error in the SRAM0. This register also functions as an interrupt or a reset mask.

41.2.7 ECC1STS : ECC 1-Bit Error Status Register

Base address: SRAM = 0x4000_2000

Offset address: 0xC3

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	ECC1ERR
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ECC1ERR	ECC 1-Bit Error Status 0: No 1-bit ECC error occurred 1: 1-bit ECC error occurred	R/(W)*1
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

Note 1. Only 0 can be written to clear the bit.

ECC1ERR bit (ECC 1-Bit Error Status)

The ECC1ERR bit indicates whether a 1-bit ECC error occurred in the SRAM0. When a 1-bit error is detected while ECC operations are enabled and error checking is selected, theECC1ERR bit is set to 1. The SRAM error signal is also asserted at this time. The 1-bit ECC error can be cleared by writing 0 to the ECC1ERR bit.

The SRAM error can be specified as a non-maskable interrupt or a reset in the ECCOAD register. Do not access the SRAM0 while writing 0 to this register.

41.2.8 ECCPRCR : ECC Protection Register

Base address: SRAM = 0x4000_2000

Offset address: 0xC4

Bit position:	7	6	5	4	3	2	1	0
Bit field:	KW[6:0]							ECCPRCR
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ECCPRCR	Register Write Control 0: Disable writes to the protected registers 1: Enable writes to the protected registers	R/W
7:1	KW[6:0]	Write Key Code 0x78: Enable write to the ECCPRCR bit Others: Disable write to the ECCPRCR bit	W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

ECCPRCR bit (Register Write Control)

The ECCPRCR bit controls the write of the ECCMODE, ECC1STSEN, and ECCOAD registers. When this bit is set to 1, writing to the ECCMODE, ECC1STSEN, and ECCOAD registers is enabled. When writing to this bit, write 0x78 to the KW[6:0] bits at the same time.

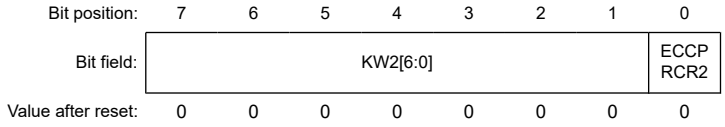
KW[6:0] bits (Write Key Code)

The KW[6:0] bits enable or disable writes to the ECCPRCR bit. When writing toECCPRCR bit, write 0x78 to the KW[6:0] bits at the same time. When a value other than 0x78 is written to theKW[6:0] bits, the ECCPRCR bit is not updated. The KW[6:0] bits are always read as 0x00.

41.2.9 ECCPRCR2 : ECC Protection Register 2

Base address: SRAM = 0x4000_2000

Offset address: 0xD0



Bit	Symbol	Function	R/W
0	ECCPRCR2	Register Write Control 0: Disable writes to the protected registers 1: Enable writes to the protected registers	R/W
7:1	KW2[6:0]	Write Key Code 0x78: Enable write to the ECCPRCR2 bit Others: Disable write to the ECCPRCR2 bit	W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

ECCPRCR2 bit (Register Write Control)

The ECCPRCR2 bit controls the write mode of the ECCETST register. When the ECCPRCR2 bit is set to 1, writes to the ECCETST register is enabled. When writing to this bit, write 0x78 to the KW2[6:0] bits at the same time.

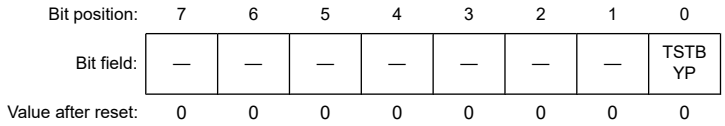
KW2[6:0] bits (Write Key Code)

The KW2[6:0] bits enable or disable writes to the ECCPRCR2 bit. When writing to ECCPRCR2 bit, write 0x78 to the KW2[6:0] bits at the same time. When a value other than 0x78 is written to the KW2[6:0] bits, the ECCPRCR2 bit is not updated. The KW2[6:0] bits are always read as 0x00.

41.2.10 ECCETST : ECC Test Control Register

Base address: SRAM = 0x4000_2000

Offset address: 0xD4



Bit	Symbol	Function	R/W
0	TSTBYP	ECC Bypass Select 0: Disable ECC bypass 1: Enable ECC bypass	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

The ECC Protection Register 2 (ECCPRCR2) protects this register against writes. Before writing to this bit, set the ECCPRCR2 bit in the ECCPRCR2 register to 1 (write protection disabled). Do not write to the ECCETST register while accessing the SRAM.

TSTBYP bit (ECC Bypass Select)

The TSTBYP bit enables direct access to the ECC code by bypassing the ECC function. When the ECC bypass function is used, the ECCMOD[1:0] bits in the ECCMODE register are set to 00b. The ECC must be accessed in 32 bits using the same address for 32-bit data. The ECC code is assigned to the lower 7 bits of the 32-bit data. When writing the ECC code, the upper 25 bits are ignored. When reading the ECC code, the upper 25 bits are undefined.

Note: For details of ECC test, see section 41.3.4. ECC Decoder Testing.

41.2.11 ECCOAD : SRAM ECC Error Operation After Detection Register

Base address: SRAM = 0x4000_2000

Offset address: 0xD8

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

Bit	Symbol	Function	R/W
0	OAD	Operation After Detection 0: Non-maskable interrupt 1: Reset	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

The ECC Protection Register (ECCPRCR) protects this register against writes. Before writing to this bit, set the ECCPRCR bit in the ECCPRCR register to 1 (write protection disabled). Do not write to the ECCOAD register while accessing the SRAM.

OAD bit (Operation After Detection)

The OAD bit selects whether to generate a reset or a non-maskable interrupt when an ECC error is detected. The OAD bit in the ECCOAD register is used for SRAM0.

41.3 Operation

41.3.1 Module Stop Function

Power consumption can be reduced by setting module stop control register A (MSTPCRA) to stop supply of the clock signal to SRAM.

SRAM0 is controlled by SRAM0 bit in MSTPCRA register and, in the case of 1, SRAM0 becomes the clock stop state.

The SRAM is thus placed in the module-stop state by stopping supply of the clock signals. The SRAM operates after a reset.

SRAM is not accessible if it is in the module-stop state. A transition to the module-stop state should not be made while access to SRAM is in progress.

Access to the SRAM in the module-stop state is prohibited. If access is attempted, correct operation is not guaranteed.

For details on the MSTPCRA register, see section 10, Low Power Modes.

41.3.2 Correction of ECC errors

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

When ECC function is enabled, 7-bit check bits are appended to 32-bit data for writing. For reading, 39-bit (data: 32 bits, check bits: 7 bits) data is read out from the SRAM.

When ECC function is enabled and error checking is selected by setting ECCMOD[1:0] in the ECCMODE register to 00b, error correction is done if a 1-bit error occurs and the ECC1ERR bit in the ECC1STS register is set to 1 if the E1STSEN bit in the ECC1STSEN register is 1. If a 2-bit error occurs, error detection is done and the ECC2ERR bit in the ECC2STS register is set to 1, though error correction is not performed.

When ECC function is enabled and the error checking is disable, error correction is done if a 1-bit error occurs but ECC1ERR bit in the ECC1STS register is not updated although E1STSEN bit in the ECC1STSEN register is 1. If a 2-bit error occurs, this error is detected but the ECC2ERR bit in the ECC2STS register is not updated, and error correction is not performed.

When ECC function is disable, neither error correction nor error detection is done although 1-bit or 2-bit error occur.

So ECC1ERR bit and ECC2ERR bit are not updated.

There is no way to confirm the location where the error was found. Therefore, when after the occurrence of an error, update all the data.

When updating all the data after the occurrence of an error, the 32-bit data writing is only supported.

Since the SRAM data is undefined after power on and release from Deep Software Standby mode, accessing the SRAM when ECC function is enabled and error checking is selected causes an ECC error to occur. Therefore, before using ECC function, initial writing with 32-bit data size to the area to be used in the SRAM should be done.

When a read access is executed in a row after a write access, read access is executed with priority. Therefore, during initialization, do not perform the read access in a row after the write access.

41.3.3 ECC Error Interrupt Function

When ECC function is enabled and error checking is applied to the SRAM, an ECC error occurs when either the ECC2ERR bit in the ECC2STS register or the ECC1ERR bit in the ECC1STS register becomes 1 to indicate that ECC checking revealed a 2-bit error or a 1-bit error, respectively.

An ECC erroris output with a pulse width of ICLK.When the ECC 1-bit error is to be masked, set the ECC1STSEN.E1STSEN bit to 0 to disable updating of the ECC1ERR bit. An ECC error will not be generated while ECC function is disabled or when ECC function is enabled but error checking is not selected.

ECC error can choose non maskable interrupt or reset by ECCOAD register. When set 1 in the OAD bit of the ECCOAD register, ECC error is output to the Reset function. When set 0 in the OAD bit of the ECCOAD register, ECC Error interrupt is output to the ICU as non-maskable interrupt.

41.3.4 ECC Decoder Testing

Figure 41.1 shows the ECC decoder testing.

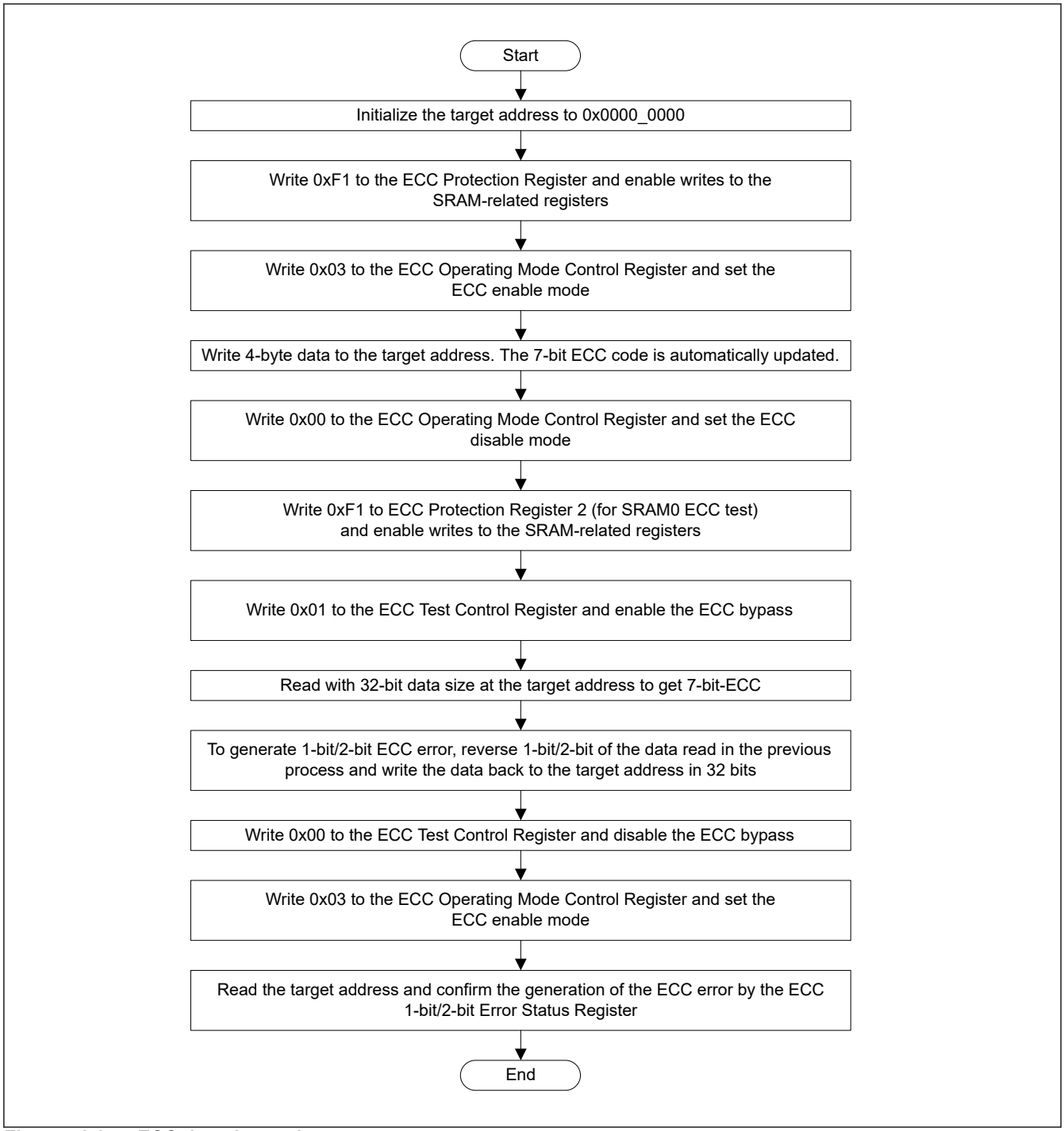


Figure 41.1 ECC decoder testing

41.3.5 TrustZone Filter function

There are two types of TrustZone Filter function for SRAM.

- TrustZone Filter for SRAM register protection
- TrustZone Filter for SRAM memory protection

41.3.5.1 TrustZone Filter for SRAM register protection

SRAM registers can be protected with a Security Attribution (SA) from Non-secure access. When SA indicates that SRAM registers are secure status, non-secure access cannot overwrite them because TrustZone Filter detects finds an error and protects the write access. SA for SRAM registers is just one to be used commonly among SRAM registers.

Table 41.2 Register protection

SA	Access status	Write access	Read access
Secure	Secure	Permit	Permit
	Non-secure	TrustZone Filter error Protected	Permit
Non-secure	Secure	Permit	Permit
	Non-secure	Permit	Permit

When TrustZone Filter error for SRAM register access occurs, no error notification and no error response occurs.

41.3.5.2 TrustZone Filter for SRAM memory protection

SRAM memory can be divided into Secure/Non secure callable/Non secure status with Memory Security Attribution (MSA) and can be protected from Non-secure access. When MSA indicates that SRAM memory region are Secure or Non secure callable status, Non-secure access can't overwrite them.

Table 41.3 Memory protection

SA	Access status	Write access	Read access
Secure / Non secure callable	Secure	Permit	Permit
	Non-secure	TrustZone Filter error • Protected • Error response occurs	TrustZone Filter error • Read data is 0 • Error response occurs
Non-secure	Secure	Permit	Permit
	Non-secure	Permit	Permit

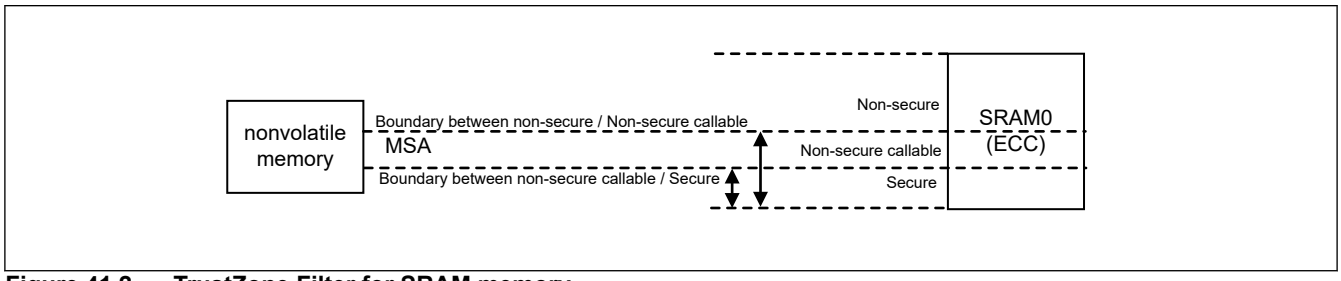


Figure 41.2 TrustZone Filter for SRAM memory

When TrustZone Filter error for SRAM memory access occurs, an error notification which become Reset request or NMI request occurs. See section 45.2. Arm TrustZone Security .

41.3.6 Interrupt Source

The SRAM interrupt source includes an ECC error and TrustZone filter error. ECC error can choose non-maskable interrupt or reset by OAD bit. When the debugger is connected, reset and non-maskable interrupt are maskable. Also, if these masks are set by the debugger, each status register is not set even if an ECC error occurs. For details on the debug mode, see section 2, CPU.

Table 41.4 SRAM Interrupt Source

Name	Interrupt Source	DTC Activation	DMAC Activation
ECCERR	ECC error	Not possible	Not possible
TZFLT	TrustZone filter error	Not possible	Not possible

41.3.7 Access Cycle

■ Number of cycles from the CPU

- When the cache is hit, access is one cycle.
- For cache off or non cacheable

Table 41.5 SRAM0

Register Setting	Read (cycles)		Write (cycles)	
	Word access	Half-word/Byte access	Word access	Half-word/Byte access
ECC Off ECCMOD[1] = 0	3		2*1	
ECC On ECCMOD[1] = 1	3		2*1	4

Note 1. For efficiency of the access, when read access occurs to the same memory after write, memory write by the precedent write command delays it until for the next idle cycle or the next write access. When read continues, it is given priority to read.

- For cache on and cacheble (When the cache miss hit)

Table 41.6 SRAM0

Register Setting	Read (cycles)		Write (cycles)	
	Word access	Half-word/Byte access	Word access	Half-word/Byte access
ECC Off ECCMOD[1] = 0	3		1*1	
ECC On ECCMOD[1] = 1	3		1*1	

Note 1. For efficiency of the access, when read access occurs to the same memory after write, memory write by the precedent write command delays it until for the next idle cycle or the next write access. When read continues, it is given priority to read.

41.3.8 ECC encode specification

The following table shows ECC encoding specifications. Add the ECC cord (eout [6:0]) formed by the following calculating formula to higher 7 bits (din [38:32]) of write data and write in it at SRAM.

Table 41.7 ECC encode

ECC code	calculation formula
eout[6]	(din[13] ^ din[12] ^ din[11] ^ din[10] ^ din[9] ^ din[8] ^ din[7] ^ din[6] ^ din[5] ^ din[4] ^ din[3] ^ din[2] ^ din[1] ^ din[0])
eout[5]	(din[23] ^ din[22] ^ din[21] ^ din[20] ^ din[19] ^ din[18] ^ din[17] ^ din[16] ^ din[15] ^ din[14] ^ din[3] ^ din[2] ^ din[1] ^ din[0])
eout[4]	(din[29] ^ din[28] ^ din[27] ^ din[26] ^ din[25] ^ din[24] ^ din[17] ^ din[16] ^ din[15] ^ din[14] ^ din[7] ^ din[6] ^ din[5] ^ din[4])
eout[3]	(din[31] ^ din[30] ^ din[26] ^ din[25] ^ din[24] ^ din[20] ^ din[19] ^ din[18] ^ din[14] ^ din[10] ^ din[9] ^ din[8] ^ din[4] ^ din[0])
eout[2]	(din[31] ^ din[30] ^ din[28] ^ din[27] ^ din[24] ^ din[22] ^ din[21] ^ din[18] ^ din[15] ^ din[12] ^ din[11] ^ din[8] ^ din[5] ^ din[1])
eout[1]	~(din[30] ^ din[29] ^ din[27] ^ din[25] ^ din[23] ^ din[21] ^ din[19] ^ din[16] ^ din[13] ^ din[11] ^ din[9] ^ din[6] ^ din[2] ^ din[0])
eout[0]	~(din[31] ^ din[29] ^ din[28] ^ din[26] ^ din[23] ^ din[22] ^ din[20] ^ din[17] ^ din[13] ^ din[12] ^ din[10] ^ din[7] ^ din[3] ^ din[0])

Note: eout[6:0] = ECC code, din[31:0] = write data

42. Standby SRAM

42.1 Overview

An on-chip SRAM is provided to retain data in Deep Software Standby mode. Table 42.1 lists the Standby SRAM specifications.

Table 42.1 Standby SRAM specifications

Item	Description
SRAM capacity	1 KB
SRAM address	0x2800_0000 to 0x2800_03FF
Access	Standby SRAM clock is the same clock as the PCLKB. See section 42.3.5. Access Cycle for details.
Data retention function	Data can be retained in Deep Software Standby mode. See section 42.3.1. Data Retention for details.
parity	Even parity (data: 8 bits, parity: 1 bit)
Module-stop function	Module-stop state can be set to reduce power consumption. See section 42.3.2. Setting for the Module-stop Function for details.
Security	Permits the read and write operations to Standby RAM following TrustZone Filter function. See section 42.3.4. TrustZone Filter function for details.

42.2 Register Descriptions

42.2.1 STBRAMSAR : Standby RAM memory Security Attribution Register

Base address: CPSCU = 0x4000_8000

Offset address: 0x014

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

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

Bit	Symbol	Function	R/W
3:0	NSBSTBR[3:0]	Security attributes of each region for Standby RAM 0x0: Region7-0 are all Secure. 0x1: Region7 is Non-secure. Region6-0 are Secure 0x2: Region7-6 are Non-secure. Region5-0 are Secure. 0x3: Region7-5 are Non-secure. Region4-0 are Secure. 0x4: Region7-4 are Non-secure. Region 3-0 are Secure. 0x5: Region7-3 are Non-secure. Region 2-0 are Secure. 0x6: Region7-2 are Non-secure. Region 1-0 are Secure. 0x7: Region7-1 are Non-Secure. Region0 is Secure. Others: Region7-0 are all Non-Secure.	R/W
31:4	—	This bit is read as 1.	R

Note: Only secure access can write to this register. Both secure access and Non-secure read access are allowed but Non-secure write access is not allowed and TrustZone access error is not generated.

Note: This register is write-protected by PRCR register.

NSBSTBR[3:0] bit (Security attributes of each region for Standby RAM)

Standby RAM is divided into 8 regions. Each region can be set as Secure or Non-secure state with NSBSTBR[3:0]

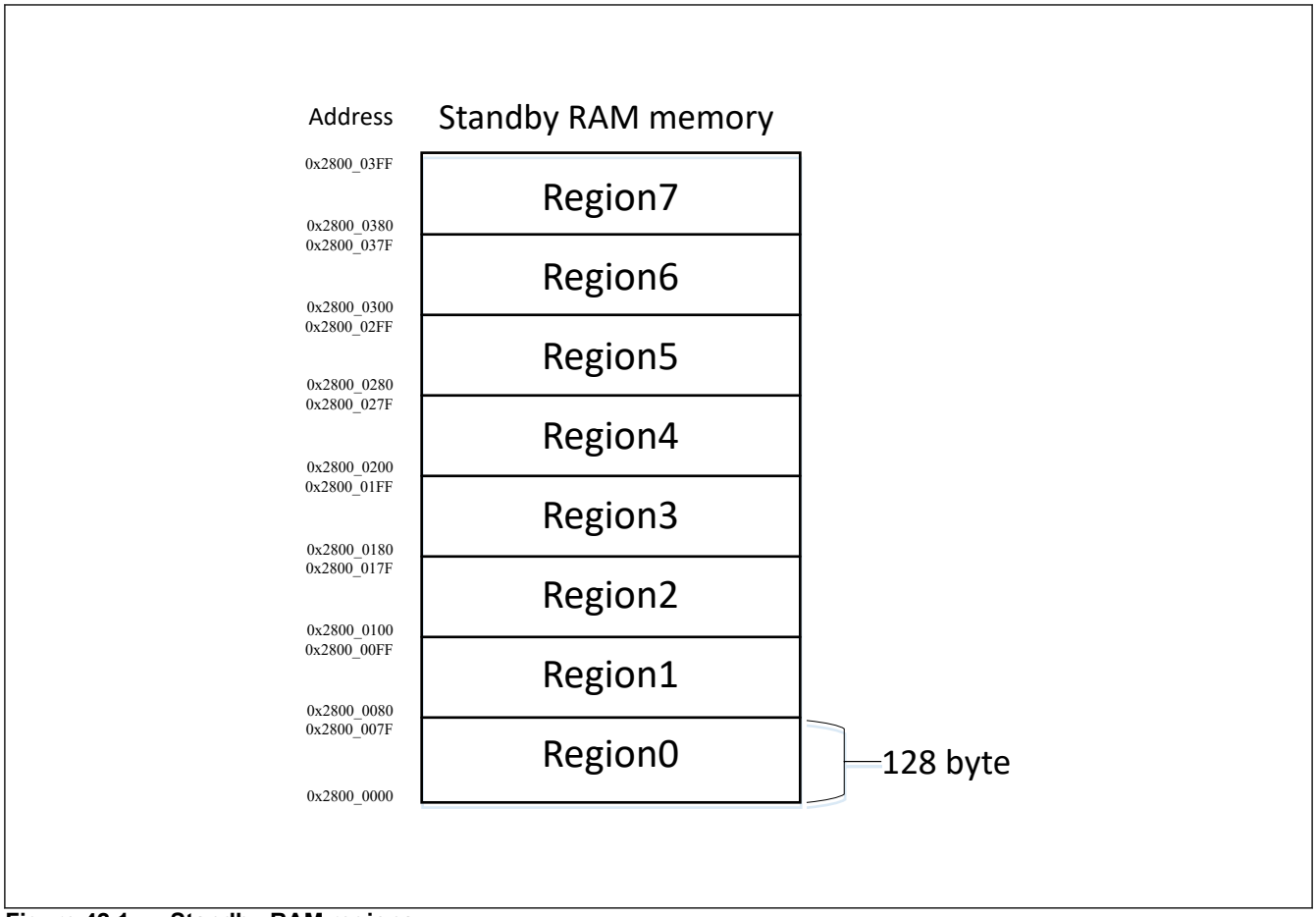


Figure 42.1 Standby RAM regions

42.3 Operation

42.3.1 Data Retention

The power supply to the Standby SRAM in Deep Software Standby mode is enabled by the DPSBYCR.DEEPCUT[1:0] bits. If the DPSBYCR.DEEPCUT[1:0] bits are set to 00b, data in the Standby SRAM is retained in Deep Software Standby mode. For details on the DPSBYCR.DEEPCUT[1:0] bits, see section 10, Low Power Modes.

42.3.2 Setting for the Module-stop Function

Power consumption can be reduced by setting module stop control register A (MSTPCRA) to stop supply of the clock signal to SRAM.

If the Standby SRAM bit in MSTPCRA is set to 1, supply of the clock signal to the Standby SRAM is stopped.

The Standby SRAM is thus placed in the module-stop state by stopping supply of the clock signals. The Standby SRAM operates after a reset.

The Standby SRAM is not accessible if it is in the module-stop state. A transition to the module-stop state should not be made while access to the standby SRAM is in progress.

For details on the MSTPCRA register, see section 10, Low Power Modes.

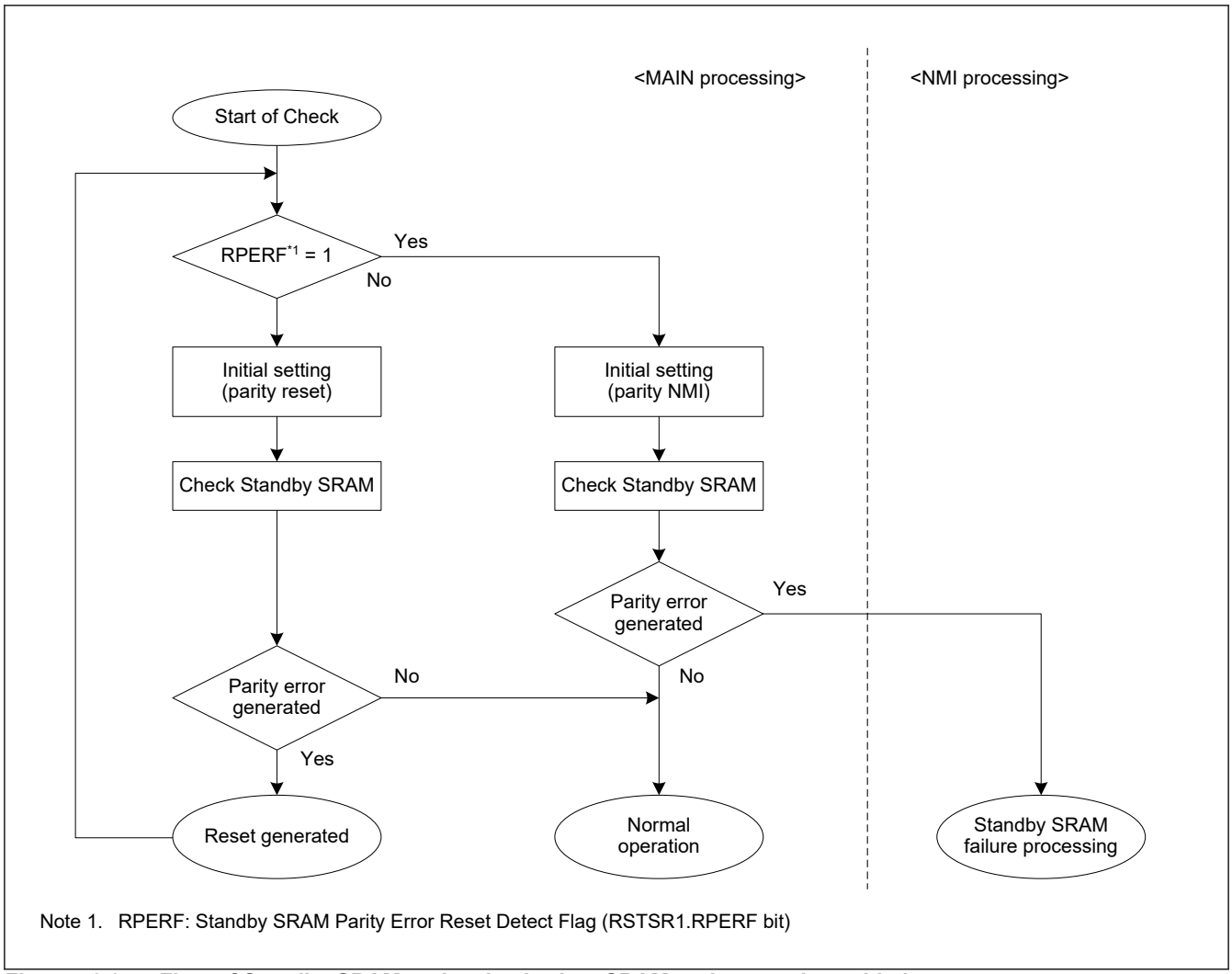
42.3.3 Parity Calculation Function

The IEC60730 standard requires the checking of SRAM data. When data is written, a parity bit is added to every 8-bit data in the Standby SRAM which has 32-bit data width, and when data is read, the parity is checked. When a parity error occurs, a parity error notification is generated. This function can also be used to trigger a reset.

The parity-error notification can be specified as a non-maskable interrupt or a reset in the OAD bit of the SRAM.PARIOAD register. When the OAD bit is set to 1, a parity error is output to the reset function. When the OAD bit is set to 0, a parity error is output to the ICU as a non-maskable interrupt.

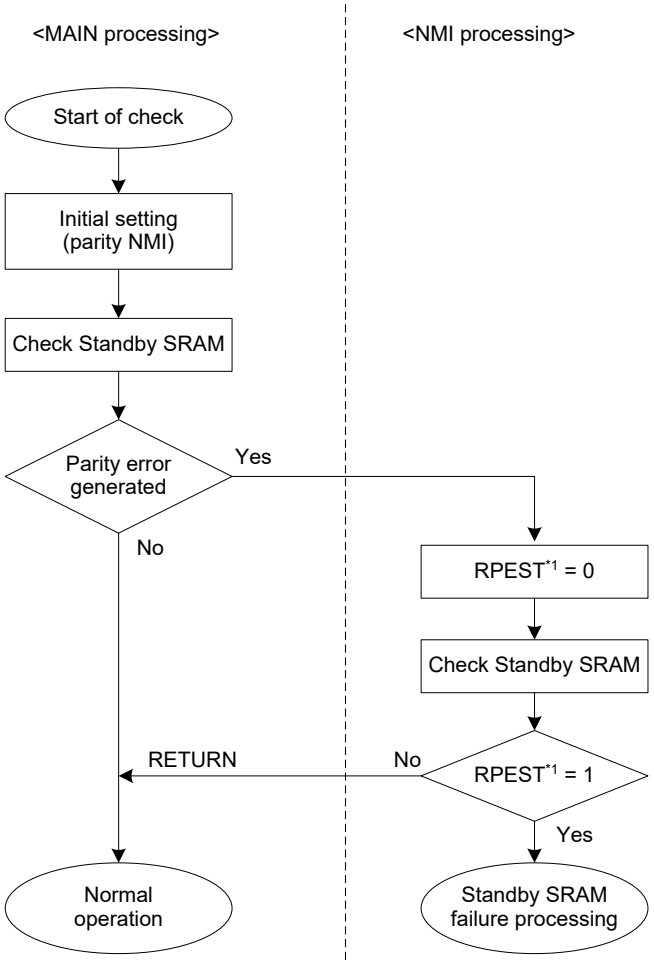
Parity errors often occur because of noise. To confirm whether the cause of the parity error is noise or corruption, follow the parity check flows shown in Figure 42.2 and Figure 42.3.

When a read access is executed in a row after a write access, read access is executed with priority. Therefore, during initialization, do not perform the read access in a row after the write access.



Note 1. RPERF: Standby SRAM Parity Error Reset Detect Flag (RSTSR1.RPERF bit)

Figure 42.2 Flow of Standby SRAM parity check when SRAM parity reset is enabled



Note 1. RPEST: SRAM Parity Error Interrupt Status Flag (NMISR.RPEST bit)

Figure 42.3 Flow of Standby SRAM parity check when SRAM parity interrupt is enabled

42.3.4 TrustZone Filter function

There is only one type of TrustZone Filter function for Standby SRAM and that is, TrustZone Filter for SRAM memory protection

42.3.4.1 TrustZone Filter for Standby SRAM Memory Protection

Standby SRAM memory can be divided into 8 regions, 128 bytes each with a Security Attribution (SA) to be protected from Non-secure access. When SA indicates that the region in Standby SRAM is secure status, non-secure access cannot overwrite them because TrustZone Filter detects finds an error and protects the write access.

Table 42.2 Security Attribution and Access status

SA	Access status	Write access	Read access
Secure	Secure	Permit	Permit
	Non-secure	TrustZone Filter error - Protected	TrustZone Filter error - Read data is 0
Non-secure	Secure	Permit	Permit
	Non-secure	Permit	Permit

When TrustZone Filter error for Standby SRAM access occurs, no error notification and no error response occurs.

42.3.5 Access Cycle

Number of cycles from the CPU.

For Standby SRAM, cache always has non-cacheable access.

Table 42.3 Standby SRAM (parity area 0x2800_0000 to 0x2800_03FF)

	Read cycle		Write cycle	
	Word access	**Halfword/Byte access**	**Word access**	**Halfword/Byte access**
ICLK ≥ PCLKB	Min.: 2 ICLK + 2 PCLKB Max.: (n + 1) ICLK + 2 PCLKB		Min.: 1 ICLK + 1 PCLKB Max.: n ICLK + 1 PCLKB	

Note: When the frequency ratio of ICLK : PCLKB is n : 1.

42.4 Usage Notes

42.4.1 Instruction Fetch from the Standby SRAM Area

When using Standby SRAM to operate a program, initialize the Standby SRAM area so that the CPU can correctly prefetch data. A parity error might occur if the CPU prefetches from an area that is not initialized. Initialize the additional 12-byte area from the end address of the program with the 4-byte boundary. Renesas recommends using the NOP instruction for data initialization.

43. Flash Memory

This MCU incorporates code flash memory, data flash memory, and option-setting memory. The code flash memory stores instructions and operands, and the data flash memory stores data. For option-setting memory, see section 6, Option-Setting Memory.

43.1 Overview

Table 43.1 lists the specifications of the flash memory, and Figure 43.1 is block diagrams of the flash memory related modules.

The I/O pins used in boot mode, see Table 43.27.

The FCU (flash control unit) controls programming and erasure of the flash memory. The FACI (flash application command interface) controls the FCU according to the specified FACI commands.

Regarding the configuration of the code flash memory, see Figure 43.2, and for the configuration of the data flash memory, see Figure 43.3.

Table 43.1 Specifications of flash memory (1 of 2)

Item	Code flash memory	Data flash memory
Memory capacity	User area: 512 Kbytes max	Data area: 16 Kbytes
Read cycle	See section 43.16.3. Access Cycle	See section 43.16.3. Access Cycle
Value after erasure	0xFF	Undefined
Programming/erasing method	• Programming and erasing the code flash memory and data flash memory, and programming the option-setting memory are handled by the FACI commands specified in the FACI command issuing area (0x407E_0000) (self-programming). • Programming/erasure through transfer by a serial-programmer via a serial interface (serial programming)	
Protection	Protects against erroneous rewriting of the flash memory	
Background operations (BGOs)	• The data flash memory can be read while the code flash memory is being programmed or erased. • The code flash memory can be read while the data flash memory is being programmed or erased.	
Units of programming and erasure	• Units of programming for the user area: 128 bytes • Units of erasure for the user area: Block units	• Unit of programming for the data area:4/8/16 bytes • Unit of erasure for the data area:64/128/256 bytes
Other functions	Interrupts can be accepted during self-programming. In the initial settings of this MCU, an expansion area of the option-setting memory can be set.	
On-board programming (three types)	Programming/erasure in boot mode (for the SCI interface) • The asynchronous serial interface (SCI9) is used. • The transfer rate is adjusted automatically. Programming/erasure in On-chip debug mode • JTAG or SWD interface is used Programming and erasure by self-programming • This allows code flash memory programming/erasure without resetting the system.	
Unique ID	A 16-byte ID code provided for each MCU	
FACI command	Program : 128 bytes Block erase: 1 block (8 KB or 32 KB) P/E suspend P/E resume Forced Stop Status Clear Configuration set (16 bytes)	Program: 4/8/16 bytes Block Erase: 1 block (64 bytes) Multi Block Erase: 64/128/256 bytes P/E suspend P/E resume Forced Stop Blank Check: 4 bytes to data flash memory capacity Status Clear

Table 43.1 Specifications of flash memory (2 of 2)

Item	Code flash memory	Data flash memory
Security function	Protects against illicit tampering with or reading out of data in flash memory Startup area select setting protection ● BTFLG and FSUACR registers are protected by the FSPR bit. Permanent block protect setting protection ● Code flash memory is permanently protected from programming/erasure operation by the permanent block protect function. Flash memory protection for TrustZone ● Protection for flash memory area (P/E) ● Protection for flash memory area (read) ● Protection for register ● Protection during FACI command operation. ● Code flash P/E mode entry protection	
Safety function	Software protection ● FACI command protection by FENTRYR register. ● Flash memory is protected by FWEPROR register ● The user area is protected by the block protect setting Error protection ● Error is detected when unintended commands or prohibited settings occur. The FACI command is not accepted after an error detection. Boot area protection ● The start-up area select function allows customer to safely update the boot firmware. The size of the start-up area is 8 KB.	
Interrupt request	● FRDYI (flash sequencer ready (processing end)) : Enabled by FRDYIE bit. ● FIFERR (flash sequencer error) : Enabled by CFAEIE/CMDLKIE/DFAEIE bits	
Address conversion	● Start-up area select function is supported	

Figure 43.1 shows how modules related to flash memory can be configured. The flash sequencer is configured with the FCU and FACI. The FCU executes basic control for rewriting of the flash memory. The FACI receives FACI commands using peripheral bus, and controls FCU operations accordingly.

In response to a reset, the FACI transfers data from the flash memory to the option byte storage registers.

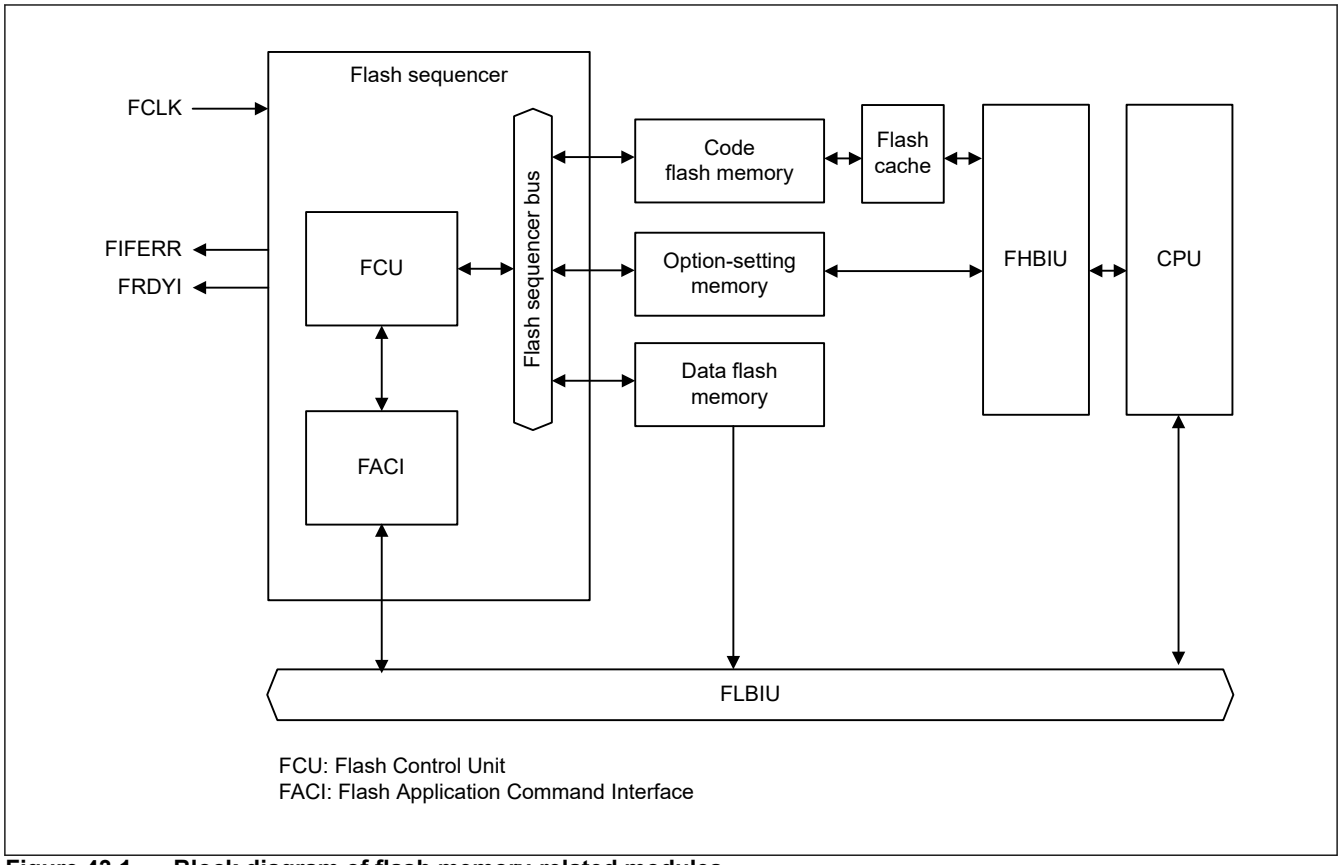


Figure 43.1 Block diagram of flash memory-related modules

43.2 Structure of Memory

Figure 43.2 shows the memory map of code flash memory.

The user area of the code flash memory in this MCU is divided into 8- and 32-Kbyte blocks, which serve as the units of erasure.

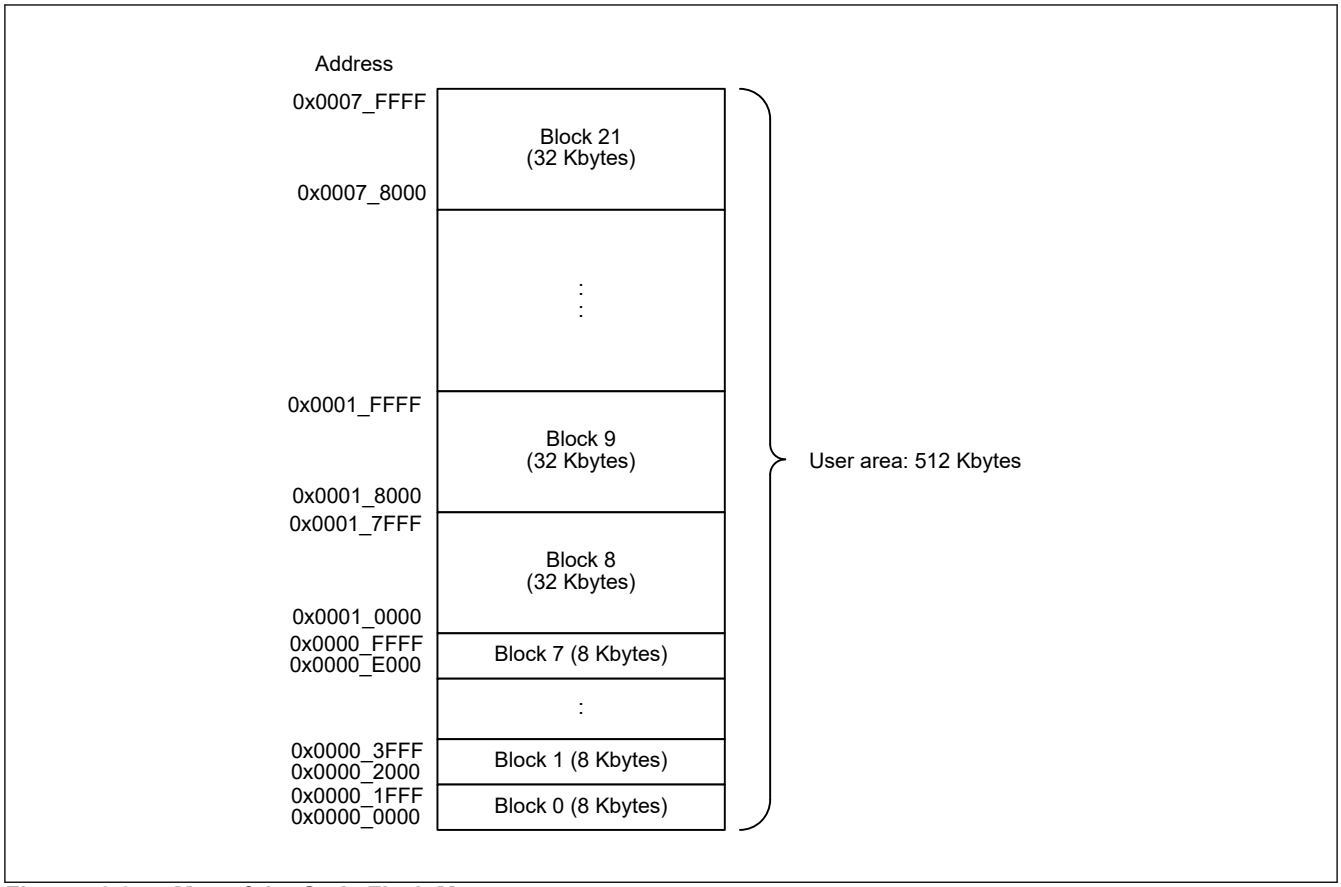


Figure 43.2 Map of the Code Flash Memory

Table 43.2 Read and programming/erasure address by product for the code flash memory

Product	Address	Number of blocks
512 Kbytes product	0x0000_0000 to 0x0007_FFFF	0 to 21
256 Kbytes product	0x0000_0000 to 0x0003_FFFF	0 to 13

The data area of the data flash memory in this MCU is divided into 64-byte blocks, with each being a unit for erasure. Figure 43.3 shows the mapping of the data flash memory.

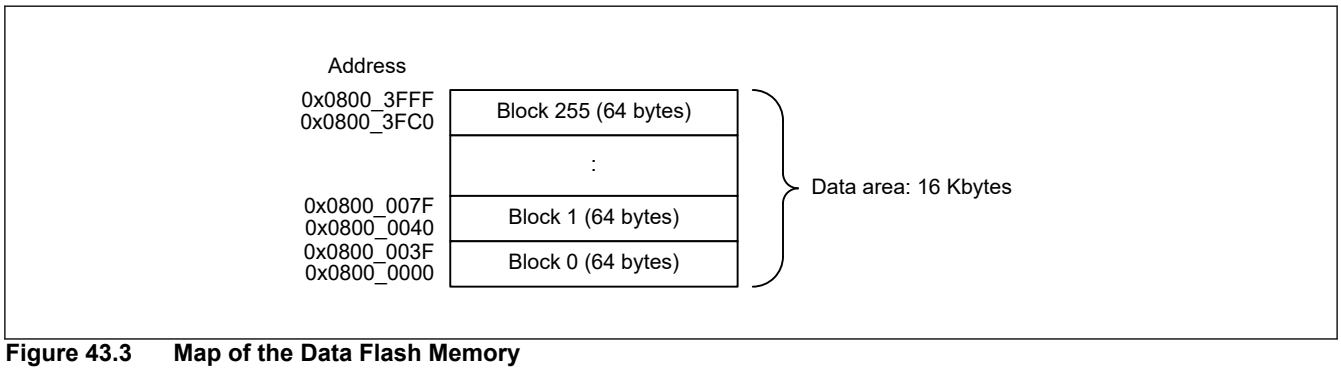


Figure 43.3 Map of the Data Flash Memory

43.3 Address Space

Using the hardware interface with flash memory requires access to all registers of the hardware, which is for issuing FACI commands. Table 43.3 provides information about the hardware interface.

Table 43.3 Information on the hardware interface area

Area	Address	Capacity
Area containing various registers of the hardware	See section 43.4. Register Descriptions.	See section 43.4. Register Descriptions.
FACI command-issuing area	0x407E_0000	4 bytes

For the address information of the flash memory, see Figure 43.2.

43.4 Register Descriptions

43.4.1 FCACHEE : Flash Cache Enable Register

Base address: FCACHE = 0x4001_C100

Offset address: 0x000

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

Bit	Symbol	Function	R/W
0	FCACHEEN	Flash Cache Enable 0: FCACHE is disabled 1: FCACHE is enabled	R/W
15:1	—	These bits are read as 0. The write value should be 0.	R/W

This register is not controlled by any security attribute register.

FCACHEEN bit (Flash Cache Enable)

FCACHEE.FCACHEEN bit enable and disables the function of Flash Cache of FCACHE1, FCACHE2 and FLPF.

FCACHEE.FCACHEEN bit dose not influence for FCACHEIV.FCACHEIV.

When FCACHE is enabled, it works for accesses marked as cacheable.

It is prohibited to disable FCACHE after enabling.

43.4.2 FCACHEIV : Flash Cache Invalidate Register

Base address: FCACHE = 0x4001_C100

Offset address: 0x004

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

Bit	Symbol	Function	R/W
0	FCACHEIV	Flash Cache Invalidate 0: Read: Do not invalidate. Write: The setting is ignored. 1: Invalidate FCACHE is invalidated.	R/W
15:1	—	These bits are read as 0. The write value should be 0.	R/W

This register is not controlled by any security attribute register.

FCACHEIV bit (Flash Cache Invalidate)

When 1 is written to FCACHEIV.FCACHEIV bit, the Flash cache data of FCACHE1, FCACHE2 and FLPF is invalidated.

Invalidate FCACHE with keeping FCACHE enabled after programming or erasing the code flash or the option setting memory.

43.4.3 FLWT : Flash Wait Cycle Register

Base address: FCACHE = 0x4001_C100

Offset address: 0x01C

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

Bit	Symbol	Function	R/W
2:0	FLWT[2:0]	Flash Wait Cycle 0 0 0: 0 wait (ICLK ≤ 120 MHz) 0 0 1: 1 wait (ICLK > 120 MHz) Others: Setting prohibited	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

FLWT[2:0] bits (Flash Wait Cycle)

The Flash Wait Cycle Register (FLWT) sets the access wait count for the flash memory.

For faster clock frequencies, set FLWT.FLWT before changing the clock frequency. For slower clock frequencies, set FLWT.FLWT after changing the clock frequency.

For information on the frequency setting, see section 8, Clock Generation Circuit.

43.4.4 FSAR : Flash Security Attribution Register

Base address: FCACHE = 0x4001_C100

Offset address: 0x040

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

Bit	Symbol	Function	R/W
0	FLWTSA	FLWT Security Attribution Target register : FLWT 0: Secure 1: Non-Secure	R/W
7:1	—	These bits are read as 1. The write value should be 1.	R/W
8	FCKMHZSA	FCKMHZ Security Attribution Target register : FCKMHZ 0: Secure 1: Non-Secure	R/W
15:9	—	These bits are read as 1. The write value should be 1.	R/W

Note: Only secure access can write to this register. Both secure access and Non-secure read access are allowed but Non-secure write access is not allowed and TrustZone access error is not generated.
Note: This register is write-protected by PRCR register.

Write access is invalid when PRCR.PRC4 bit is 0. See section 11, Register Write Protection.

FLWTSA bit (FLWT Security Attribution)

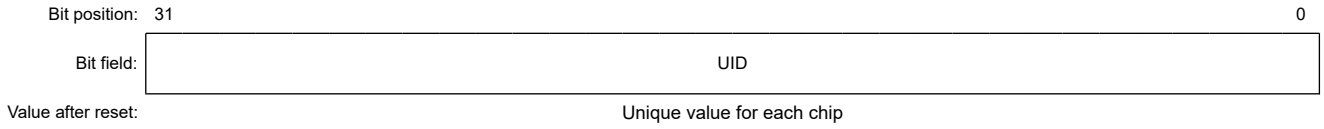
This bit sets the security attribute of FLWT.

FCKMHZSA bit (FCKMHZ Security Attribution)

This bit sets the security attribute of FCKMHZ.

43.4.5 UIDRn : Unique ID Registers n (n = 0 to 3)

Address: 0x0100_8190 + n × 4

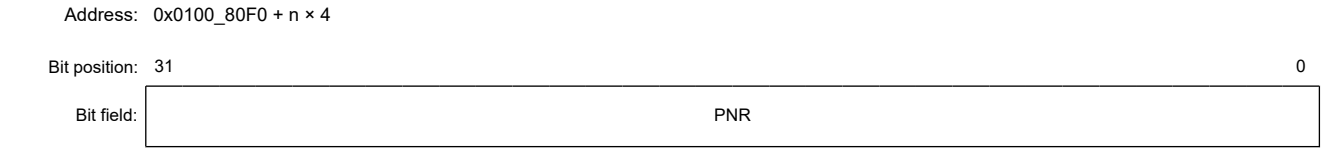


Bit	Symbol	Function	R/W
31:0	UID	Unique ID	R

The UIDRn is a read-only register that stores a 16-byte ID code (unique ID) for identifying the individual MCU. The UIDRn register should be read in 32-bit units. When reading by the signature request command of the serial programming interface, the data is read in order from the data with the large address. That is, the data in 0x0100_819F is read first, and in 0x0100_8190 is read last.

43.4.6 PNRn : Part Numbering Register n (n = 0 to 3)

Address: 0x0100_80F0 + n × 4

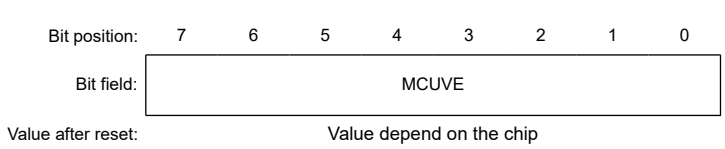


Bit	Symbol	Function	R/W
31:0	PNR	Part Number	R

The PNRn is a read-only register that stores a 16-byte part numbering. The PNRn register should be read in 32-bit units. Each byte corresponds to the ASCII code representation of the product part number as detailed in Table 1.12. The first character ("R", 0x52 in ASCII code) of the part number is stored in the byte with the smallest address (0x0100_80F0). When reading by the signature request command of the serial programming interface, the data is read in order from the data with the small address. That is, the data in 0x0100_80F0 is read first, and in 0x0100_80FF is read last.

43.4.7 MCUVER : MCU Version Register

Address: 0x0100_81B0



Bit	Symbol	Function	R/W
7:0	MCUVE	MCU Version	R

The MCUVER is a read-only register that stores a MCU version. The MCUVER register should be read in 8-bit units.

43.4.8 FWEPROR : Flash P/E Protect Register

Base address: SYSC = 0x4001_E000

Offset address: 0x416

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

Bit	Symbol	Function	R/W
1:0	FLWE[1:0]	Flash Programming and Erasure 0 0: Prohibits Program, Block Erase, Multi Block Erase, Blank Check, and Configuration set command processing. 0 1: Permits Program, Block Erase, Multi Block Erase, Blank Check, and Configuration set command processing. 1 0: Prohibits Program, Block Erase, Multi Block Erase, Blank Check, and Configuration set command processing. 1 1: Prohibits Program, Block Erase, Multi Block Erase, Blank Check, and Configuration set command processing.	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W

It is possible that Program, Block Erase, Multi Block Erase, Blank Check, and Configuration set command processing are prohibited by software.

The FWEPROR register is initialized by a reset from the following:

- All reset source
- Transition to Deep Software Standby mode
- Transition to Software Standby mode.

FLWE[1:0] bits (Flash Programming and Erasure)

The FLWE[1:0] bits are used to set the flash P/E protection. The value after reset is 10b.

If these bits are set to other than 01b that does not allow programming and erasure of the flash memory, the following commands cannot be executed. Issuing any of the following commands leads to setting of the FLWEERR bit in the FSTATR register to 1.

Program / Block Erase / Multi Block Erase / Blank Check / Configuration set command

43.4.9 FASTAT : Flash Access Status Register

Base address: FACI = 0x407F_E000

Offset address: 0x10

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

Bit	Symbol	Function	R/W
2:0	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
3	DFAE	Data Flash Memory Access Violation Flag 0: No data flash memory access violation has occurred 1: A data flash memory access violation has occurred.	R/W*1
4	CMDLK	Command Lock Flag 0: The flash sequencer is not in the command-locked state 1: The flash sequencer is in the command-locked state.	R
6:5	—	These bits are read as 0. The write value should be 0.	R/W
7	CFAE	Code Flash Memory Access Violation Flag 0: No code flash memory access violation has occurred 1: A code flash memory access violation has occurred.	R/W*1

Note 1. Only 0 can be written to clear the flag after 1 is read.

The FASTAT register indicates whether a code flash or data flash memory access violation has occurred. If any of the CFAE, CMDLK, and DFAE bits is set to 1, the flash sequencer enters the command-locked state (see section 43.11.2. Error Protection). To release it from the command-locked state, issue a status clear command or Forced Stop command to the flash sequencer.

DFAE bit (Data Flash Memory Access Violation Flag)

The DFAE bit indicates whether a data flash memory access violation occurred. When this bit is set to 1, the ILGLERR bit in the FSTATR register is set to 1, placing the flash sequencer in the command-locked state.

[Setting conditions]

FACI commands issued in the data flash P/E mode are as follows:

- The setting of the FSADDR or FEADDR register is the reserved portion of the data area
- FACI commands of non-secure access are issued while the setting of the FSADDR or FEADDR register is the secure region address.

[Clearing conditions]

- When this bit is written as 0 after it is set to 1
- When the flash sequencer starts to process the Status Clear or Forced Stop command.

CMDLK bit (Command Lock Flag)

The CMDLK bit indicates that the flash sequencer is in the command-locked state.

[Setting conditions]

- The flash sequencer detects an error and enters the command-locked state.

[Clearing conditions]

- When the flash sequencer starts to process the Status Clear or Forced Stop command.

CFAE bit (Code Flash Memory Access Violation Flag)

The CFAE bit indicates whether a code flash memory access violation has occurred. When this bit is set to 1, the ILGLERR bit in the FSTATR register is set to 1, placing the flash sequencer in the command-locked state.

[Setting conditions]

FACI commands issued in the code flash P/E mode are as follows:

- The setting of the FSADDR register is the reserved portion of the user area
- The Configuration set command is issued while the setting of the FSADDR register is from 0x0000A100 to 0x0000A2F0 in self-programming mode
- FACI commands of non-secure access are issued while the setting of the FSADDR register is the secure region address.

[Clearing conditions]

- When this bit is written as 0 after it is set to 1

- When the flash sequencer starts to process the Status Clear or Forced Stop command.

43.4.10 FAEINT : Flash Access Error Interrupt Enable Register

Base address: FACI = 0x407F_E000

Offset address: 0x14

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

Bit	Symbol	Function	R/W
2:0	—	These bits are read as 0. The write value should be 0.	R/W
3	DFAEIE	Data Flash Memory Access Violation Interrupt Enable 0: Generation of an FIFERR interrupt request is disabled when FASTAT.DFAE is set to 1 1: Generation of an FIFERR interrupt request is enabled when FASTAT.DFAE is set to 1.	R/W
4	CMDLKIE	Command Lock Interrupt Enable 0: Generation of an FIFERR interrupt request is disabled when FASTAT.CMDLK is set to 1 1: Generation of an FIFERR interrupt request is enabled when FASTAT.CMDLK is set to 1.	R/W
6:5	—	These bits are read as 0. The write value should be 0.	R/W
7	CFAEIE	Code Flash Memory Access Violation Interrupt Enable 0: Generation of an FIFERR interrupt request is disabled when FASTAT.CFAE is set to 1 1: Generation of an FIFERR interrupt request is enabled when FASTAT.CFAE is set to 1.	R/W

The FAEINT register enables or disables generation of a flash access error (FIFERR) interrupt request.

DFAEIE bit (Data Flash Memory Access Violation Interrupt Enable)

The DFAEIE bit enables or disables generation of an FIFERR interrupt request when a data flash memory access violation occurs, setting the DFAE bit in the FASTAT register to 1.

CMDLKIE bit (Command Lock Interrupt Enable)

The CMDLKIE bit enables or disables generation of an FIFERR interrupt request when the flash sequencer enters the command-locked state, setting the CMDLK bit in the FASTAT register to 1.

CFAEIE bit (Code Flash Memory Access Violation Interrupt Enable)

The CFAEIE bit enables or disables generation of an FIFERR interrupt request when a code flash memory access violation occur, setting the CFAE bit in the FASTAT register to 1.

43.4.11 FRDYIE : Flash Ready Interrupt Enable Register

Base address: FACI = 0x407F_E000

Offset address: 0x18

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

Bit	Symbol	Function	R/W
0	FRDYIE	Flash Ready Interrupt Enable 0: Generation of an FRDY interrupt request is disabled 1: Generation of an FRDY interrupt request is enabled.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

The FRDYIE register enables or disables generation of a flash ready (FRDY) interrupt request.

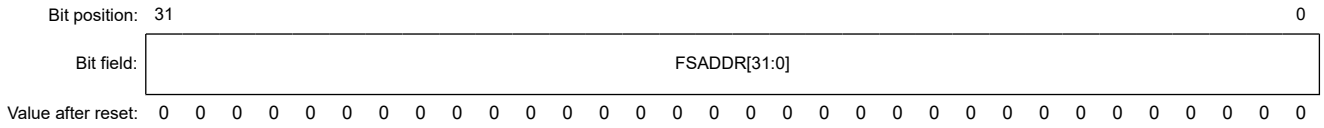
FRDYIE bit (Flash Ready Interrupt Enable)

The FRDYIE bit enables or disables generation of an FRDY interrupt request when the FRDY bit in the FSTATR register is changed from 0 to 1 on completion of processing by the flash sequencer of the Program, Block Erase, Multi Block Erase, Blank Check, and Configuration set command.

43.4.12 FSADDR : FACI Command Start Address Register

Base address: FACI = 0x407F_E000

Offset address: 0x30



Bit	Symbol	Function	R/W
31:0	FSADDR[31:0]	Start Address for FACI Command Processing	R/W*1

Note 1. These bits can be written when the FRDY bit in the FSTATR register is 1. Writing to these bits are ignored when the FRDY bit is 0. Note that b0 and b1 are read-only.

Table 43.4 FACI command addres boundary

Command	Address Boundary
Program (code flash memory)	128-byte
Program (data flash memory)	4, 8, 16 -byte
Block Erase (code flash memory)	8-KB or 32-KB
Block Erase (data flash memory)	64-byte
Multi Block Erase (data flash memory)	64-byte
Blank Check (data flash memory)	4-byte
Configuration set	16-byte

The FSADDR register specifies the address where the target area for command processing starts when the FACI command for Program, Block Erase, Multi Block Erase, Blank Check, or Configuration set is issued.

The FSADDR value is initialized when the SUINIT bit in the FSUINITR register is set to 1. It is also initialized by a reset.

FSADDR[31:0] bits (Start Address for FACI Command Processing)

The FSADDR[31:0] bits specify the start address for FACI command processing. Bits [31:24] are ignored in FACI command processing for the code flash memory. Bits [31:17] are ignored in FACI command processing for the data flash memory. Bits associated with the address bits of lower order than the address boundary listed in Table 43.4 are also ignored.

For information on the addresses of the code flash memory and the data flash memory, see section 43.2. Structure of Memory.

For information on the addresses of the configuration setting, see section 43.9.3.15. Configuration Set Command.

43.4.13 FEADDR : FACI Command End Address Register

Base address: FACI = 0x407F_E000

Offset address: 0x34

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

Bit	Symbol	Function	R/W
31:0	FEADDR[31:0]	End Address for FACI Command Processing	R/W*1

Note 1. These bits can be written when the FRDY bit in the FSTATR register is 1. Writing to these bits are ignored when the FRDY bit is 0. Note that bit [0] and bit [1] are read-only.

The FEADDR register specifies the end address of the target area for Multi Block Erase and Blank Check command processing. When incremental mode is selected as the addressing mode for Blank Checking (when FBCCNT.BCDIR = 0), the address specified in the FSADDR register should be equal to or smaller than the address in the FEADDR register. Conversely, the address in the FSADDR register should be equal to or larger than the address in the FEADDR register when decremental mode is selected as the addressing mode for Blank Check (i.e. when FBCCNT.BCDIR = 1). If the BCDIR, FSADDR, and FEADDR bit settings are inconsistent with the specified rules, the flash sequencer enters the command-locked state (see section 43.11.2. Error Protection).

The FEADDR value is initialized when the SUINIT bit in the FSUINITR register is set to 1. It is also initialized by a reset.

FEADDR[31:0] bits (End Address for FACI Command Processing)

The FEADDR[31:0] bits specify the end address for Multi Block Erase and Blank Check command processing. In command processing, bits 31 to 17 and any bits that do not reach the address boundaries listed in the section 43.4.12. FSADDR : FACI Command Start Address Register are ignored.

For information on the addresses of the flash memory, see section 43.2. Structure of Memory.

43.4.14 FMEPROT : Flash P/E Mode Entry Protection Register

Base address: FACI = 0x407F_E000

Offset address: 0x44

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

Bit	Symbol	Function	R/W
0	CEPROT	Code Flash P/E Mode Entry Protection 0: FENTRYC bit is not protected 1: FENTRYC bit is protected.	R/W*1 *2 *4
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code	W*3

Note 1. Writing to this bit is only possible when the FRDY bit in the FSTATR register is 1. Writing to this bit while the FRDY bit = 0 is ignored.
Note 2. Writing to this bit is only possible when 16 bits are written and the value written to the KEY bits is 0xD9.
Note 3. Written values are not retained by these bits (always read as 0x00).
Note 4. Only secure access can write to this register. Both secure access and non-secure read access are allowed. Non-secure write access is denied, but TrustZone access error is not generated.

CEPROT bit (Code Flash P/E Mode Entry Protection)

The CEPROT bit specifies the protection setting of the FRNTRYC bit in the FENTRYR register.

[Setting condition]

- 1 being written to the CEPROT bit while writing to FMEPROT is enabled.

[Clearing condition]

- 0 being written to the CEPROT bit while writing to FMEPROT is enabled.

43.4.15 FBPROT0 : Flash Block Protection Register

Base address: FACI = 0x407F_E000

Offset address: 0x78

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	KEY[7:0]								—	—	—	—	—	—	—	BPCN0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BPCN0	Block Protection for Non-secure Cancel 0: Block protection is enabled 1: Block protection is disabled.	R/W*1 *2
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code	W*3

Note 1. This bit can be written when the FRDY bit in the FSTATR register is 1. Writing to this bit is ignored when the FRDY bit is 0.
Note 2. Writing to this bit is only possible when 16 bits are written and the value written to the KEY[7:0] bits is 0x78.
Note 3. Written values are not retained by these bits (always read as 0x00).

The FBPROT0 register is used to disable the block protect function for non-secure. When the block protect setting is locked by the permanent block setting, it cannot be disabled by this register.

The FBPROT0 value is initialized when the SUINIT bit in the FSUINITR is set to 1, because the FENTRYR value is initialized to 0x0000. It is also initialized by a reset.

BPCN0 bit (Block Protection for Non-secure Cancel)

The BPCN0 bit disables the block protect setting for non-secure function.

[Setting condition]

- When the write-enabling conditions are satisfied and the FENTRYR is not 0x0000, write 1 to this bit.

[Clearing conditions]

- 8 bits being written to FBPROT0 while the FRDY bit is 1.
- A value other than 0x78 specified in the KEY bits and 16 bits are written to FBPROT0 while the FRDY bit is 1.
- 0 being written to the BPCN0 bit while writing to FBPROT0 is enabled.
- The FENTRYR register value is 0x0000.

43.4.16 FBPROT1 : Flash Block Protection for Secure Register

Base address: FACI = 0x407F_E000

Offset address: 0x7C

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	KEY[7:0]								—	—	—	—	—	—	—	BPCN1
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BPCN1	Block Protection for Secure Cancel 0: Block protection is enabled 1: Block protection is disabled.	R/W*1 *2
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code	W*3

Note 1. Writing to this bit is only possible when the FRDY bit in the FSTATR register is 1. Writing to this bit while FRDY bit = 0 is ignored.
Note 2. Writing to this bit is only possible when 16 bits are written and the value written to the KEY[7:0] bits is 0xB1.
Note 3. Written values are not retained by these bits (always read as 0x00).

The FBPROT1 register is used to disable the block protect function for secure developer. When the block protect setting is locked by the permanent block setting, it cannot be disabled by this register.

The FBPROT1 value is initialized when the SUINIT bit in the FSUINITR is set to 1, because the FENTRYR value is initialized to 0x0000. It is also initialized by a reset.

BPCN1 bit (Block Protection for Secure Cancel)

The BPCN1 bit disables the block protect setting for secure function.

[Setting condition]

- When the write-enabling conditions are satisfied and the FENTRYR is not 0x0000, write 1 to BPCN1.

[Clearing conditions]

- 8 bits being written to FBPROT1 while the FRDY bit is 1.
- A value other than 0xB1 specified in the KEY bits and 16 bits are written to FBPROT1 while the FRDY bit is 1.
- 0 being written to the BPCN1 bit while writing to FBPROT1 is enabled.
- The FENTRYR register value is 0x0000.

43.4.17 FSTATR : Flash Status Register

Base address: FACI = 0x407F_E000

Offset address: 0x80

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	FRDY	ILGLERR	ERSERR	PRGERR	SUSRDY	DBFULL	ERSSPD	PRGSPD	—	FLWEERR	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	—	These bits are read as 0. The write value should be 0.	R/W
6	FLWEERR	Flash Write/Erase Protect Error Flag 0: An error has not occurred 1: An error has occurred.	R
7	—	These bits are read as 0. The write value should be 0.	R/W
8	PRGSPD	Programming Suspend Status Flag 0: The flash sequencer is not in the programming suspension processing state or programming suspended state 1: The flash sequencer is in the programming suspension processing state or programming suspended state.	R

Bit	Symbol	Function	R/W
9	ERSSPD	Erasure Suspend Status Flag 0: The flash sequencer is not in the erasure suspension processing state or the erasure suspended state 1: The flash sequencer is in the erasure suspension processing state or the erasure suspended state.	R
10	DBFULL	Data Buffer Full Flag 0: The data buffer is empty 1: The data buffer is full.	R
11	SUSRDY	Suspend Ready Flag 0: The flash sequencer cannot receive P/E suspend commands 1: The flash sequencer can receive P/E suspend commands.	R
12	PRGERR	Programming Error Flag 0: Programming has completed successfully 1: An error has occurred during programming.	R
13	ERSERR	Erasure Error Flag 0: Erasure has completed successfully 1: An error has occurred during erasure.	R
14	ILGLERR	Illegal Command Error Flag 0: The flash sequencer has not detected an illegal FACI command or illegal flash memory access 1: The flash sequencer has detected an illegal FACI command or illegal flash memory access.	R
15	FRDY	Flash Ready Flag 0: Program, Block Erase, Multi Block Erase, P/E suspend, P/E resume, Forced Stop, Blank Check, or Configuration set command processing is in progress 1: None of the above is in progress.	R
19:16	—	These bits are read as 0. The write value should be 0.	R/W
20	OTERR	Other Error 0: A status clear or forced stop command processing is complete 1: An error has occurred.	R
21	SECERR	Security Error 0: A status clear or forced stop command processing is complete 1: An error has occurred.	R
22	FESETERR	FENTRY Setting Error 0: A status clear or forced stop command processing is complete 1: An error has occurred.	R
23	ILGCOMERR	Illegal Command Error 0: A status clear or forced stop command processing is complete 1: An error has occurred.	R
31:24	—	These bits are read as 0. The write value should be 0.	R/W

The FSTATR register indicates the state of the flash sequencer.

FLWEERR flag (Flash Write/Erase Protect Error Flag)

The FLWEERR flag indicates a violation of the flash memory overwrite protection setting in the FWEPROR register. When this flag is 1, the flash sequencer is in the command-locked state.

[Setting condition]

- An error has occurred.

[Clearing condition]

- The flash sequencer starts processing the Forced Stop command.

PRGSPD flag (Programming Suspend Status Flag)

The PRGSPD flag indicates that the flash sequencer is in the programming suspension processing state or programming suspended state.

[Setting condition]

- The flash sequencer starts processing in response to the programming suspend command.

[Clearing conditions]

- Reception of the P/E resume command by the flash sequencer (after write access to the FACI command-issuing area is complete)
- The flash sequencer starts processing the Forced Stop command.

ERSSPD flag (Erasure Suspend Status Flag)

The ERSSPD flag indicates that the flash sequencer is in the erasure suspension processing state or erasure suspended state.

[Setting condition]

- The flash sequencer starts processing in response to an erasure suspend command.

[Clearing condition]

- Reception of the P/E resume command by the flash sequencer (after write access to the FACI command-issuing area is complete)
- The flash sequencer starts processing of the Forced Stop command.

DBFULL flag (Data Buffer Full Flag)

The DBFULL flag indicates the state of the data buffer when the program command is issued. The flash sequencer incorporates a buffer for write data (data buffer). When data for writing to the flash memory are written to the FACI command-issuing area while the data buffer is full, the flash sequencer inserts a wait cycle in the peripheral bus.

[Setting condition]

- The data buffer becomes full while program commands are issued.

[Clearing condition]

- The data buffer becomes empty.

SUSRDY flag (Suspend Ready Flag)

The SUSRDY flag indicates whether the flash sequencer can receive a P/E suspend command.

[Setting condition]

- After starting programming/erasure processing, the flash sequencer enters a state in which P/E suspend commands can be received.

[Clearing conditions]

- Reception of the P/E suspend command or Forced Stop command by the flash sequencer (after write access to the FACI command-issuing area is complete)
- During programming or erasure, the flash sequencer enters the command-locked state
- Programming or erasure has completed.

PRGERR flag (Programming Error Flag)

The PRGERR flag indicates the result of programming of the flash memory. When this flag is 1, the flash sequencer is in the command-locked state.

[Setting condition]

- An error has occurred during programming.

[Clearing condition]

- The flash sequencer starts processing of the Status Clear or Forced Stop command.

ERSERR flag (Erasure Error Flag)

The ERSERR flag indicates the result of erasure of the flash memory. When this flag is 1, the flash sequencer is in the command-locked state.

[Setting condition]

- An error has occurred during erasure.

[Clearing condition]

- The flash sequencer starts processing of the Status Clear or Forced Stop command.

ILGLERR flag (Illegal Command Error Flag)

The ILGLERR flag indicates that the flash sequencer has detected an illegal FACI command or flash memory access. If this flag is 1, the flash sequencer is in the command-locked state.

[Setting conditions]

- See section 43.11.2. Error Protection.

[Clearing condition]

- The flash sequencer starts processing of the Status Clear or Forced Stop command.

FRDY flag (Flash Ready Flag)

The FRDY flag indicates the command processing state of the flash sequencer.

[Setting conditions]

- The flash sequencer completes command processing
- The flash sequencer receives a P/E suspend command and suspends programming of the flash memory
- The flash sequencer received the Forced Stop command and ended command processing.

[Clearing conditions]

- The flash sequencer received an FACI command
- For Program and Configuration setting, the first write access to the FACI command-issuing area
- For other commands, the last write access to the FACI command-issuing area.

OTERR flag (Other Error)

See Table 43.21. When this flag is 1, the flash sequencer is in the command-lock state.

[Setting condition]

- An error has occurred.

[Clearing condition]

- The status clear or forced stop command processing is complete.

SECERR flag (Security Error)

See Table 43.21. When this flag is 1, the flash sequencer is in the command-lock state.

[Setting condition]

- An error has occurred.

[Clearing condition]

- The status clear or forced stop command processing is complete.

FESETERR flag (FENTRY Setting Error)

See Table 43.21. When this flag is 1, the flash sequencer is in the command-lock state.

[Setting condition]

- An error has occurred.

[Clearing condition]

- The status clear or forced stop command processing is complete.

ILGCOMERR flag (Illegal Command Error)

See Table 43.21. When this flag is 1, the flash sequencer is in the command-lock state.

[Setting condition]

- An error has occurred.

[Clearing condition]

- The status clear or forced stop command processing is complete.

43.4.18 FENTRYR : Flash P/E Mode Entry Register

Base address: FACI = 0x407F_E000

Offset address: 0x84

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

Bit	Symbol	Function	R/W
0	FENTRYC	Code Flash P/E Mode Entry 0: Code flash is in read mode 1: Code flash is in P/E mode.	R/W*1*2
6:1	—	These bits are read as 0. The write value should be 0.	R/W
7	FENTRYD	Data Flash P/E Mode Entry 0: Data flash is in read mode 1: Data flash is in P/E mode.	R/W*1*2
15:8	KEY[7:0]	Key Code	W*3

Note 1. These bits can be written when the FRDY bit in the FSTATR register is 1. Writing to these bits are ignored when the FRDY bit is 0.
Note 2. Writing to these bits is only possible when 16 bits are written and the value written to the KEY[7:0] bits is 0xAA.
Note 3. Written values are not retained by these bits (always read 0x00).

FENTRYR is used to specify code flash P/E mode or data flash P/E mode. To specify the code flash P/E mode or data flash P/E mode so that the flash sequencer can receive FACI commands, set either the FENTRYD or FENTRYC bit to 1 to place the flash sequencer in P/E mode.

FENTRYR is initialized when the SUINIT bit in FSUINITR is set to 1. It is also initialized by a reset.

Note: Writing a value of 0XAA81 to this register causes the ILGLERR bit in the FSTATR register to be set to 1, resulting in the flash sequencer being placed in the command-locked state.

FENTRYC bit (Code Flash P/E Mode Entry)

The FENTRYC bit specifies P/E mode for the code flash memory.

[Setting condition]

- Write 1 to the FENTRYC bit while writing to FENTRYR is enabled and FENTRYR is 0x0000.

[Clearing conditions]

- Write 8 bits to FENTRYR while the FRDY bit is 1
- A value other than 0xAA is specified in the KEY[7:0] bits and 16 bits are written to FENTRYR while the FRDY bit is 1
- Write 0 to the FENTRYC bit while writing to FENTRYR is enabled
- Write to FENTRYR while writing is enabled and its value is other than 0x0000

- The protection of FMEPROT register is enabled.

FENTRYD bit (Data Flash P/E Mode Entry)

The FENTRYD bit specifies P/E mode for the data flash memory.

[Setting condition]

- Write 1 to the FENTRYD bit while writing to FENTRYR is enabled and FENTRYR is 0x0000.

[Clearing conditions]

- Write 8 bits to FENTRYR while the FRDY bit is 1
- Writing of 16 bits to FENTRYR with a value other than 0xAA specified for the KEY[7:0] bits while the FRDY bit is 1
- Write 0 to the FENTRYD bit while writing to FENTRYR is enabled
- Write to FENTRYR while writing is enabled and its value is other than 0x0000.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits control writing permission to the FENTRYD or FENTRYC bits.

43.4.19 FSUINITR : Flash Sequencer Setup Initialization Register

Base address: FACI = 0x407F_E000

Offset address: 0x8C

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

Bit	Symbol	Function	R/W
0	SUINIT	Set-Up Initialization 0: The FSADDR, FEADDR, FBPROT0, FBPROT1, FENTRYR, FBCCNT, and FCPSR flash sequencer setup registers keep their current values 1: The FSADDR, FEADDR, FBPROT0, FBRPOT1, FENTRYR, FBCCNT, and FCPSR flash sequencer setup registers are initialized.	R/W*1*2
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code	W*3

Note 1. This bit can be written when the FRDY bit in the FSTATR register is 1. Writing to this bit is ignored when the FRDY bit is 0.
Note 2. Writing to these bits is only possible when 16 bits are written and the value written to the KEY[7:0] bits is 0x2D.
Note 3. Written values are not retained by these bits (always read 0x00).

FSUINITR is used for initialization of the flash sequencer setup.

SUINIT bit (Set-Up Initialization)

The SUINIT bit initializes the following flash sequencer setup registers:

- FSADDR
- FEADDR
- FBPROT0
- FBPROT1
- FENTRYR
- FBCCNT
- FCPSR.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits control writing permission to the SUINIT bit.

43.4.20 FCMDR : FACI Command Register

Base address: FACI = 0x407F_E000

Offset address: 0xA0

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

Bit	Symbol	Function	R/W
7:0	PCMDR[7:0]	Pre-command Flag The command just before the latest command is stored.	R
15:8	CMDR[7:0]	Command Flag The latest command is stored.	R

FCMDR records the two most recent commands accepted by the flash sequencer.

PCMDR[7:0] bits (Pre-command Flag)

The PCMDR[7:0] bits indicate the command received immediately before the latest command received by the flash sequencer.

CMDR[7:0] bits (Command Flag)

The CMDR[7:0] bits indicate the latest command received by the flash sequencer.

Table 43.5 States of FCMDR after receiving commands

Command	CMDR	PCMDR
Program	0xE8	Previous command
Block erase	0xD0	0x20
Multi block erase	0xD0	0x21
P/E suspend	0xB0	Previous command
P/E resume	0xD0	Previous command
Status Clear	0x50	Previous command
Forced Stop	0xB3	Previous command
Blank Check	0xD0	0x71
Configuration set	0x40	Previous command

43.4.21 FBCCNT : Blank Check Control Register

Base address: FACI = 0x407F_E000

Offset address: 0xD0

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

Bit	Symbol	Function	R/W
0	BCDIR	Blank Check Direction 0: Blank checking is executed from the lower addresses to the higher addresses (incremental mode) 1: Blank checking is executed from the higher addresses to the lower addresses (decremental mode).	R/W

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

FBCCNT specifies the addressing mode in processing the Blank Check command. FBCCNT is initialized when the SUINIT bit in FSUINITR is set to 1. It is also initialized by a reset.

BCDIR bit (Blank Check Direction)

The BCDIR bit specifies the addressing mode for Blank Check.

43.4.22 FBCSTAT : Blank Check Status Register

Base address: FACI = 0x407F_E000

Offset address: 0xD4

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

Bit	Symbol	Function	R/W
0	BCST	Blank Check Status Flag 0: The target area is in the non-programmed state, that is, the area has been erased but has not yet been reprogrammed 1: The target area has been programmed with 0s or 1s.	R
7:1	—	These bits are read as 0. The write value should be 0.	R/W

FBCSTAT stores the results of checking in response to the Blank Check command.

BCST flag (Blank Check Status Flag)

The BCST flag indicates the results of checking in response to the Blank Check command.

43.4.23 FPSADDR : Data Flash Programming Start Address Register

Base address: FACI = 0x407F_E000

Offset address: 0xD8

Bit position:	31														17	16														0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PSADR[16:0]														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
16:0	PSADR[16:0]	Programmed Area Start Address The address of the first programmed area	R
31:17	—	These bits are read as 0. The write value should be 0.	R/W

FPSADDR indicates the address of the first programmed area found in processing of the Blank Check command.

PSADR[16:0] bits (Programmed Area Start Address)

The PSADR[16:0] bits indicate the address of the first programmed area found in processing of the Blank Check command. The address is an offset from the address where the data flash memory starts. The setting of these bits is only valid if the BCST bit in the FBCSTAT register is 1 and while the FRDY bit in the FSTATR register is 1. When the BCST bit in the FBCSTAT register is 0, the PSADR[16:0] bits hold the address produced by the previous check.

43.4.24 FSUASMON : Flash Startup Area Select Monitor Register

Base address: FACI = 0x407F_E000

Offset address: 0xDC

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

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

Bit	Symbol	Function	R/W
14:0	—	These bits are read as 0. The write value should be 0.	R
15	FSPR	Protection Programming Flag to set Boot Flag and Startup Area Control 0: Protected state 1: Non-protected state.	R
30:16	—	These bits are read as 0. The write value should be 0.	R
31	BTFLG	Flag of Startup Area Select for Boot Swap 0: The startup area is the alternate block (block 1) 1: The startup area is the default block (block 0).	R

FSPR bit (Protection Programming Flag to set Boot Flag and Startup Area Control)

The FSPR bit indicates the protection state against the configuration set command for the BTFLG bit, and FSUACR Register.

In response to a reset or configuration set command, the FACI transfers data from flash memory to this register.

BTFLG bit (Flag of Startup Area Select for Boot Swap)

The BTFLG bit indicates whether the address of the startup area is exchanged for the boot swap function or not.

In response to a reset or configuration set command, the FACI transfers data from flash memory to this register.

43.4.25 FCPSR : Flash Sequencer Processing Switching Register

Base address: FACI = 0x407F_E000

Offset address: 0xE0

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

Bit	Symbol	Function	R/W
0	ESUSPMD	Erasure Suspend Mode 0: Suspension priority mode 1: Erasure priority mode.	R/W
15:1	—	These bits are read as 0. The write value should be 0.	R/W

FCPSR selects the erasure suspension mode. FCPSR is initialized when the SUINIT bit in FSUINITR is set to 1. It is also initialized by a reset.

ESUSPMD bit (Erasure Suspend Mode)

The ESUSPMD bit selects the erasure suspension mode when a P/E suspend command is issued while the flash sequencer is executing erasure processing (see section 43.9.3.10. P/E Suspend Command). This bit should be set before issuing Block Erase or Multi Block Erase command.

43.4.26 FPCKAR : Flash Sequencer Processing Clock Notification Register

Base address: FACI = 0x407F_E000

Offset address: 0xE4

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

Bit	Symbol	Function	R/W
7:0	PCKA[7:0]	Flash Sequencer Operating Clock Notification These bits are used to set the operating frequency of the flash sequencer while processing FACI commands.	R/W*1*2
15:8	KEY[7:0]	Key Code	W*3

Note 1. This bit can be written when the FRDY bit in the FSTATR register is 1. Writing to this bit is ignored when the FRDY bit is 0.
Note 2. Writing to these bits is only possible when 16 bits are written and the value written to the KEY[7:0] bits is 0x1E.
Note 3. Written values are not retained by these bits (always read 0x00).

FPCKAR specifies the operating frequency of the flash sequencer while processing FACI commands. The highest operating frequency for the given product is set as the initial value.

PCKA[7:0] bits (Flash Sequencer Operating Clock Notification)

The PCKA[7:0] bits specify the operating frequency of the flash sequencer while processing FACI commands. Set the desired frequency for these bits before issuing an FACI command. Specifically, convert the frequency in MHz to a binary number and set it for these bits.

Example:

Frequency is 35.9 MHz (PCKA = 0x24)

Round up the first decimal place of 35.9 MHz to a whole number (= 36) and convert it into a binary number.

If the value set in these bits is smaller than the actual operating frequency of the flash sequencer, the flash memory programming/erasure characteristics cannot be guaranteed. If the value set in these bits is greater than the actual operating frequency of the flash sequencer, the flash memory programming/erasure characteristics can be guaranteed but the FACI command processing time such as the time programming/erasure takes will increase. The minimum FACI command processing time is obtained when the operating frequency of the flash sequencer is the same as the PCKA value.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits control writing permission to the PCKA bit.

43.4.27 FSUACR : Flash Startup Area Control Register

Base address: FACI = 0x407F_E000

Offset address: 0xE8

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

Bit	Symbol	Function	R/W
1:0	SAS[1:0]	Startup Area Select 0 0: Startup area is selected by BTFLG bit 0 1: Startup area is selected by BTFLG bit 1 0: Startup area is temporarily switched to the default area (block 0) 1 1: Startup area is temporarily switched to the alternate area (block 1).	R/W*1 *3
7:2	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code	W*2

Note 1. Following described the write condition of thisese bits (these conditions are required at the same time).
1. Access size to this register is 16 bits
2. The value of KEY[7:0] is 0x66
3. The FSPR bit is 1.

Note 2. Written values are not retained by these bits (always read 0x00).

Note 3. Only secure access can write to this register. Both secure access and non-secure read access are allowed. Non-secure write access is denied, but TrustZone access error is not generated.

FSUACR sets the startup area for the boot swap function.

SAS[1:0] bits (Startup Area Select)

The SAS[1:0] bits select the startup area. Three methods are available for changing the startup area.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits control writing permission to the SAS [1:0] bits.

43.4.28 FCKMHZ : Data Flash Access Frequency Register

Base address: FLAD = 0x407F_C000

Offset address: 0x40

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

Bit	Symbol	Function	R/W
7:0	FCKMHZ[7:0]	Data Flash Access Frequency Register These bits optimize the speed of reading the data flash memory.	R/W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

This register optimizes the speed of reading the data flash memory.

Set the frequency of the peripheral module clock (FCLK) of internal peripheral bus which is the clock for access to the data flash memory, in MHz units. For example, 35.9 MHz should be rounded up and set the frequency to 36. Number of cycles required for access to the data flash memory are inserted according to the frequency. When changing the frequency of the FCLK, follow the procedure below to modify the value of the data flash access frequency register (FCKMHZ) in either of the following ways according to whether operation is at a lower frequency before or after the change.

- When changing the speed from low to high: Modify FCKMHZ. After confirming the change by reading FCKMHZ, change the frequency.
- When changing the speed from high to low: Change the frequency. After the frequency is changed, modify FCKMHZ.

43.5 Flash Cache

43.5.1 Feature of flash cache

The FCACHE (Flash Cache) speeds up read access from bus master to the flash memory. The FCACHE includes:

- FCACHE1, for CPU instruction fetches
- FCACHE2, for CPU operand access
- FLPF, for the prefetch access in CPU instruction fetches

Table 43.6 Flash Cache 1 (FCACHE1) overview

Cache Target Region	0x0000_0000 - 0x007F_FFFF
Target Bus Master	CPU instruction Fetch
Capacity	256 Bytes
Associativity	8WAY set associative
	128 bits/entry (128 bit aligned data), 2 entries/way
Access Cycle	Cache Hit : 0 wait Cache Miss : wait number of Flash Wait Cycle Register

Table 43.7 Flash Cache 2 (FCACHE2) overview

Cache Target Region	0x0000_0000 - 0x007F_FFFF
Target Bus Master	CPU Operand Access
Capacity	16 Bytes
Associativity	Full Associative
	128 bits/entry (128 bit aligned data), 1 entry
Access Cycle	Cache Hit : 0 wait Cache Miss : wait number of Flash Wait Cycle Register

Table 43.8 Prefetch Buffer (FLPF) overview

Cache Target Region	0x0000_0000 - 0x007F_FFFF
Capacity	32 Bytes
Associativity	Full Associative
	128 bits/entry (128 bit aligned data), 2 entries
Request Address	Next address of previous CPU Instruction
Access Cycle	Cache Hit : 0 wait Cache Miss : wait number of Flash Wait Cycle Register

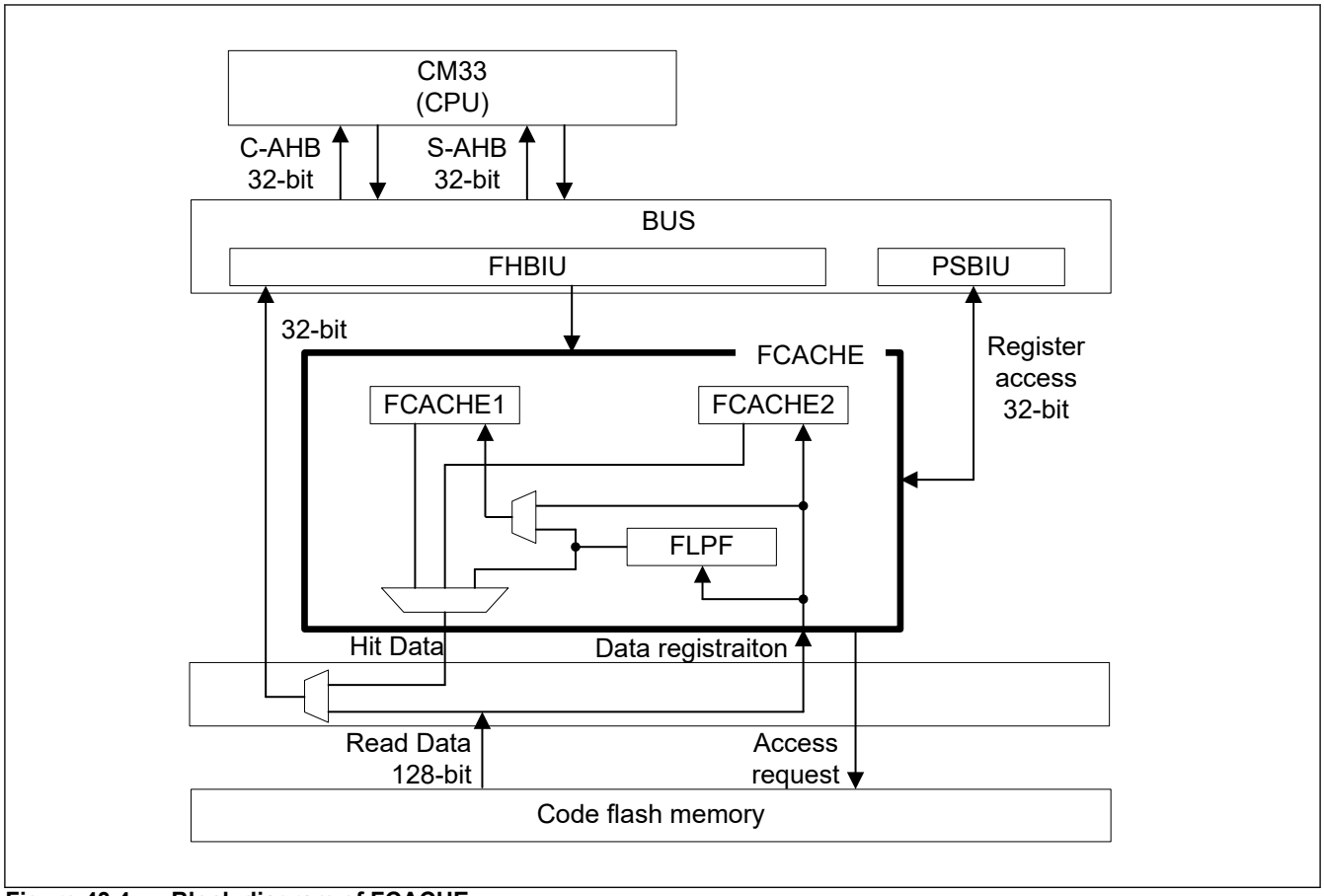


Figure 43.4 Block diagram of FCACHE

43.6 Operating Modes Associated with Flash Memory

Figure 43.5 is a diagram of the mode transitions associated with the flash memory. For the procedures for setting the modes, see section 6, Option-Setting Memory.

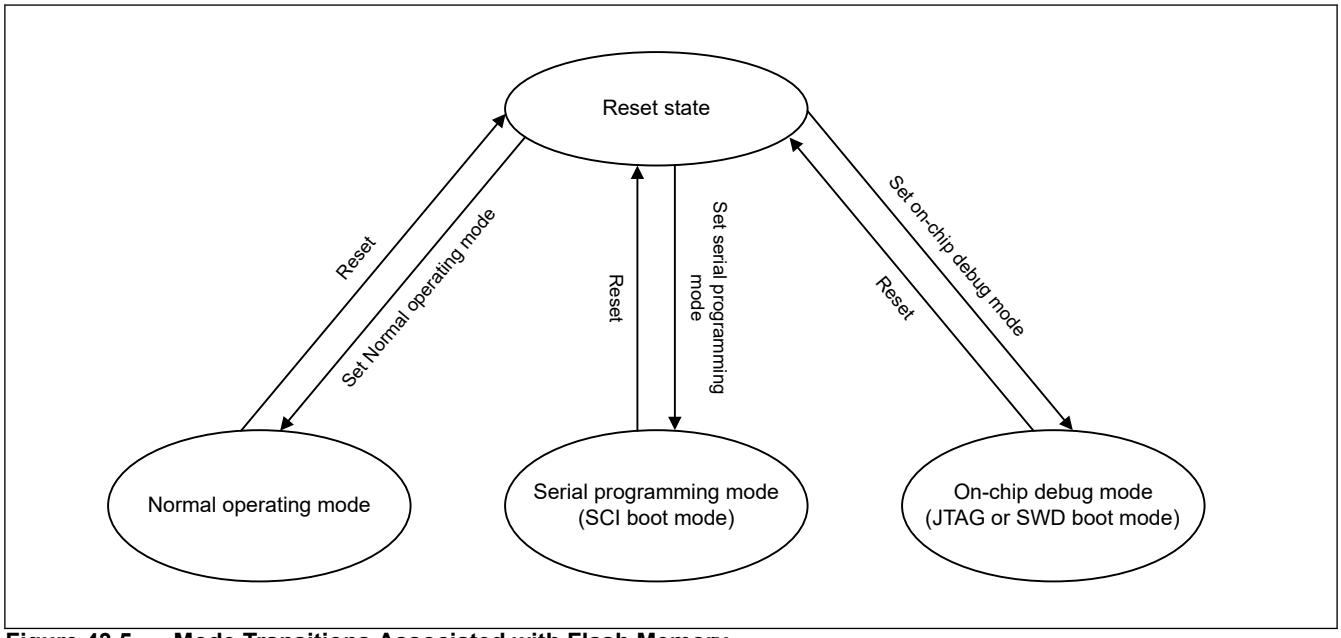


Figure 43.5 Mode Transitions Associated with Flash Memory

The flash memory area where programming and erasure are permitted and the boot program after a reset are different according to each mode. The differences between modes are listed in Table 43.9.

Table 43.9 Differences between Modes

Parameter	Normal operating mode	Serial programming mode (SCI boot mode)	On-chip debug mode (JTAG or SWD boot mode)
Programmable and erasable areas	• Code flash memory • Data flash memory • Option setting memory (programming only)	• Code flash memory • Data flash memory • Option-setting memory (programming only)	• Code flash memory • Data flash memory • Option setting memory (programming only)
Erasure in block units	Possible	Possible	Possible
Boot program at a reset	User area program	Embedded program for serial programming	Depends on debug command

43.7 Overview of Functions

By using a dedicated flash-memory programmer to program the flash memory through a serial interface (serial programming) or JTAG/SWD interface (on-chip debug mode), the device can be rewritten regardless of whether this is before or after it is mounted on the target system.

Furthermore, security functions to prohibit rewriting or reading of the user program written to the flash memory are incorporated, and this can prevent falsification and illicit reading of the programs by third parties.

Programming by the user program (self-programming) is available to suit applications where the application on the target system may require updating after manufacturing or shipment. Protection features for the safe rewriting of the flash memory are also incorporated. Furthermore, interrupt processing during self-programming is supported, so programming can proceed at the same time as processing for external communications, etc., and this is the case in various situations. Table 43.10 lists the overview of the methods of programming and the corresponding operating modes.

Table 43.10 Programming methods

Programming method	Functional overview	Operating mode
Serial programming	A dedicated flash-memory programmer through the SCI interface enables on-board programming of the flash memory after the device is mounted on the target system.	Serial programming mode
	A dedicated flash-memory programmer through the SCI interface and a dedicated programming adapter board allow off-board programming of the flash memory, for example, programming of the device before it is mounted on the target system.	
Self-programming	A user program written to memory in advance of serial programming execution can also program the flash memory. The background operation capability makes it possible to fetch instructions or otherwise read data from the code flash memory while the data flash memory is programmed. As a result, a program resident in the code flash memory is able to program the data flash memory. For background operations that are not possible, instructions in the code flash memory cannot be fetched and data cannot be accessed while the code flash memory is being programmed by self-programming. In such cases, a program for programming from the internal SRAM must be transferred in advance and executed.	Normal operating mode
JTAG or SWD programming	A dedicated flash-memory programmer or an on-chip debugger through JTAG or SWD enables on-board programming of the flash memory after the device is mounted on the target system. A dedicated flash-memory programmer or an on-chip debugger through JTAG or SWD and a dedicated programming adapter board allow off-board programming of the flash memory, for example, programming of the device before it is mounted on the target system.	On-chip debug mode

Table 43.11 lists the functions of the flash memory. Serial programmer commands realize each function of serial programming, while reading of the flash memory by an FACI command or the user program realizes each function of self-programming.

Table 43.11 Basic Functions

Function	Functional overview	Availability	
		Serial programming	Self-programming
Blank check	Checks a specified block to ensure that writing to it has not already proceeded. Results of reading from data flash memory to which nothing is written after erasure are not guaranteed, so use blank checking to confirm that writing to memory has not proceeded after erasure.	Not supported	Supported (data flash programming only)
Block erasure	Erases the memory contents in the specified block	Supported	Supported
Programming	Writes to the specified address	Supported	Supported
CRC	Calculates the CRC in the specified range of the flash memory and transfers it to the flash programmer	Supported	Non supported
Read	Reads data programmed in the flash memory	Supported	Not supported (read by user program is possible)
Start-up program protection functions	Configures the start-up program protection functions	Supported	Supported
Option function selection	Selects the option function, and modifies the initial setting of this MCU	Supported	Supported
Block protection	Setting block protection	Supported	Supported
Device lifecycle transition	Transitions the device lifecycle	Supported	Not supported
Memory security attribution	Setting the memory security attribution	Supported	Not supported
Key	Injects key	Supported	Supported (except the key related to device lifecycle transition)
All erasure	Erase the flash memory to the state after shipment	Supported	Not supported

The flash memory supports various security functions.

Table 43.12 lists the security functions supported by the flash memory.

Table 43.12 Lists of Security Functions

Function	Description
Security flag for Start-up area select	Start-up area selection can be protected by setting of security flag (FSPR).
Permanently block protection	Programming or erasure of each block of code flash memory can be protected permanently.
Protection for TrustZone	Programming or erasure area, readable area, register access, and FACI command operation are protected by ARM TrustZone security.
Programming or erasure mode protection	Only secure developer can enter the programming or erasure mode for code flash.

43.8 Operating Modes of the Flash Sequencer

The flash sequencer has three operating modes as shown in Figure 43.6. Transitions between modes are initiated by changing the value of the FENTRYR register.

When the value of the FENTRYR register is 0x0000, the flash sequencer is in read mode. In this mode, it does not receive FACI commands. The code flash memory and data flash memory are both readable.

When the value of the FENTRYR register is 0x0001, the flash sequencer is in code flash P/E mode where the code flash memory can be programmed or erased by FACI commands. In this mode, the data flash memory is readable.

When the value of the FENTRYR register is 0x0080, the flash sequencer is in data flash P/E mode where the data flash memory can be programmed or erased by FACI commands. In this mode, the data flash memory is not readable. However, the code flash memory is readable.

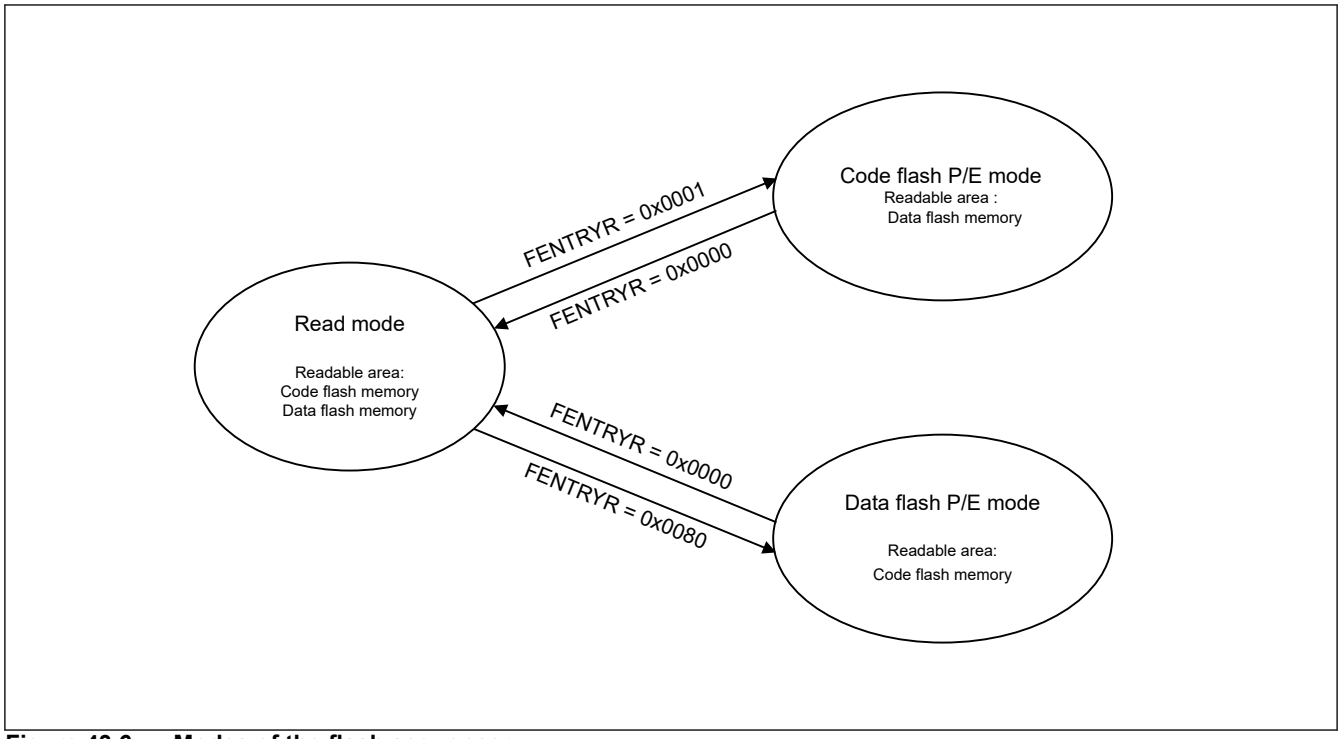


Figure 43.6 Modes of the flash sequencer

43.9 FACI Commands

43.9.1 List of FACI Commands

The FACI controls the FCU according to the specified FACI commands.

This section describes information about the FACI commands and Table 43.13 lists the FACI commands.

Table 43.13 FACI commands

FACI command	Function
Program	Programs the user area and data area. Units of programming are 128 bytes for the user area and 4, 8, or 16 bytes for the data area.
Block erase	Erases user area and data area. The erase unit is 8 KB or 32 KB for user area, and 64 bytes for data flash.
Multi block erase	Erases data area. The erase unit is 64, 128, 256 bytes for data flash.
P/E suspend	Suspends programming or erasure processing.
P/E resume	Resumes suspended programming or erasure processing.
Status clear	Initializes the ILGLERR, ERSERR, PRGERR, ILGCOMERR, FESETERR, SECERR, and OTERR bits in the FSTATR register and the CMDLK, CFAE, and DFAE bits in the FASTAT register, and the flash sequencer released from command-locked state.
Forced stop	Forcibly stops processing of FACI commands and initializes the FSTATR and FASTAT registers.
Blank check	Checks if data areas are blank. Units of Blank Check: 4 bytes to data flash memory capacity (specified in 4-byte units).
Configuration set	Sets the option-setting memory. Units of setting: 16 bytes.

The FACI commands are issued by writing to the FACI command-issuing area (see Table 43.3). When write access as shown in Table 43.14 proceeds in the specified state, the flash sequencer executes the processing associated with the given command (see section 43.9.2. Relationship between the Flash Sequencer State and FACI Commands).

Table 43.14 FACI command formats

FACI commands	Number of write access	Write data to the FACI command-issuing area			
		1st access	2nd access	3rd to (N+2)th access	(N+3)th access
Program (user area) N = 64	67	0xE8	0x40 (=N)	WD1 to WD64	0xD0
Program (data area) 4-byte programming: N = 2 8-byte programming: N = 4 16-byte programming: N = 8	N+3	0xE8	0x02 (=N) 0x04 (=N) 0x08 (=N)	WD1 to WDN	0xD0
Block Erase (user area 8K/32K Bytes)	2	0x20	0xD0	—	—
Block Erase (data area 64 bytes)	2	0x20	0xD0	—	—
Multi block erase (data area 64/128/256 bytes)	2	0x21	0xD0	—	—
P/E suspend	1	0xB0	—	—	—
P/E resume	1	0xD0	—	—	—
Status Clear	1	0x50	—	—	—
Forced Stop	1	0xB3	—	—	—
Blank Check	2	0x71	0xD0	—	—
Configuration set N = 8	11	0x40	0x08 (=N)	WD1 to WD8	0xD0

Note: WDN (N = 1, 2, …): Nth 16-bit data to be programmed.

The flash sequencer clears the FSTATR.FRDY bit to 0 at the start of a command processing other than the Status Clear command, and sets this bit to 1 on completion.

If the FRDYIE.FRDYIE bit setting is 1, a flash ready (FRDY) interrupt is generated when the FSTATR.FRDY bit is set to 1.

43.9.2 Relationship between the Flash Sequencer State and FACI Commands

The FACI commands are accepted according to the mode/state of the flash sequencer. FACI commands should be issued after transitioning of the flash sequencer to the code flash P/E mode or data flash P/E mode and after checking the state of the flash sequencer.

Use the FSTATR and FASTAT registers to check the state of the flash sequencer. In addition, the occurrence of errors in general can be checked by reading the CMDLK bit in the FASTAT register. The value of the CMDLK bit is the logical OR of the following bits in the FSTATR register:

- ILGLERR
- ILGCOMERR
- FESETERR
- SECERR
- OTERR
- ERSERR
- PRGERR
- FLWEERR.

Table 43.15 lists the available FACI commands in each operating mode.

Table 43.15 Operating mode and available FACI commands

Operating mode	FENTRYR	Available FACI commands
Read mode	0x0000	None
Code flash P/E mode	0x0001	Program Block erase P/E suspend P/E resume Status Clear Forced Stop Configuration set
Data flash P/E mode	0x0080	Program Block erase Multi block erase P/E suspend P/E resume Status Clear Forced Stop Blank Check

Table 43.16 shows the state of the flash sequencer and acceptable FACI commands. An appropriate mode is assumed to have been set before the commands are executed.

Table 43.16 Acceptable FACI commands and state of the flash sequencer

	Program, block erase or multi block erase command processing	Configuration set command processing	Program, block erase or multi block erase command suspension processing	Blank check command processing	Programming suspended	Erasure suspended	Programming while erasure is suspended	Command-locked state (FRDY = 1)	Command-locked state (FRDY = 0)	Processing of forced stop command	Other state
FRDY bit	0	0	0	0	1	1	0	1	0	0	1
SUSRDY bit	1	0	0	0	0	0	0	0	0	0	0
ERSSPD bit	0	0	0/1	0/1	0	1	1	0/1	0/1	0	0
PRGSPD bit	0	0	0/1	0/1	1	0	0	0/1	0/1	0	0
CMDLK bit	0	0	0	0	0	0	0	1	1	0	0
Program	X	X*4	X	X	X	O*3	X	X	X	X	O
Block erase or multi block erase	X	X*4	X	X	X	X	X	X	X	X	O
P/E suspend	O	X*4	X	X	X	X	X	—	X	X	—
P/E resume	X	X*4	X	X	O	O	X	X	X	X	X
Status clear	X	X*4	X	X	O	O	X	O	X	X	O
Forced stop	O	O*4	O	O	O	O	O	O	O	O	O
Blank check	X	X*4	X	X	O*1	O*1	X	X	X	X	O*1
Configuration set	X	X*4	X	X	X	X	X	X	X	X	O*2

Note: O: Acceptable
X: Not acceptable (places the sequencer in the command-locked state)

—: Ignored
Note 1. Only acceptable in data flash P/E mode.
Note 2. Only acceptable in code flash P/E mode
Note 3. Acceptable when programming area is other than erase suspending block.
Note 4. When configuration set is processing and when FSTATR.DBFULL bit is 1, do not issue this command.

43.9.3 Usage of FACI Commands

43.9.3.1 Overview of Command Usage in Code Flash P/E Mode

Figure 43.7 show an overview of FACI command usage in code flash P/E mode. For the available commands in code flash P/E mode, see Table 43.15.

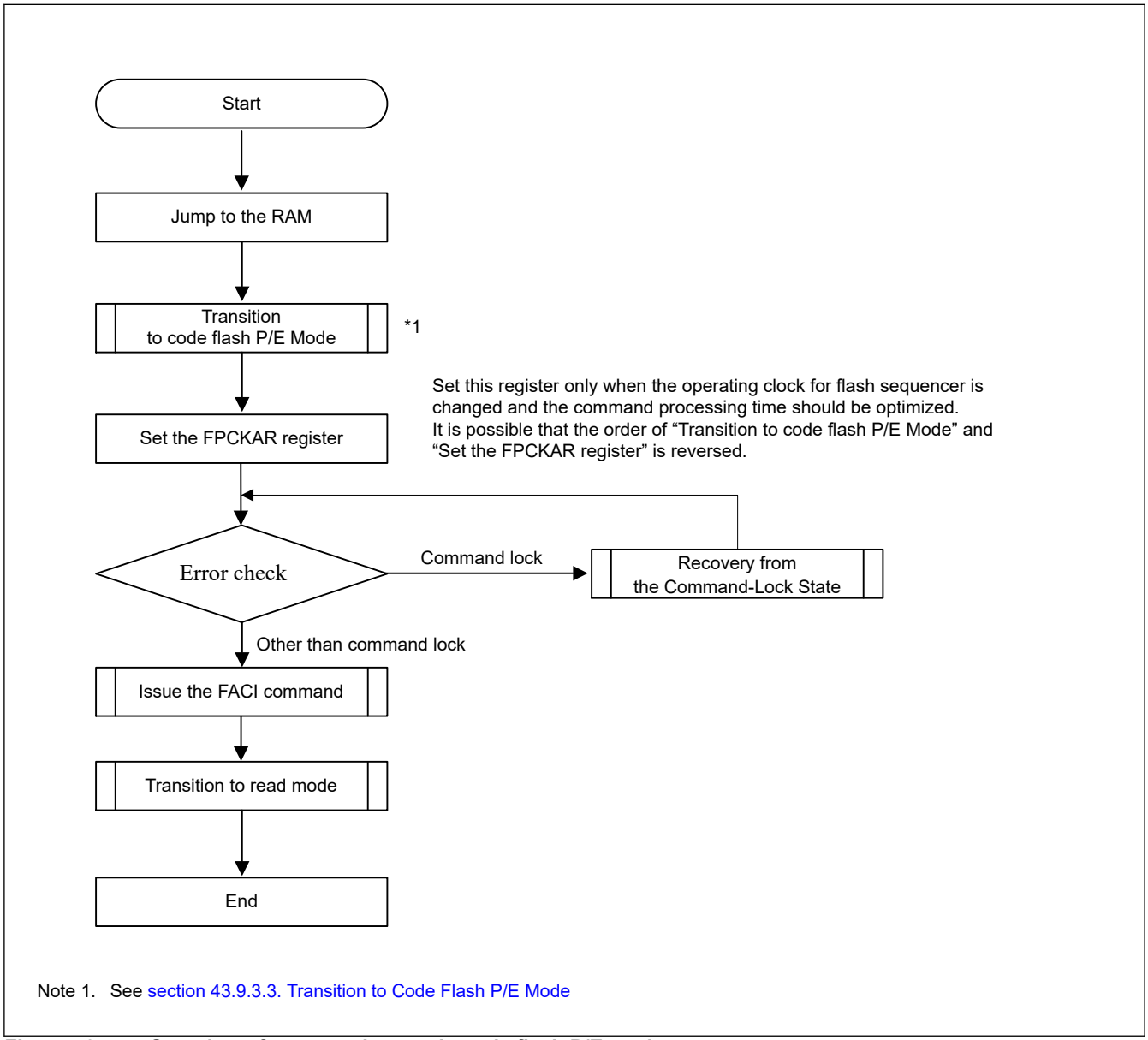
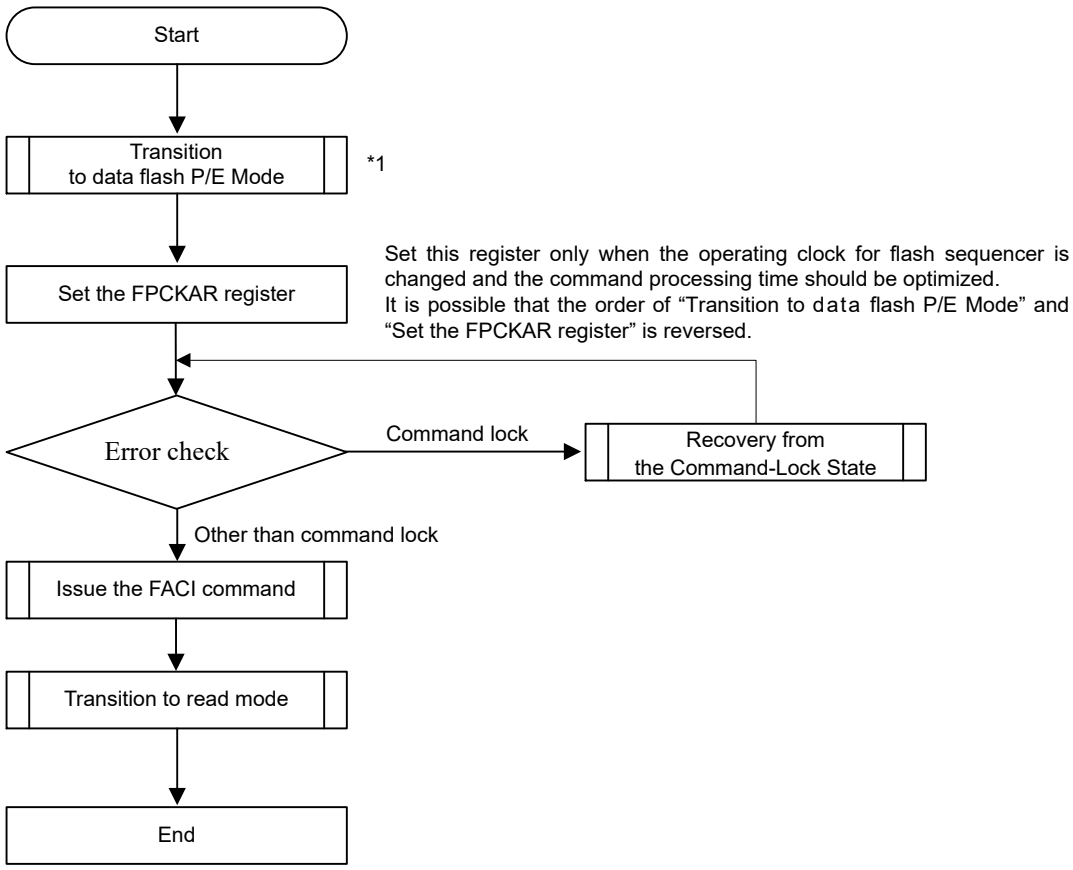


Figure 43.7 Overview of command usage in code flash P/E mode

43.9.3.2 Overview of Command Usage in Data Flash P/E Mode

Figure 43.8 shows an overview of FACI command usage in data flash P/E and Table 43.15 lists the available commands in data flash P/E mode.



Note 1. See section 43.9.3.4. Transition to Data Flash P/E Mode

Figure 43.8 Overview of command usage in data flash P/E mode

43.9.3.3 Transition to Code Flash P/E Mode

To issue FACI commands for the code flash memory, a transition to code flash P/E mode is required by setting the FENTRYC bit in the FENTRYR register to 1.

Figure 43.9 shows the procedure to transition to code flash P/E mode.

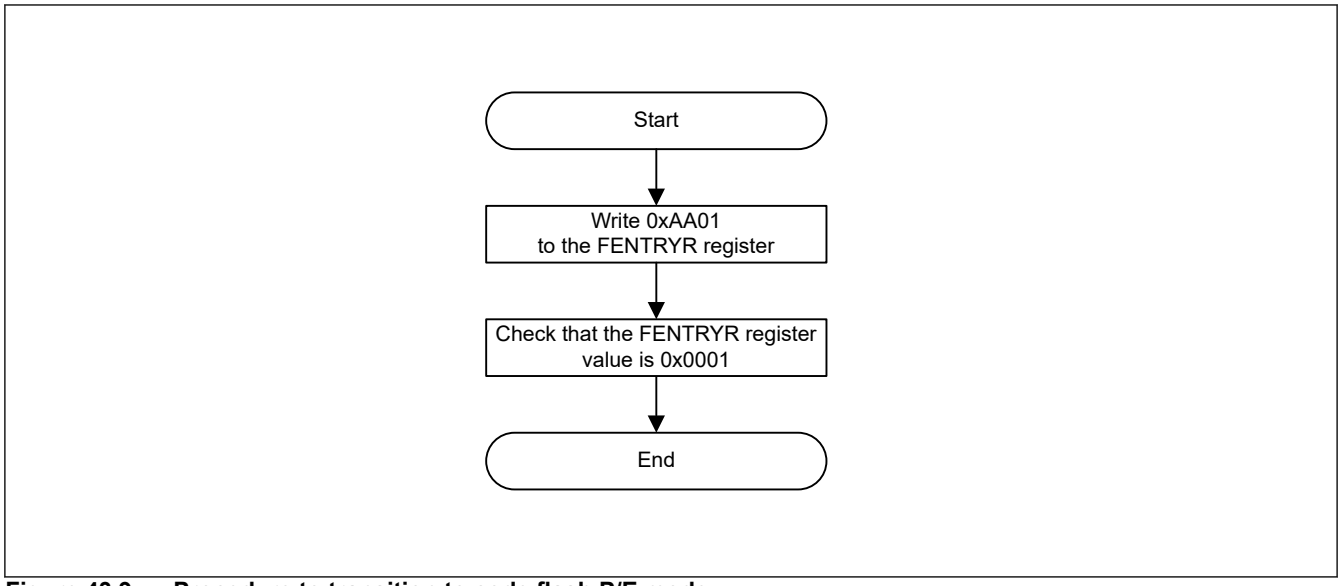


Figure 43.9 Procedure to transition to code flash P/E mode

43.9.3.4 Transition to Data Flash P/E Mode

To issue FACI commands for the data flash memory, a transition to data flash P/E mode is required by setting the FENTRYD bit in the FENTRYR register to 1.

Figure 43.10 shows the procedure to transition to data flash P/E mode.

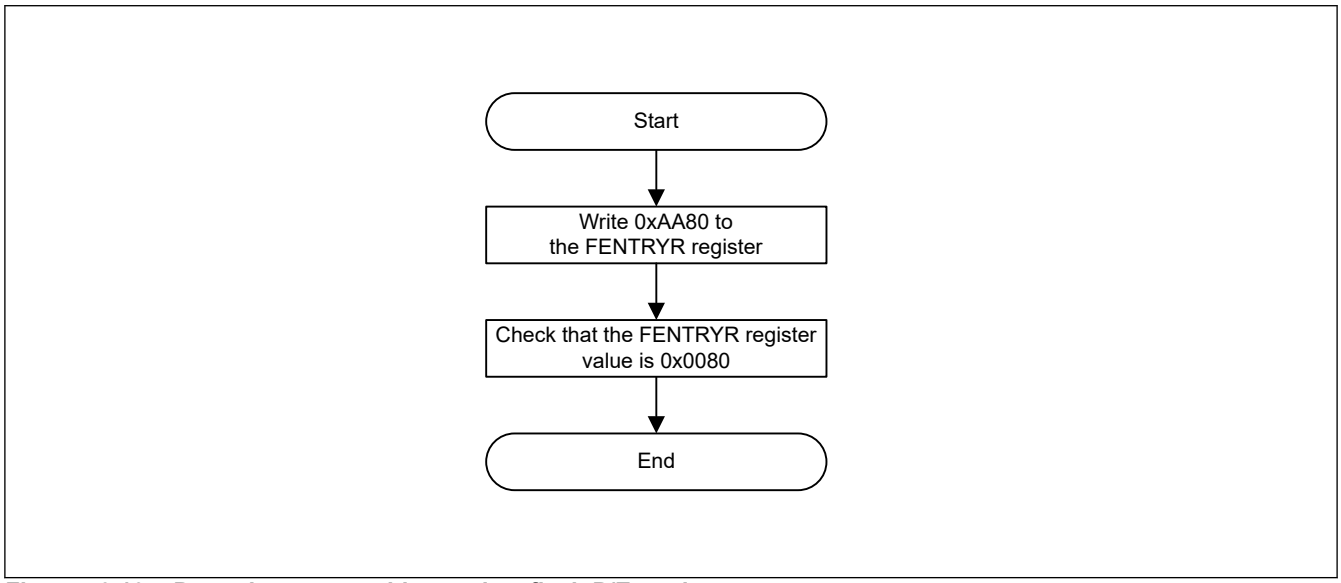
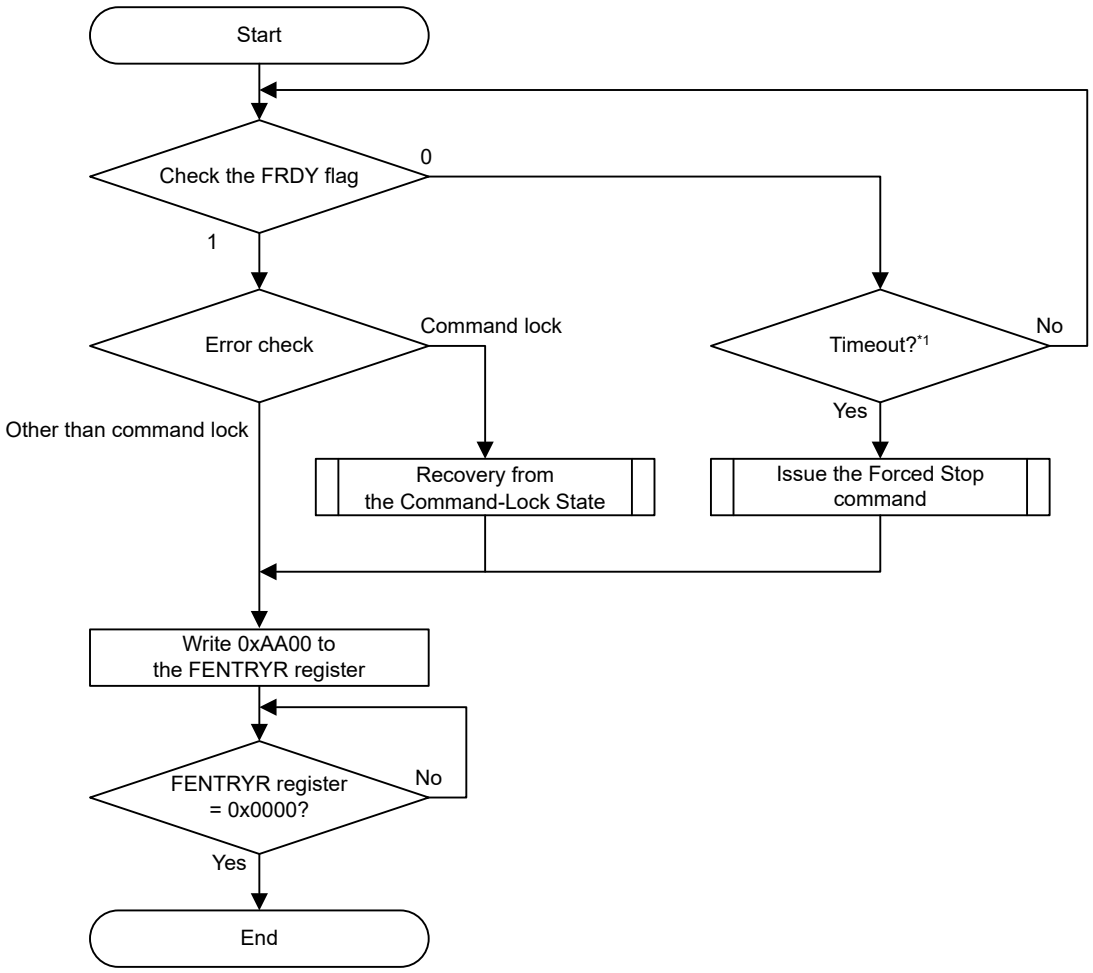


Figure 43.10 Procedure to transition to data flash P/E mode

43.9.3.5 Transition to Read Mode

To read the flash memory, a transition to read mode is required by setting the FENTRYR register to 0x0000. The transition to read mode should be made after the flash sequencer completes the processing and while operation is not in the command-locked state.

Figure 43.11 shows the procedure to transition to read mode.



Note 1. Judgment of the timeout is based on 1.1 times the maximum time for processing of the FACI command that is in progress (see section 46, Electrical Characteristics)

Figure 43.11 Procedure to transition to read mode

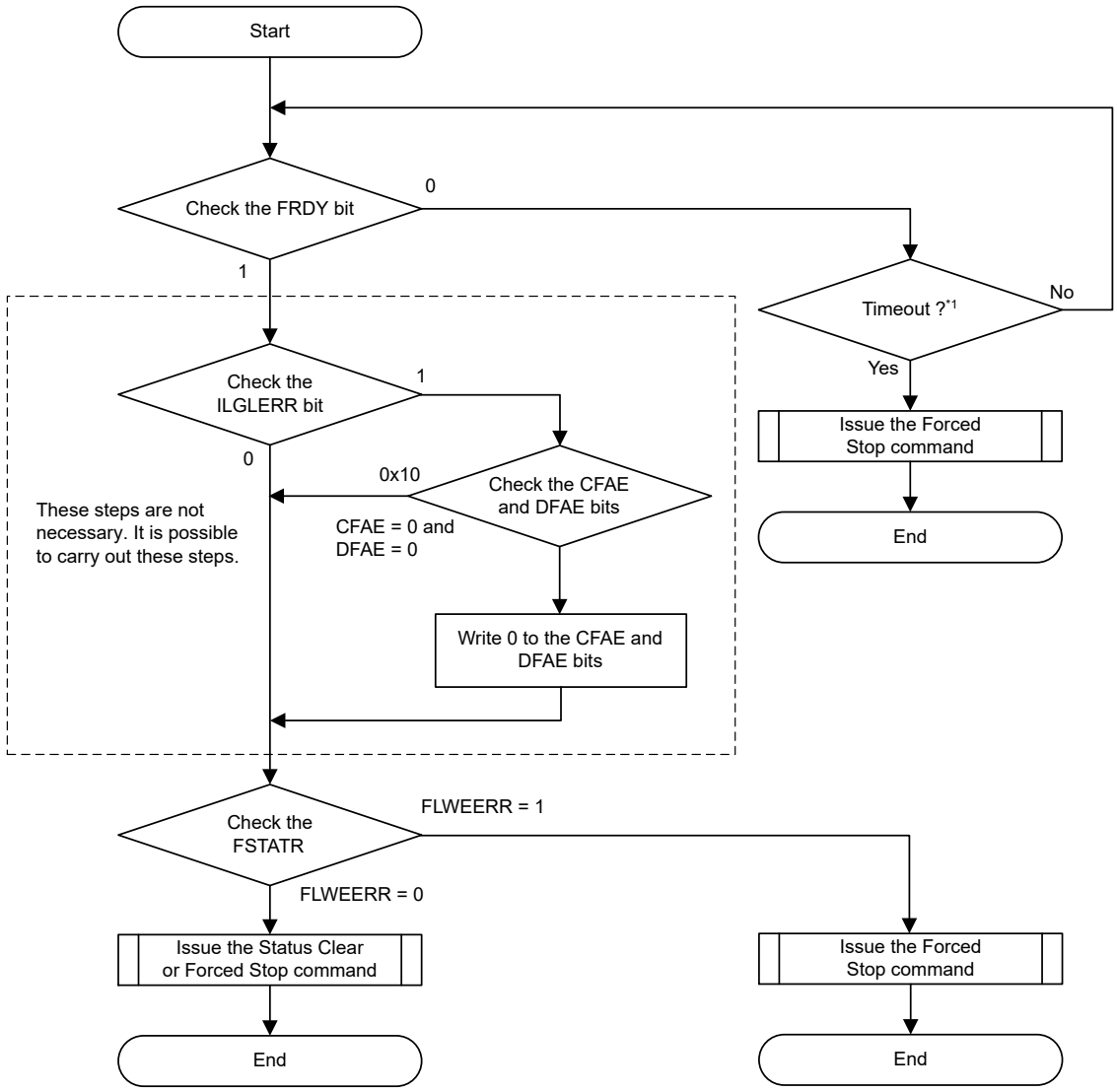
43.9.3.6 Recovery from the Command-Locked State

When the flash sequencer enters the command-locked state, FACI commands cannot be accepted. To release the sequencer from the command-locked state, use the status clear command, forced stop command, or FASTAT register.

When the command-locked state is detected by checking for an error before issuing the P/E suspend command, the FRDY bit in the FSTATR register might be 0 even though command processing has not completed. If processing is not complete by the maximum programming/erasure time specified in the electrical characteristics, this is a timeout and the flash sequencer must be stopped with the forced stop command.

The FLWEERR bit in the FSTATR register does not change from 1 to 0 with the status clear command. When these bits are set to 1, use the forced stop command to release from the command-locked state. Bits other than FRDY and FLWEERR in FSTATR register that indicate the command-locked state can be changed from 1 to 0 with the status clear or forced stop command.

Figure 43.12 shows the recovery flow from the command-locked state.



Note 1. Judgment of the timeout is based on 1.1 times the maximum time for processing of the FACI command that is in progress (see section 46, Electrical Characteristics)

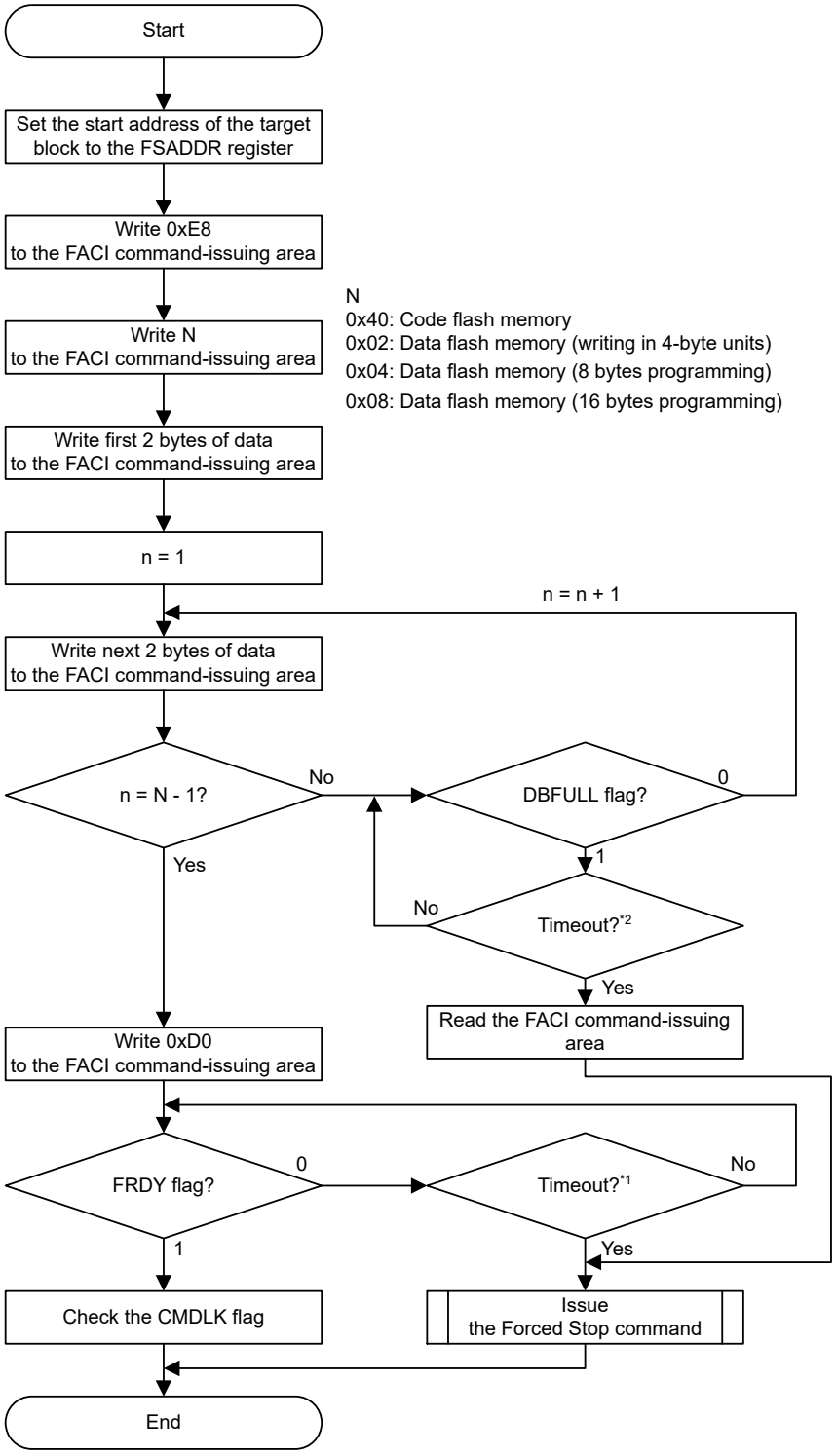
Figure 43.12 Recovery flow from the command-locked state

43.9.3.7 Program Command

The program command is used for writing to the user area and data area. Before issuing the FACI program command, set the first address of the target block in the FSADDR register. Writing 0xD0 at the final access of the FACI command-issuing area starts the program command processing. If the target area of program command processing contains area that are not for writing, write 0xFFFF to the corresponding area.

Issuing the program command while the FACI internal data buffer is full leads to a wait on the peripheral bus that might affect communications performance of other peripheral modules. To avoid a wait, set the DBFULL bit in the FSTATR register to 0 when issuing the FACI command. Writing to the data area does not lead to the data buffer becoming full.

Figure 43.13 shows the usage of the program command.



Note 1. Judgment of the timeout is based on 1.1 times the maximum time of programming. See section 46, Electrical Characteristics about those time

Note 2. 1.4 µs (f_{FCLK} ≥ 20 MHz)

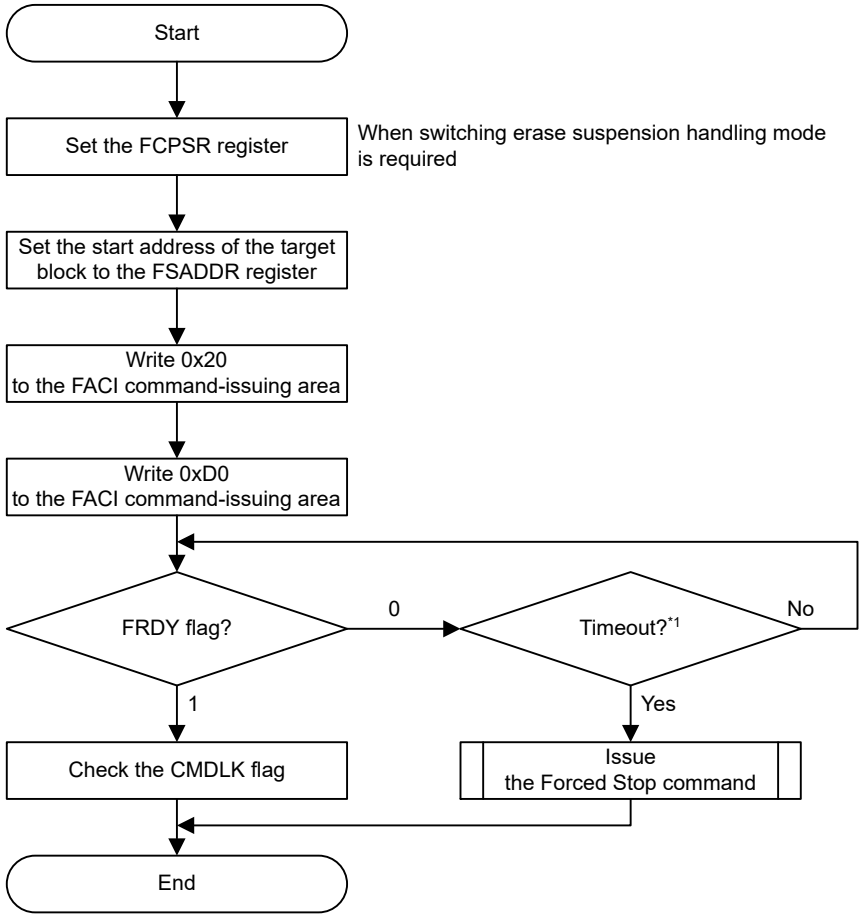
Figure 43.13 Usage flow of the program command

43.9.3.8 Block Erase Command

The block erase command is used for erasing user area or data area. The erase unit is one block. Before issuing a block erase command, set the first address of the target block to FSADDR register. Writing 0xD0 at the second write access of the FACI command triggers the FACI to start the block erase command processing. Completion of command processing can be confirmed with the FRDY bit of FSTATR register.

Set the FCPSR registers before issuing the block erase command. Additionally, FCPSR must be set when the erasure-suspended mode is to be switched.

Figure 43.14 shows the usage of the block erase command.



Note 1. Judgment of the timeout is based on 1.1 times the maximum time of erasure. See section 46, Electrical Characteristics.

Figure 43.14 Usage flow of the block erase command

43.9.3.9 Multi Block Erase Command

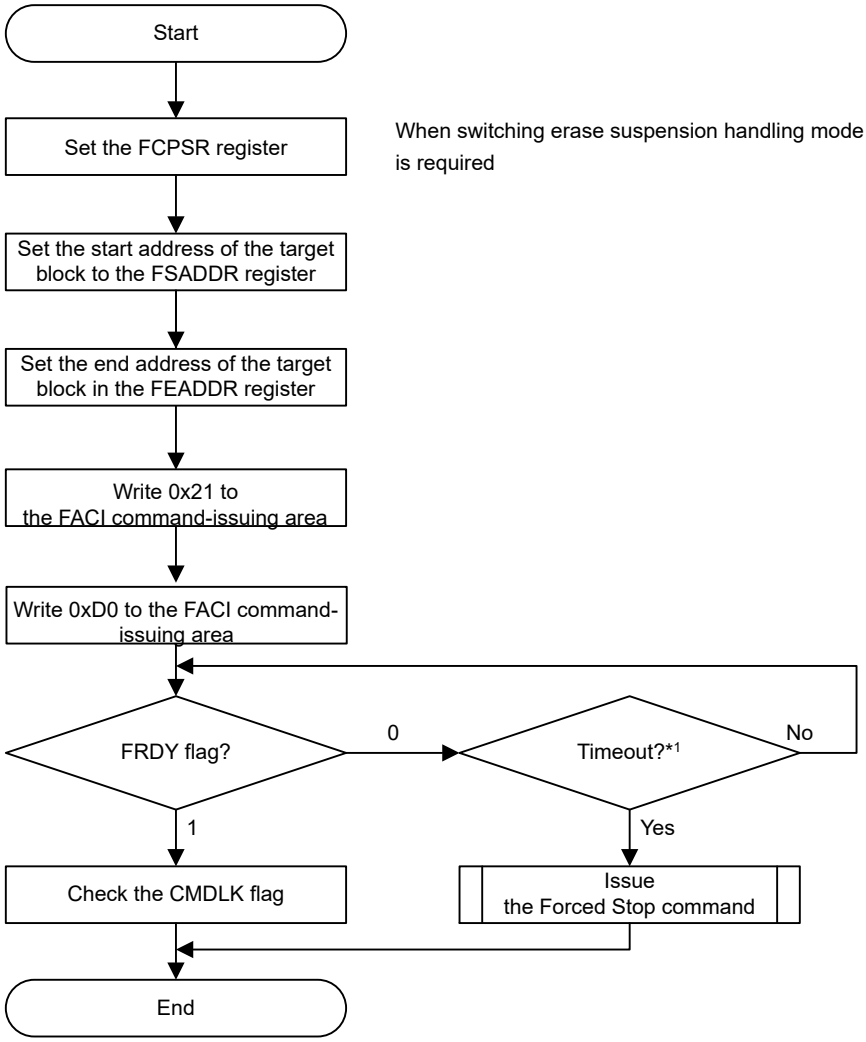
The multi block erase command is used for erasing data area. The erase unit is 64, 128, or 256 bytes. Before issuing the multi block erase command, set the start address to FSADDR register and the end address to FEADDR register. Writing 0xD0 at the second write access of the FACI command triggers FACI to start the multi block erase command processing. Completion of command processing can be confirmed with the FRDY bit of FSTATR register.

Set the FCPSR registers before issuing the multi block erase command. Additionally, FCPSR must be set when the erasure-suspended mode is to be switched.

The erase size is specified by both the FSADDR and FEADDR settings. Table 43.17 describes how to set the FSADDR and FEADDR.

Table 43.17 Settings for the erase size

Erase size	FSADDR	FEADDR
64 bytes	FSA0 to FSA5 = 0 (64 byte-boundary)	FSADDR + 0x3C
128 bytes	FSA0 to FSA6 = 0 (128 byte-boundary)	FSADDR + 0x7C
256 bytes	FSA0 to FSA7 = 0 (256 byte-boundary)	FSADDR + 0xFC



Note 1. Judgment of the timeout is based on 1.1 times the maximum time of erasure. See section 46, Electrical Characteristics.

Figure 43.15 Usage flow of the multi block command

43.9.3.10 P/E Suspend Command

The P/E suspend command is used to suspend programming/erasure. Before issuing a P/E suspend command, check that the CMDLK bit in the FASTAT register is 0, and that the execution of programming/erasure is performed normally. To confirm that the P/E suspend command can be received, check that the SUSRDY bit in the FSTATR register is 1. After issuing a P/E suspend command, read the CMDLK bit to confirm that no error occurs.

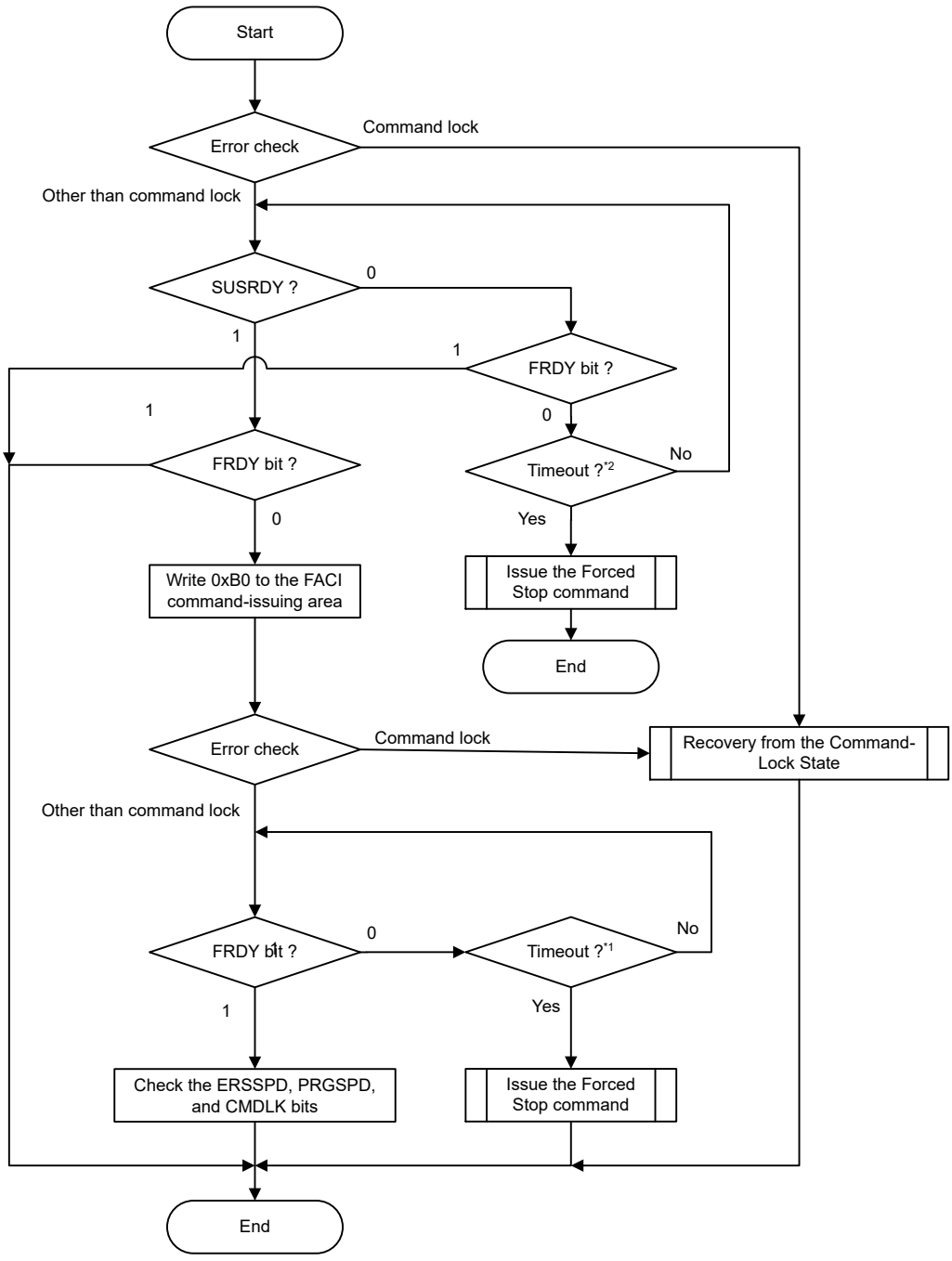
If an error occurs during programming/erasure, the CMDLK bit is set to 1. When programming/erasure processing has finished from the time when the SUSRDY bit is 1 to when the P/E suspend command is received, no error occurs and the

suspended state is not entered (the FRDY bit in the FSTATR register is 1 and the ERSSPD and PRGSPD bits in FSTATR are 0).

When a P/E suspend command is received and the programming/erasure suspend processing finishes normally, the flash sequencer enters the suspended state, the FRDY bit is set to 1, and the ERSSPD or PRGSPD bit is 1. After issuing a P/E suspend command, check that the ERSSPD or PRGSPD bit is 1 and the suspended state is entered, then proceed with the subsequent flow. If a P/E resume command is issued in the subsequent flow even when the suspended state is not entered, an illegal command error occurs and the flash sequencer shifts to the command-locked state (see section 43.11.2. Error Protection).

If the erasure suspended state is entered, programming to blocks other than an erasure target block can be performed. Additionally, the programming and erasure suspended states can shift to read mode by clearing the FENTRYR register.

Figure 43.16 shows the usage of the P/E suspend command.



Note 1. Judgment of the timeout is based on 1.1 times the maximum time of suspend delay during erasure for both code flash and data flash.
See section 46, Electrical Characteristics.
Note 2. 4.4µs (f_{FCLK} >= 20MHz)

Figure 43.16 Usage flow of the P/E suspend command

(1) Suspension during Programming

When issuing a P/E suspend command during flash memory programming, the flash sequencer suspends programming processing. Figure 43.17 shows the suspend programming operation. When receiving programming-related command, the flash sequencer clears the FRDY bit in the FSTATR register to 0 to start programming. If the flash sequencer enters the state

in which the P/E suspend command can be received after programming starts, it sets the SUSRDY bit in the FSTATR register to 1.

When a P/E suspend command is issued, the flash sequencer receives the command and clears the SUSRDY bit to 0. If the flash sequencer receives a P/E suspend command while a programming pulse is applied, the flash sequencer continues with the pulse. After the specified pulse application time, the flash sequencer finishes pulse application, starts the programming suspend processing, and sets the PRGSPD bit in the FSTATR register to 1.

When a suspended processing finishes, the flash sequencer sets the FRDY bit to 1 to enter the programming suspended state. When receiving a P/E resume command in the programming suspended state, the flash sequencer clears the FRDY and PRGSPD bits to 0 and resumes programming.

Figure 43.17 shows the timing for suspension during programming.

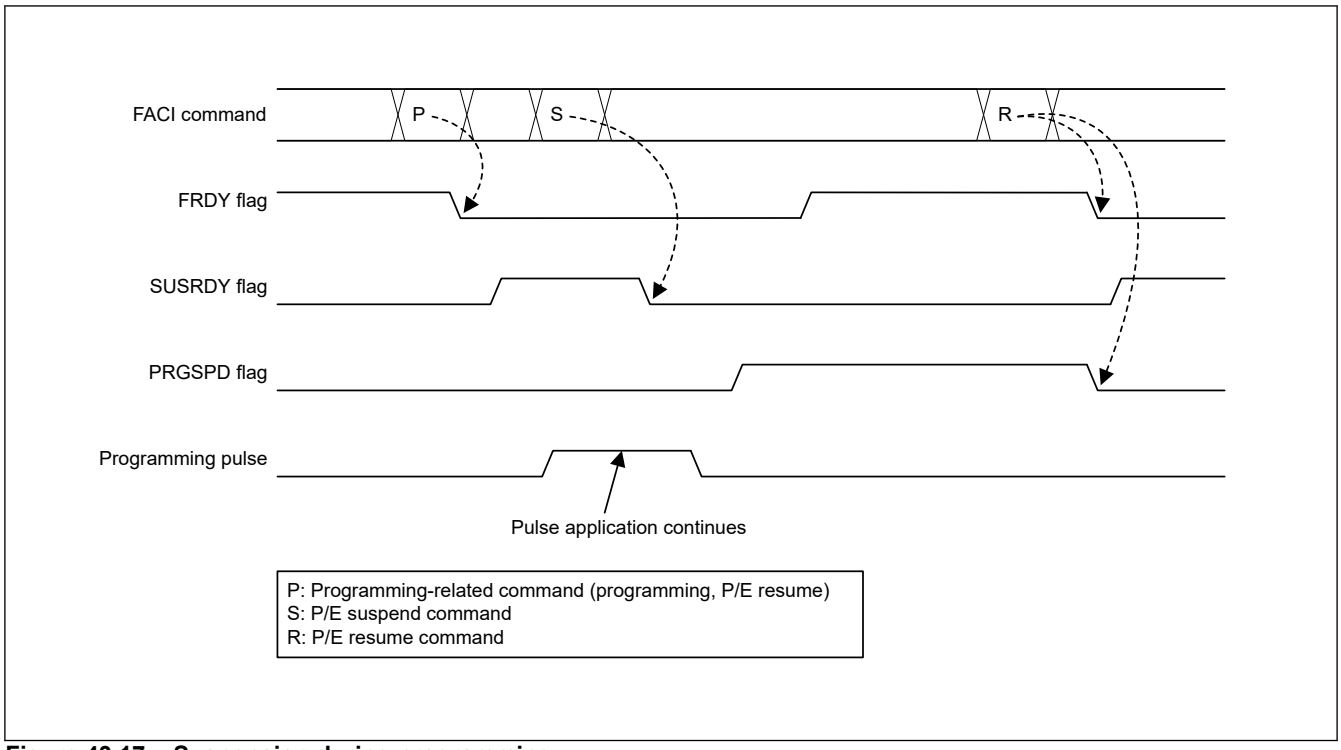


Figure 43.17 Suspension during programming

(2) Suspension during Erasure (Suspension Priority Mode)

The flash sequencer has a suspension priority mode for the suspension of erasure. Figure 43.18 shows the suspend operation of erasure when the erasure suspend mode is set to the suspension priority mode (the ESUSPMD bit in the FCPSR register is 0).

When receiving an erasure-related command, the flash sequencer clears the FRDY bit in the FSTATR register to 0 to start erasure. If the flash sequencer enters the state in which the P/E suspend command can be received after erasure starts, it sets the SUSRDY bit in the FSTATR register to 1.

When a P/E suspend command is issued, the flash sequencer receives the command and clears the SUSRDY bit to 0.

When receiving a suspend command during erasure, the flash sequencer starts the suspend processing and sets the ERSSPD bit in the FSTATR register to 1 even when it is applying an erasure pulse. When the suspended processing finishes, the flash sequencer sets the FRDY bit to 1 to enter the erasure suspended state. When receiving a P/E resume command in the erasure suspended state, the flash sequencer clears the FRDY and ERSSPD bits to 0 and resumes erasure. Operations of the FRDY, SUSRDY, and ERSSPD bits at the suspension and resumption of erasure are the same, regardless of the erasure suspend mode.

The setting of the erasure suspend mode affects the control method of erasure pulses. In suspension priority mode, when receiving a P/E suspend command while erasure pulse A that has not been previously suspended is being applied, the flash sequencer suspends the application of erasure pulse A and enters the erasure suspended state. When receiving a P/E suspend command while reapplying erasure pulse A after erasure is resumed with a P/E resume command, the flash sequencer

continues to apply erasure pulse A. After the specified pulse application time, the flash sequencer finishes erasure pulse application and enters the erasure suspended state.

When the flash sequencer receives a P/E resume command next and erasure pulse B is being applied, the flash sequencer receives a P/E suspend command again, and the application of erasure pulse B is then suspended. In suspension priority mode, delays due to suspension can be minimized because the application of an erasure pulse is suspended once per pulse, and priority is given to the suspend processing.

If the interval of suspension after resume is longer than t_{REST1} (Resume time: priority on suspension, resume after the 1st suspend for the same pulse), suspend delay will be always t_{SESD1} (Suspend delay: priority on suspension, the 1st suspend for the same pulse).

If the interval of suspension after resume is shorter than t_{REST1}, suspend delay becomes either t_{SESD1} or t_{SESD2} (Suspend delay: priority on suspension, the 2nd suspend for the same pulse).

(The value of t_{REST1} / t_{SESD1} / t_{SESD2}, see section 46, Electrical Characteristics.)

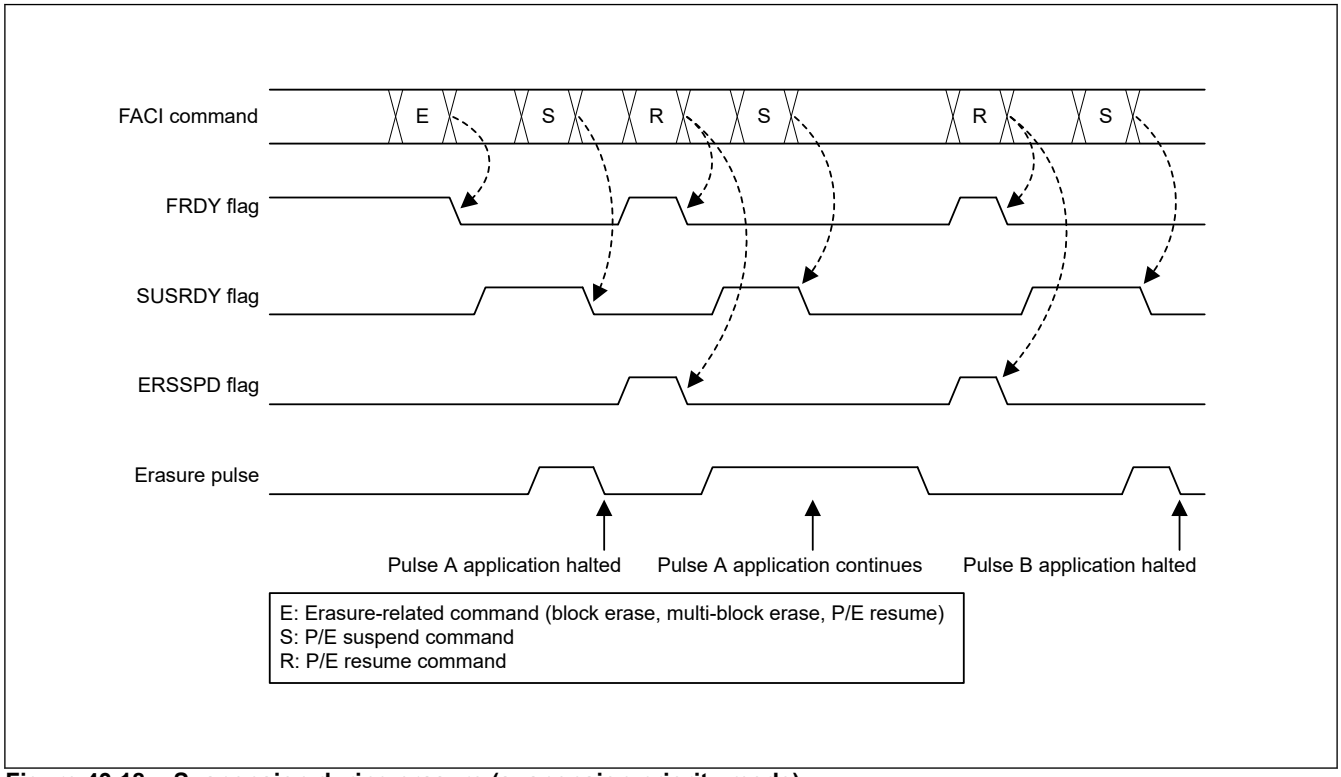


Figure 43.18 Suspension during erasure (suspension priority mode)

(3) Suspension during Erasure (Erasure Priority Mode)

The flash sequencer has an erasure priority mode for the suspension of erasure. Figure 43.19 shows the suspend operation of erasure when the erasure suspend mode is set to the erasure priority mode (the ESUSPMD bit in the FCPSR register is 1). The control method of erasure pulses in erasure priority mode is the same as that of programming pulses for the programming suspend processing.

If the flash sequencer receives a P/E suspend command while an erasure pulse is applied, the flash sequencer continues to apply the pulse. In this mode, the required time for the erasure processing can be reduced compared to the suspension priority mode because the re-application of erasure pulses does not occur when a P/E resume command is issued.

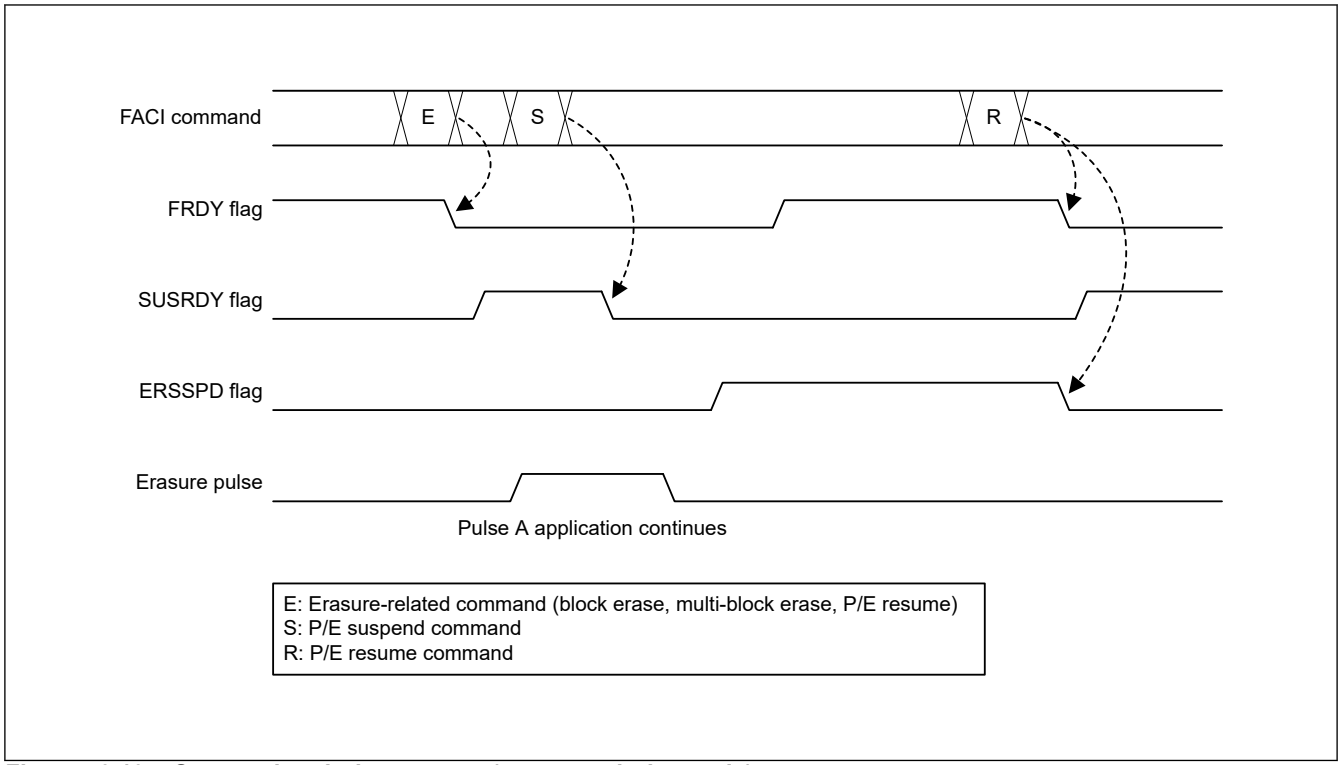
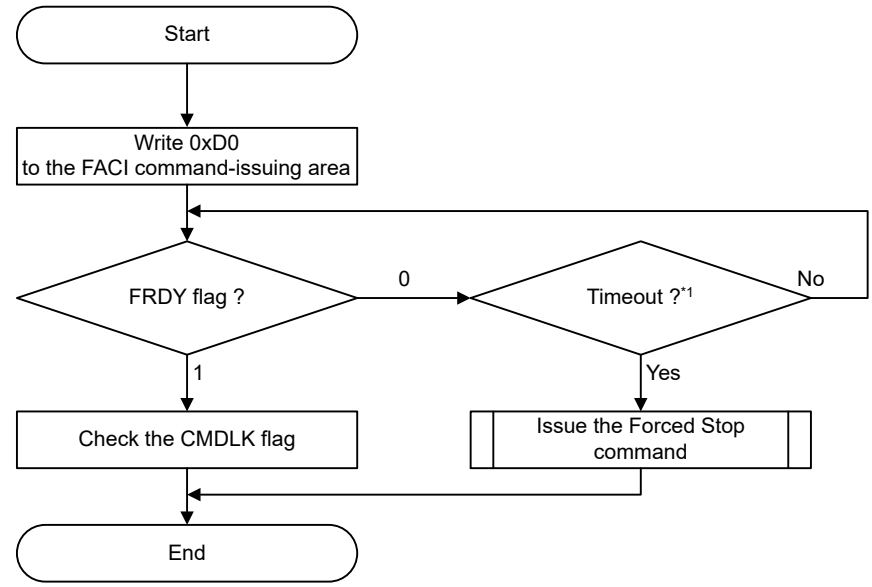


Figure 43.19 Suspension during erasure (erasure priority mode)

43.9.3.11 P/E Resume Command

The P/E resume command is used to resume suspended programming or erasure. If the FENTRYR setting has been modified during suspension, issue a P/E resume command only after resetting FENTRYR to the previous value that was held before the P/E suspend command was issued. Figure 43.20 shows usage of the P/E resume command.



Note 1. Judgment of the timeout is based on 1.1 times the maximum time for halted processing of the FACI command. See section 46, Electrical Characteristics.

Figure 43.20 Usage flow of the P/E resume command

43.9.3.12 Status Clear Command

The status clear command is used to clear the command-locked state (see section 43.9.3.6. Recovery from the Command-Locked State).

You can use the status clear command to clear the following bits in the FSTATR register in the command-locked state:

- ILGLERR
- ILGCOMERR
- FESETERR
- SECERR
- OTERR
- ERSERR
- PRGERR

Figure 43.21 shows usage of the status clear command.

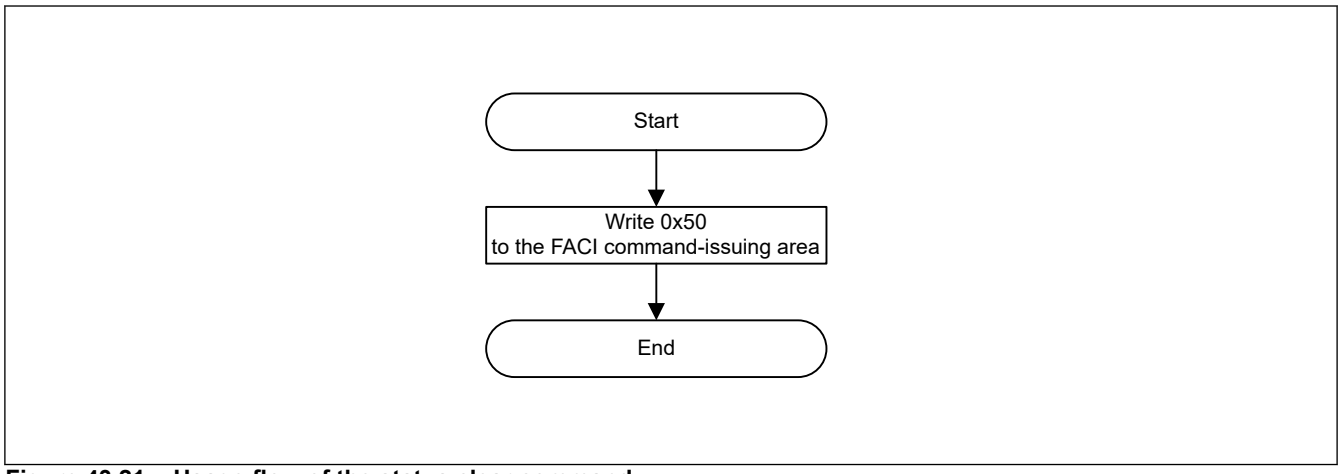


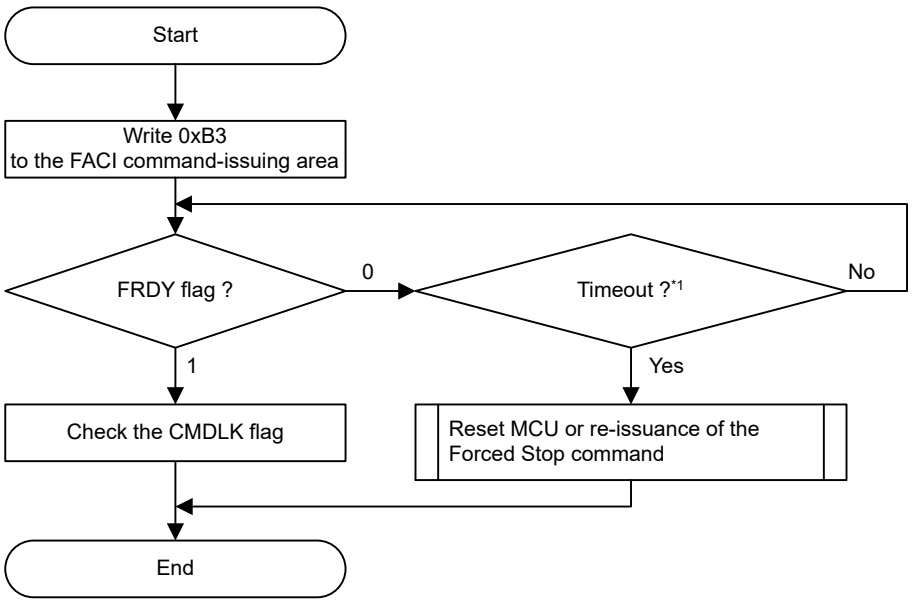
Figure 43.21 Usage flow of the status clear command

43.9.3.13 Forced Stop Command

The forced stop command is used to forcibly end command processing by the flash sequencer. Although this command halts command processing more quickly than the P/E suspension command, values from the programming or erasure that are in progress are not guaranteed. Additionally, resumption of processing is not possible. Processing of programming or erasure that is halted by the forced stop command is also defined as one programming round.

Executing the forced stop command also initializes part of the FACI, the whole FCU, the FSTATR and FASTAT registers. This command can be used in the procedure for recovery from the command-locked state and for processing in response to a timeout of the flash sequencer (see section 43.9.3.6. Recovery from the Command-Locked State).

Figure 43.22 shows usage of the forced stop command.



Note 1. Judgment of the timeout is based on 1.1 times the maximum time of the Forced Stop command latency. See section 46, Electrical Characteristics.

Figure 43.22 Usage flow of the forced stop command

(1) Notes on Using the Forced Stop Command during Command Issue

When using the forced stop command at the timeout occurrence by DBFULL bit of the program command, writing in the FACI command-issuing area is sometimes processed as writing in data of the program command. See Table 43.3 in section 43.3. Address Space for information on the FACI command-issuing area to force a command lock. Then issue a forced stop command with return method from the command lock status (see Figure 43.13). Locking commands is possible in any case where the unit for reading the FACI command issuing area is 8, 16, or 32 bits.

43.9.3.14 Blank Check Command

The blank check command is used to confirm that an area is in the non-programmed state. Values read from the data flash memory that have been erased but not yet programmed again that is in the non-programmed state, are undefined.

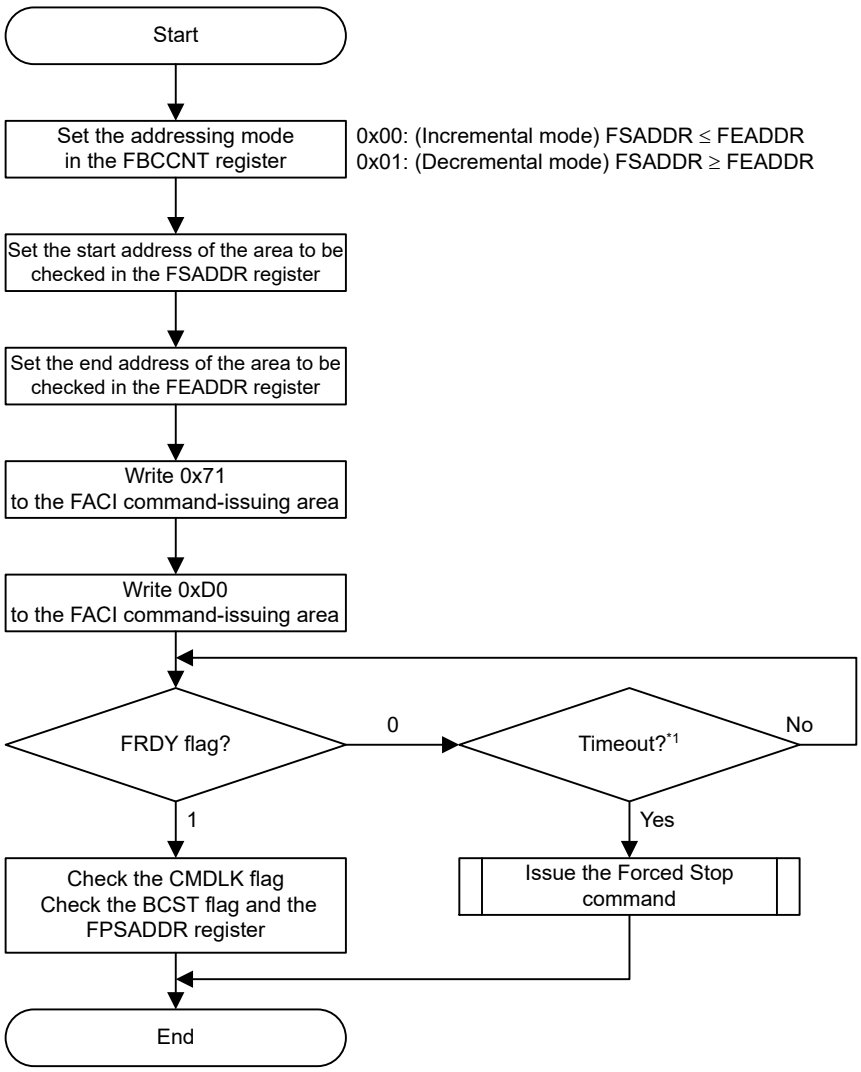
Before issuing the Blank Check command, set addressing mode, start address, and end address of the target area for Blank Check to the FBCCNT, FSADDR, and FEADDR registers. When Blank Check addressing mode is set to decremental mode (i.e. FBCCNT.BCDIR = 1), address specified in FSADDR should be equal to or larger than address in FEADDR.

On the other hand, the address in FSADDR should be equal to or smaller than address in FEADDR when Blank Check addressing mode is set to incremental mode (i.e. FBCCNT.BCDIR = 0).

If the settings of the BCDIR bit, FSADDR, and FEADDR are inconsistent, the flash sequencer enters the command-locked state. The size of the target area for Blank Check is in the range from 4 bytes to the data flash memory capacity and is set in units of 4 bytes.

Write 0x71 and 0xD0 to the FACI command-issuing area to start Blank Check. Completion of processing can be confirmed by the FRDY bit of the FSTATR register. At the end of processing, the result of Blank Check is stored in the BCST bit in the FBCSTAT register. If non-programmed data exists within the target area for Blank Check, flash sequencer stops Blank Check command operation. In this case, address of non-programmed data is indicated to FPSADDR register.

Figure 43.23 shows usage of the blank check command.



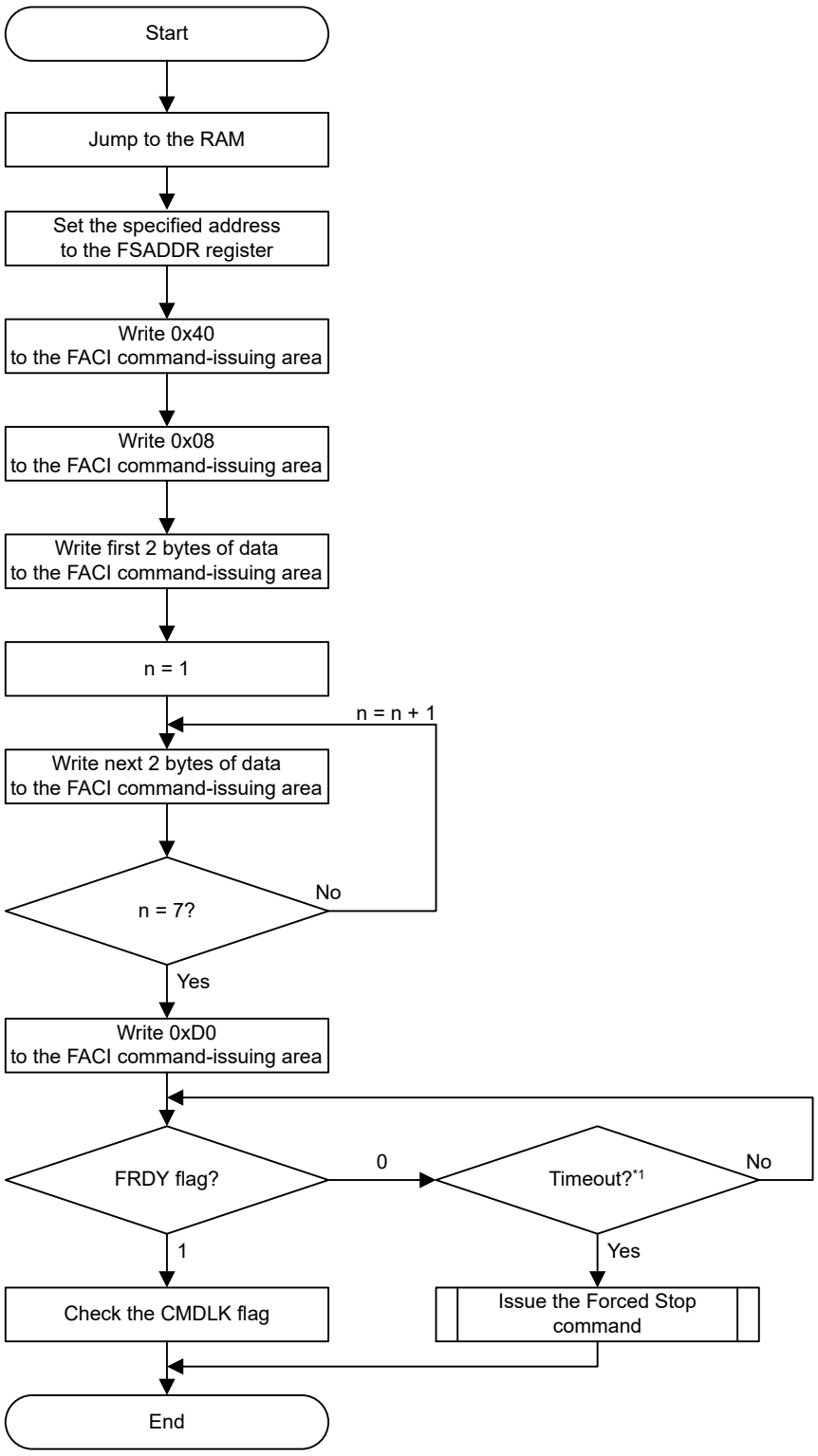
Note 1. Judgment of the timeout is based on 1.1 times the maximum time of Blank Checking for the whole target range. See section 46, Electrical Characteristics.

Figure 43.23 Usage flow of the blank check command

43.9.3.15 Configuration Set Command

The Configuration set command is used to set option-setting memory. Before issuing the Configuration set command, set the specified address (shown in Table 43.18) in the FSADDR register. Writing 0xD0 to the FACI command-issuing area in the final access for issuing the FACI command starts FACI processing of the Configuration set command.

Figure 43.24 shows usage of the configuration set command.



Note 1. Judgment of the timeout is based on 1.1 times the maximum time of programing in option-setting memory. See section 46, Electrical Characteristics.

Figure 43.24 Usage flow of the configuration set command

The correspondence between the possible target data for configuration setting and the address value set in the FSADDR register is shown in Table 43.18. For details, see section 43.4.12. FSADDR : FACI Command Start Address Register.

Table 43.18 Address Used by Configuration Set Command

Address	FSADDR Register Value	Setting Data	Operation of additional writing		Timing when the Setting is Enabled
			SAS.FSPR bit is 1	SAS.FSPR bit is 0	
0x0100_A100	0x0100_A100	Option Function Select Register 0 (OFS0)	Writable	Writable	At a reset
0x0100_A134	0x0100_A130	Start-up Area Setting Register (SAS)	Writable	Not writable*1	When a reset or command is executed
0x0100_A180	0x0100_A180	Option Function Select Register 1 (OFS1)	Writable	Writable	At a reset
0x0100_A1C0	0x0100_A1C0	Block Protect Setting Register (BPS)	Writable*2	Writable*2	When a reset or command is executed
0x0100_A1E0	0x0100_A1E0	Permanent Block Protect Setting Register (PBPS)	Writable*3 (from 1 to 0 only)	Writable*3 (from 1 to 0 only)	When a reset or command is executed
0x0100_A200	0x0100_A200	Option Function Select Register 1 Secure (OFS1_SEC)	Writable	Writable	At a reset
0x0100_A240	0x0100_A240	Block Protect Setting Register Secure (BPS_SEC)	Writable*4	Writable*4	When a reset or command is executed
0x0100_A260	0x0100_A260	Permanent Block Protect Setting Register Secure (PBPS_SEC)	Writable*5 (from 1 to 0 only)	Writable*5 (from 1 to 0 only)	When a reset or command is executed
0x0100_A280	0x0100_A280	Option Function Select Register 1 Select (OFS1_SEL)	Writable	Writable	At a reset
0x0100_A2C0	0x0100_A2C0	Block Protect Setting Register Select (BPS_SEL)	Writable	Writable	At a reset

Note 1. The SAS.FSPR bit cannot be restored to 1 by using the Configuration set command once it is set to 0. Therefore, setting the start-up area select flags again becomes impossible. (when the Configuration set command is issued to the address of 0x0100A134, the command is locked.) Exercise extra caution when handling the SAS.FSPRbit.
Note 2. Once PBPS[n] bit is set to 0, the BPS[n] bit cannot be restored to 1 by using the Configuration set command.
Note 3. Once these bits are set to 0, the bits cannot be restored to 1 by using the Configuration set command. The PBPS[n] bit cannot be set to 0 by using the Configuration set command when the BPS[n] bit is 1.
Note 4. Once PBPS_SEC[n] bit is set to 0, the BPS_SEC[n] bit cannot be restored to 1 by using the Configuration set command.
Note 5. Once these bits are set to 0, the bits cannot be restored to 1 by using the Configuration set command. The PBPS_SEC[n] bit cannot be set to 0 by using the Configuration set command when the BPS_SEC[n] bit is 1.

43.10 Suspend Operation

Reading from the flash memory is not possible during programming or erasure if the conditions for background operation given in Table 43.29 are not satisfied. When a P/E suspend command is issued to suspend the programming or erasure of the flash memory, reading from the flash memory is enabled. Regarding P/E suspend commands, there are one suspend command mode for programming and two suspend command modes for erasure (suspension priority mode and erasure priority mode). To resume suspended programming or erasure, the P/E resume command is available. For details on the suspend operation, see Figure 43.16.

43.11 Protection Function

43.11.1 Software Protection

Software protection disables programming and erasure of the code flash memory through the settings of control registers and block protect setting in the user area. If an attempt is made to issue an FACI command against software protection, the flash sequencer enters the command-locked state.

43.11.1.1 Protection through FWEPROR

Unless the FWEPROR.FLWE[1:0] bits are set to 01b, programming cannot proceed in any mode.

43.11.1.2 Protection by FENTRYR

When the FENTRYR register is set to 0x0000, the flash sequencer enters read mode. In read mode, FACI commands cannot be accepted. If an attempt is made to issue an FACI command in read mode, the flash sequencer enters the command-locked state.

43.11.1.3 Protection by Block Protect Setting

Each block in user area has the block protect setting (BPS or BPS_SEC). When the FBPROT0 or FBPROT1 register is 0x0000 and the block protect bit is 0, issuing the Program or Block Erase command to user area of the code flash causes the command-locked state. To program or erase the block whose block protect bit is 0, set the FBPROT0 or FBPROT1 register to 0x0001.

The block protect setting can be locked by the permanent block protect setting (PBPS or PBPS_SEC). When the permanent block protect setting and the block protect setting are 0, issuing a Program or Block erase command to user area of the code flash causes the flash sequencer to enter the command-locked state regardless of the FBPROT0 and FBPROT1 register settings.

Valid block protect setting (BPS or BPS_SEC) depends on the Block Protect Select bit (BPS_SEL).

See section 43.12.2. Permanent Block Protect Setting for details of the block protect setting and permanent block protect setting. See section 43.4.15. FBPROT0 : Flash Block Protection Register and section 43.4.16. FBPROT1 : Flash Block Protection for Secure Register for more information.

For details of block protect setting (BPS or BPS_SEC) and block protect select (BPS_SEL), see section 6, Option-Setting Memory.

The protected area by the block protect setting is always determined by the address of the FSADDR register setting regardless of the address swapping function setting (the startup area select). Table 43.19 to Table 43.20 show the relation of user area and the block protect setting in each function setting.

- BPS[0] to BPS[n] or BPS_SEC[0] to BPS_SEC[n] are assigned to the block of user area (for example, address is 0x00_0000 to the last block address).
- BPS[0]/BPS_SEC[0] and BPS[1]/BPS_SEC[1] are assigned to the block of user area depending on the startup area select setting (SAS.BTFLG bit). (See section 43.11.3. Start-Up Program Protection).

Table 43.19 shows the block protect setting when the startup area select is disabled (not swapping).

Table 43.20 show example of the block protect setting when the address conversion function is used.

Table 43.19 Example of Block Protect setting when SAS.BTFLG is 1

FSADDR[23:0]	Block size	Block protect setting	User area block number	Notes
The last block address	32 KB	BPS[n] or BPS_SEC[n]	Block n	—
⋮	⋮	⋮	⋮	—
0x01_8000 to 0x01_FFFF	32 KB	BPS[9] or BPS_SEC[9]	Block 9	—
0x01_0000 to 0x01_7FFF	32 KB	BPS[8] or BPS_SEC[8]	Block 8	—
0x00_E000 to 0x00_FFFF	8 KB	BPS[7] or BPS_SEC[7]	Block 7	—
0x00_C000 to 0x00_DFFF	8 KB	BPS[6] or BPS_SEC[6]	Block 6	—
⋮	⋮	⋮	⋮	—
0x00_2000 to 0x00_3FFF	8 KB	BPS[1] or BPS_SEC[1]	Block 1	Not swap block 0 and block 1 in this startup area select setting
0x00_0000 to 0x00_1FFF	8 KB	BPS[0] or BPS_SEC[0]	Block 0	Not swap block 0 and block 1 in this startup area select setting

Table 43.20 Example of Block Protect setting when SAS.BTFLG is 0

FSADDR[23:0]	Block size	Block protect setting	User area block number	Notes
The last block address	32 KB	BPS[n] or BPS_SEC[n]	Block n	—
⋮	⋮	⋮	⋮	—
0x01_8000 to 0x01_FFFF	32 KB	BPS[9] or BPS_SEC[9]	Block 9	—
0x01_0000 to 0x01_7FFF	32 KB	BPS[8] or BPS_SEC[8]	Block 8	—
0x00_E000 to 0x00_FFFF	8 KB	BPS[7] or BPS_SEC[7]	Block 7	—
0x00_C000 to 0x00_DFFF	8 KB	BPS[6] or BPS_SEC[6]	Block 6	—
⋮	⋮	⋮	⋮	—
0x00_2000 to 0x00_3FFF	8 KB	BPS[1] or BPS_SEC[1]	Block 0	Swap block 0 and block 1 in this startup area select setting
0x00_0000 to 0x00_1FFF	8 KB	BPS[0] or BPS_SEC[0]	Block 1	Swap block 0 and block 1 in this startup area select setting

43.11.2 Error Protection

Error protection detects the issuing of illegal FACI commands, illegal access, and flash sequencer malfunction. FACI command acceptance is disabled (command-locked state) in response to the detection of these errors. The flash memory cannot be programmed or erased while the flash sequencer is in the command-locked state. For release from the command-locked state, issue the Status Clear or Forced Stop command. The Status Clear command can only be used while the FRDY bit in the FSTATR register is 1. The Forced Stop command can be used regardless of the value of the FRDY bit. While the CMDLKIE bit in the FAEINT register is 1, a flash access error (FIFERR) interrupt is generated if the flash sequencer enters the command-locked state (the CMDLK bit in the FASTAT register is set to 1).

If the flash sequencer enters the command-locked state in response to a command other than the P/E suspend command during programming or erasure processing, the flash sequencer continues the processing for programming or erasure. In this state, the P/E suspend command cannot be used to suspend the processing for programming or erasure. If a command is issued in the command-locked state, the ILGLERR bit becomes 1 and the other bits retain the values set from previous error detection.

Table 43.21 shows the error protection types and status bit values after error detections.

Table 43.21 Error protection type (1 of 3)

Error type	Description	ILGCOMERR	FESETERR	SECERR	OTERR	ILGLERR	ERSERR	PRGERR	FLWEERR	CFAE	DFAE
FENTRYR setting error	The value set in FENTRYR is not 0x0000, 0x0001, or 0x0080	0	1	0	0	1	0	0	0	0	0
	The FENTRYR setting at suspension is different from that at resumption	0	1	0	0	1	0	0	0	0	0

Table 43.21 Error protection type (2 of 3)

Error type	Description	ILGCOMERR	FESETERR	SECERR	OTERR	ILGLERR	ERSERR	PRGERR	FLWEERR	CFAE	DFAE
Illegal command error	An undefined size is specified in the first cycle of the command. (not byte-write)	1	0	0	0	1	0	0	0	0	0
	An undefined code is written in the first access of the FACI command	1	0	0	0	1	0	0	0	0	0
	The value specified in the last access of the multiple-access FACI command is not 0xD0	1	0	0	0	1	0	0	0	0	0
	The value (N) specified in the second write access of the FACI command in the program or configuration set command is wrong	1	0	0	0	1	0	0	0	0	0
	Blank Check command is issued with inconsistent BCDIR, FSADDR, and FEADDR settings (see section 43.4.13. FEADDR : FACI Command End Address Register)	1	0	0	0	1	0	0	0	0	0/1 *1
	A multi block erase command is issued with inconsistent FSADDR and FEADDR settings. • FSADDR > FEADDR • FEADDR is set to reserved area.	1	0	0	0	1	0	0	0	0	0/1 *1
	An FACI command not acceptable in each mode is issued (see Table 43.15)	1	0	0	0	1	0	0	0	0	0
	An FACI command is issued when command acceptance conditions are not satisfied (see Table 43.16)	0/1	0/1	0/1	0/1	1	0/1	0/1	0/1	0/1	0/1
	A program or block erase command is issued against the area protected by the block protect setting (see section 43.11.1.3. Protection by Block Protect Setting)	1	0	0	0	1	0	0	0	0	0
	A program command is issued against the erase area in erase suspend	1	0	0	0	1	0	0	0	0	0
Erasure error	An error occurs during erasure	0	0	0	0	0	1	0	0	0	0
Programming error	An error occurs during programming	0	0	0	0	0	0	1	0	0	0
Code flash memory access violation	An FACI command is issued to the reserved portion of the user area in code flash P/E mode	0	0	0	0	1	0	0	0	1	0
	Configuration set command is issued to the reserved option-setting memory	0	0	0	0	1	0	0	0	1	0
	Configuration set command of non-secure access is issued to the secure region of TrustZone in the code flash	0	0	0	0	1	0	0	0	1	0
	Program or block erase command of non-secure access is issued to the secure region of user area.	0	0	0	0	1	0	0	0	1	0

Table 43.21 Error protection type (3 of 3)

Error type	Description	ILGCOMERR	FESETERR	SECERR	OTERR	ILGLERR	ERSERR	PRGERR	FLWEERR	CFAE	DFAE
Data flash memory access violation	A program or block erase command is issued to the reserved data area in data flash P/E mode	0	0	0	0	1	0	0	0	0	1
	A multi block erase command is issued to the reserved data area in data flash P/E mode. (FSADDR is set to reserved data area).	1	0	0	0	1	0	0	0	0	1
	Blank Check command is issued to reserved data area in data flash P/E mode. (FSADDR is set to reserved data area).	1	0	0	0	1	0	0	0	0	1
	A program, block erase, multi block erase, or blank check command of non-secure access is issued to the secure region of data area.	0	0	0	0	1	0	0	0	0	1
Security error	Configuration set command for the SAS.BTFLG bit setting is issued when the SAS.FSPR bit is 0 (see section 43.9.3.15. Configuration Set Command)	0	0	1	0	1	0	0	0	0	0
Others	An FACI command-issuing area is accessed in read mode	0	0	0	1	1	0	0	0	0	0
	An FACI command-issuing area is read in code flash P/E mode or data flash P/E mode	0	0	0	1	1	0	0	0	0	0
Flash write erase protection error	A flash memory write protection error is detected by the FWEPROR register setting*2 during command processing by the flash sequencer	0	0	0	0	0	0/1	0/1	1	0	0

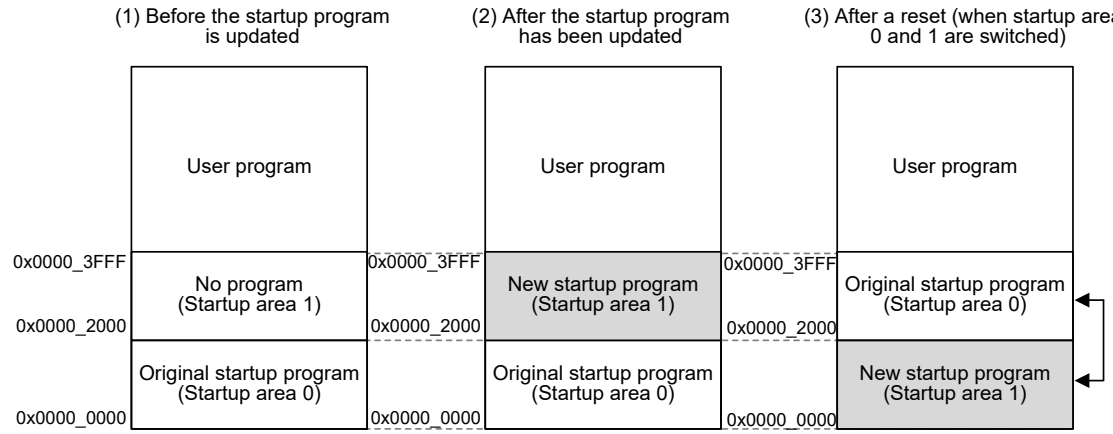
Note 1. DFAE value depends on the FSADDR setting.
Note 2. For details on the FWEPROR register, see section 43.4.8. FWEPROR : Flash P/E Protect Register.

43.11.3 Start-Up Program Protection

Protection of the startup program is for protection of the program to be started after a reset (the startup program). This function provides a way to safely update the startup program when rewriting is suspended during a reset.

The startup area is 8 Kbytes in size and is assigned to the user area in the code flash memory. This function uses the values of the SAS.BTFLG bit and the FSUACR.SAS[1:0] bits to change the area where the startup program is stored in block units (see Figure 43.25 to Figure 43.28).

In protection of the startup program, the state of the selection of the startup area can be fixed by the FSPR bit. However, the SAS.FSPR bit never be restored to 1 once the flag is set to 0. Exercise extra caution when handling the SAS.FSPR bit.



Note: The addresses are used for reading, programming, and erasure. Startup area 0 is erased when Block Erase command is issued for the address 0x00000000 with allocation shown under 3.

1. Before the startup program is updated
   When the SAS.BTFLG bit is 1 and the FSUACR.SAS[1:0] bits are 0xb, the address range from 0x00002000 to 0x00003FFF is selected as startup area 1 and the address range from 0x00000000 to 0x00001FFF is selected as startup area 0.
2. After the startup program has been updated
   A new startup program can be written to startup area 1. At this time, startup area 0 will still contain the previous startup program, so rewriting by starting up from startup area 0 remains possible if rewriting to the startup area 1 fails.
3. After a reset (when startup areas 0 and 1 are switched)
   When rewriting startup area 1 is successfully completed, switch the assignment of startup areas 0 and 1 by rewriting the startup area selection bit (the SAS.BTFLG bit) with Configuration set command. Afterwards, the program is started up from startup area 1 after a reset.

Figure 43.25 Concept of Protection of the Startup Program

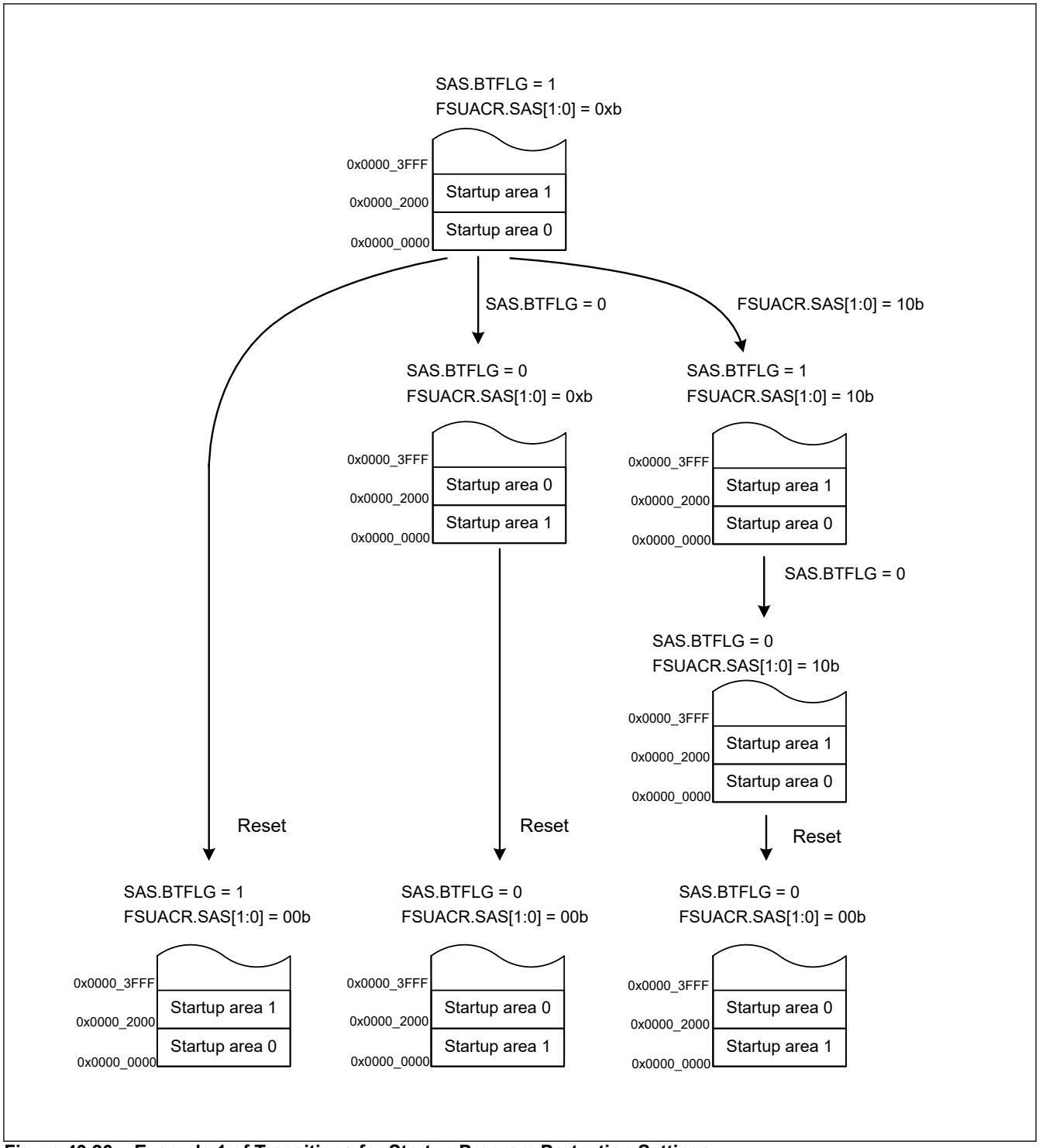


Figure 43.26 Example 1 of Transitions for Startup Program Protection Settings

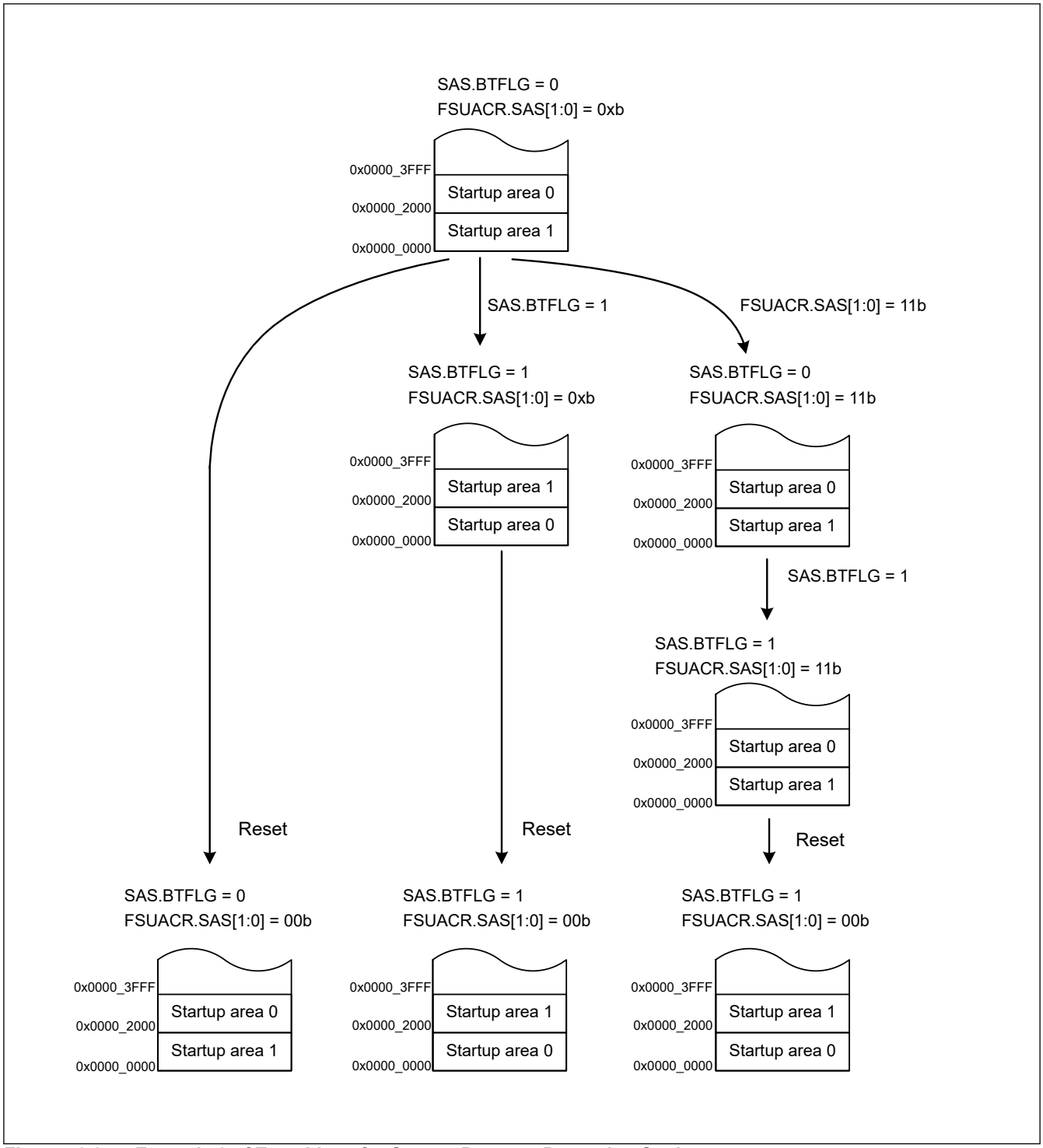
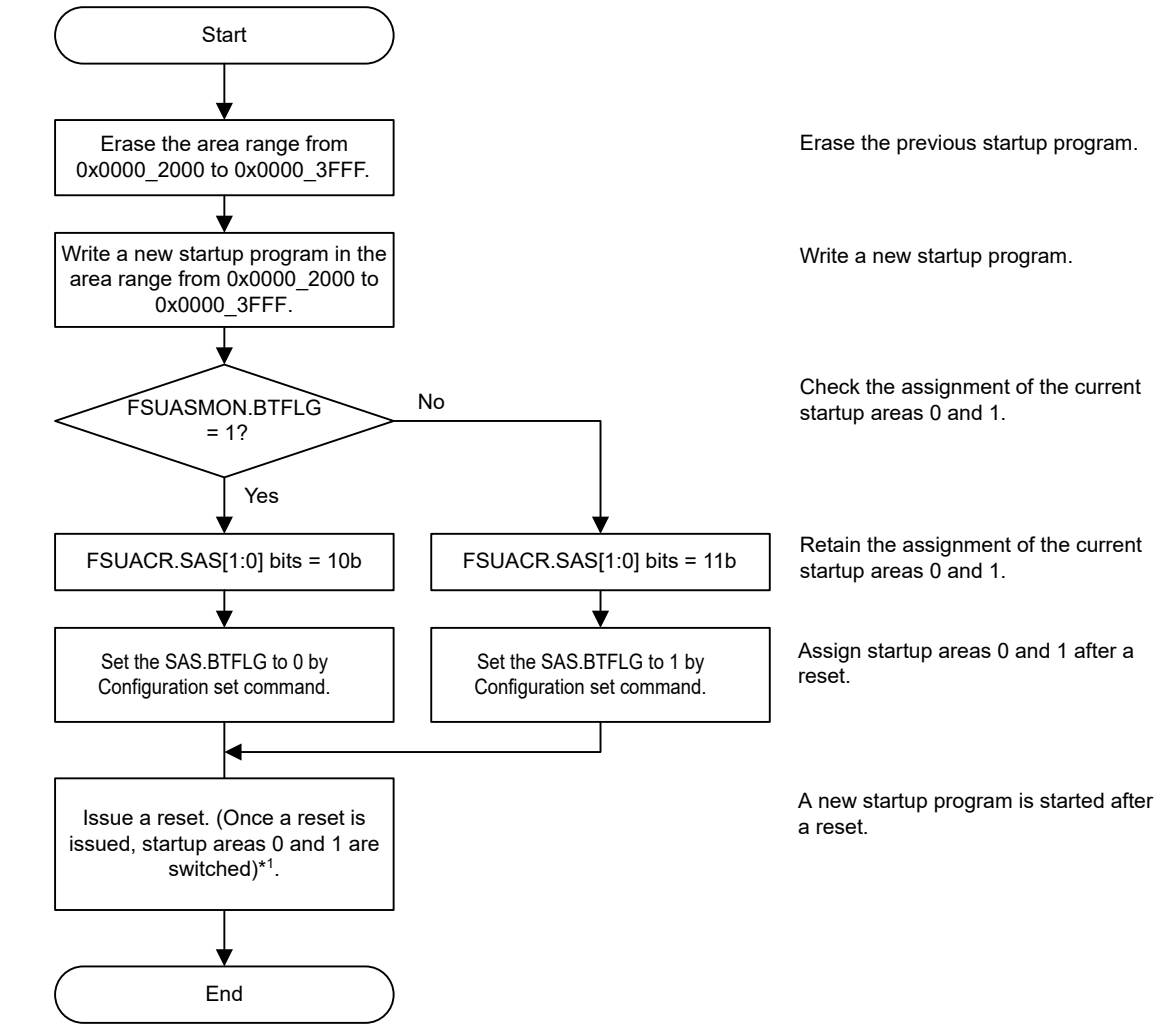


Figure 43.27 Example 2 of Transitions for Startup Program Protection Settings



Note 1. For details, see Figure 43.26 and Figure 43.27.

Figure 43.28 Concept of Protection of the Startup Program

43.12 Security Function

The flash sequencer supports the following security functions:

- Security flag for startup area
- Permanent block protect setting
- Flash memory protection for TrustZone

43.12.1 Security Flag for Startup Area Select

The security flag (SAS.FSPR) for the startup area is located in the option-setting memory.

When the SAS.FSPR bit is 0, issuing the configuration set command to change the SAS.BTFLG bit causes the flash sequencer to be in the command-locked state. Also, when the SAS.FSPR bit is 0, it is invalid to write to the Startup Area Select bits SAS[1:0] in the FSUACR register. The SAS.FSPR bit enables protection.

43.12.2 Permanent Block Protect Setting

The permanent block protect setting is the clear protection for the block protection setting. User area cannot be permanently updated by the FACI command when the permanent block protect setting is enabled. See section 43.11.1.3. Protection by Block Protect Setting for more details.

The block protect setting and the permanent block protect setting have the write/clear protection against the configuration set command. The flash sequencer does not detect an error when the configuration set command is issued to the write/clear protected settings.

Figure 43.29 and Table 43.22 show the write/clear protection against the block protect setting (BPS[n]) and the permanent protect setting (PBPS[n]). Figure 43.30 and Table 43.23 show the write/clear protection against the block protect setting for secure (BPS_SEC[n]) and permanent protect setting for secure (PBPS_SEC[n]).

Effective permanent block protect setting (PBPS or PBPS_SEC) depends on block protect select (BPS_SEL). For details of permanent block protect setting (PBPS or PBPS_SEC) and block protect select (BPS_SEL), see section 6, Option-Setting Memory.

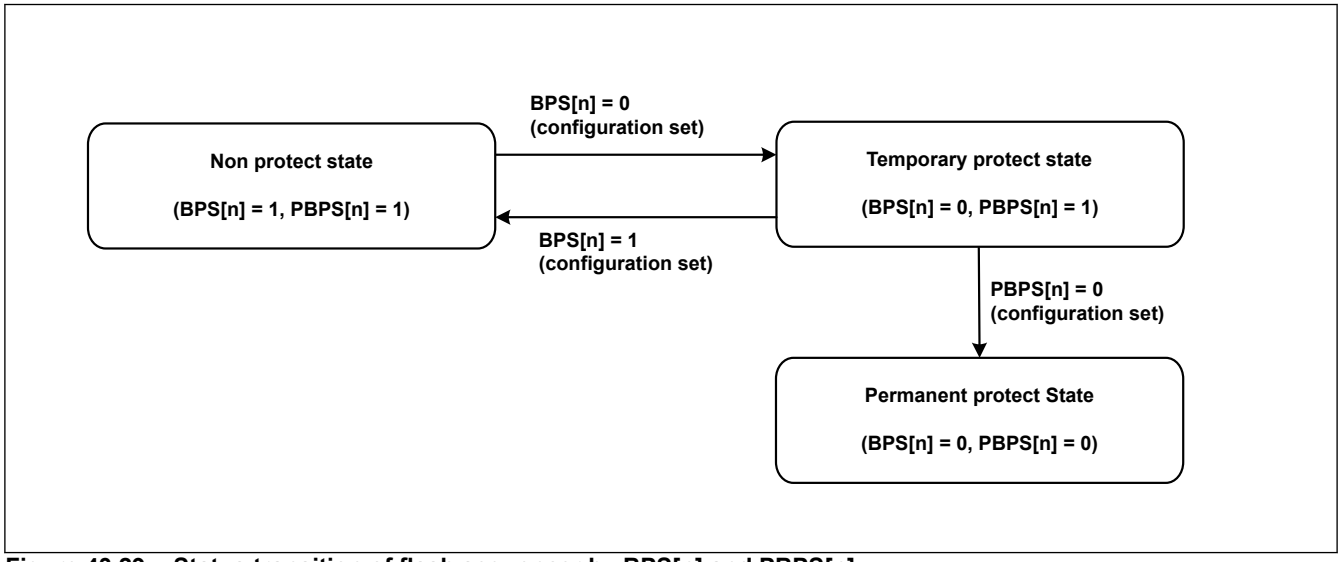


Figure 43.29 Status transition of flash sequencer by BPS[n] and PBPS[n]

Table 43.22 Write/clear protection of BPS[n] and PBPS[n]

Current state		Updatable state by configuration set command			
BPS[n]	PBPS[n]	BPS[n] = 1	BPS[n] = 0	PBPS[n] = 1	PBPS[n] = 0
1	1	✓	✓	✓	X
1	0	—	—	—	—
0	1	✓	✓	✓	✓
0	0	X	✓	X	✓

Note:
- ✓ indicates updatable by configuration set command.
- X indicates not updatable by configuration set command (error does not occur).
- — indicates not reaching to this state.

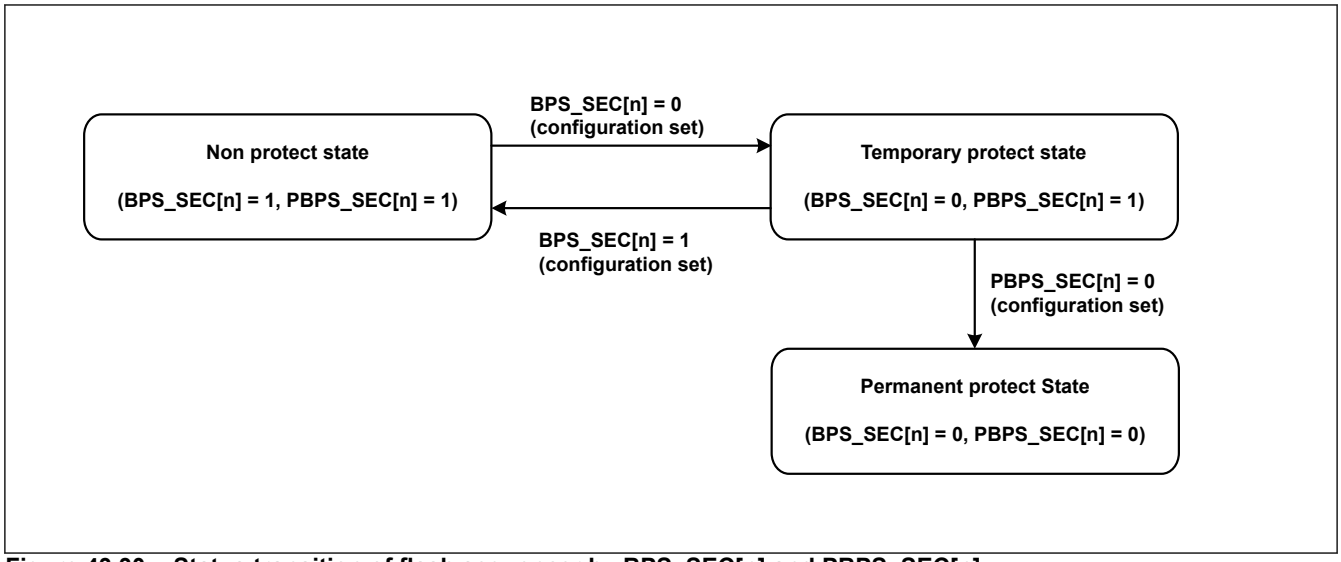


Figure 43.30 Status transition of flash sequencer by BPS_SEC[n] and PBPS_SEC[n]

Table 43.23 Write/clear protection of BPS_SEC[n] and PBPS_SEC[n]

Current state		Updatable state by configuration set command			
BPS_SEC[n]	PBPS_SEC[n]	BPS_SEC[n] = 1	BPS_SEC[n] = 0	PBPS_SEC[n] = 1	PBPS_SEC[n] = 0
1	1	✓	✓	✓	X
1	0	—	—	—	—
0	1	✓	✓	✓	✓
0	0	X	✓	X	✓

Note:
- ✓ indicates updatable by configuration set command.
- X indicates not updatable by configuration set command (error does not occur).
- — indicates not reaching to this state.

43.12.3 Flash Memory Protection for TrustZone

Information in this section focuses on the flash sequencer operation.

The flash memory provides the following types of protect function against non-secure access:

- Protection for flash memory area (P/E)
- Protection for flash memory area (read)
- Protection for registers
- Protection during FACI command operation
- Code flash P/E mode entry protection

43.12.3.1 Protection for Flash Memory Area (P/E)

This function protects the secure region of the code flash and data flash from FACI commands of non-secure access. The condition of protection depends on the FACI command, the access attribution, and the memory boundary setting.

For details of secure region, see section 45, Security Features .

See Table 43.24 for information on protection of the flash memory area (P/E).

Table 43.24 Protection for the flash memory area (P/E)

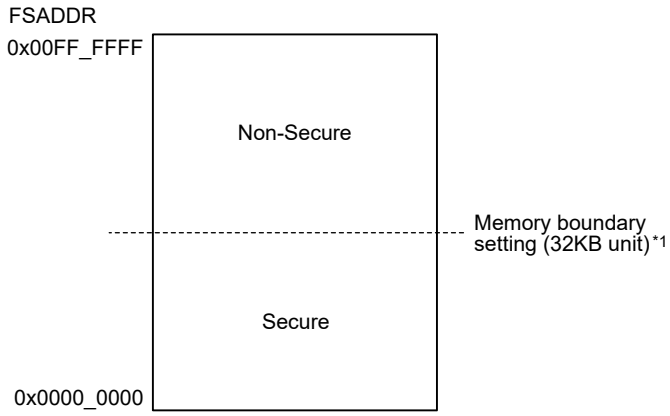
FACI command	Target area		Issuing of FACI command by non-secure access	Issuing of FACI command by secure access
Program Block erase	Code flash memory	User area (non-secure area)	✓	✓
		User area (secure area)	X	✓
	Data flash memory	Data area (non-secure area)	✓	✓
		Data area (secure area)	X	✓
Multi block erase Blank check	Data flash memory	Data area (non-secure area)	✓	✓
		Data area (secure area)	X	✓
Configuration set	Code flash memory	option-setting memory (non-secure area)	✓	✓
		option-setting memory (secure area)	X	✓

Note:
- ✓ indicates FACI command operation is not prohibited.
- X indicates FACI command operation is prohibited. Error occurs when the area is selected, and the FACI command is executed.

When the target area of FACI command is the user area of code flash, the flash sequencer compares the FSADDR register setting with the memory boundary setting of the code flash and determines whether the target area is in the secure region.

The memory boundary can be set to 0x0000_0000 to 0x00FF_8000 in 32 KB unit.

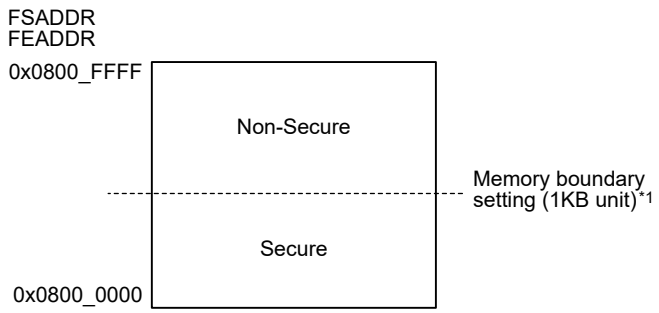
Figure 43.31 shows details of the non-secure/secure attribute of user area in the code flash.



Note 1. Comparing FSADDR with memory boundary setting of user area and judges whether the target area is in secure area.

Figure 43.31 Secure/non-secure region in user area

When the target area of the issuing FACI command is the data area of data flash, the flash sequencer compares the FSADDR/FEADDR register setting with the memory boundary setting of the data flash and determines whether the target area is in secure region. The memory boundary can be set to 0x0800_0000 to 0x0800_FC00 in 1 KB unit. Figure 43.32 shows details of the non-secure/secure attribute of data area in the data flash.



Note 1. Comparing FSADDR/FEADDR with memory boundary setting of data area and judges whether the target area is in secure area.

Figure 43.32 Secure/non-secure region in data area

See Figure 43.33 for details of non-secure/secure region of option-setting memory. The flash sequencer judges that target area is secure region from the FSADDR register setting.

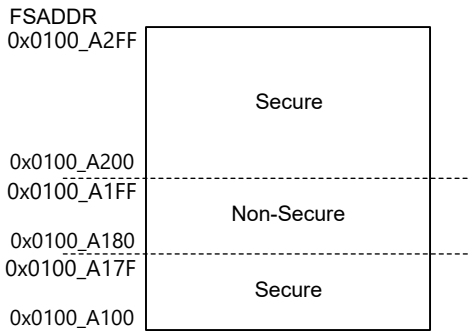


Figure 43.33 Secure/non-secure region in option-setting memory

43.12.3.2 Protection for Flash Memory Area (Read)

This function protects the secure region of code flash and data flash from non-secure bus access.

For details of secure region, see section 45, Security Features .

43.12.3.3 Protection for Register

The flash sequencer registers have write-access protection against non-secure access. Table 43.25 shows details of the protected registers of the flash sequencer.

Table 43.25 Protected registers of the flash sequencer for TrustZone (1 of 2)

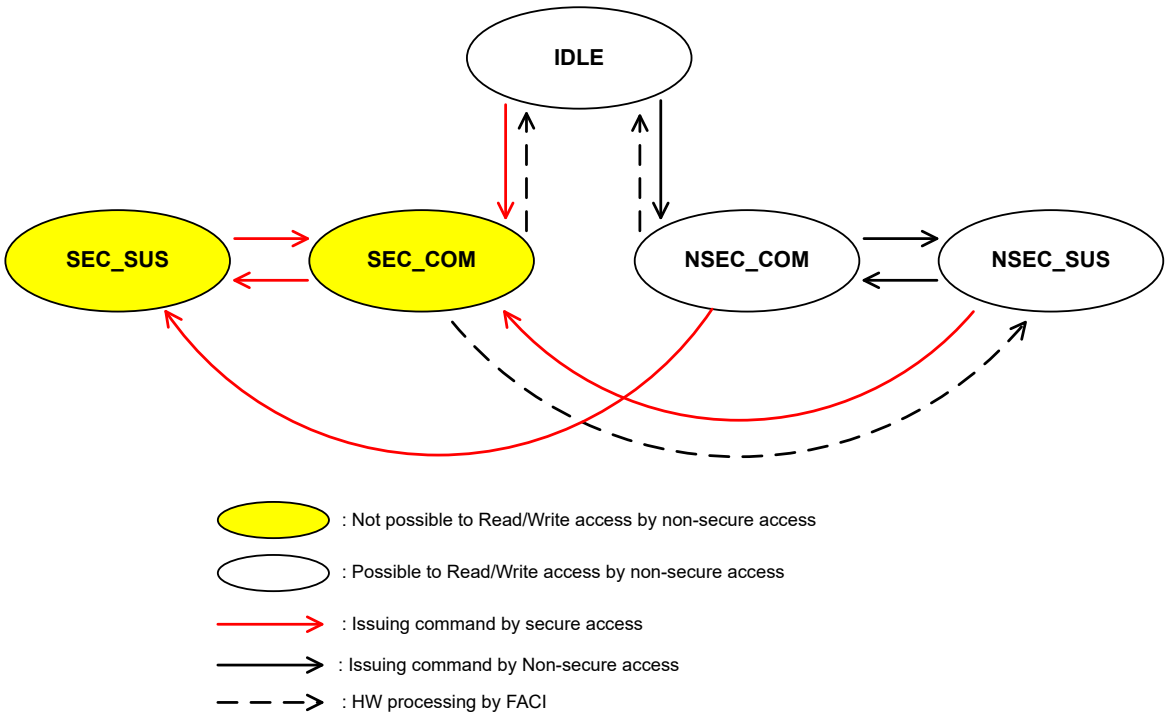
Protection target register	Security attribute setting	Notes
FCKMHZ	Security attribution register setting (FSAR.FCKMHZSA)	See section 43.4.4. FSAR : Flash Security Attribution Register
FMEPROT	Always secure	See section 43.4.14. FMEPROT : Flash P/E Mode Entry Protection Register

Table 43.25 Protected registers of the flash sequencer for TrustZone (2 of 2)

Protection target register	Security attribute setting	Notes
FBPROT1	Always secure	See section 43.4.16. FBPROT1 : Flash Block Protection for Secure Register
FSUACR	Always secure	See section 43.4.27. FSUACR : Flash Startup Area Control Register
FACI command-issuing area and all registers of FACI (Base address is FACI) and FWEPROR register	During FACI command processing by secure access	See section 43.12.3.4. Protection during FACI Command Operation

43.12.3.4 Protection during FACI Command Operation

This function protects read/write access to the FACI command-issuing area, including all registers of FACI (Base address is FACI) and FWEPROR register by the non-secure access during the FACI command processing of the secure access. The protect condition includes the suspending period of the program, block erase, or multi block erase command by the P/E suspend command of the secure access. See Figure 43.34 and Table 43.26 for details of the protection during the FACI command operation.



Note: IDLE: State of not executing FACI command
SEC_COM: State of executing FACI command by secure access
SEC_SUS: State of "P/E Suspend" by secure access
NSEC_COM: State of executing FACI command by non-secure access
NSEC_SUS: State of P/E Suspend by non-secure access

Figure 43.34 State of protection during FACI command operation

Table 43.26 Protection during FACI command operation

	Flash sequencer is not operating	Program, Block erase, Multi block erase, Blank check, or Configuration set command processing		Command lock state		Forced stop command processing		While suspend Program, Block erase, or Multi block erase command		Program command processing while suspend Block erase or Multi block erase command by secure access		Program command processing while suspend Block erase or Multi block erase command by non-secure access		P/E resume command processing while suspend Program, Block erase, or Multi block erase command by secure access		P/E Resume command processing while suspend Program, Block erase, or Multi block erase command by non-secure access	
FACI command attribute	—	S	NS	S	NS	S	NS	S	NS	S	NS *1	S	NS	S	NS *1	S	NS
FRDY bit	1	0	0	1	1	0	0	1	1	0	0	0	0	1	1	0	0
PRGSPD or ERSSPD bit	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0
CMDLK bit	0	0	0	1	1	1/0	1/0	1/0	1/0	0	0	0	0	0	0	0	0
Non-secure access	✓	X	✓	✓	✓	X	✓	X	✓	X	✓	X	✓	X	✓	X	✓

Note:
- S indicates the FACI command by the secure access.
- NS indicates the FACI command by the non-secure access.
- ✓ indicates read/write access is possible by the non-secure access.
- X indicates read/write access is not possible by the non-secure access. Write data is ignored and read data is always 0.

Note 1. The FACI command issued by the non-secure access is not allowed.

Code flash programming/erasure can be protected by the FMEPROT register of secure function. Therefore, it does not assume that secure function issues P/E suspend command during code flash programming/erasure of non-secure function.

Data flash programming/erasure of non-secure can be suspended by secure function. If secure function issues P/E suspend command during data flash programming/erasure of non-secure function, secure function should issue P/E resume command. When secure function issues P/E resume command, secure function should notify non-secure function that data flash programming/erasure is complete and return to non-secure function. See Figure 43.35 and Figure 43.36 in example of issuing P/E suspend of secure function during programming/erasure of non-secure function.

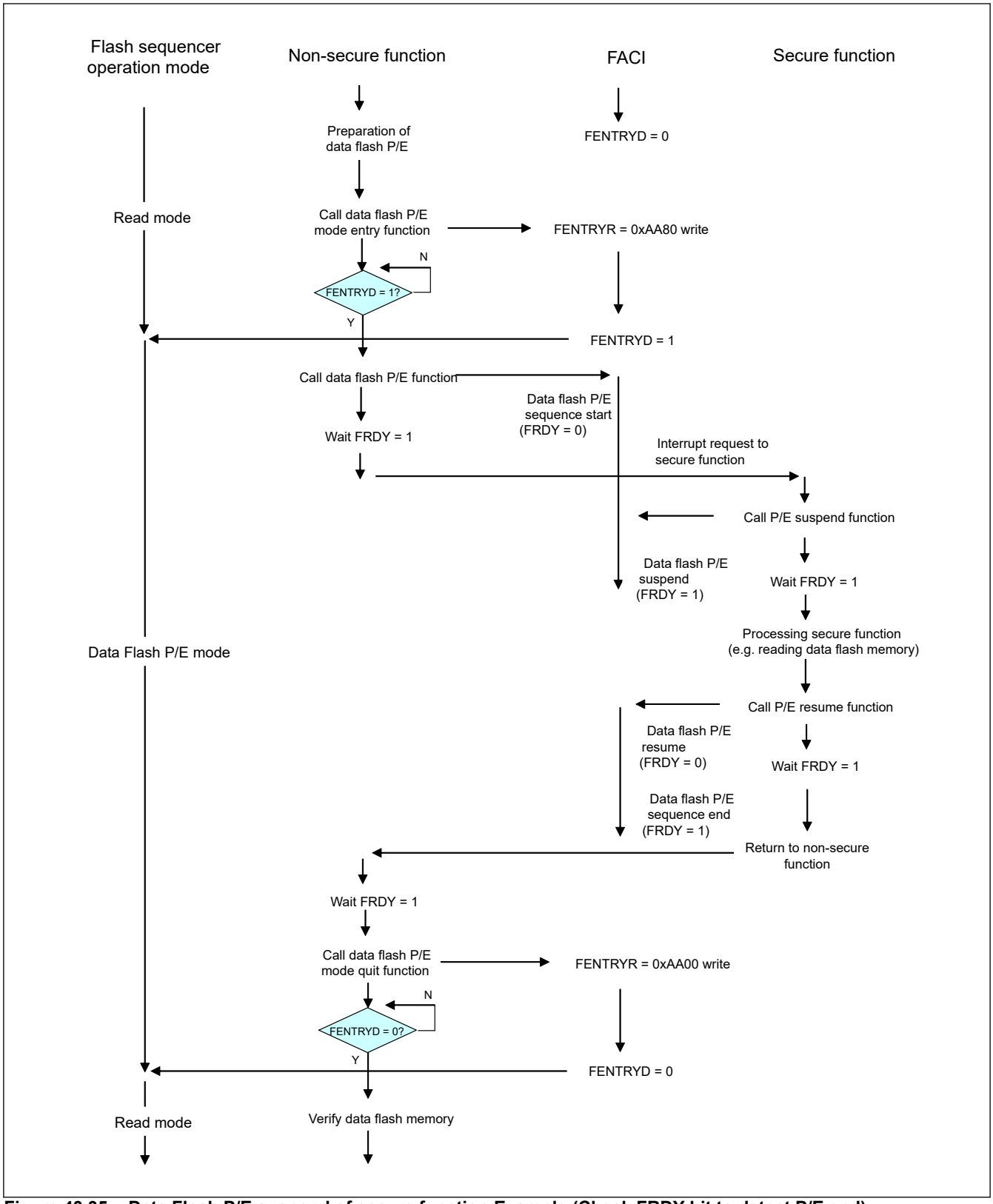


Figure 43.35 Data Flash P/E suspend of secure function Example (Check FRDY bit to detect P/E end)

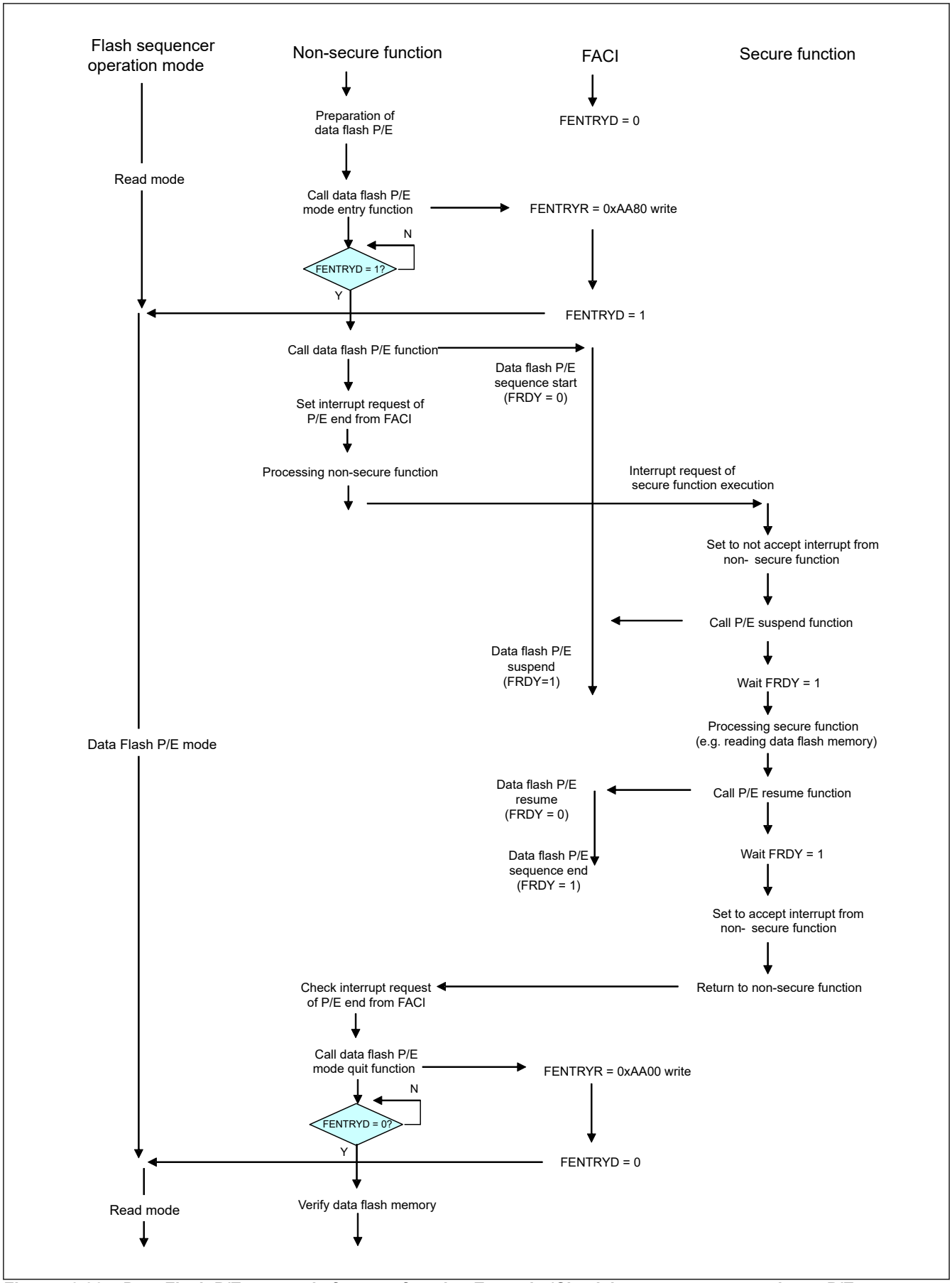


Figure 43.36 Data Flash P/E suspend of secure function Example (Check interrupt request to detect P/E end)

43.12.3.5 Code Flash P/E Mode Entry Protection

The flash sequencer has protection function of code flash P/E by the FMEPROT register for the secure developer. Secure function can prevent disturbance of reading code flash memory by this protection function. See section 43.4.14. FMEPROT : Flash P/E Mode Entry Protection Register.

For applications that do not require non-secure region programming/erasure other than from secure function, it is recommended to always disable non-secure function of code flash programming/erasure by enabling the protection function of FMEPROT register.

For details, see Figure 43.37 of the code flash P/E sequence example by non-secure function.

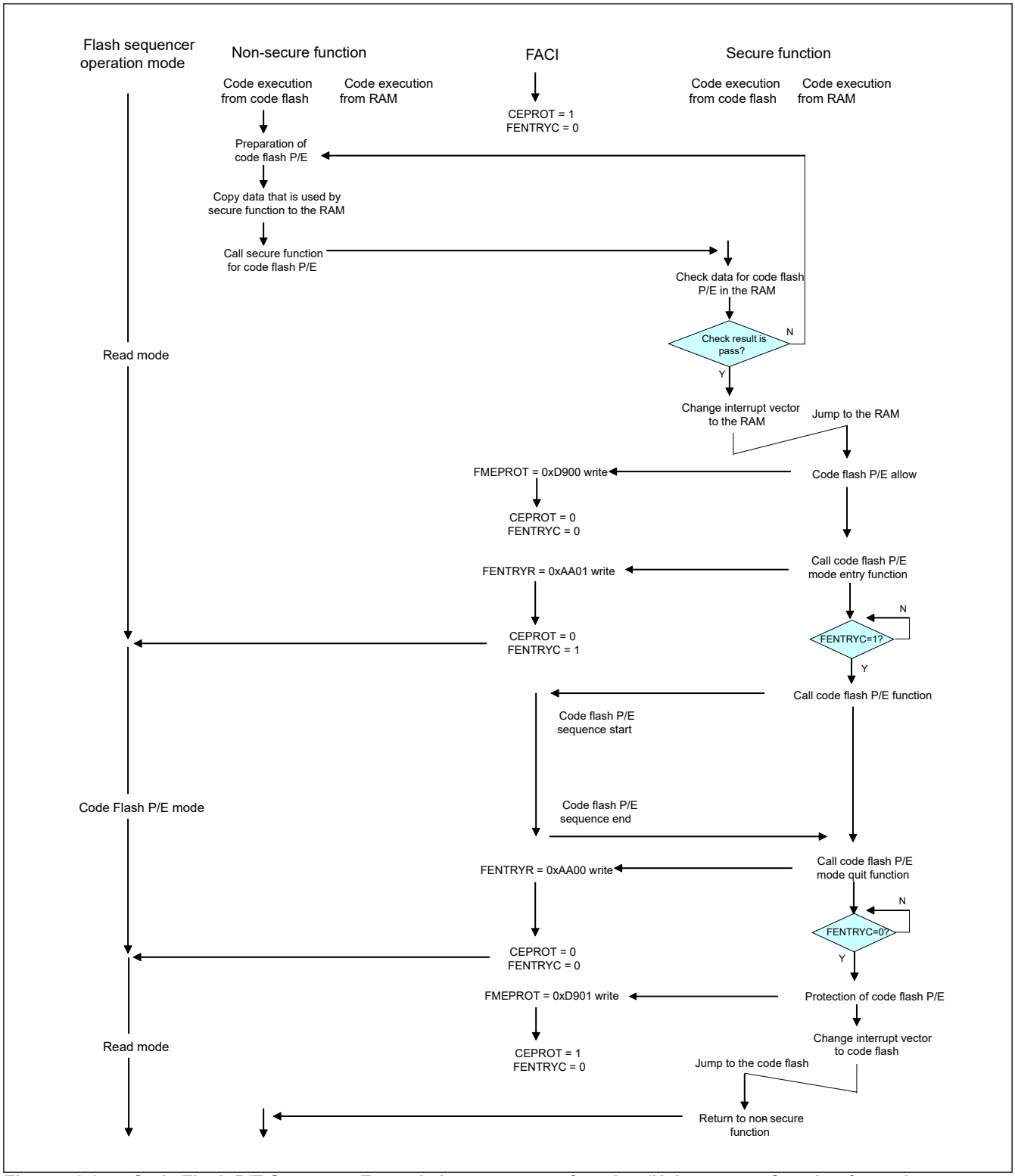


Figure 43.37 Code Flash P/E Sequence Example by non-secure function (Using secure function for code flash P/E)

43.13 Boot Mode

There are two serial programming modes; the boot mode (for the SCI interface) with SCI9. Table 43.27 lists the I/O pins used in boot mode. Table 43.28 lists the available communication interface used in the boot mode.

Table 43.27 I/O Pins Used in Boot Mode

Pin Name	I/O	Mode to be Used	Use
MD	Input	Boot mode (for the SCI interface)	Selection of operating mode
PA15/RXD9	Input	Boot mode (for the SCI interface)	For host communication (to receive data through SCI)
PB03/TXD9	Output		For host communication (to transmit data through SCI)

Table 43.28 Available Communication Interface Used in Boot Mode

Main clock oscillator or external clock is connected	Yes	No	No
Tool connection time*1	Up to 1 second	Up to 2 seconds	Up to 3 seconds

Note 1. See the boot firmware manual for the detail connection time.

43.13.1 Boot Mode (for the SCI Interface)

In boot mode (for the SCI interface), the host sends control commands and data for programming, and the flash memory is programmed or erased accordingly. An on-chip SCI handles transfer between the host and this MCU in asynchronous mode. Tools for transmission of control commands and the data for programming must be prepared in the host.

When this MCU is activated in boot mode (for the SCI interface), the program on the dedicated area the MCU is executed. The boot program automatically adjusts the bit rate of the SCI and controls programming/erasure by receiving control commands from the host.

Figure 43.38 shows the system configuration for operations in boot mode (for the SCI interface).

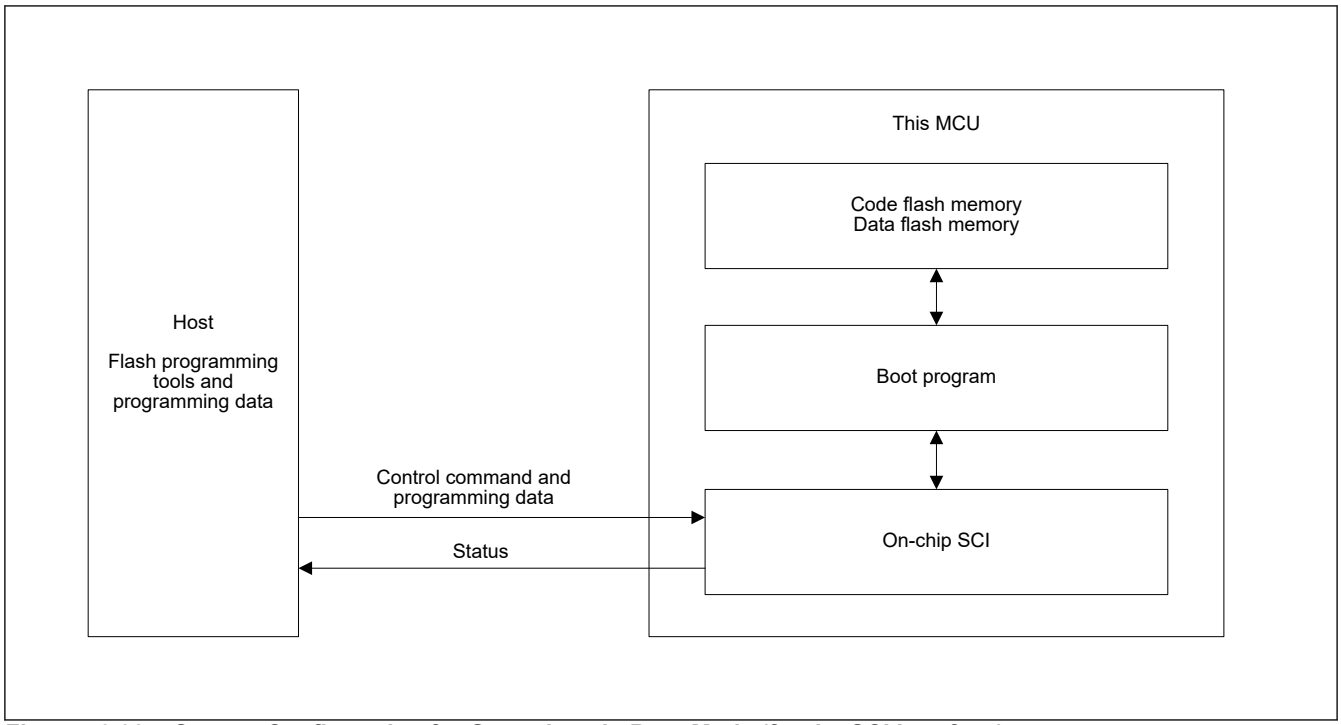


Figure 43.38 System Configuration for Operations in Boot Mode (for the SCI Interface)

43.14 Using the Serial Programmer for Rewriting

A serial programmer can be used to rewrite flash memory in boot mode.

(1) Serial Programming

This MCU is mounted on the system board at the time of serial programming. Providing a connector to the board enables rewriting of this MCU by the serial programmer to proceed.

43.14.1 Environments for Serial Programming

The recommended environments for rewriting the flash memory of the MCU with data are described below.

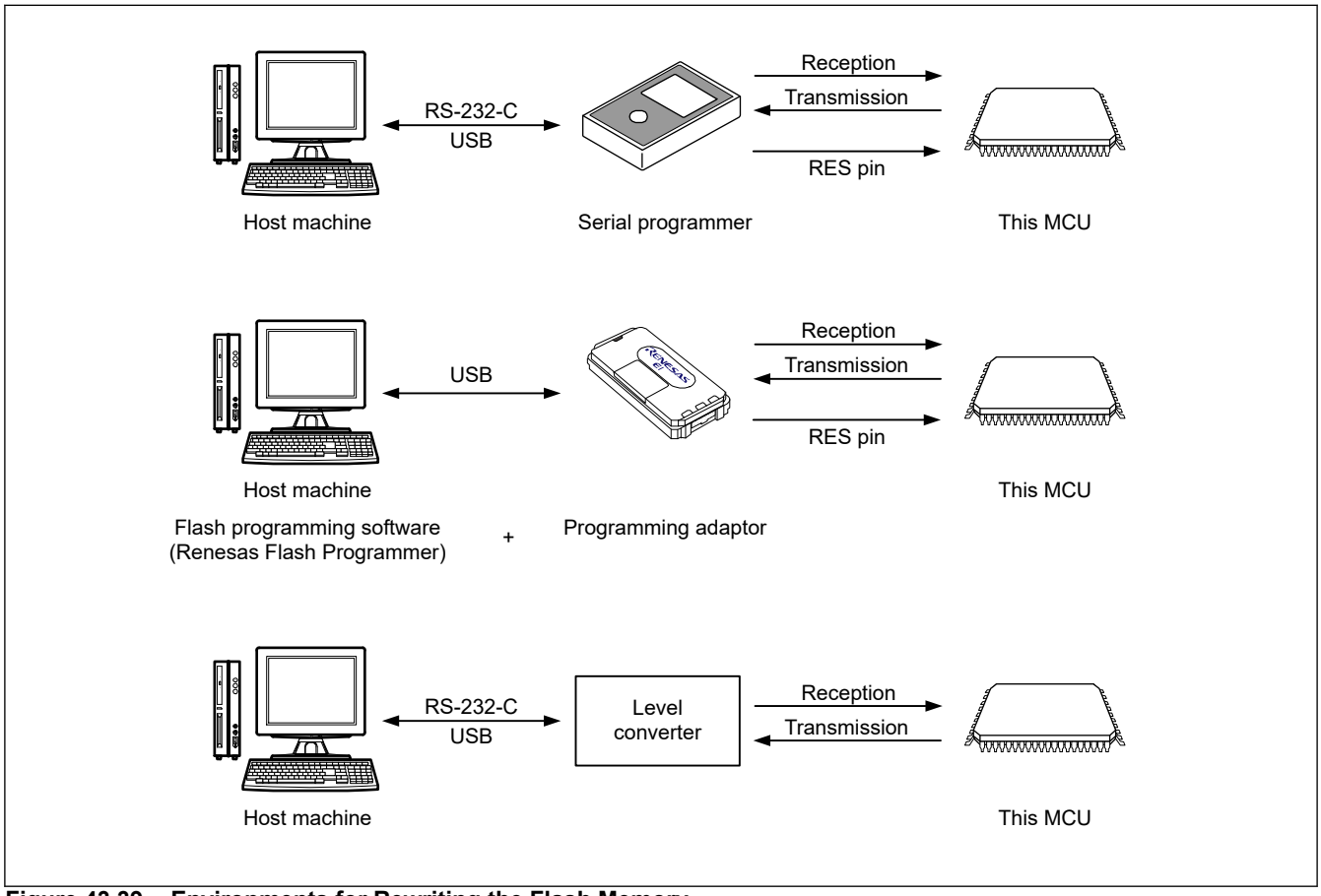


Figure 43.39 Environments for Rewriting the Flash Memory

43.15 Programming through Self-Programming

43.15.1 Overview

This MCU supports programming of the flash memory by the user program itself. The FACI commands can be used with user programs for writing to the flash memory. This allows upgrading of user programs and rewriting of constant data fields.

The program for rewriting must be transferred to the internal RAM in advance when the BGO is not available or when rewriting the option-setting memory.

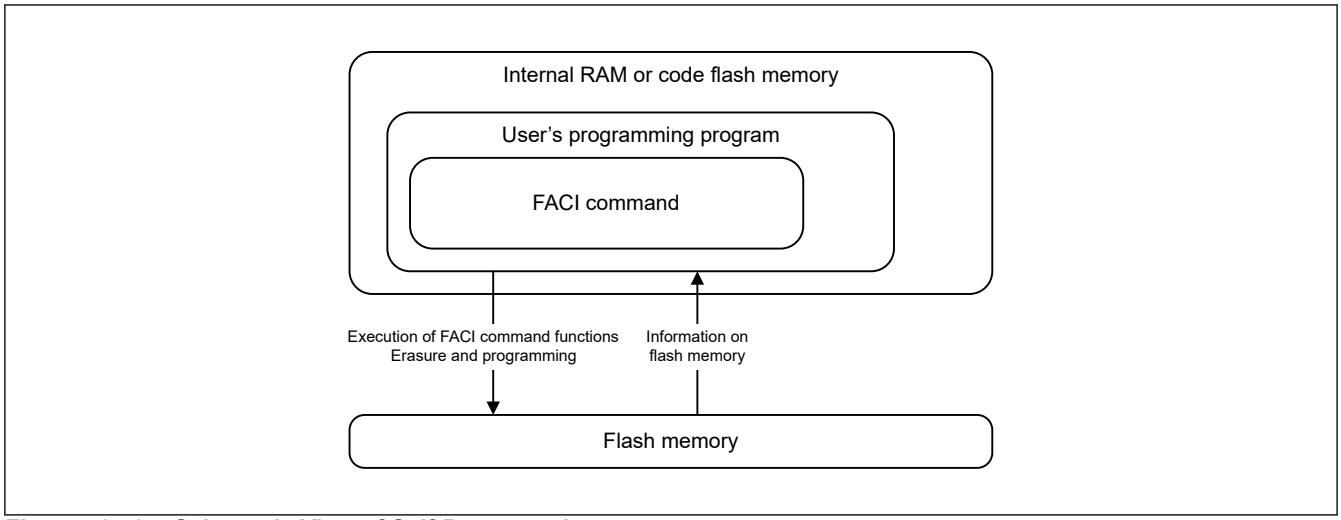


Figure 43.40 Schematic View of Self-Programming

For comprehensive information on the self-programming, see section 43.9. FACI Commands.

43.15.2 Background Operation

The background operation (BGO) can be used to execute the flash rewrite routine on the code flash memory when the data flash memory is rewritten.

Background operations can be used when the combination of the flash memory for rewriting and the flash memory for reading is any of those listed below.

Table 43.29 Conditions under which Background Operation is Usable

	Range for rewriting	Range for reading
Common	Code flash memory	Data flash memory
	Data flash memory	Code flash memory

43.16 Reading Flash Memory

43.16.1 Reading Code Flash Memory

Special settings are not required to read code flash memory after release from the reset state. Data can simply be read out through access to addresses in the code flash memory.

When reading code flash memory that has been erased but not yet been programming again (i.e. that is in the non-programmed state), all bits are read as 1.

43.16.2 Reading Data Flash Memory

Special settings are not required to read data flash memory after release from the reset state. Data can simply be read out through access to addresses in the data flash memory.

Values read from data flash memory that has been erased but not yet been programming again (i.e. that is in the non-programmed state) are undefined. Use blank checking when you need to confirm that an area is in the non-programmed state.

43.16.3 Access Cycle

When the CPU cache is hit, access is one cycle.

When the CPU cache is missed while CPU cache operation is enabled or when CPU cache is disabled, the access cycle is the following.

Table 43.30 Code Flash Memory

Flash Cache Operation	FLWT Register Setting	Read cycle (ICLK)
enable and hit	—	3
disable or miss	0x00	3
	0x01	4

Table 43.31 Data Flash Memory

FCKMHZ Register Setting	Read (cycle)
0x00 to 0x09	Min: 2 ICLK + 3 FCLK Max: (n + 1) ICLK + 3 FCLK
0x0A to 0x13	Min: 2 ICLK + 4 FCLK Max: (n + 1) ICLK + 4 FCLK
0x14 to 0x1D	Min: 2 ICLK + 5 FCLK Max: (n + 1) ICLK + 5 FCLK
0x1E to 0x27	Min: 2 ICLK + 6 FCLK Max: (n + 1) ICLK + 6 FCLK
0x28 to 0x31	Min: 2 ICLK + 7 FCLK Max: (n + 1) ICLK + 7 FCLK
0x32 to 0x3B	Min: 2 ICLK + 8 FCLK Max: (n + 1) ICLK + 8 FCLK
0x3C	Min: 2 ICLK + 9 FCLK Max: (n + 1) ICLK + 9 FCLK

Note: When the frequency ratio of ICLK : FCLK is n : 1

43.17 Usage Notes

(1) Reading Area Where Programming/Erasure was Interrupted and Area Targeted for Suspension

The data stored in the area where programming or erasure has been suspended or the area where programming or erasure has been suspended by using the suspend command are undefined. To avoid faulty operation caused by reading undefined data, take care not to fetch instructions or read data from areas where programming or erasure was suspended and where programming or erasure was suspended by using the suspend command.

(2) Suspension During Programming/Erasure

When processing of programming/erasure is stopped by issuing the P/E suspend command, the programming/erasure processing can be resumed by issuing the P/E resume command. If the flash sequencer enters the command-locked state for any reason and issues the forced stop command after the suspended processing is normally completed and the ERSSPD flag or PRGSPD flag is set to 1, the suspended processing cannot be resumed. In addition, the values in the area where the processing was suspended are not guaranteed. Erase that area

(3) Prohibition of Additional Programming

Programming a given area of the code flash memory or data flash memory twice is not possible. To program the code flash memory or data flash memory where has been programed, erase the target area. Programming can be added to the option-setting memory.

(4) Resets During Programming/Erasure, or Blank Checking

In the case of a reset due to the signal on the RES pin during programming/erasure, or blank checking of the flash memory, wait for at least t_{RESW} (see section 46, Electrical Characteristics) of the reset input period once the operating voltage is within the range stipulated in the electrical characteristics, then release the device from the reset state.

(5) Allocation of Vectors for Interrupts and Other Exceptions During Programming/Erasure

Generation of an interrupt or other exception during programming/erasure may lead to fetching of the vector from the code flash memory. Under conditions where BGO cannot be used, set the address of the vector to an address that is not in the

code flash memory. Alternatively, make sure that no handling of interrupts or exceptions proceeds during programming/erasure.

(6) Items Prohibited During Programming/Erasure, or Blank Checking

High voltage is applied to the flash memory during programming/erasure, or blank checking. To prevent damage to the flash memory, do not perform the following operations.

- Have the operating voltage from the power supply go beyond the permitted range.
- Change the FWEPROR.FLWE[1:0]bits.
- Change the OPCCR.OPCM[2:0] bits.
- Change the SCKDIVCR.FCK[2:0]bits.
- Change the SCKSCR.CKSEL[2:0]bits.
- Transition to the software standby mode, or Deep Software Standby mode.

(7) Programming/Erasure in Low-Speed Modes

Do not programming/erasure the flash memory when low-speed mode is selected with the operating power control register (OPCCR).

(8) Emulator Connection

Renesas provides the emulator which supports both debugging using SWD or JTAG communication and serial programming using SCI communication. This emulator makes it easy to switch between debugging and serial programming.

Table 43.32 shows the pinout of 10 pin or 20 pin socket pinouts when using this emulator. The pinout of SWD and JTAG is ARM standard, and MD, TXD, RXD pins are added for the serial programming using SCI communication.

The serial programming interface must be used to program the TrustZone IDAU boundary register settings.

It is recommended to connect PA14/SWCLK/TCK and P201/MD pins using wired OR circuit on the board to use both debugging and serial programming

Table 43.32 Pin assign for emulator

Pin No.	SWD	JTAG	Serial Programming using SCI
1	VCC	VCC	VCC
2	PA13/SWDIO	PA13/TMS	NC
4	PA14/SWCLK	PA14/TCK	P201/MD
	Wired OR with P201/MD	Wired OR with P201/MD	
6	PB03/SWO/TXD9	PB03/TDO/TXD9	PB03/TXD9
8	PA15/RXD9	PA15/TDI/RXD9	PA15/RXD9
9	GNDdetect	GNDdetect	GNDdetect
10	nRESET	nRESET	nRESET
12	PE02/TCLK	PE02/TCLK	NC
14	PE03/TDATA[0]	PE03/TDATA[0]	NC
16	PE04/TDATA[1]	PE04/TDATA[1]	NC
18	PE05/TDATA[2]	PE05/TDATA[2]	NC
20	PE06/TDATA[3]	PE06/TDATA[3]	NC
3, 5, 15, 17, 19	GND	GND	GND
7	NC	NC	NC
11, 13	NC	NC	NC

44. Internal Voltage Regulator

44.1 Overview

The MCU includes one internal voltage regulator:

- Linear regulator (LDO)

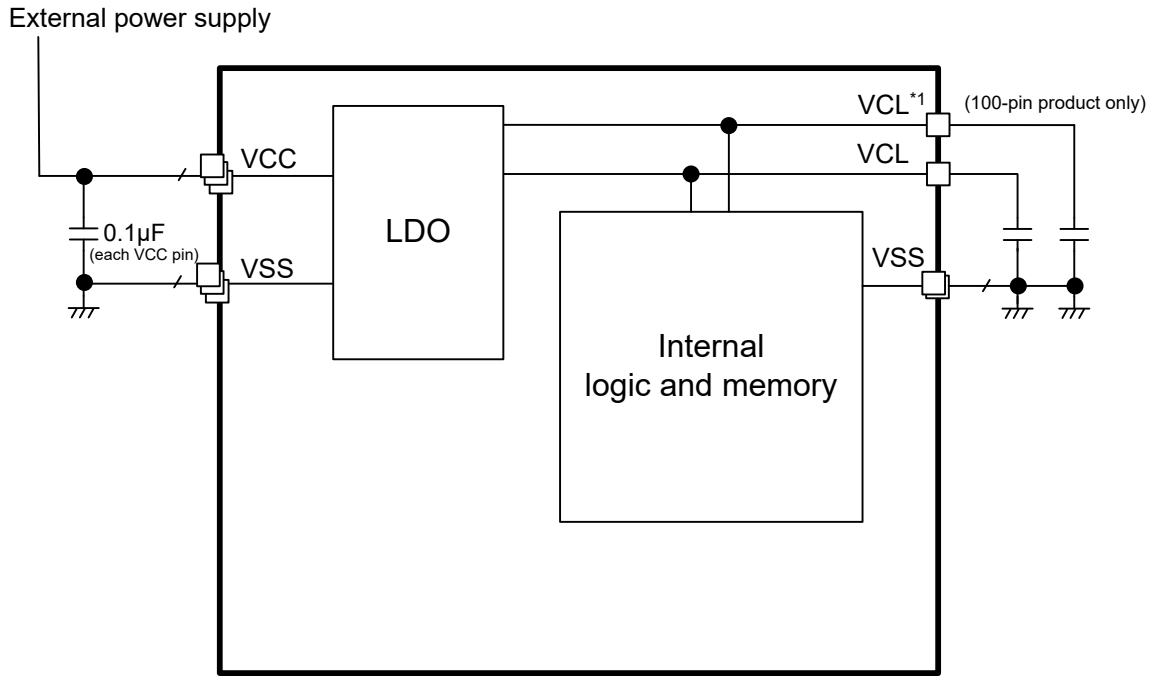
This regulator supplies voltage to all internal circuits and memory except for I/O and analog power domains.

44.2 Operation

Table 44.1 lists the LDO mode pin settings, and Figure 44.1 shows the LDO mode settings. In LDO mode, the internal voltage is generated from VCC.

Table 44.1 LDO mode pin

Pins	Setting descriptions
All VCC	• Connect each pin to the system power supply. • Connect each pin to VSS through a 0.1-µF multilayer ceramic capacitor. Place the capacitor close to the pin.
All VCL (100-pin product)	Connect the each pin to VSS through a 0.1-µF multilayer ceramic capacitor. Place the capacitor close to the pin.
VCL (64, 48-pin products)	Connect the pin to VSS through a 0.22-µF multilayer ceramic capacitor. Place the capacitor close to the pin.



Note 1. Only 100-pin product has two VCL pins.

Figure 44.1 LDO mode settings

45. Security Features

45.1 Features

- ARMv8-M TrustZone security
  - Eight regions IDAU for memory space
    - Up to three regions for the code flash
    - Up to two regions for the data flash
    - Up to three regions for the SRAM
    - IDAU setting is common for the CPU, DMAC, and DTC
  - SAU is not implemented
  - Secure or Non-secure region for the Standby SRAM
  - Individual Secure or Non-secure security attribution for each peripheral
  - Some peripherals support both Secure and Non-secure security attributions
- Device lifecycle management
- Three debug access levels
  - DBG2: The debugger connection is allowed, and no restriction to access memories and peripherals
  - DBG1: The debugger connection is allowed, and restricted to access only non-secure memory regions and peripherals
  - DBG0: The debugger connection is not allowed
- Key injection
- Cryptographic accelerator
  - See section 35, Secure Cryptographic Engine (SCE5)

45.2 Arm TrustZone Security

45.2.1 Arm TrustZone Technology

Arm TrustZone technology divides the system and the application into Secure and Non-secure domains. Secure application can access both Secure and Non-secure memory and resources. Non-secure application can only access Non-secure memory and resources.

The system starts up in Secure state by default. The security state of CPU can be either Secure or Non-secure.

45.2.2 Memory Security Attribution

The code flash, the data flash, and the SRAM are divided into Secure (S), Non-secure (NS) and Non-secure callable (NSC) regions. These memory security attributions are set into the nonvolatile memory by the serial programming command when the device lifecycle is SSD state. These memory security attributions are loaded into the IDAU and the memory controller before application execution. These memory security attributions cannot be updated by application but can be through the dedicated registers.

The code flash can be divided in up to three regions. The data flash can be divided in up to two regions. The SRAM can be divided in up to three regions. Figure 45.1 shows the memory mapping. Table 45.1 shows the size of memory region.

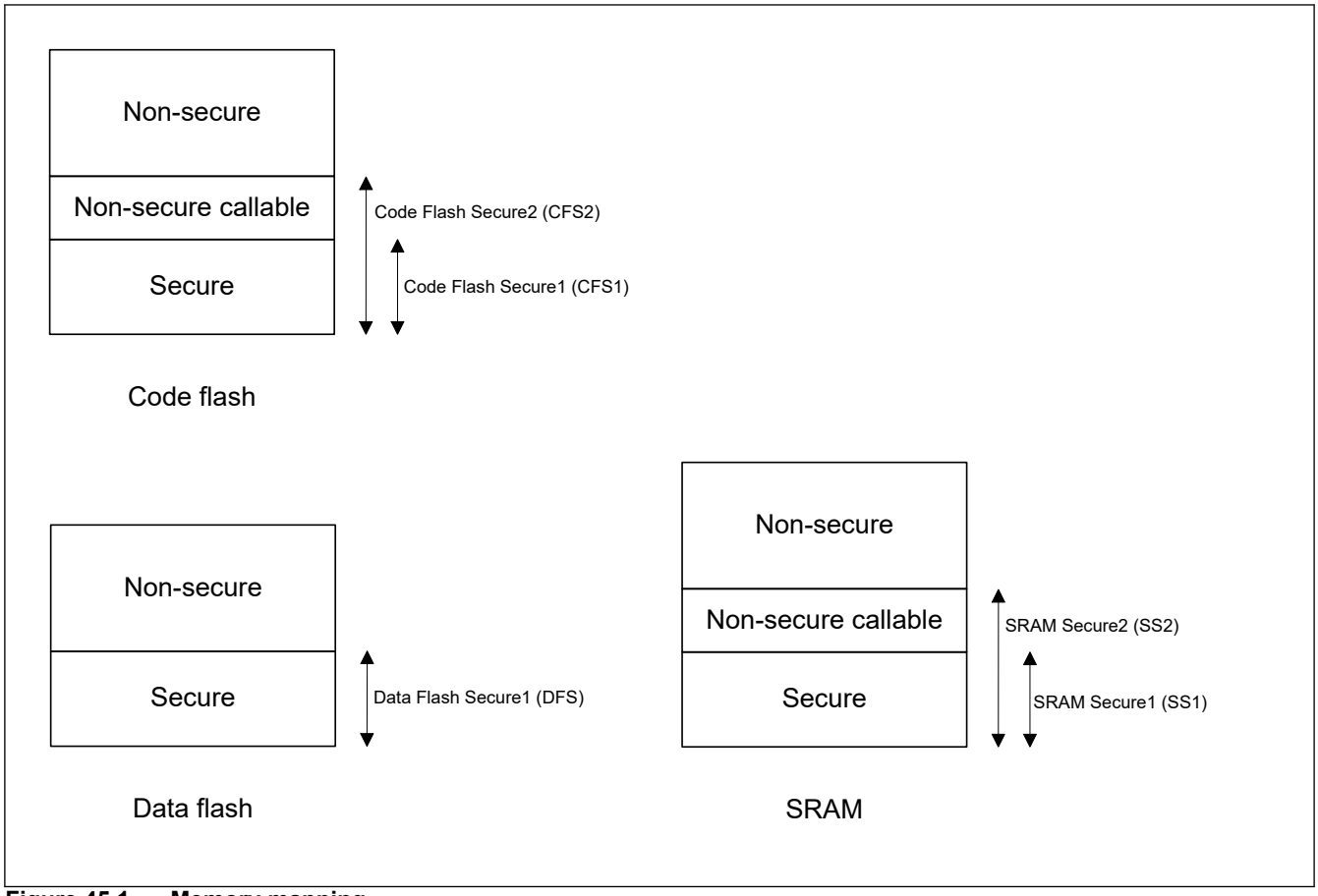


Figure 45.1 Memory mapping

Table 45.1 Memory Region Size

Memory Region	Start Address	Size
Code flash secure	0x0000_0000	CFS1 × 1 KB
Code flash non-secure callable	CFS1 × 1 KB	CFS2 × 32 KB - CFS1 × 1 KB
Code flash non-secure	CFS2 × 32 KB	Code flash size - CFS2 × 32 KB
Data flash secure	0x0800_0000	DFS × 1 KB
Data flash non-secure	0x0800_0000 + DFS × 1 KB	Data flash size - DFS × 1 KB
SRAM secure	0x2000_0000	SS1 × 1 KB
SRAM non-secure callable	0x2000_0000 + SS1 × 1 KB	SS2 × 8 KB - SS1 × 1 KB
SRAM non-secure	0x2000_0000 + SS2 × 8 KB	SRAM size - SS2 × 8 KB

The Standby SRAM is divided 8 regions. Security attribution can be set for each region, but both secure region and non-secure region must be contiguous. In other words, the Standby SRAM can have one contiguous secure region and one contiguous non-secure region. The Standby SRAM security attribution is set to the dedicated register by the secure application. See section 42, Standby SRAM for the details.

Table 45.2 shows the access permission of the memory.

Table 45.2 Access Permission of Memory (1 of 2)

Memory	Secure access	Non-secure access
Code flash, Data flash, SRAM configured as Secure or Non-secure callable	allowed	Write ignored / Read ignored TrustZone Access error is generated
Code flash, Data flash, SRAM configured as non-secure	allowed	allowed

Table 45.2 Access Permission of Memory (2 of 2)

Memory	Secure access	Non-secure access
Standby SRAM configured as Secure	allowed	Write ignored / Read 0x00 TrustZone Access error is not generated
Standby SRAM configured as Non-secure	allowed	allowed

45.2.3 Peripheral Security Attribution

Each peripheral can be configured to be Secure or Non-secure.

Peripherals are divided into two types.

Type-1 peripherals has the one security attribution. Access to all registers is controlled by one security attribution. Type-1 peripheral security attribution is set to the PSARx (x = B to E) register by the secure application.

Type-2 peripherals has the security attribution for each register or for each bit. Access to each register or bit field is controlled according to these security attributions. Type-2 peripheral security attribution is set to the Security Attribution register in each module by the secure application. For the Security Attribution register, see sections in the user manual for each peripheral.

Table 45.3 shows the classification of peripheral type.

Table 45.3 Peripheral Type Classification

Type	Peripheral
Type-1	SCI, SPI, CANFD, IIC, SCE5, DOC, CRC, CAC, TSN, ADC, DAC12, POEG, AGT, IWDT, WDT, IIRFA, TFU, ACMPHS, KINT
Type-2	System control (Resets, LVD, Clock Generation Circuit, Low Power Modes), FLASH CACHE, SRAM controller, CPU CACHE, DMAC, DTC, ICU, MPU, BUS, Security setting, ELC, I/O ports
Always Non-secure	GPT, PDG

Table 45.4 shows the access permission of type-1 peripherals. The access permission of type-2 peripherals is different by peripherals. See section Register Description of each peripherals.

Table 45.4 The access permission of type-1 peripherals

Permission	Secure access	Non-secure access
Peripheral configured as secure	allowed	Write ignored / Read ignored TrustZone Access error is generated
Peripheral configured as non-secure	allowed	allowed

45.2.4 Flash Sequencer Security Attribution

The flash sequencer is used to program or erase the flash.

The flash sequencer has the special security attribution. Table 45.5 shows the access permission of flash sequencer.

Table 45.5 Access Permissions of Flash Sequencer (1 of 2)

	Secure access	Non-secure access
FACI command issuing area	allowed	When the FACI command is issued to the secure region of code flash, data flash and option-setting memory ● Issued FACI command is invalid ● Flash sequencer error is generated When the FACI command is issued to the non-secure region of code flash, data flash and option-setting memory ● Issued FACI command is valid
FBPROT1, FSUACR, FMEPROT registers	allowed	Write ignored / Readable TrustZone Access error is not generated

Table 45.5 Access Permissions of Flash Sequencer (2 of 2)

	Secure access	Non-secure access
FCKMHZ register	allowed	Configured by Flash Security Attribution register When configured as Secure, ● Write ignored / Readable ● TrustZone Access error is not generated. When configured as Non-secure ● allowed
Other registers	allowed	During programming/erasure or during suspend programming/erasure by secure application ● Write ignored / Read 0x00 ● TrustZone Access error is not generated In other state ● allowed

45.2.5 Address Space Security Attribution

Table 45.6 shows the security attribution of the address space.

Table 45.6 Address Space Security Attribution

Region	Attribution
Code flash secure	Secure
Code flash non-secure callable	Non-secure callable
Code flash non-secure	Non-secure
Data flash secure	Secure
Data flash non-secure	Non-Secure
SRAM secure	Secure
SRAM non-secure callable	Non-secure callable
SRAM non-secure	Non-secure
Peripherals	Exempt
Other area	Exempt

Note: Exempt: No check will be done. All bus transactions are propagated.

45.2.6 TrustZone Access Error

Table 45.7 shows the behavior when TrustZone access error. The behavior varies depending on the master or slave area to be accessed.

Table 45.7 The Behavior When TrustZone Access Error

Area	CPU	DMAC/DTC
Code flash, Data flash, SRAM	Detect SecureFault exception*2	● Transfer does not start ● Occur NMI or reset*1 ● Occur interrupt (DMA_TRANSERR)
Other area	● Detect BusFault exception*2 *3 ● Occur NMI or reset*1*2 *3	● Stop transfer ● Occur NMI or reset *1 ● Occur interrupt (DMA_TRANSERR)

Note 1. NMI or reset is selected with TZFOAD.OAD bit.
Note 2. When TrustZone access error occurs by the debugger access, exception, NMI, or reset does not occurs.Only the error response is returned.
Note 3. These error behaviors does not occur for write access to the PHBIU or PLBIU address space which memory attribute is set to "Early Write Acknowledgment" by the ARM MPU.

45.3 Device Lifecycle Management

Device lifecycle identifies the current phase of the device and controls the capabilities of the debug interface, the serial programming interface and Renesas test mode. Figure 45.2 is the illustration of the device lifecycle. Table 45.8 shows the lifecycle definition and capability in each lifecycle.

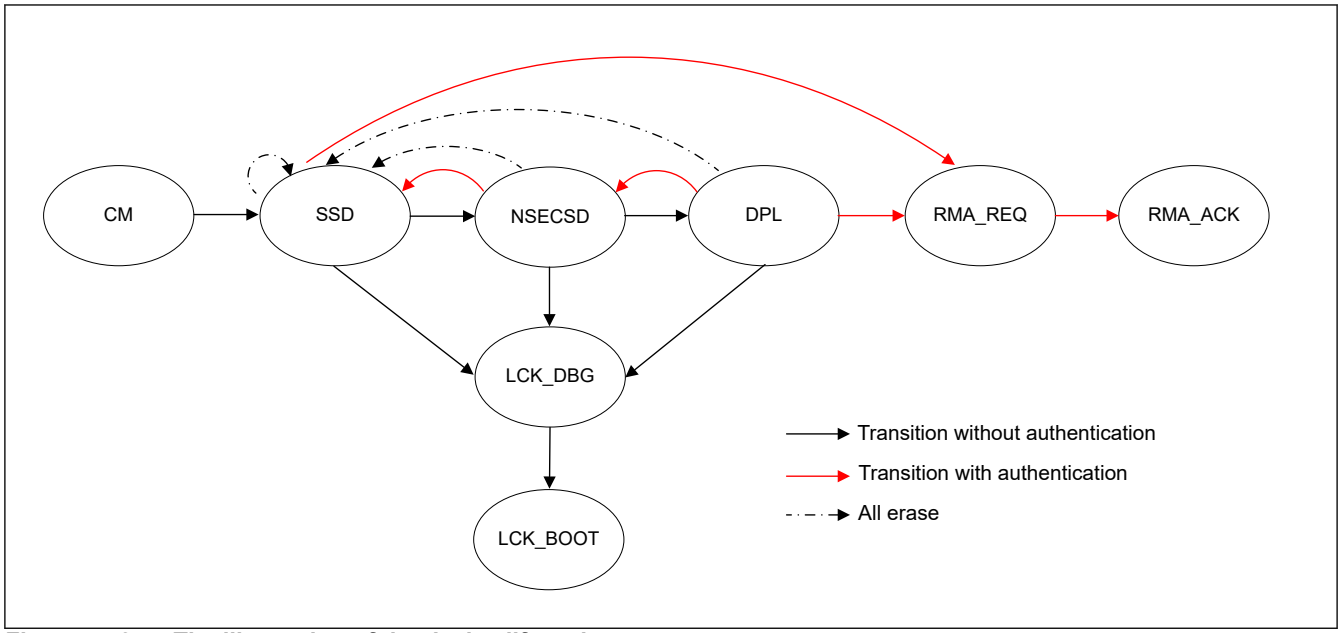


Figure 45.2 The illustration of the device lifecycle

Table 45.8 The lifecycle definition and the capability can be used in each lifecycle

Lifecycle	Definition	Debug level	Serial programming	Renesas test mode
CM	“Chip Manufacturing” The device is in Renesas factory. The state when the customer received the device.	DBG2	Available cannot access code/data flash area	Not available
SSD	“Secure Software Development” The secure part of application is being developed.	DBG2	Available can program/erase/read all code/data flash area	Not available
NSECSD	“Non-SECure Software Development” The non-secure part of application is being developed.	DBG1	Available can program/erase/read only non-secure code/data flash area	Not available
DPL	“DePLoyed” The device is in-field.	DBG0	Available cannot access code/data flash area	Not available
LCK_DBG	“LoCKed DeBuG” The debug interface is permanently disabled.	DBG0	Available cannot access code/data flash area	Not available
LCK_BOOT	“LoCKed BOOT interface” The debug interface and the serial programming interface are permanently disabled.	DBG0	Not available	Not available
RMA_REQ	“Return Material Authorization REQuest” Request for RMA. The customer must send the device to Renesas in this state.	DBG0	Available cannot access code/data flash area	Not available
RMA_ACK	“Return Material Authorization ACKnowledged” Failure analysis in Renesas	DBG2	Available cannot access code/data flash area	Available

45.3.1 Changing the Lifecycle State

Use the serial programming commands to change the device lifecycle state. See the boot firmware application note for the detail of command. The lifecycle cannot be updated by application but can read through the dedicated registers.

As shown in Figure 45.2, there are three types lifecycle transition.

The first one is to change to lower debug access level or restrict the serial programming mode. This change can be done with no restriction.

Note: The debug interface is permanently disabled in LCK_DBG. After changed to LCK_DBG, the debug interface cannot be used forever.

Note: The debug interface and serial programming interface are permanently disabled in LCK_BOOT. After changed to LCK_BOOT, the debug interface and the serial programming interface cannot be used forever.

The second one is to change to higher debug access level or request for RMA. This change needs key authentication. The key length is 128 bits. The secure developer needs to inject two keys when the lifecycle is SSD state. One is "SECDBG_KEY" which is used to authentication to change the lifecycle from NSECSD to SSD. Other one is "RMA_KEY" which is used to authentication to change the lifecycle from SSD or DPL to RMA_REQ. The non-secure developer needs to inject one key when the lifecycle is NSECSD state. This is "NONSECDBG_KEY" which is used to authentication to change the lifecycle from DPL to NSECSD. See section 45.4. Key Injection for the detail of how to inject the key. The key authentication uses a challenge and response authentication or authentication using the unique ID. The authentication using the unique ID is available only transition to RMA_REQ. The following is the process of how to calculate the response, challenge and response authentication, or the authentication code using unique ID.

Response = AES128-CMAC (KEY, 128bits challenge)

Authentication code = AES128-CMAC (KEY, 128bits unique ID)

Note: In case the key is not injected, these lifecycle changes cannot be done.

Note: In the lifecycle transition from NSECSD to SSD or form DPL to NSECSD, the contents on the flash memory are not erased.

Note: MCU does not respond after changing to higher debug access level or RMA_REQ. If you continue to use the serial programming commands, need to re-enter the boot mode after a reset. See the boot firmware application note for the detail.

Note: In the lifecycle transition to RMA_REQ, the contents on the flash memory except permanently locked block or setting or BPS_SEL register are erased. The contents in the permanently locked block or register can be read by Renesas at failure analysis. Permanently locked block means the block which programming and erasure is disabled permanently by PBPS, PBPS_SEC and BPS_SEL register. Permanently locked register means SAS register which programming and erasure is disabled permanently by FSPR bit.

The third one is all erase. This is done by an initialize command unless an initialize command itself is disabled. The lifecycle is back to SSD and the contents on the flash memory is erased. If there is permanently locked block or register, an initialize command does not execute. In case of the all bits of PBPS and PBPS_SEC register are 1 and FSPR bit is1, an initialize command is executable.

Note: The initialize command can be issued by everyone, so contents on the flash memory are easily erased. Developers who do not want this can invalidate the initialize command permanently by parameter setting command.

Note: MCU does not respond after executing the initialize command. If you continue to use the serial programming commands, need to re-enter the boot mode after a reset. See the boot firmware application note for the detail.

45.3.2 Debug access level

There are three debug access levels, and the debug access level changes according to the lifecycle state.

- DBG2: The debugger connection is allowed, and no restriction to access memories and peripherals
- DBG1: The debugger connection is allowed, and restricted to access only non-secure memory regions and peripherals
- DBG0: The debugger connection is not allowed

45.3.3 Serial Programming

Whether a serial programmer can be connected and the range of flash memory that can be accessed depends on the lifecycle state as shown in Table 45.8. And the accepted serial programming command differs depending on the lifecycle state. See the boot firmware application note for the detail of command.

45.3.4 Lifecycle changing example

The following is a typical lifecycle changing example.

Secure developer

- Change the lifecycle from CM to SSD by using the serial programming command.
- Set the memory security attribution of the code flash, data flash and SRAM by using the serial programming command.
- Program the secure application by using the serial programming interface and debug the secure application. Debug is possible if the lifecycle is CM, but it is impossible to set the memory security attribution in CM state. If the memory security attribution is not set, all area of the code flash, data flash and SRAM is Secure.

Note: Need to configure the registers listed in Table 45.10 as Non-secure only in NSECSD state. See section 45.6.1. Restrictions on setting the security attribution for details.

- Inject SECDBG_KEY and RMA_KEY by using the serial programming command (if need).
- Disable the all erase by using the serial programming command (if need).
- Change the lifecycle from SSD to NSECSD by using the serial programming command.

Non-secure developer

- Program the Non-secure application by using the serial programming interface and debug the Non-secure application.
- Inject NONSECDBG_KEY by using the serial programming command (if need).
- Disable the all erase by using the serial programming command (if need).
- Change the lifecycle to DPL by using the serial programming command.

45.3.5 Failure analysis

If the customer requests the failure analysis to Renesas, it is necessary to send the device after changing the lifecycle to RMA_REQ. If the lifecycle is not RMA_REQ, Renesas cannot do the failure analysis. Because RMA_REQ is permanent state, it cannot back to the another state after changing to RMA_REQ. It is assumed to change to SSD or NSECSD and analyze before changing to RMA_REQ.

Devices sent to Renesas will not be returned to customers. The device will be discarded.

Note: As described in the section 45.3.1. Changing the Lifecycle State, RMA_KEY is needed to change the lifecycle to RMA_REQ. If the customer forgets the RMA_KEY, Renesas cannot do the failure analysis.

45.4 Key Injection

There are three steps required to inject a user key into the MCU.

First, the customer needs to create the 256 bits installation key. This key is called User Factory Programming Key (UFPK) and used to encrypt a user key. The customer gets the key of the wrapped version (W-UFPK) through the Renesas Key Wrapping Service.

Second the customer encrypts the user key using UFPK as the AES key.

Last the customer sends W-UFPK and the encrypted user key to the MCU by using serial programming interface. The sent user key is decrypted, wrapped with the hardware unique key, and then stored in the nonvolatile memory.

Figure 45.3 is the illustration of key injection. Table 45.9 shows the keys that can be injected by serial programming interface.

User Key is used for authentication during the life cycle transition.

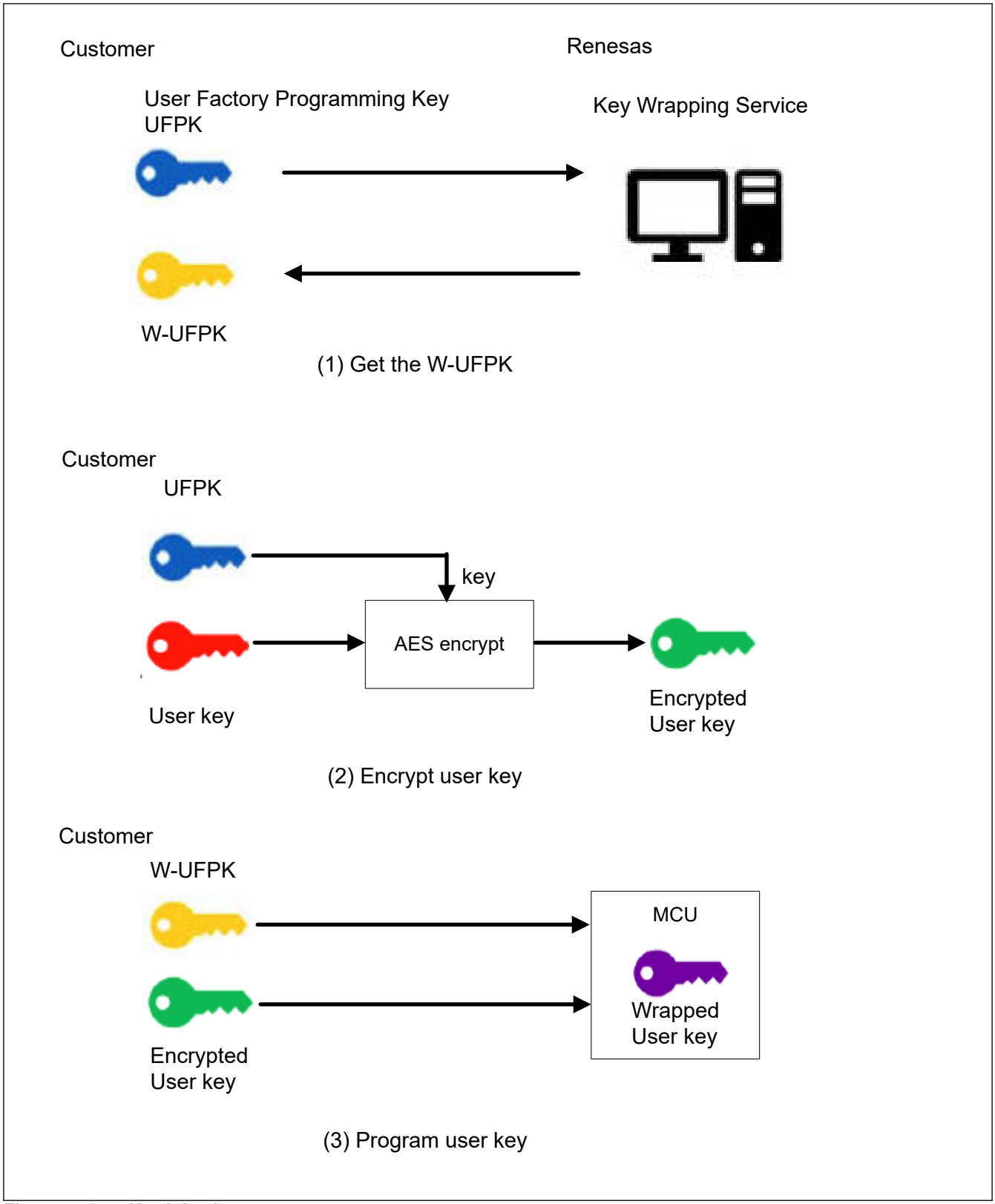


Figure 45.3 Key Injection

Table 45.9 The keys that can be injected by serial programming

Lifecycle transition	SECDBG_KEY, NONSECDBG_KEY, RMA_KEY

45.5 Register Description

45.5.1 PSARB : Peripheral Security Attribution Register B

Base address: PSCU = 0x400E_0000

Offset address: 0x04

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	PSARB31	PSARB30	PSARB29	PSARB28	PSARB27	—	—	—	—	PSARB22	—	—	PSARB19	PSARB18	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	PSARB9	PSARB8	—	—	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	—	This bit is read as 1. The write value should be 1.	R/W
1	—	This bit is read as 1. The write value should be 1.	R/W
2	—	This bit is read as 1. The write value should be 1.	R/W
3	—	This bit is read as 1. The write value should be 1.	R/W
5:4	—	These bits are read as 1. The write value should be 1.	R/W
6	—	This bit is read as 1. The write value should be 1.	R/W
7	—	This bit is read as 1. The write value should be 1.	R/W
8	PSARB8	IIC1 and the MSTPCRB.MSTPB8 bit security attribution 0: Secure 1: Non-secure	R/W
9	PSARB9	IIC0 and the MSTPCRB.MSTPB9 bit security attribution 0: Secure 1: Non-secure	R/W
10	—	This bit is read as 1. The write value should be 1.	R/W
11	—	This bit is read as 1. The write value should be 1.	R/W
12	—	This bit is read as 1. The write value should be 1.	R/W
14:13	—	These bits are read as 1. The write value should be 1.	R/W
15	—	This bit is read as 1. The write value should be 1.	R/W
16	—	This bit is read as 1. The write value should be 1.	R/W
17	—	This bit is read as 1. The write value should be 1.	R/W
18	PSARB18	SPI1 and the MSTPCRB.MSTPB18 bit security attribution 0: Secure 1: Non-secure	R/W
19	PSARB19	SPI0 and the MSTPCRB.MSTPB19 bit security attribution 0: Secure 1: Non-secure	R/W
21:20	—	These bits are read as 1. The write value should be 1.	R/W
22	PSARB22	SCI9 and the MSTPCRB.MSTPB22 bit security attribution 0: Secure 1: Non-secure	R/W
23	—	This bit is read as 1. The write value should be 1.	R/W
24	—	This bit is read as 1. The write value should be 1.	R/W

Bit	Symbol	Function	R/W
25	—	This bit is read as 1. The write value should be 1.	R/W
26	—	This bit is read as 1. The write value should be 1.	R/W
27	PSARB27	SCI4 and the MSTPCRB.MSTPB27 bit security attribution 0: Secure 1: Non-secure	R/W
28	PSARB28	SCI3 and the MSTPCRB.MSTPB28 bit security attribution 0: Secure 1: Non-secure	R/W
29	PSARB29	SCI2 and the MSTPCRB.MSTPB29 bit security attribution 0: Secure 1: Non-secure	R/W
30	PSARB30	SCI1 and the MSTPCRB.MSTPB30 bit security attribution 0: Secure 1: Non-secure	R/W
31	PSARB31	SCI0 and the MSTPCRB.MSTPB31 bit security attribution 0: Secure 1: Non-secure	R/W

Note: A bit undefined in this table is reserved bit. The reserved bit should be kept the initial value.
Note: Only secure access can write to this register. Both secure access and Non-secure read access are allowed but Non-secure write access is not allowed and TrustZone access error is not generated.
Note: This register is write-protected by PRCR register.

The PSARB specifies the security attribution for each module and the corresponding bit in Module Stop Control Register.

45.5.2 PSARC : Peripheral Security Attribution Register C

Base address: PSCU = 0x400E_0000

Offset address: 0x08

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	PSARC31	—	—	—	PSARC27	—	—	—	—	—	PSARC21	PSARC20	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	PSARC13	—	—	—	—	—	—	—	—	—	—	—	PSARC1	PSARC0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	PSARC0	CAC and the MSTPCRC.MSTPC0 bit security attribution 0: Secure 1: Non-secure	R/W
1	PSARC1	CRC and the MSTPCRC.MSTPC1 bit security attribution 0: Secure 1: Non-secure	R/W
2	—	This bit is read as 1. The write value should be 1.	R/W
3	—	This bit is read as 1. The write value should be 1.	R/W
7:4	—	These bits are read as 1. The write value should be 1.	R/W
8	—	This bit is read as 1. The write value should be 1.	R/W
11:9	—	These bits are read as 1. The write value should be 1.	R/W
12	—	This bit is read as 1. The write value should be 1.	R/W

Bit	Symbol	Function	R/W
13	PSARC13	DOC and the MSTPCRC.MSTPC13 bit security attribution 0: Secure 1: Non-secure	R/W
19:14	—	These bits are read as 1. The write value should be 1.	R/W
20	PSARC20	TFU and the MSTPCRC.MSTPC20 bit security attribution 0: Secure 1: Non-secure	R/W
21	PSARC21	IIRFA and the MSTPCRC.MSTPC21 bit security attribution 0: Secure 1: Non-secure	R/W
26:22	—	These bits are read as 1. The write value should be 1.	R/W
27	PSARC27	CANFD and the MSTPCRC.MSTPC27 bit security attribution 0: Secure 1: Non-secure	R/W
30:28	—	These bits are read as 1. The write value should be 1.	R/W
31	PSARC31	SCE5 and the MSTPCRC.MSTPC31 bit security attribution 0: Secure 1: Non-secure	R/W

Note: Only secure access can write to this register. Both secure access and Non-secure read access are allowed but Non-secure write access is not allowed and TrustZone access error is not generated.
Note: This register is write-protected by PRCR register.

The PSARC specifies the security attribution for each module and the corresponding bit in Module Stop Control Register.

45.5.3 PSARD : Peripheral Security Attribution Register D

Base address: PSCU = 0x400E_0000

Offset address: 0x0C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	PSARD28	PSARD27	PSARD26	PSARD25	—	—	PSARD22	—	PSARD20	PSARD19	—	—	PSARD16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	PSARD14	PSARD13	PSARD12	PSARD11	—	—	—	—	—	—	—	PSARD3	PSARD2	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
1:0	—	These bits are read as 1. The write value should be 1.	R/W
2	PSARD2	AGT1 and the MSTPCRD.MSTPD2 bit security attribution 0: Secure 1: Non-secure	R/W
3	PSARD3	AGT0 and the MSTPCRD.MSTPD3 bit security attribution 0: Secure 1: Non-secure	R/W
10:4	—	These bits are read as 1. The write value should be 1.	R/W
11	PSARD11	POEG Group D and the MSTPCRD.MSTPD11 bit security attribution 0: Secure 1: Non-secure	R/W
12	PSARD12	POEG Group C and the MSTPCRD.MSTPD12 bit security attribution 0: Secure 1: Non-secure	R/W

Bit	Symbol	Function	R/W
13	PSARD13	POEG Group B and the MSTPCRD.MSTPD13 bit security attribution 0: Secure 1: Non-secure	R/W
14	PSARD14	POEG Group A and the MSTPCRD.MSTPD14 bit security attribution 0: Secure 1: Non-secure	R/W
15	—	This bit is read as 1. The write value should be 1.	R/W
16	PSARD16	ADC and the MSTPCRD.MSTPD16 bit security attribution 0: Secure 1: Non-secure	R/W
18:17	—	These bits are read as 1. The write value should be 1.	R/W
19	PSARD19	DAC12 unit1 and the MSTPCRD.MSTPD19 bit security attribution 0: Secure 1: Non-secure	R/W
20	PSARD20	DAC12 unit0 and the MSTPCRD.MSTPD20 bit security attribution 0: Secure 1: Non-secure	R/W
21	—	This bit is read as 1. The write value should be 1.	R/W
22	PSARD22	TSN and the MSTPCRD.MSTPD22 bit security attribution 0: Secure 1: Non-secure	R/W
24:23	—	These bits are read as 1. The write value should be 1.	R/W
25	PSARD25	ACMPHS3 and the MSTPCRD.MSTPD25 bit security attribution 0: Secure 1: Non-secure	R/W
26	PSARD26	ACMPHS2 and the MSTPCRD.MSTPD26 bit security attribution 0: Secure 1: Non-secure	R/W
27	PSARD27	ACMPHS1 and the MSTPCRD.MSTPD27 bit security attribution 0: Secure 1: Non-secure	R/W
28	PSARD28	ACMPHS0 and the MSTPCRD.MSTPD28 bit security attribution 0: Secure 1: Non-secure	R/W
31:29	—	These bits are read as 1. The write value should be 1.	R/W

Note: Only secure access can write to this register. Both secure access and Non-secure read access are allowed but Non-secure write access is not allowed and TrustZone access error is not generated.
Note: This register is write-protected by PRCR register.

The PSARD specifies the security attribution for each module and the corresponding bit in Module Stop Control Register.

45.5.4 PSARE : Peripheral Security Attribution Register E

Base address: PSCU = 0x400E_0000

Offset address: 0x10

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	PSARE4	—	—	PSARE1	PSARE0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	PSARE0	WDT security attribution 0: Secure 1: Non-secure	R/W
1	PSARE1	IWDT security attribution 0: Secure 1: Non-secure	R/W
3:2	—	These bits are read as 1. The write value should be 1.	R/W
4	PSARE4	KINT and the MSTPCRE.MSTPE4 bit security attribution 0: Secure 1: Non-secure	R/W
31:5	—	These bits are read as 1. The write value should be 1.	R/W

Note: Only secure access can write to this register. Both secure access and Non-secure read access are allowed but Non-secure write access is not allowed and TrustZone access error is not generated.
Note: This register is write-protected by PRCR register.

The PSARE specifies the security attribution for each module and the corresponding bit in Module Stop Control Register.

45.5.5 MSSAR : Module Stop Security Attribution Register

Base address: PSCU = 0x400E_0000

Offset address: 0x14

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MSSAR3	MSSAR2	MSSAR1	MSSAR0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	MSSAR0	The MSTPCRC.MSTPC14 bit security attribution 0: Secure 1: Non-secure	R/W
1	MSSAR1	The MSTPCRA.MSTPA22 bit security attribution 0: Secure 1: Non-secure	R/W
2	MSSAR2	The MSTPCRA.MSTPA7 bit security attribution 0: Secure 1: Non-secure	R/W
3	MSSAR3	The MSTPCRA.MSTPA0 bit security attribution 0: Secure 1: Non-secure	R/W
31:4	—	These bits are read as 1. The write value should be 1.	R/W

Note: Only secure access can write to this register. Both secure access and Non-secure read access are allowed but Non-secure write access is not allowed and TrustZone access error is not generated.
Note: This register is write-protected by PRCR register.

The MSSAR specifies the security attribution for the corresponding bit in Module Stop Control Register.

45.5.6 CFSAMONA : Code Flash Security Attribution Monitor Register A

Base address: PSCU = 0x400E_0000

Offset address: 0x18

Bit position:	31								23								15								0
Bit field:	—	—	—	—	—	—	—	—	CFS2[8:0]								—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	The value set by user								0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
14:0	—	These bits are read as 0.	R
23:15	CFS2[8:0]	Code Flash Secure area 2 Indicate the total area of secure region and non-secure callable region for code flash.	R
31:24	—	These bits are read as 0.	R

Note: The CFSAMONA does not have security attribution.

45.5.7 CFSAMONB : Code Flash Security Attribution Monitor Register B

Base address: PSCU = 0x400E_0000

Offset address: 0x1C

Bit position:	31								23								10								0
Bit field:	—	—	—	—	—	—	—	—	CFS1[13:0]								—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	The value set by user								0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	—	These bits are read as 0.	R
23:10	CFS1[13:0]	Code Flash Secure area 1 Indicate the area of secure region for code flash.	R
31:24	—	These bits are read as 0.	R

Note: The CFSAMONB does not have security attribution.

45.5.8 DFSAMON : Data Flash Security Attribution Monitor Register

Base address: PSCU = 0x400E_0000

Offset address: 0x20

Bit position:	31																15								10								0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DFS[5:0]								—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	The value set by user								0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	—	These bits are read as 0.	R
15:10	DFS[5:0]	Data flash Secure area Indicate the area of Secure region for data flash.	R
31:16	—	These bits are read as 0.	R

Note: The DFSAMON does not have security attribution.

45.5.9 SSAMONA : SRAM Security Attribution Monitor Register A

Base address: PSCU = 0x400E_0000

Offset address: 0x24

Bit position:	31											20												13												0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	SS2[7:0]												—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	The value set by user												0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
12:0	—	These bits are read as 0.	R
20:13	SS2[7:0]	SRAM Secure area 2 Indicate the total area of Secure region and non-secure callable region for SRAM.	R
31:21	—	These bits are read as 0.	R

Note: The SSAMONA does not have security attribution.

45.5.10 SSAMONB : SRAM Security Attribution Monitor Register B

Base address: PSCU = 0x400E_0000

Offset address: 0x28

Bit position:	31											20										10										0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	SS1[10:0]										—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	The value set by user										0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	—	These bits are read as 0.	R
20:10	SS1[10:0]	SRAM secure area 1 Indicate the area of secure region for SRAM.	R
31:21	—	These bits are read as 0.	R

Note: The SSAMONB does not have security attribution.

45.5.11 DLMMON : Device Lifecycle Management State Monitor Register

Base address: PSCU = 0x400E_0000

Offset address: 0x2C

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

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

Bit	Symbol	Function	R/W
3:0	DLMMON[3:0]	Device Lifecycle Management State Monitor 0x1: CM 0x2: SSD 0x3: NSECSD 0x4: DPL 0x5: LCK_DBG 0x6: LCK_BOOT 0x7: RMA_REQ 0x8: RMA_ACK Others: Reserved	R
31:4	—	These bits are read as 0. The write value should be 0.	R

Note: The DLMMON does not have security attribution.

45.5.12 TZFSAR : TrustZone Filter Security Attribution Register

Base address: CPSCU = 0x4000_8000

Offset address: 0x180

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TZFSA0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

Bit	Symbol	Function	R/W
0	TZFSA0	Security attributes of registers for TrustZone Filter 0: Secure 1: Non-secure	R/W
31:1	—	These bits are read as 1.	R

Note: Only secure access can write to this register. Both secure access and Non-secure read access are allowed but Non-secure write access is not allowed and TrustZone access error is not generated.

Note: This register is write-protected by PRCR register.

TZFSA0 bit (Security attributes of registers for TrustZone Filter)

Security attributes of register for TZFOAD and TZFPT registers.

45.5.13 TZFOAD : TrustZone Filter Operation After Detection Register

Base address: TZF = 0x4000_0E00

Offset address: 0x00

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

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

Bit	Symbol	Function	R/W
15:8	KEY[7:0]	KeyCode This bit is used to enable or disable writing of the OAD bit.	W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

OAD bit (Operation after detection)

The OAD bit is specified to generate either reset or non-maskable interrupt when the access to the protect region is detected by the TrustZone Filter.

When the OAD bit is set, write 0xA5 in the KEY[7:0] bits at the same time.

KEY[7:0] bits (KeyCode)

The KEY[7:0] bits are used to enable or disable writing of the OAD bit. When writing the OAD bit, write 0xA5 in the KEY[7:0] bits at the same time.

When the KEY[7:0] bits value except 0xA5 is written in, the OAD bit is not updated.

The KEY[7:0] bits are read always as 0x00.

45.5.14 TZFPT : TrustZone Filter Protect Register

Base address: TZF = 0x4000_0E00

Offset address: 0x04

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

Bit	Symbol	Function	R/W
0	PROTECT	Protection of register 0: All Bus TrustZone Filter register writing is protected. Read is possible. 1: All Bus TrustZone Filter register writing is possible.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	KeyCode This bit is used to enable or disable writing of the PROTECT bit.	W

Note: If the security attribution is configured as secure:
- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:
- Secure and Non-secure access are allowed.

PROTECT bit (Protection of register)

The PROTECT bit controls enable or disable writing to the corresponding registers to be protected. TZFOAD register is protected by PROTECT.

When the PROTECT bit is set, write 0xA5 in the KEY[7:0] bits at the same time.

KEY[7:0] bits (KeyCode)

The KEY[7:0] bits are used to enable or disable writing of the PROTECT bit. When writing the PROTECT bit, write 0xA5 in the KEY[7:0] bits at the same time.

When the KEY[7:0] bits value except 0xA5 is written in, the PROTECT bit is not updated.

The KEY[7:0] bits are read always as 0x00.

45.6 Usage Notes

45.6.1 Restrictions on setting the security attribution

To set the software breakpoint, the debugger need to re-program the flash. Table 45.10 shows the registers that the debugger sets to re-program the flash. If the security attribution of the register listed in Table 45.10 is configured as secure, the debugger cannot set the software breakpoint in NSECSD state because the debugger cannot change the register setting. Secure developper need to configure the registers listed in Table 45.10 as non-secure only in NSECSD state.

Table 45.10 The registers that the debugger sets to re-program the flash

Function name	Register name
Clock Generation Circuit	SCKDIVCR, SCKCR, PLLCCR, PLLCR, HOCOCR, MOCOCR
Low-Power modes	OPCCR

45.6.2 SAU setting

After reset, all of address space is marked as Secure by SAU default setting. SAU_CTRL register should be set to 0x2 to enable the IDAU security attribution. That is, after setting SAU_CTRL register to 0x2, the address space security attribution becomes as shown in Table 45.6.

45.6.3 Non-secure exception during the setting of FACI registers

As shown in Table 45.5, the registers related to FACI are protected from non-secure access only during programming/erasure or during suspend programming/erasure. Outside of this state, the access from non-secure region is not protected. For example, when programming by the secure user, the non-secure user can rewrite the FSADDR if a non-secure exception occurs immediately after “Set the start address of the target block to the FSADDR register” flow in Figure 43.13. If the FACI command is issued after the non-secure exception processing is completed and the CPU state returns to the secure state, data will be programmed to an address not intended by the secure user.

To prevent such a things, secure user needs to set not to accept the non-secure exception during the following period.

- Set not to accept the non-secure exeption before setting FWEPROR to 0x01 or setting FENTRYR to other than 0x0000, that is before lereasing the protection of FWEPROR or FENTRYR.
- Set to accept the non-secure exception after all write access to the FACI command-issuing area is completed.

45.6.4 FCU interrupt usage

It is recommended that secure users do not use the FCU interrupts, but rather use the register polling. Because non-secure users can program/erase the data flash without calling the secure gateway, if secure user uses FCU interrupts, the unintentional exception handliing may be exeuted when data flash is programmed/erased by a non-secure user.

46. Electrical Characteristics

Unless otherwise specified, minimum and maximum values are guaranteed by either design simulation, characterization results or test in production.

Supported peripheral functions and pins differ from one product name to another.

Unless otherwise specified, the electrical characteristics of the MCU are defined under the following conditions:

- VCC = AVCC0 = 2.7 to 3.6 V
- 2.7 V ≤ VREFH0 ≤ AVCC0
- VSS = AVSS0 = VREFL0 = 0 V
- $T_a = T_{opr}$

Figure 46.1 shows the timing conditions.

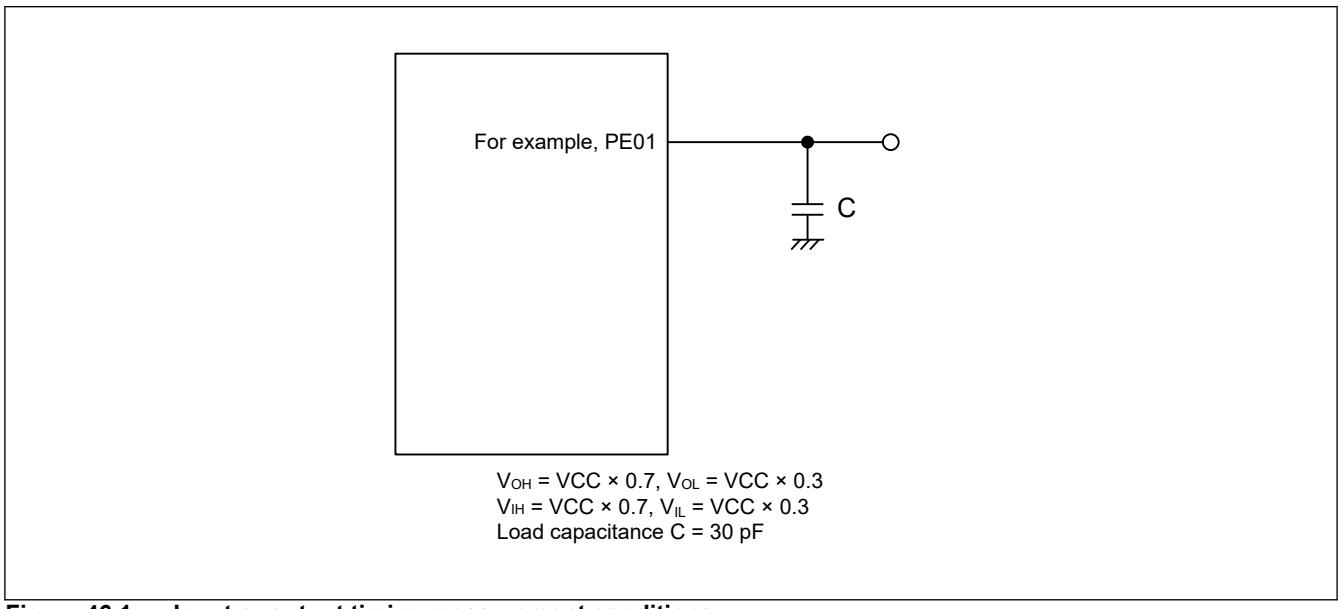


Figure 46.1 Input or output timing measurement conditions

The recommended measurement conditions for the timing specification of each peripheral provided are for the best peripheral operation. Make sure to adjust the driving abilities of each pin to meet your conditions.

46.1 Absolute Maximum Ratings

Table 46.1 Absolute maximum ratings (1 of 2)

Parameter	Symbol	Value	Unit
Power supply voltage	VCC	–0.3 to +4.0	V
Input voltage (except for 5 V-tolerant ports*1)	V_{in}	–0.3 to VCC + 0.3	V
Input voltage (5 V-tolerant ports*1)	V_{in}	–0.3 to VCC + 4.0 (max. 5.8)	V
Reference power supply voltage	VREFH0	–0.3 to VCC + 0.3	V
Analog power supply voltage	AVCC0*2	–0.3 to +4.0	V
Analog input voltage (except for PA00 to PA05, PB02, P002)	V_{AN}	–0.3 to AVCC0 + 0.3	V
Analog input voltage (PA00 to PA05, PB02, P002) when PGA differential input is disabled	V_{AN}	–0.3 to AVCC0 + 0.3	V
Analog input voltage (PA00, PA02, PA04, PB02) when PGA differential input is enabled	V_{AN}	–1.3 to AVCC0 + 0.3	V
Analog input voltage (PA01, PA03, PA05, P002) when PGA differential input is enabled	V_{AN}	–0.8 to AVCC0 + 0.3	V

Table 46.1 Absolute maximum ratings (2 of 2)

Parameter	Symbol	Value	Unit
Operating temperature*3 *4	T_{opr}	–40 to +105	°C
Storage temperature	T_{stg}	–55 to +125	°C

Note 1. Ports PA12 to PA15, PB03, PB05 to PB09, PC10 to PC12, PC14, PC15, PD00 to PD07, PE00, and PE01 are 5 V tolerant.
Note 2. Connect AVCC0 to VCC.
Note 3. See section 46.2.1. Tj/Ta Definition.
Note 4. Contact a Renesas Electronics sales office for information on derating operation when Ta = +85°C to +105°C. Derating is the systematic reduction of load for improved reliability.

Caution: Permanent damage to the MCU might result if absolute maximum ratings are exceeded.

Table 46.2 Recommended operating conditions

Parameter	Symbol	Min	Typ	Max	Unit
Power supply voltages	VCC	2.7	—	3.6	V
	VSS	—	0	—	V
Analog power supply voltages	AVCC0*1	—	VCC	—	V
	AVSS0	—	0	—	V

Note 1. Connect AVCC0 to VCC. When the A/D converter and the D/A converter are not in use, do not leave the AVCC0, VREFH0, AVSS0, and VREFL0 pins open. Connect the AVCC0 and VREFH0 pins to VCC, and the AVSS0 and VREFL0 pins to VSS, respectively.

46.2 DC Characteristics

46.2.1 Tj/Ta Definition

Table 46.3 DC characteristics

Parameter	Symbol	Typ	Max	Unit	Test conditions
Permissible junction temperature	T_j	—	125	°C	High-speed mode Low-speed mode

Note: Make sure that $T_j = T_a + \theta ja \times$ total power consumption (W), where total power consumption = $(VCC - V_{OH}) \times \Sigma I_{OH} + V_{OL} \times \Sigma I_{OL} + I_{CC}max \times VCC$.

46.2.2 I/O V_{IH}, V_{IL}

Table 46.4 I/O V_{IH}, V_{IL}

Parameter			Symbol	Min	Typ	Max	Unit
Input voltage (except for Schmitt trigger input pins)	Peripheral function pin	EXTAL (external clock input), SPI (except RSPCK)	V_{IH}	VCC × 0.8	—	—	V
			V_{IL}	—	—	VCC × 0.2	
		IIC (SMBus)*1	V_{IH}	2.1	—	—	
			V_{IL}	—	—	0.8	
		IIC (SMBus)*2	V_{IH}	2.1	—	VCC + 3.6 (max 5.8)	
			V_{IL}	—	—	0.8	
Schmitt trigger input voltage	Peripheral function pin	IIC (Except for SMBus)*1	V_{IH}	VCC × 0.7	—	—	
			V_{IL}	—	—	VCC × 0.3	
			ΔV_T	VCC × 0.05	—	—	
		IIC (Except for SMBus)*2	V_{IH}	VCC × 0.7	—	VCC + 3.6 (max 5.8)	
			V_{IL}	—	—	VCC × 0.3	
			ΔV_T	VCC × 0.05	—	—	
		5 V-tolerant ports*3 *7	V_{IH}	VCC × 0.8	—	VCC + 3.6 (max 5.8)	
			V_{IL}	—	—	VCC × 0.2	
			ΔV_T	VCC × 0.05	—	—	
		Other input pins*4	V_{IH}	VCC × 0.8	—	—	
			V_{IL}	—	—	VCC × 0.2	
			ΔV_T	VCC × 0.05	—	—	
	Ports	5 V-tolerant ports*5 *7	V_{IH}	VCC × 0.8	—	VCC + 3.6 (max 5.8)	
			V_{IL}	—	—	VCC × 0.2	
			ΔV_T	VCC × 0.05	—	—	
		Other input pins*6	V_{IH}	VCC × 0.8	—	—	
			V_{IL}	—	—	VCC × 0.2	
			ΔV_T	VCC × 0.05	—	—	

Note 1. SCL0_C, SDA0_C, SCL0_D, SDA0_D, SCL0_E, SDA0_E, SCL0_F, SDA0_F, SCL1_C, SDA1_C, SCL1_D, SDA1_D, SCL1_E, SDA1_E (total 14 pins). This is the value when IIC function is selected.
Note 2. SCL0_A, SDA0_A, SCL0_B, SDA0_B, SCL1_A, SDA1_A, SCL1_B, SDA1_B (total 8 pins). This is the value when IIC function is selected.
Note 3. RES and peripheral function pins associated with PA12 to PA15, PB03, PB05 to PB09, PC10 to PC12, PC14, PC15, PD00 to PD07, PE00, and PE01 (total 26 pins).
Note 4. All input pins except for the peripheral function pins already described in the table.
Note 5. PA12 to PA15, PB03, PB05 to PB09, PC10 to PC12, PC14, PC15, PD00 to PD07, PE00, and PE01 (total 25 pins).
Note 6. All input pins except for the ports already described in the table.
Note 7. When VCC is less than 2.7 V, the input voltage of 5 V-tolerant ports should be less than 3.6 V, otherwise breakdown may occur because 5 V-tolerant ports are electrically controlled so as not to violate the break down voltage.

46.2.3 I/O I_{OH}, I_{OL}

Table 46.5 I/O I_{OH}, I_{OL}

Parameter			Symbol	Min	Typ	Max	Unit
Permissible output current (average value per pin)	IIC pins	Standard mode*1	I_{OL}	—	—	3.0	mA
		Fast mode*1	I_{OL}	—	—	6.0	mA
		Fast mode plus*2	I_{OL}	—	—	20	mA
		High speed mode*2	I_{OL}	—	—	3.0	mA
	Other output pins*3	Low drive*4	I_{OH}	—	—	–2.0	mA
			I_{OL}	—	—	2.0	mA
		Middle drive*5	I_{OH}	—	—	–4.0	mA
			I_{OL}	—	—	4.0	mA
		High drive*6	I_{OH}	—	—	–10	mA
			I_{OL}	—	—	10	mA
		High speed high drive*7	I_{OH}	—	—	–10	mA
			I_{OL}	—	—	10	mA
		High current drive*8	I_{OH}	—	—	–10	mA
			I_{OL}	—	—	20	mA
Permissible output current (max value per pin)	IIC pins	Standard mode*1	I_{OL}	—	—	3.0	mA
		Fast mode*1	I_{OL}	—	—	6.0	mA
		Fast mode plus*2	I_{OL}	—	—	20	mA
		High speed mode*2	I_{OL}	—	—	3.0	mA
	Other output pins*3	Low drive*4	I_{OH}	—	—	–2.0	mA
			I_{OL}	—	—	2.0	mA
		Middle drive*5	I_{OH}	—	—	–4.0	mA
			I_{OL}	—	—	4.0	mA
		High drive*6	I_{OH}	—	—	–16	mA
			I_{OL}	—	—	16	mA
		High speed high drive*7	I_{OH}	—	—	–16	mA
			I_{OL}	—	—	16	mA
		High current drive*8	I_{OH}	—	—	–16	mA
			I_{OL}	—	—	20	mA
Permissible output current (max value of total of all pins)	Maximum of all output pins		ΣI_{OH} (max)	—	—	–80	mA
			ΣI_{OL} (max)	—	—	80	mA

Note 1. SCL0_A, SDA0_A, SCL1_A, SDA1_A (total 4 pins). This is the value when IIC function is selected.
Note 2. SCL0_A, SDA0_A (total 2 pins). This is the value when IIC function is selected.
Note 3. Except for P000 to P002, PA00 to PA07, PB00 to PB02, PC00 to PC05, PC13, which are input ports.
Note 4. This is the value when low driving ability is selected in the Port Drive Capability bit in the PmnPFS register. The selected driving ability is retained in Deep Software Standby mode.
Note 5. This is the value when middle driving ability is selected in the Port Drive Capability bit in the PmnPFS register. The selected driving ability is retained in Deep Software Standby mode.
Note 6. This is the value when high driving ability is selected in the Port Drive Capability bit in the PmnPFS register. The selected driving ability is retained in Deep Software Standby mode.

Note 7. This is the value when high speed high driving ability is selected in the Port Drive Capability bit in the PmnPFS register. The selected driving ability is retained in Deep Software Standby mode.
Note 8. This is the value when high current driving ability is selected in the Port Drive Capability bit in the PmnPFS register. The selected driving ability is retained in Deep Software Standby mode.

Caution: To protect the reliability of the MCU, the output current values should not exceed the values in this table. The average output current indicates the average value of current measured during 100 µs.

46.2.4 I/O V_{OH}, V_{OL}, and Other Characteristics

Table 46.6 I/O V_{OH}, V_{OL}, and other characteristics

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Output voltage	IIC*1	V_{OL}	—	—	0.4	V	I_{OL} = 3.0 mA
		V_{OL}	—	—	0.6		I_{OL} = 6.0 mA
	IIC*2	V_{OL}	—	—	0.4		I_{OL} = 15.0 mA (BFCTL.FMPE = 1)
		V_{OL}	—	0.4	—		I_{OL} = 20.0 mA (BFCTL.FMPE = 1)
		V_{OL}	—	—	0.4		I_{OL} = 3.0 mA (BFCTL.HSME = 1)
	Ports PA08 to PA11, PB12 to PB15, PC06 to PC09, PD08 to PD15, PE10 to PE15*3	V_{OH}	VCC – 0.5	—	—		I_{OH} = –1.0 mA
		V_{OL}	—	—	0.6		I_{OL} = 20 mA
	Other output pins	V_{OH}	VCC – 0.5	—	—		I_{OH} = –1.0 mA
		V_{OL}	—	—	0.5		I_{OL} = 1.0 mA
Input leakage current	RES	$\|I_{in}\|$	—	—	5.0	µA	V_{in} = 0 V V_{in} = 5.5 V
	Port P000, P001, PA06, PA07, PB00, PB01, PC00 to PC05, PC13		—	—	1.0		V_{in} = 0 V V_{in} = VCC
	Port PA00, PA02, PA04, PB02 (PGA input pins)		—	—	1.0		V_{in} = 0 V V_{in} = VCC
	Port PA01, PA03, PA05, P002 (PGAVSS pins)*4		—	—	1.0		V_{in} = 0 V V_{in} = VCC
Three-state leakage current (off state)	5 V-tolerant ports	$\|I_{TSI}\|$	—	—	5.0	µA	V_{in} = 0 V V_{in} = 5.5 V
	Other ports (except for input ports)		—	—	1.0		V_{in} = 0 V V_{in} = VCC
Input pull-up MOS current	Ports P0, P2, PA to PE (except for ports P002, PA00 to PA05, PB02)	I_p	–300	—	–10	µA	VCC = 2.7 to 3.6 V V_{in} = 0 V
Pull-up current serving as the SCL current source	IIC*5	I_{CS}	3	—	12	mA	VCC = 3.0 to 3.6 V V_{in} = 0.3 × VCC to 0.7 × VCC
Input capacitance	All input pins	C_{in}	—	—	8	pF	Vbias = 0 V Vamp = 20 mV f = 1 MHz Ta = 25°C

Note 1. SCL0_A, SDA0_A, SCL1_A, SDA1_A (total 4 pins). This is the value when IIC function is selected.
Note 2. SCL0_A, SDA0_A (total 2 pins). This is the value when IIC function is selected.
Note 3. This is the value when high current driving ability is selected in the Port Drive Capability bit in the PmnPFS register. The selected driving ability is retained in Deep Software Standby mode.
Note 4. This is the value when the pseudo-differential input on the PGAn pin is disabled (single-ended input).
Note 5. SCL0_A (1 pin). This is the value when IIC high speed mode is selected.

46.2.5 Operating and Standby Current

Table 46.7 Operating and standby current

Parameter				Symbol	Min	Typ	Max	Unit	Test conditions
Supply current[*1]	High-speed mode	Maximum[*2]		I_{CC}[*3]	—	—	150	mA	ICLK = 240 MHz PCLKA = 120 MHz PCLKB = 60 MHz PCLKC = 60 MHz PCLKD = 120 MHz FCLK = 60 MHz
		CoreMark®[*5] [*6]			—	34	—		
		Normal mode	All peripheral clocks enabled, while (1) code executing from flash[*4]		—	44	—		
			All peripheral clocks disabled, while (1) code executing from flash[*5] [*6]		—	28	—		
		Sleep mode[*5] [*6]			—	13	78		
		Increase during BGO operation	Data flash P/E		—	6	—		
			Code flash P/E		—	8	—		
	Low-speed mode[*5] [*10]				—	5	—		ICLK = 1 MHz
	Software Standby mode		SNZCR.RXDREQEN = 1		—	—	63		ICLK = 32.768 kHz
			SNZCR.RXDREQEN = 0		—	5.1	—		—
	Deep Software Standby mode	Power supplied to Standby SRAM			—	22.7	60	µA	—
		Power not supplied to SRAM	Power-on reset circuit low power function disabled		—	11.3	30		—
			Power-on reset circuit low power function enabled		—	4.4	20		—
	Inrush current on returning from deep software standby mode		Inrush current[*7]	I_{RUSH}	—	160	—	mA	—
			Energy of inrush current[*7]	E_{RUSH}	—	1.0	—	µC	—
Analog power supply current	During A/D conversion (1unit)		Without SH	AI_{CC}	—	4.9	6.0	mA	—
			With SH		—	8.4	11.5	mA	—
	PGA (1channel)				—	1	3	mA	—
	ACMPHS (1unit)				—	0.1	0.2	mA	—
	Temperature sensor				—	0.1	0.2	mA	—
	During D/A conversion (1channel)[*8]		Without AMP output		—	0.2	0.3	mA	—
			With AMP output		—	0.8	1.3	mA	—
	Waiting for A/D, D/A conversion (all units)				—	3.8	4.5	mA	—
	ADC, DAC12 in standby modes (all units)[*9]				—	0.7	10	µA	—
Reference power supply current (VREFH0)	During A/D conversion (1unit)		SAR mode	AI_{REFH0}	—	21	50	µA	—
			Oversampling mode and Hybrid mode		—	100	160	µA	—
	Waiting for A/D conversion (all units)				—	18	50	µA	—
	ADC in standby modes (all units)[*9]				—	0.03	1	µA	—

Note 1. Supply current values are with all output pins unloaded and all input pull-up MOSs in the off state.
Note 2. Measured with clocks supplied to the peripheral functions. This does not include the BGO operation.
Note 3. I_{CC} depends on f (ICLK) as follows.
I_{CC} Max. = 0.34 × f + 67 (max. operation in high-speed mode)
I_{CC} Typ. = 0.095 × f + 4.7 (normal operation in high-speed mode, all peripheral clocks disabled)
I_{CC} Typ. = 0.9 × f + 4.1 (low-speed mode)
I_{CC} Max. = 0.045 × f + 67 (sleep mode)
Note 4. This does not include the BGO operation.
Note 5. Supply of the clock signal to peripherals is stopped in this state. This does not include the BGO operation.
Note 6. FCLK, PCLKA, PCLKB, PCLKC, and PCLKD are set to divided by 64 (3.75 MHz).

Note 7. Reference value
Note 8. The DAC12 includes the Reference current in the analog power supply current.
Note 9. When the MCU is in Software Standby mode or the MSTPCRD.MSTPD16 (A/D Converter Module Stop bit) is in the module-stop state.
Note 10. FCLK, PCLKA, PCLKB, PCLKC, and PCLKD are set to divided by 64 (15.6 kHz).

Table 46.8 Coremark and normal mode current

Parameter			Symbol	Typ	Unit	Test conditions
Supply Current*1	Coremark		I_{CC}	139	µA/MHz	ICLK = 240 MHz PCLKA = PCLKB = PCLKC = PCLKD = FCLK = 3.75 MHz
	Normal mode	All peripheral clocks disabled, cache on, while (1) code executing from flash*2		139		
		All peripheral clocks disabled, cache off, while (1) code executing from flash*2		115		

Note 1. Supply current values are with all output pins unloaded and all input pull-up MOSs in the off state.
Note 2. Supply of the clock signal to peripherals is stopped in this state. This does not include the BGO operation.

46.2.6 VCC Rise and Fall Gradient and Ripple Frequency

Table 46.9 VCC rise and fall gradient characteristics

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
VCC rising gradient	Voltage monitor 0 reset disabled at startup	SrVCC	0.0084	—	20	ms/V	—
	Voltage monitor 0 reset enabled at startup		0.0084	—	—		—
	SCI boot mode*1		0.0084	—	20		—
VCC falling gradient		SfVCC	0.0084	—	—	ms/V	—

Note 1. At boot mode, the reset from voltage monitor 0 is disabled regardless of the value of the OFS1.LVDAS bit.

Table 46.10 VCC rising and falling gradient and ripple frequency characteristics

The ripple voltage must meet the allowable ripple frequency $f_{r(VCC)}$ within the range between the VCC upper limit (3.6 V) and lower limit (2.7 V). When the VCC change exceeds VCC ±10%, the allowable voltage change rising and falling gradient dt/dVCC must be met.

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
Allowable ripple frequency	$f_{r\ (VCC)}$	—	—	10	kHz	Figure 46.2 $V_{r\ (VCC)} \leq VCC \times 0.2$
		—	—	1	MHz	Figure 46.2 $V_{r\ (VCC)} \leq VCC \times 0.08$
		—	—	10	MHz	Figure 46.2 $V_{r\ (VCC)} \leq VCC \times 0.06$
Allowable voltage change rising and falling gradient	dt/dVCC	1.0	—	—	ms/V	When VCC change exceeds VCC ±10%

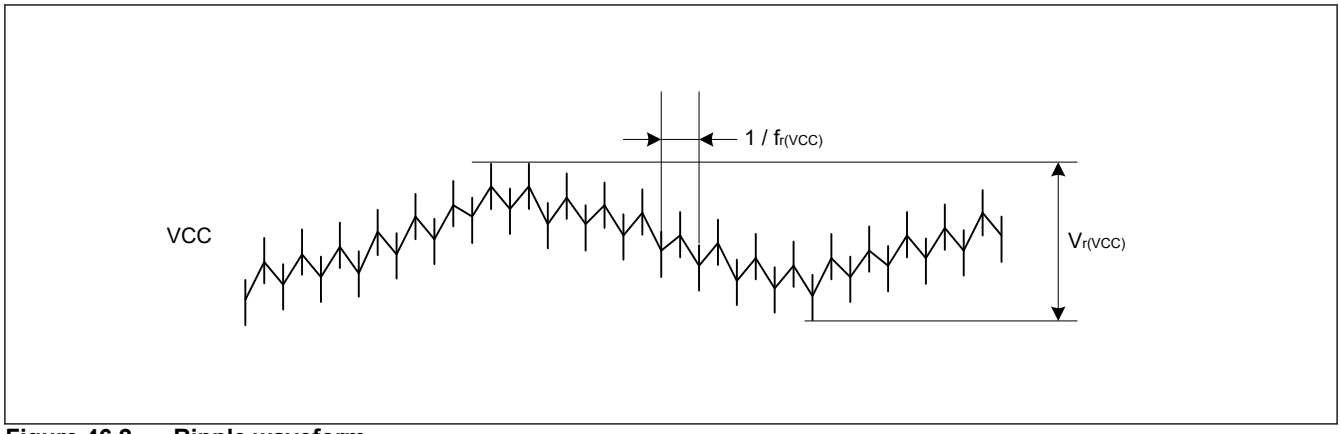


Figure 46.2 Ripple waveform

46.2.7 Thermal Characteristics

Maximum value of junction temperature (Tj) must not exceed the value of section 46.2.1. Tj/Ta Definition.

Tj is calculated by either of the following equations.

- Tj = Ta + θja × Total power consumption
- Tj = Tt + Ψjt × Total power consumption
  - Tj : Junction Temperature (°C)
  - Ta : Ambient Temperature (°C)
  - Tt : Top Center Case Temperature (°C)
  - θja : Thermal Resistance of "Junction"-to-"Ambient" (°C/W)
  - Ψjt : Thermal Resistance of "Junction"-to-"Top Center Case" (°C/W)
- Total power consumption = Voltage × (Leakage current + Dynamic current)
- Leakage current of IO = Σ (I_{OL} × V_{OL}) /Voltage + Σ (|I_{OH}| × |VCC – V_{OH}|) /Voltage
- Dynamic current of IO = Σ IO (C_{in} + C_{load}) × IO switching frequency × Voltage
  - C_{in}: Input capacitance
  - C_{load}: Output capacitance

Regarding θja and Ψjt, see Table 46.11.

Table 46.11 Thermal Resistance

Parameter	Package	Symbol	Value*1	Unit	Test conditions
Thermal Resistance	100-pin LQFP (PLQP0100KB-B)	θja	36	°C/W	JESD 51-2 and 51-7 compliant
	64-pin LQFP (PLQP0064KB-C)		39		
	64-pin QFN (PWQN0064LB-A)		26		
	48-pin LQFP (PLQP0048KB-B)		60		
	48-pin QFN (PWQN0048KC-A)		28		
	100-pin LQFP (PLQP0100KB-B)	Ψjt	0.65	°C/W	JESD 51-2 and 51-7 compliant
	64-pin LQFP (PLQP0064KB-C)		0.69		
	64-pin QFN (PWQN0064LB-A)		0.15		
	48-pin LQFP (PLQP0048KB-B)		2.01		
	48-pin QFN (PWQN0048KC-A)		0.17		

Note 1. The values are reference values when the 4-layer board is used. Thermal resistance depends on the number of layers or size of the board. For details, refer to the JEDEC standards.

46.2.7.1 Calculation guide of ICCmax

Table 46.12 shows the power consumption of each unit.

Table 46.12 Power consumption of each unit

Dynamic current/Leakage current	MCU Domain	Category	Item	Frequency [MHz]	Current [uA/ MHz]	Current*1 [mA]
Leakage current	Analog	LDO and Leak*2	Ta = 75 °C*3	—	—	37.8
			Ta = 85 °C*3	—	—	46.4
			Ta = 95 °C*3	—	—	56.1
			Ta = 105 °C*3	—	—	68.0
Dynamic current	CPU	Operation with Flash and SRAM	Coremark	240	105.324	25.28
	Peripheral Unit	Timer	GPT32 (10ch)*4	120	29.697	3.56
			POEG (4 Groups)*4	60	1.483	0.09
			AGT (2ch)*4	60	3.09	0.19
			WDT	60	0.641	0.04
			IWDT	60	0.225	0.01
		Communication interfaces	SCI (6ch)*4	120	27.683	3.32
			IIC (2ch)*4	120	5.304	0.64
			CANFD	60	5.763	0.35
			SPI (2ch)*4	120	5.738	0.69
		Data processing accelerator	TFU	240	1.188	0.03
			IIRFA	240	34.252	8.22
		Data processing	DOC	120	0.221	0.03
			CRC	120	0.508	0.06
		Analog	ADC (2 Units)*4	60	172.958	10.38
			DAC12 (4ch)*4	120	1.097	0.13
			ACMPHS (4ch)*4	60	0.641	0.04
			TSN	60	0.111	0.01
		Event link	ELC	60	1.852	0.11
		Security	SCE5	120	68.404	8.21
		System	CAC	60	0.63	0.04
			KINT	60	0.072	0.004
		DMA	DMAC	240	5.073	1.22
			DTC	240	4.18	1

Note 1. The values are guaranteed by design.
Note 2. LDO and Leak are internal voltage regulator's current and MCU's leakage current. It is selected according to the temperature of Ta.
Note 3. Δ(Tj-Ta) = 20 °C is considered to measure the current.
Note 4. To determine the current consumption per channel or unit, divide Current [mA] by the number of channels, groups or units.

Table 46.13 Outline of operation for each unit (1 of 2)

Peripheral	Outline of operation
GPT	Operating modes is set to saw-wave PWM mode. GPT is operating with PCLKD.

Table 46.13 Outline of operation for each unit (2 of 2)

Peripheral	Outline of operation
POEG	Only clear module stop bit.
AGT	AGT is operating with PCLKB.
WDT	WDT is operating with PCLKB.
IWDT	IWDT is operating with IWDTCLK.
SCI	SCI is transmitting data in clock synchronous mode.
IIC	Communication format is set to I2C-bus format. IIC is transmitting data in master mode.
CANFD	CANFD is transmitting and receiving data in self-test mode 1.
SPI	SPI mode is set to SPI operation (4-wire method). SPI master/slave mode is set to master mode. SPI is transmitting 8-bit width data.
TFU	Performs sincos operations.
IIRFA	Channel 0 perfoms 32 stages of channel processing.
DOC	DOC is operating in data addition mode.
CRC	CRC is generating CRC code using 32-bit CRC32-C polynomial.
ADC	Resolution is set to 12-bit accuracy. Conversion Data Operation Control B Register is set to 16 times average mode. ADC is converting the analog input in continuous scan mode. ADC is operating with PCLKC.
DAC12	DAC12 is outputting the conversion result while updating the value of data register.
ACMPHS	Compare between IVCMP2 and IVREF0 and enable compare output.
TSN	TSN is operating.
ELC	Only clear module stop bit.
SCE5	SCE5 is executing built-in self test.
DMAC	Bit length of transfer data is set to 32 bits. Transfer mode is set to block transfer mode. DMAC is transferring data from SRAM0 to SRAM0.
DTC	Bit length of transfer data is set to 32 bits. Transfer mode is set to block transfer mode. DTC is transferring data from SRAM0 to SRAM0.
CAC	Measurement target clocks is set to PCLKB. Measurement reference clocks is set to PCLKB. CAC is measuring the clock frequency accuracy.
KINT	Only clear module stop bit.

46.2.7.2 Example of Tj calculation

Assumption:

- Package 100-pin LQFP: θja = 36.0 °C/W
- Ta = 100 °C
- I_{CC}max = 80 mA
- VCC = 3.5 V (VCC = AVCC0)
- I_{OH} = 1 mA, V_{OH} = VCC – 0.5 V, 12 Outputs
- I_{OL} = 20 mA, V_{OL} = 1.0 V, 8 Outputs
- I_{OL} = 1 mA, V_{OL} = 0.5 V, 12 Outputs
- C_{in} = 8 pF, 16 pins, Input frequency = 10 MHz

- C_{load} = 30 pF, 16 pins, Output frequency = 10 MHz

Leakage current of IO = Σ (V_{OL} × I_{OL}) / Voltage + Σ ((VCC - V_{OH}) × I_{OH}) / Voltage
= (20 mA × 1 V) × 8 / 3.5 V + (1 mA × 0.5 V) × 12 / 3.5 V + ((VCC - (VCC - 0.5 V)) × 1 mA) × 12 / 3.5 V
= 45.7 mA + 1.71 mA + 1.71 mA
= 49.1 mA

Dynamic current of IO = Σ IO (C_{in} + C_{load}) × IO switching frequency × Voltage
= ((8 pF × 16) × 10 MHz + (30 pF × 16) × 10 MHz) × 3.5 V
= 21.3 mA

Total power consumption = Voltage × (Leakage current + Dynamic current)
= (80 mA × 3.5 V) + (49.1 mA + 21.3 mA) × 3.5 V
= 526 mW (0.526 W)

Tj = Ta + θja × Total power consumption
= 100 °C + 36.0 °C/W × 0.526W
= 118.9 °C

46.3 AC Characteristics

46.3.1 Frequency

Table 46.14 Operation frequency value in high-speed mode

Parameter		Symbol	Min	Typ	Max	Unit
Operation frequency	System clock (ICLK)*2	f	—	—	240	MHz
	Peripheral module clock (PCLKA)*2		—	—	120	
	Peripheral module clock (PCLKB)*2		—	—	60	
	Peripheral module clock (PCLKC)*2		—*3	—	60	
	Peripheral module clock (PCLKD)*2		—	—	120	
	Flash interface clock (FCLK)*2		—*1	—	60	

Note 1. FCLK must run at a frequency of at least 4 MHz when programming or erasing the flash memory.
Note 2. See section 8, Clock Generation Circuit for the relationship between the ICLK, PCLKA, PCLKB, PCLKC, PCLKD, and FCLK frequencies.
Note 3. When the ADC is used, the PCLKC frequency must be at least 1 MHz.

Table 46.15 Operation frequency value in low-speed mode

Parameter		Symbol	Min	Typ	Max	Unit
Operation frequency	System clock (ICLK)*2	f	—	—	1	MHz
	Peripheral module clock (PCLKA)*2		—	—	1	
	Peripheral module clock (PCLKB)*2		—	—	1	
	Peripheral module clock (PCLKC)*2 *3		—*3	—	1	
	Peripheral module clock (PCLKD)*2		—	—	1	
	Flash interface clock (FCLK)*1 *2		—	—	1	

Note 1. Programming or erasing the flash memory is disabled in low-speed mode.
Note 2. See section 8, Clock Generation Circuit for the relationship between the ICLK, PCLKA, PCLKB, PCLKC, PCLKD, and FCLK frequencies.
Note 3. When the ADC is used, the PCLKC frequency must be set to at least 1 MHz.

46.3.2 Clock Timing

Table 46.16 Clock timing

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
EXTAL external clock input cycle time	t_{EXcyc}	41.66	—	—	ns	Figure 46.3
EXTAL external clock input high pulse width	t_{EXH}	15.83	—	—	ns	
EXTAL external clock input low pulse width	t_{EXL}	15.83	—	—	ns	
EXTAL external clock rise time	t_{EXr}	—	—	5.0	ns	
EXTAL external clock fall time	t_{EXf}	—	—	5.0	ns	
Main clock oscillator frequency	f_{MAIN}	8	—	24	MHz	—
Main clock oscillation stabilization wait time (crystal)*1	$t_{MAINOSCWT}$	—	—	—*1	ms	Figure 46.4
LOCO clock oscillation frequency	f_{LOCO}	29.4912	32.768	36.0448	kHz	—
LOCO clock oscillation stabilization wait time	t_{LOCOWT}	—	—	60.4	µs	Figure 46.5
ILOCO clock oscillation frequency	f_{ILOCO}	13.5	15	16.5	kHz	—
MOCO clock oscillation frequency	F_{MOCO}	6.8	8	9.2	MHz	—
MOCO clock oscillation stabilization wait time	t_{MOCOWT}	—	—	15.0	µs	—
HOCO clock oscillator oscillation frequency	f_{HOCO16}	15.78	16	16.22	MHz	–20 ≤ Ta ≤ 105°C
	f_{HOCO18}	17.75	18	18.25		
	f_{HOCO20}	19.72	20	20.28		
	f_{HOCO16}	15.71	16	16.29		–40 ≤ Ta ≤ –20°C
	f_{HOCO18}	17.68	18	18.32		
	f_{HOCO20}	19.64	20	20.36		
HOCO clock oscillation stabilization wait time*2	t_{HOCOWT}	—	—	64.7	µs	—
HOCO period jitter	—	—	±85	—	ps	—
PLL clock frequency	f_{PLL}	120	—	240	MHz	—
PLL2 clock frequency	f_{PLL2}	120	—	240	MHz	—
PLL/PLL2 clock oscillation stabilization wait time	t_{PLLWT}	—	—	174.9	µs	Figure 46.6
PLL/PLL2 period jitter	—	—	±100	—	ps	—
PLL/PLL2 long term jitter	—	—	±300	—	ps	Term: 1µs, 10µs

Note 1. When setting up the main clock oscillator, ask the oscillator manufacturer for an oscillation evaluation, and use the results as the recommended oscillation stabilization time. Set the MOSCWTCR register to a value equal to or greater than the recommended value.
After changing the setting in the MOSCCR.MOSTP bit to start main clock operation, read the OSCSF.MOSCSF flag to confirm that it is 1, and then start using the main clock oscillator.

Note 2. This is the time from release from reset state until the HOCO oscillation frequency (f_{HOCO}) reaches the range for guaranteed operation.

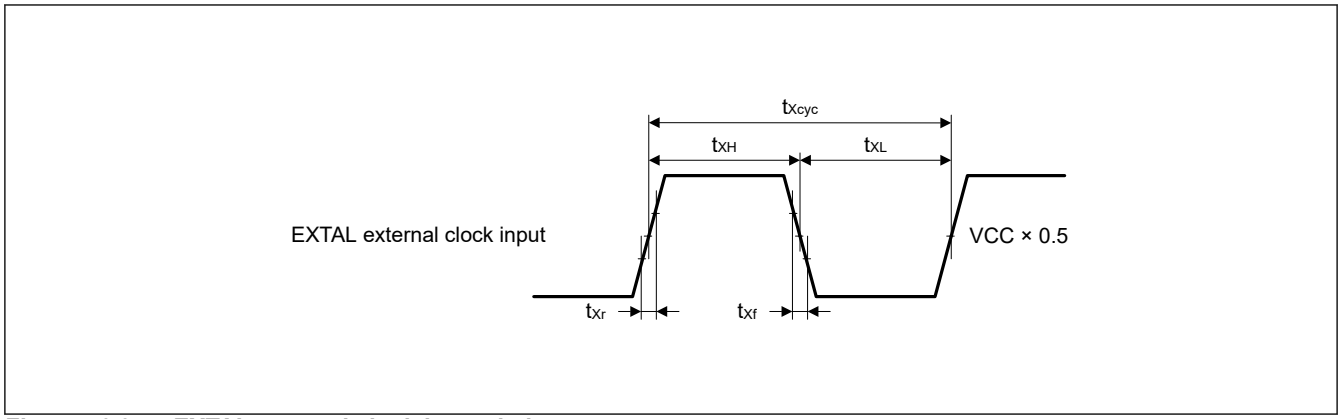


Figure 46.3 EXTAL external clock input timing

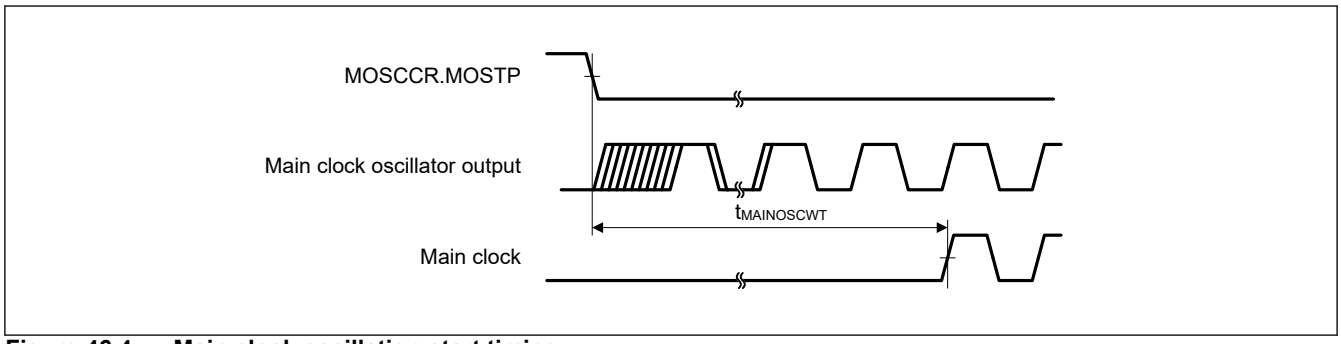


Figure 46.4 Main clock oscillation start timing

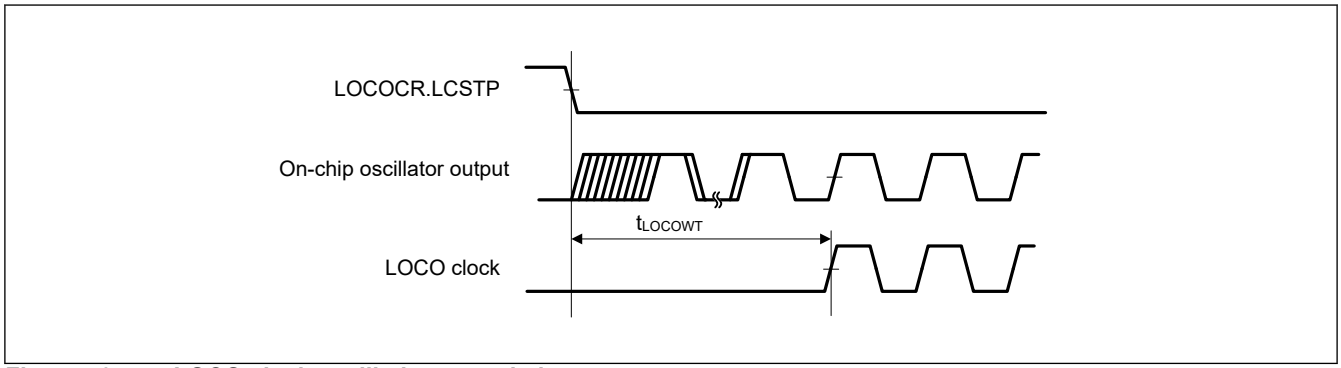


Figure 46.5 LOCO clock oscillation start timing

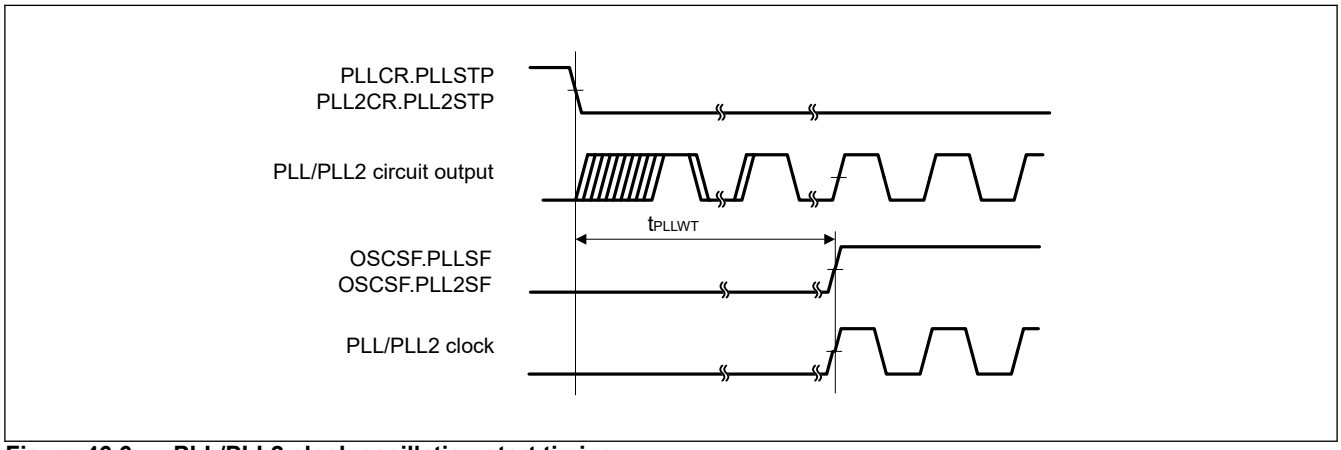


Figure 46.6 PLL/PLL2 clock oscillation start timing

46.3.3 Reset Timing

Table 46.17 Reset timing

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
RES pulse width	Power-on	t_{RESWP}	0.7	—	—	ms	Figure 46.7
	Deep Software Standby mode	t_{RESWD}	0.6	—	—	ms	Figure 46.8
	Software Standby mode	t_{RESWS}	0.3	—	—	ms	
	All other	t_{RESW}	200	—	—	µs	
Wait time after RES cancellation		t_{RESWT}	—	37.3	41.2	µs	Figure 46.7
Wait time after internal reset cancellation (IWDT reset, WDT reset, software reset, SRAM parity error reset, bus master MPU error reset, TrustZone error reset, Cache parity error reset)		t_{RESW2}	—	324	397.7	µs	—

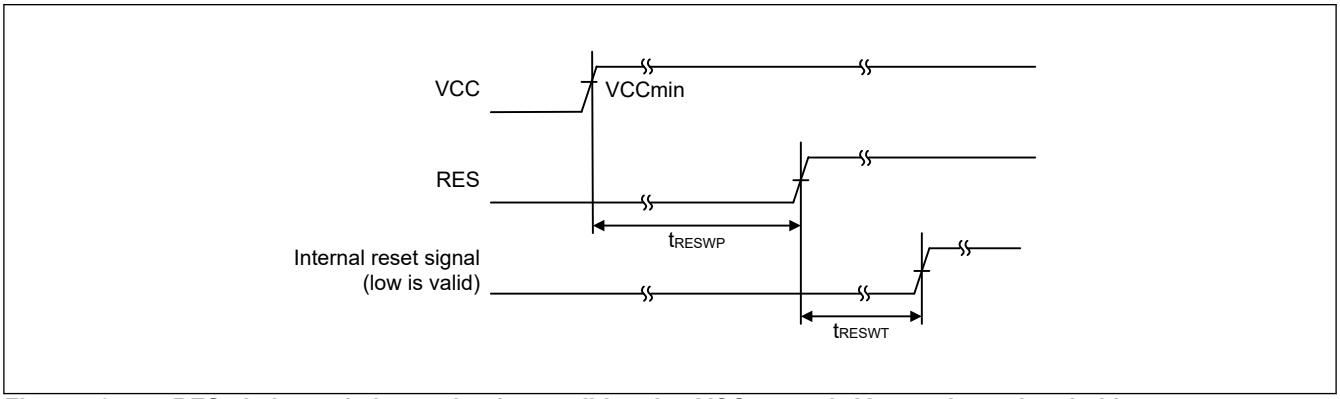


Figure 46.7 RES pin input timing under the condition that VCC exceeds V_{POR} voltage threshold

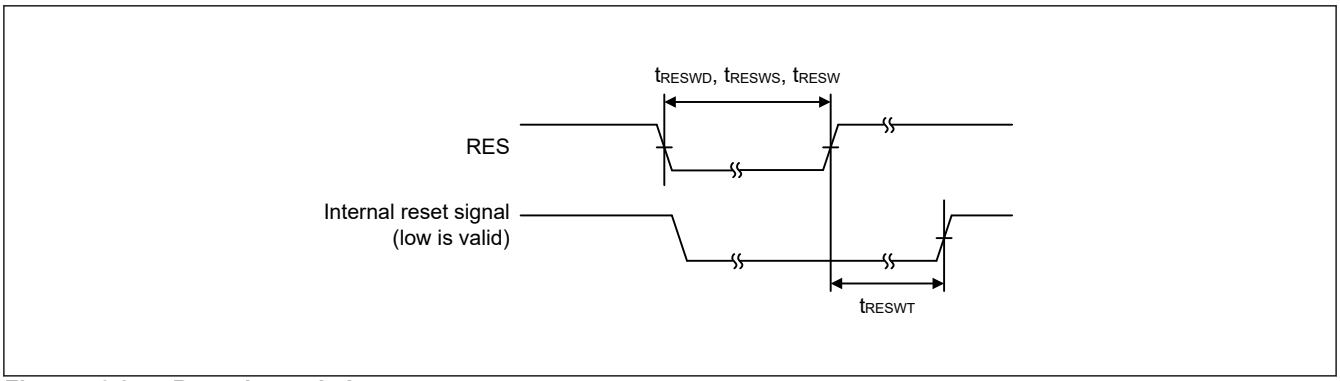


Figure 46.8 Reset input timing

46.3.4 Wakeup Timing

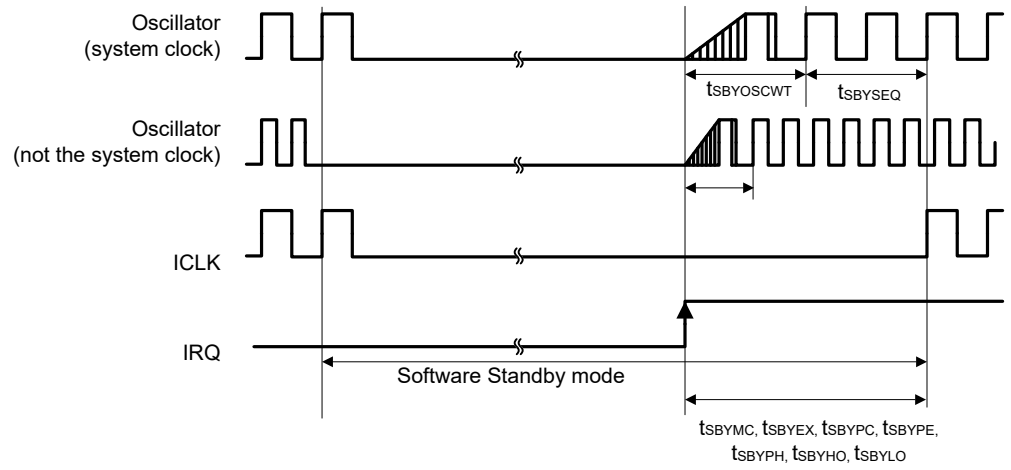
Table 46.18 Timing of recovery from low power modes

Parameter			Symbol	Min	Typ	Max	Unit	Test conditions
Recovery time from Software Standby mode*1	Crystal resonator connected to main clock oscillator	System clock source is main clock oscillator*2	t_{SBYMC}*11	—	2.1	2.4	ms	Figure 46.9 The division ratio of all oscillators is 1.
		System clock source is PLL with main clock oscillator*3	t_{SBYPC}*11	—	2.2	2.6	ms	
	External clock input to main clock oscillator	System clock source is main clock oscillator*4	t_{SBYEX}*11	—	45	125	µs	
		System clock source is PLL with main clock oscillator*5	t_{SBYPE}*11	—	170	255	µs	
	System clock source is LOCO*6		t_{SBYLO}*11	—	0.7	0.9	ms	
	System clock source is HOCO clock oscillator*7		t_{SBYHO}*11	—	55	130	µs	
	System clock source is PLL with HOCO*8		t_{SBYPH}*11	—	175	265	µs	
	System clock source is MOCO clock oscillator*9		t_{SBYMO}*11	—	35	65	µs	
Recovery time from Deep Software Standby mode	DPSBYCR.DEEPCUT[1] = 0 and DPSWCR.WTSTS[5:0] = 0x0E		t_{DSBY}	—	0.38	0.54	ms	Figure 46.10
	DPSBYCR.DEEPCUT[1] = 1 and DPSWCR.WTSTS[5:0] = 0x19		t_{DSBY}	—	0.55	0.73	ms	
Wait time after cancellation of Deep Software Standby mode			t_{DSBYWT}	56	—	57	t_{cyc}	
Recovery time from Software Standby mode to Snooze mode	High-speed mode when system clock source is HOCO (20 MHz)		t_{SNZ}	—	35*10	70*10	µs	Figure 46.11
	High-speed mode when system clock source is MOCO (8 MHz)		t_{SNZ}	—	11*10	14*10	µs	

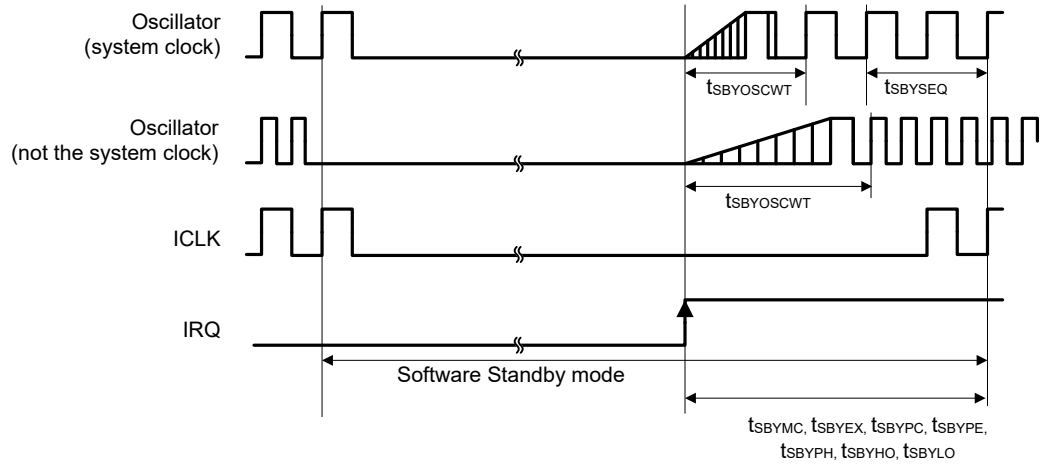
Note 1. The recovery time is determined by the system clock source. When multiple oscillators are active, the recovery time can be determined with the following equation:
Total recovery time = recovery time for an oscillator as the system clock source + the longest tSBYOSCWT in the active oscillators - tSBYOSCWT for the system clock + 2 LOCO cycles (when LOCO is operating)
Note 2. When the frequency of the crystal is 24 MHz (Main Clock Oscillator Wait Control Register (MOSCWTCR) is set to 0x05) and the greatest value of the internal clock division setting is 1.
Note 3. When the frequency of PLL is 240 MHz (Main Clock Oscillator Wait Control Register (MOSCWTCR) is set to 0x05) and the greatest value of the internal clock division setting is 4.
Note 4. When the frequency of the external clock is 24 MHz (Main Clock Oscillator Wait Control Register (MOSCWTCR) is set to 0x00) and the greatest value of the internal clock division setting is 1.
Note 5. When the frequency of PLL is 240 MHz (Main Clock Oscillator Wait Control Register (MOSCWTCR) is set to 0x00) and the greatest value of the internal clock division setting is 4.
Note 6. The LOCO frequency is 32.768 kHz and the greatest value of the internal clock division setting is 1.
Note 7. The HOCO frequency is 20 MHz and the greatest value of the internal clock division setting is 1.
Note 8. The PLL frequency is 240 MHz and the greatest value of the internal clock division setting is 4.
Note 9. The MOCO frequency is 8 MHz and the greatest value of the internal clock division setting is 1.
Note 10. When the SNZCR.RXDREQEN bit is set to 0, the following time is added as the power supply recovery time: 16 µs (typical), 48 µs (maximum).
Note 11. The recovery time can be calculated with the equation of tSBYOSCWT + tSBYSEQ. And they can be determined with the following value and equation. For n, the greatest value is selected from among the internal clock division settings.

Wakeup time	TYP		MAX		Unit
	tSBYOSCWT	tSBYSEQ	tSBYOSCWT	tSBYSEQ	
tSBYMC	(MSTS[7:0]*32 + 3) / 0.262	35 + 18 / fICLK + 4n / fMAIN	(MSTS[7:0]*32 + 14) / 0.236	62 + 18 / fICLK + 4n / fMAIN	µs
tSBYPC	(MSTS[7:0]*32 + 34) / 0.262	35 + 18 / fICLK + 4n / fPLL	(MSTS[7:0]*32 + 45) / 0.236	62 + 18 / fICLK + 4n / fPLL	µs

Wakeup time	TYP		MAX		Unit
	tSBYOSCWT	tSBYSEQ	tSBYOSCWT	tSBYSEQ	
tSBYEX	10	35 + 18 / fICLK + 4n / fEXMAIN	62	62 + 18 / fICLK + 4n / fEXMAIN	µs
tSBYPE	135	35 + 18 / fICLK + 4n / fPLL	192	62 + 18 / fICLK + 4n / fPLL	µs
tSBYLO	0	35 + 18 / fICLK + 4n / fLOCO	0	62 + 18 / fICLK + 4n / fLOCO	µs
tSBYHO	20	35 + 18 / fICLK + 4n / fHOCO	67	62 + 18 / fICLK + 4n / fHOCO	µs
tSBYPH	140	35 + 18 / fICLK + 4n / fPLL	202	62 + 18 / fICLK + 4n / fPLL	µs
tSBYMO	0	35 + 18 / fICLK + 4n / fMOCO	0	62 + 18 / fICLK + 4n / fMOCO	µs



When stabilization of the system clock oscillator is slower



When stabilization of an oscillator other than the system clock is slower

Figure 46.9 Software Standby mode cancellation timing

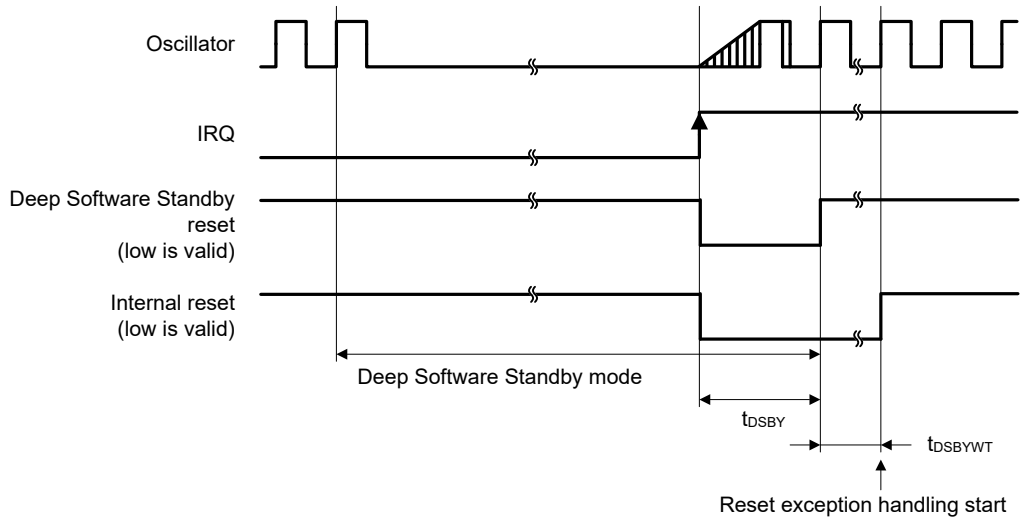
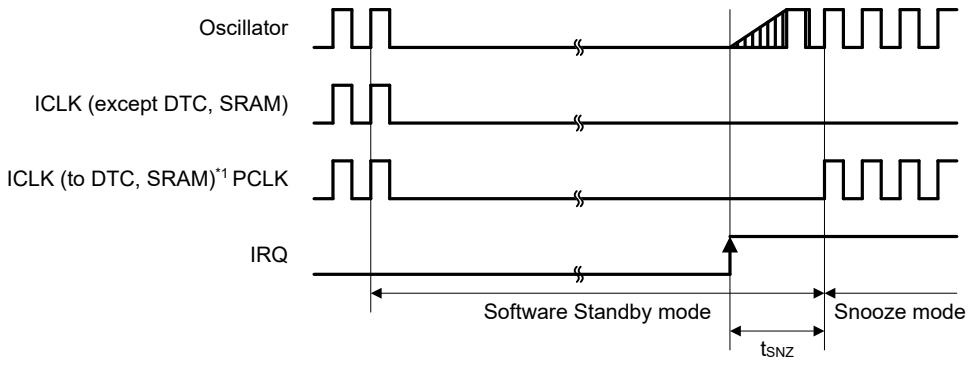


Figure 46.10 Deep Software Standby mode cancellation timing



Note 1. When SNZCR.SNZDTCEN bit is set to 1, ICLK is supplied to DTC and SRAM.

Figure 46.11 Recovery timing from Software Standby mode to Snooze mode

46.3.5 NMI and IRQ Noise Filter

Table 46.19 NMI and IRQ noise filter

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions	
NMI pulse width	t_{NMIW}	200	—	—	ns	NMI digital filter disabled	$t_{Pcyc} \times 2 \leq 200$ ns
		$t_{Pcyc} \times 2$*1	—	—			$t_{Pcyc} \times 2 > 200$ ns
		200	—	—		NMI digital filter enabled	$t_{NMICK} \times 3 \leq 200$ ns
		$t_{NMICK} \times 3.5$*2	—	—			$t_{NMICK} \times 3 > 200$ ns
IRQ pulse width	t_{IRQW}	200	—	—	ns	IRQ digital filter disabled	$t_{Pcyc} \times 2 \leq 200$ ns
		$t_{Pcyc} \times 2$*1	—	—			$t_{Pcyc} \times 2 > 200$ ns
		200	—	—		IRQ digital filter enabled	$t_{IRQCK} \times 3 \leq 200$ ns
		$t_{IRQCK} \times 3.5$*3	—	—			$t_{IRQCK} \times 3 > 200$ ns

Note: 200 ns minimum in Software Standby mode.
Note: If the clock source is switched, add 4 clock cycles of the switched source.
Note 1. t_{Pcyc} indicates the PCLKB cycle.
Note 2. t_{NMICK} indicates the cycle of the NMI digital filter sampling clock.
Note 3. t_{IRQCK} indicates the cycle of the IRQi digital filter sampling clock.

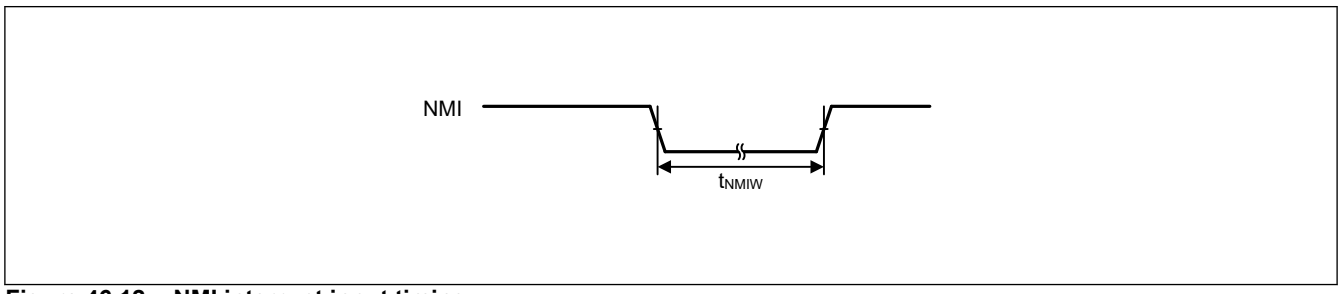


Figure 46.12 NMI interrupt input timing

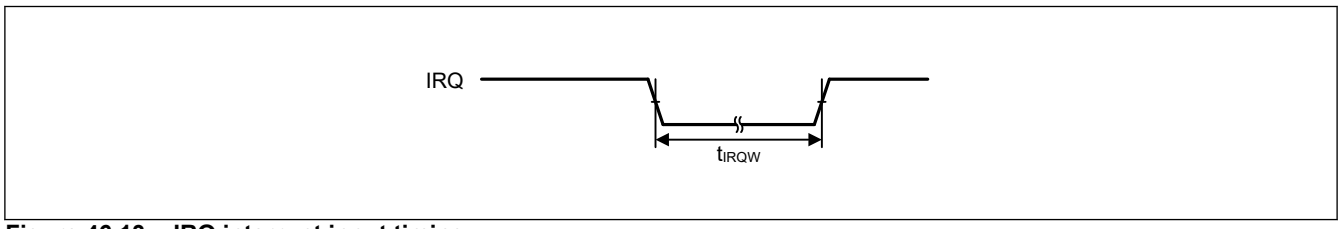


Figure 46.13 IRQ interrupt input timing

46.3.6 I/O Ports, POEG, GPT, AGT, KINT and ADC Trigger Timing

Table 46.20 I/O ports, POEG, GPT, AGT, KINT and ADC trigger timing (1 of 4)

GPT Conditions:
High drive output is selected in the Port Drive Capability bit in the PmnPFS register.
AGT Conditions:
Middle drive output is selected in the Port Drive Capability bit in the PmnPFS register.

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
I/O ports	Input data pulse width	t_{PRW}	1.5	—	—	t_{Icyc}	Figure 46.14

Table 46.20 I/O ports, POEG, GPT, AGT, KINT and ADC trigger timing (2 of 4)

GPT Conditions:
High drive output is selected in the Port Drive Capability bit in the PmnPFS register.
AGT Conditions:
Middle drive output is selected in the Port Drive Capability bit in the PmnPFS register.

Parameter			Symbol	Min	Typ	Max	Unit	Test conditions
POEG	GTETRGn input pulse width		t_{POEW}	1.5	—	—	t_{Pcyc}	Figure 46.15
	Output disable time	Input level detection of the GTETRGn pin (via flag)	t_{POEGDI}	—	—	3 PCLKB + 0.34	µs	Figure 46.16 When the digital noise filter is not in use (POEGGn.NFEN = 0 (n = A to D))
		Detection of the output stopping signal from GPT (deadtime error, simultaneous high output, or simultaneous low output)	t_{POEGDE}	—	—	0.5	µs	Figure 46.17
		Edge detection signal from a comparator	t_{POEGDC}	—	—	4 PCLKB + 0.5	µs	Figure 46.18 The time is that when the noise filter for ACMPHS is not in use (CMPCTL.CDFS[1:0] = 00) and excludes the time for detection by ACMPHS.
		Register setting	t_{POEGDS}	—	—	1 PCLKB + 0.3	µs	Figure 46.19 Time for access to the register is not included.
		Oscillation stop detection*3	$t_{POEGDOS}$	—	≤ 1	—	µs	Figure 46.20
		Input level detection of the GTETRGn pin (direct path)	$t_{POEGDDI}$	—	—	2 PCLKB + 1 PCLKD + 0.34	µs	Figure 46.21
		Level detection signal from a comparator	$t_{POEGDDC}$	—	—	3 PCLKD + 0.3	µs	Figure 46.22 The time is that when the noise filter for ACMPHS is not in use (CMPCTL.CDFS[1:0] = 00) and excludes the time for detection by ACMPHS.

Table 46.20 I/O ports, POEG, GPT, AGT, KINT and ADC trigger timing (3 of 4)

GPT Conditions:
High drive output is selected in the Port Drive Capability bit in the PmnPFS register.
AGT Conditions:
Middle drive output is selected in the Port Drive Capability bit in the PmnPFS register.

Parameter				Symbol	Min	Typ	Max	Unit	Test conditions
GPT	Input capture pulse width	Single edge		t_{GTICW}	1.5	—	—	t_{PDcyc}	Figure 46.23
		Dual edge			2.5	—	—		
	GTIOCxY output skew (x = 0 to 3, Y = A or B)	Middle drive buffer		t_{GTISK}*1	—	—	4	ns	Figure 46.24
		High drive buffer			—	—	4		
		High current output buffer			—	—	4		
	GTIOCxY output skew (x = 4 to 6, Y = A or B)	Middle drive buffer			—	—	4		
		High drive buffer			—	—	4		
		High current output buffer			—	—	4		
	GTIOCxY output skew (x = 7 to 9, Y = A or B)	Middle drive buffer			—	—	4		
		High drive buffer			—	—	4		
		High current output buffer			—	—	4		
	GTIOCxY output skew (x = 0 to 9, Y = A or B)	Middle drive buffer			—	—	6		
		High drive buffer			—	—	6		
		High current output buffer			—	—	6		
	OPS output skew GTOUUP, GTOULO, GTOVUP, GTOVLO, GTOWUP, GTOWLO			t_{GTOSK}	—	—	5	ns	Figure 46.25
	External trigger input pulse width	Synchronous clock	Single-edge setting	t_{GTEW}	1.5	—	—	t_{Pcyc}	Figure 46.26
			Both-edge setting		2.5	—	—		
		Asynchronous clock	Single-edge setting		2.5	—	—		
			Both-edge setting		3.5	—	—		
	Timer clock pulse width	Synchronous clock	Single-edge setting	t_{GTCKWH}, t_{GTCKWL}	1.5	—	—	t_{Pcyc}	Figure 46.27
			Both-edge setting		2.5	—	—		
		Asynchronous clock	Single-edge setting		2.5	—	—		
			Both-edge setting		3.5	—	—		
GPT (PWM Delay Generation Circuit)	GTIOCxY_Z skew (x = 0 to 3, Y = A or B, Z = A to D)			t_{HRSK}*2	—	—	4.0	ns	Figure 46.28
AGT	AGTIO, AGTEE input cycle			t_{ACYC}*2	50	—	—	ns	Figure 46.29
	AGTIO, AGTEE input high width, low width			t_{ACKWH}, t_{ACKWL}	20	—	—	ns	
	AGTIO, AGTO, AGTOA, AGTOB output cycle			t_{ACYC2}	33.3	—	—	ns	
KINT	KRn (n = 00 to 07) pulse width			t_{KR}	250	—	—	ns	Figure 46.30

Table 46.20 I/O ports, POEG, GPT, AGT, KINT and ADC trigger timing (4 of 4)

GPT Conditions:
High drive output is selected in the Port Drive Capability bit in the PmnPFS register.
AGT Conditions:
Middle drive output is selected in the Port Drive Capability bit in the PmnPFS register.

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
ADC	ADC trigger input pulse width	t_{TRGW}	1.5	—	—	t_{ADcyc}	Figure 46.31

Note: t_{Icyc}: ICLK cycle, t_{Pcyc}: PCLKB cycle, t_{PDcyc}: GTCLK cycle, t_{ADcyc}: ADCLK cycle.
Note 1. This skew applies when the same driver I/O is used. If the I/O of the middle and high drivers is mixed, operation is not guaranteed.
Note 2. Constraints on input cycle:
When not switching the source clock: $t_{Pcyc} \times 2 < t_{ACYC}$ should be satisfied.
When switching the source clock: $t_{Pcyc} \times 6 < t_{ACYC}$ should be satisfied.
Note 3. Reference value.

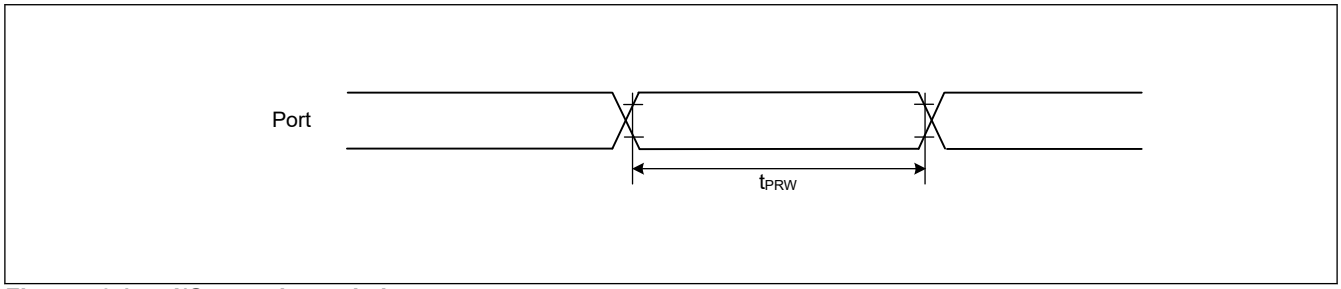


Figure 46.14 I/O ports input timing

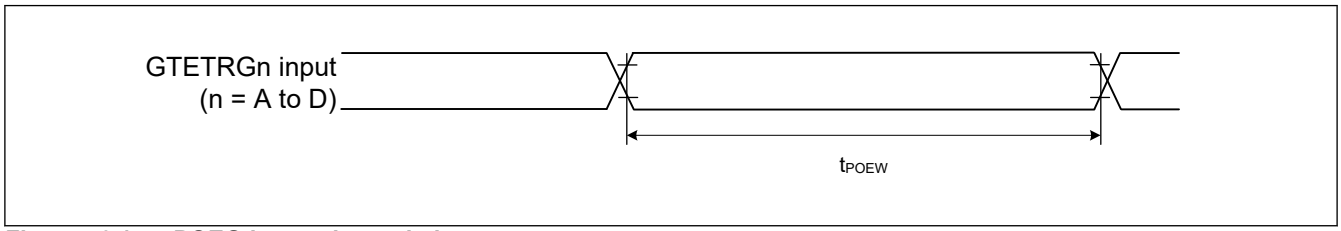


Figure 46.15 POEG input trigger timing

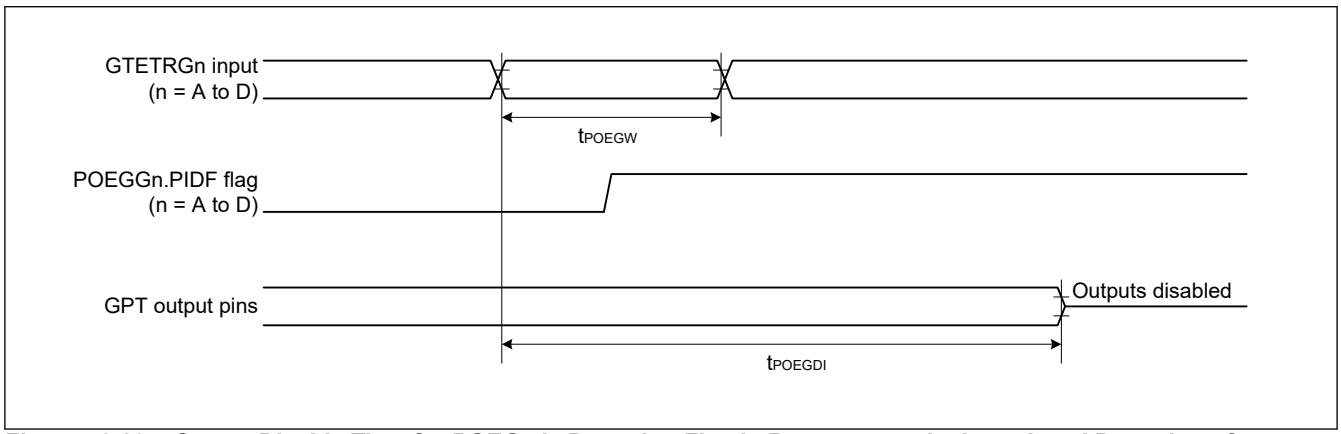
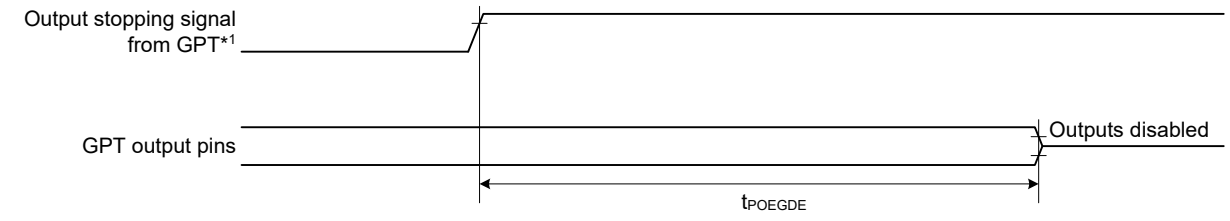


Figure 46.16 Output Disable Time for POEG via Detection Flag in Response to the Input Level Detection of the GTETRGn pin



Note 1. GPT32n.GTST.DTEF (dead time error flag), GPT32n.GTST.OABLF (simultaneous low output flag), or GPT32n.GTST.OABHF (simultaneous high output flag)

Figure 46.17 Output Disable Time for POEG in Response to Detection of the Output Stopping Signal from GPT

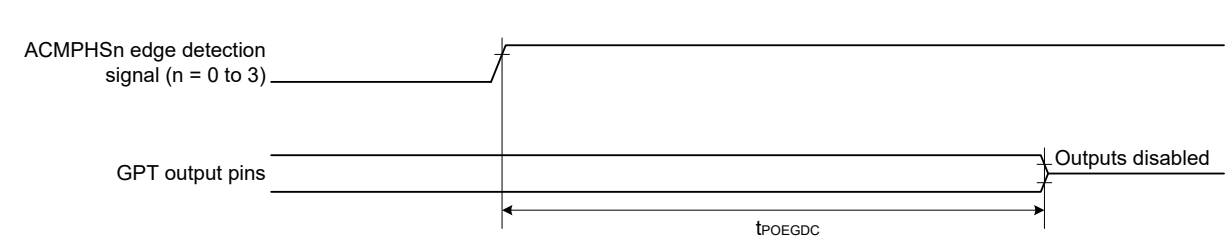


Figure 46.18 Output Disable Time for POEG in Response to Edge Detection Signal from ACMPHS

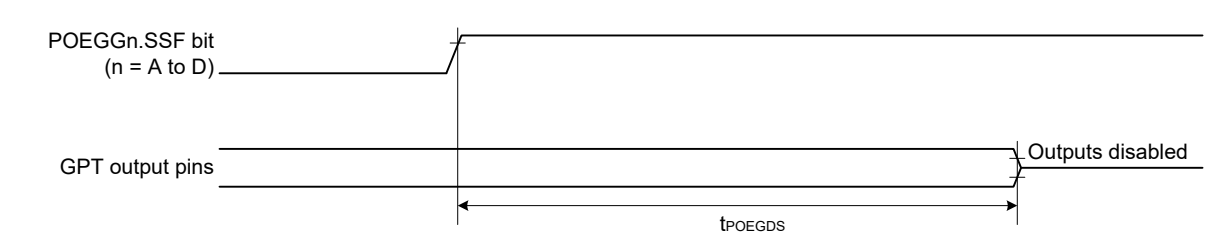


Figure 46.19 Output Disable Time for POEG in Response to the Register Setting

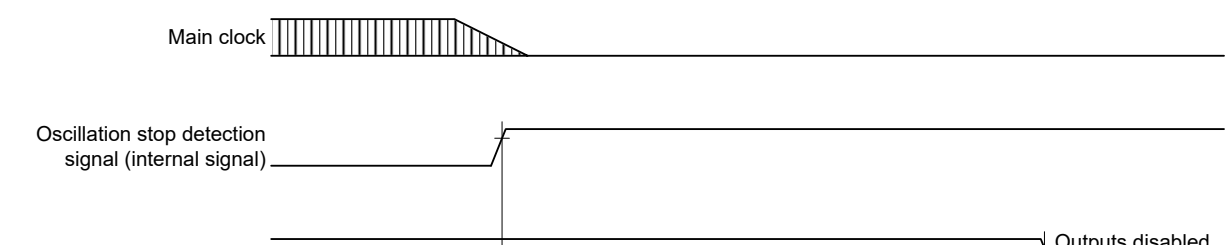


Figure 46.20 Output Disable Time of POEG in Response to the Oscillation Stop Detection

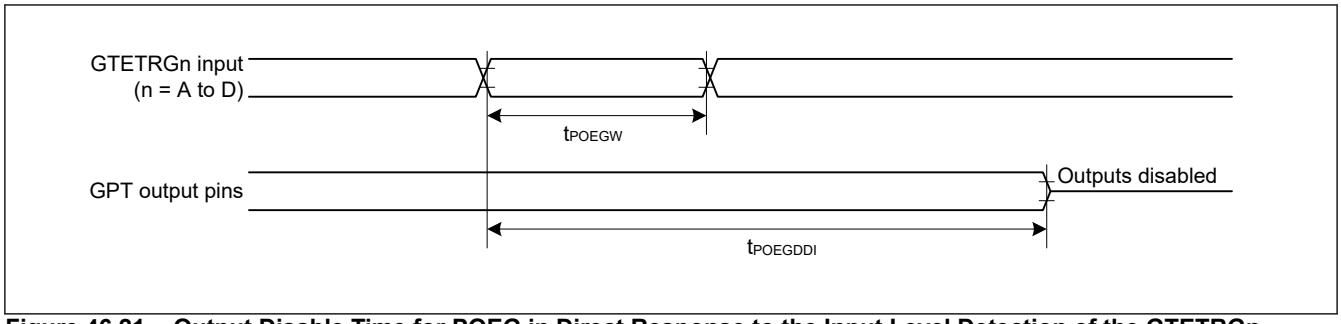


Figure 46.21 Output Disable Time for POEG in Direct Response to the Input Level Detection of the GTETRGn pin

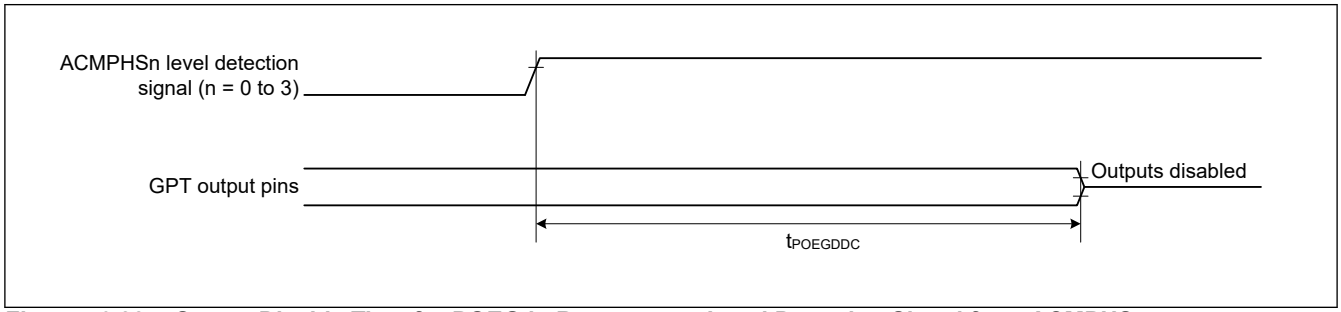


Figure 46.22 Output Disable Time for POEG in Response to Level Detection Signal from ACMPHS

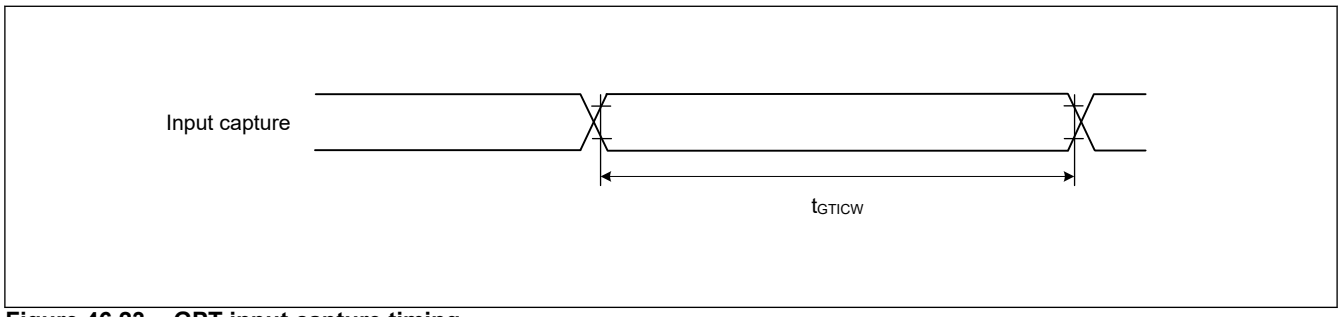


Figure 46.23 GPT input capture timing

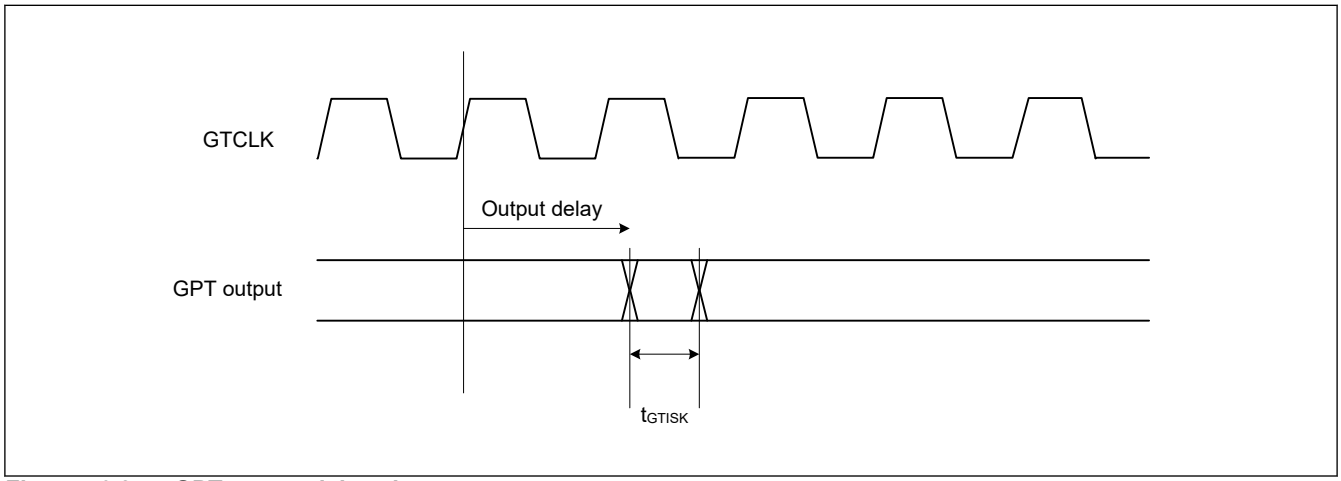


Figure 46.24 GPT output delay skew

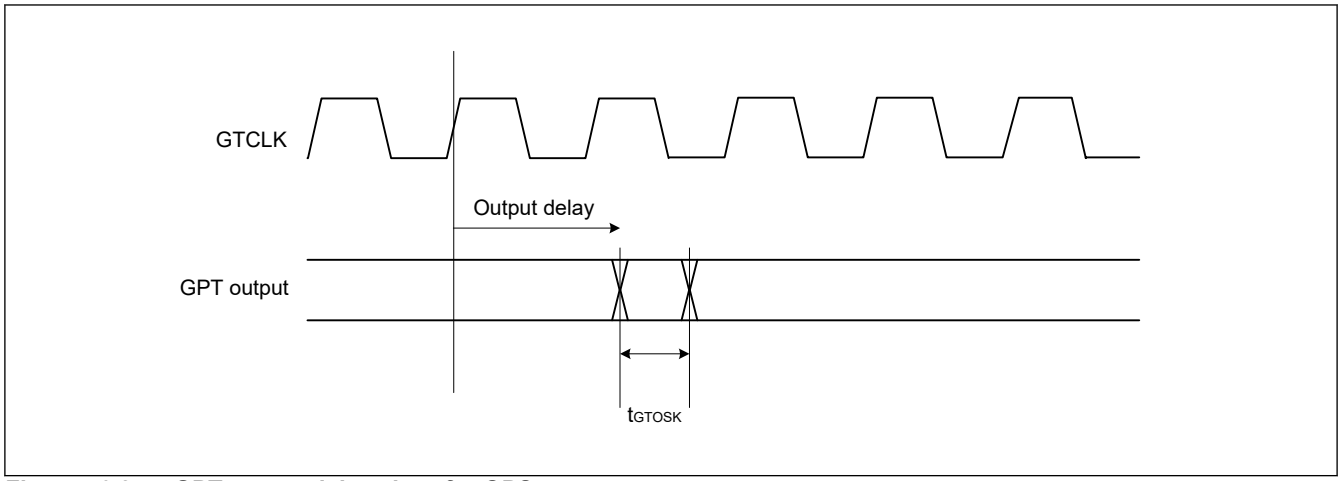


Figure 46.25 GPT output delay skew for OPS

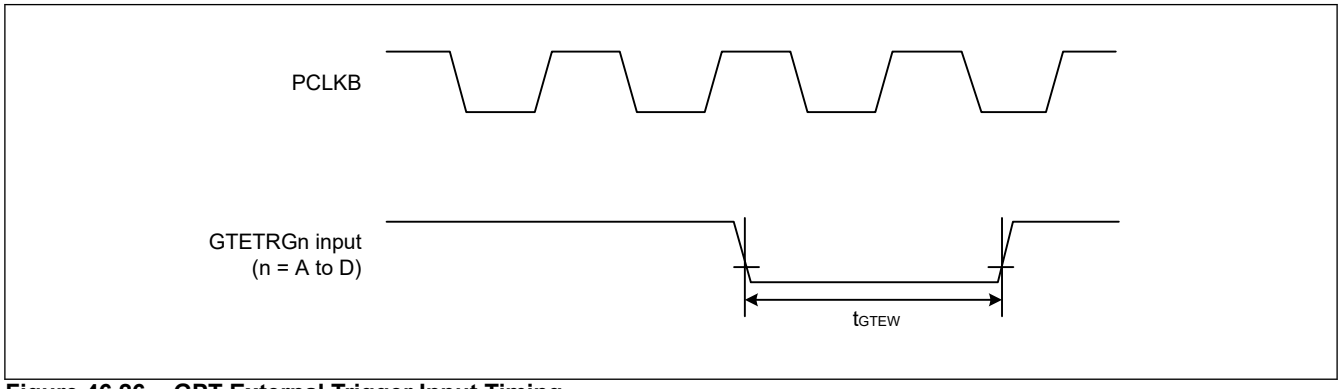


Figure 46.26 GPT External Trigger Input Timing

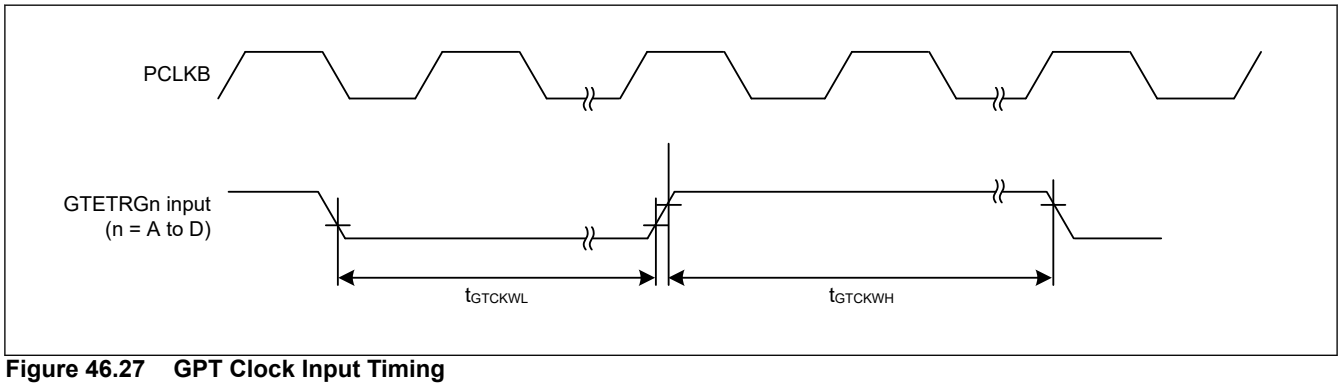


Figure 46.27 GPT Clock Input Timing

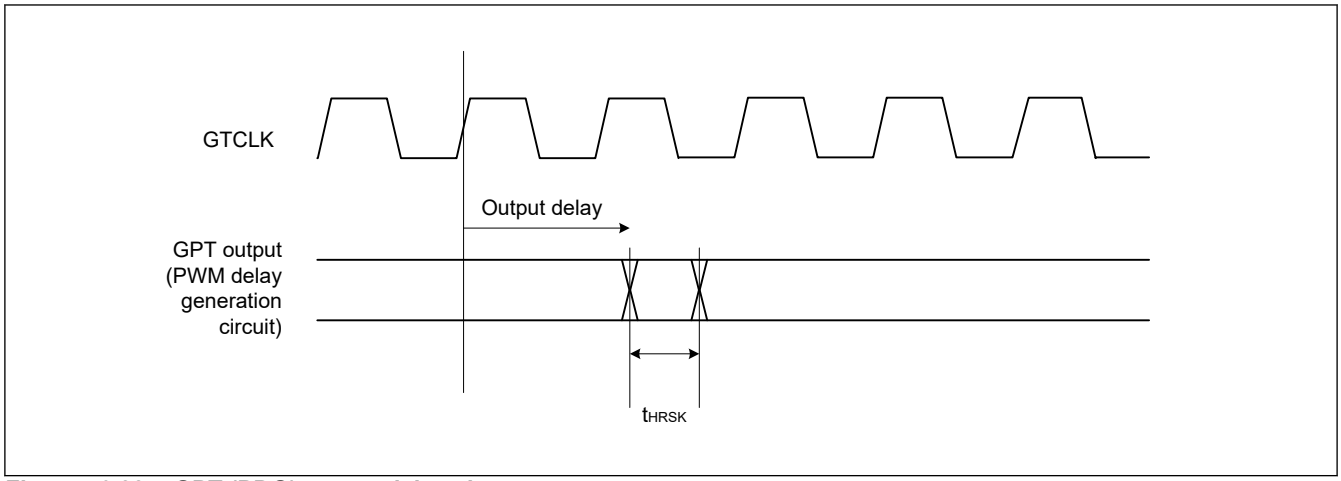


Figure 46.28 GPT (PDG) output delay skew

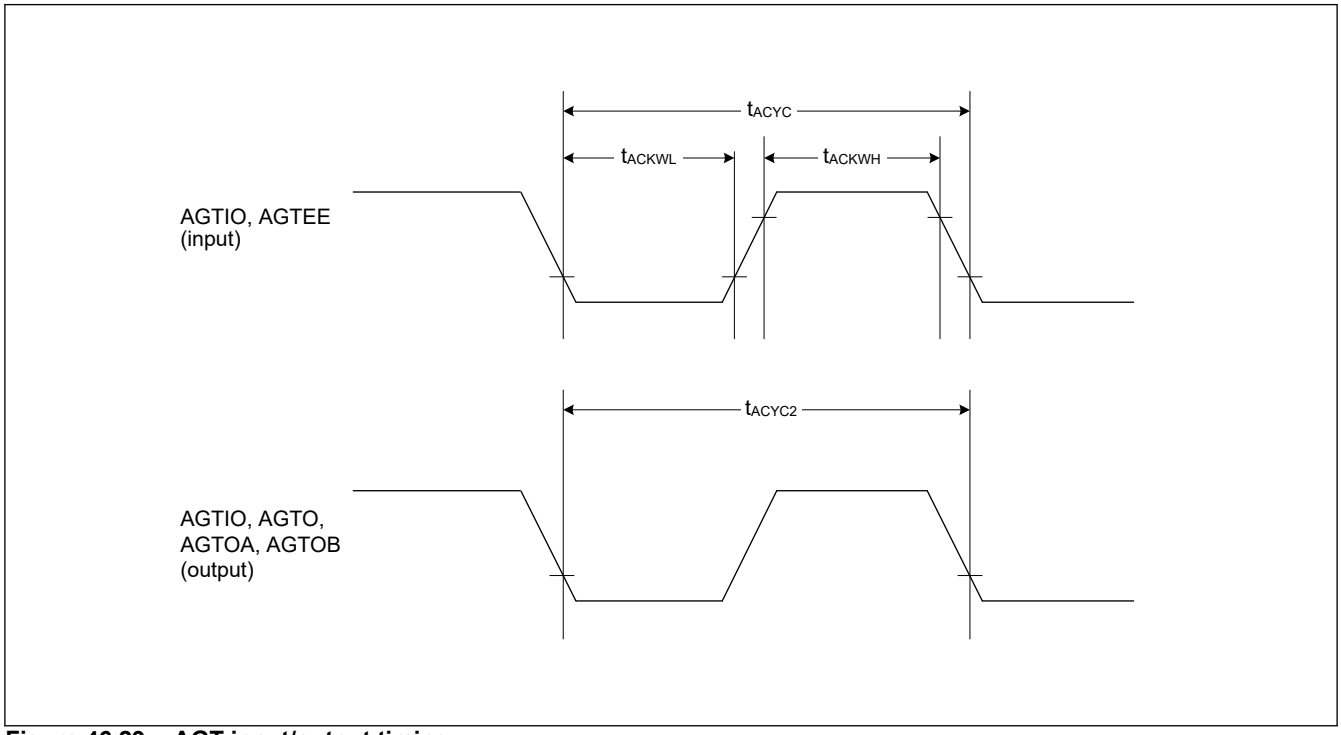


Figure 46.29 AGT input/output timing

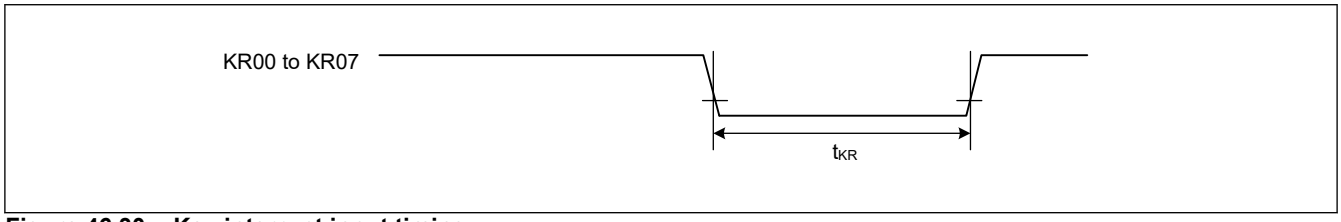


Figure 46.30 Key interrupt input timing

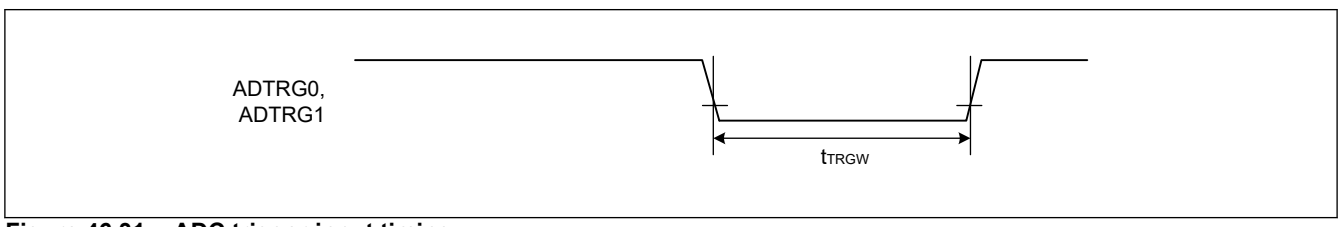


Figure 46.31 ADC trigger input timing

46.3.7 PDG Timing

Table 46.21 PDG timing

Parameter	Min	Typ	Max	Unit	Test conditions
Operation frequency	80	—	200	MHz	—
Resolution	—	156	—	ps	GPTCLK = 200 MHz
DNL*1	—	±2.0	—	LSB	—

Note 1. This value normalizes the differences between lines in 1-LSB resolution.

46.3.8 CAC Timing

Table 46.22 CAC timing

Parameter			Symbol	Min	Typ	Max	Unit	Test conditions
CAC	CACREF input pulse width	$t_{PBcyc} \leq t_{cac}$*1	t_{CACREF}	$4.5 \times t_{cac} + 3 \times t_{PBcyc}$	—	—	ns	—
		$t_{PBcyc} > t_{cac}$*1		$5 \times t_{cac} + 6.5 \times t_{PBcyc}$	—	—	ns	

Note: t_{PBcyc}: PCLKB cycle.
Note 1. t_{cac}: CAC count clock source cycle.

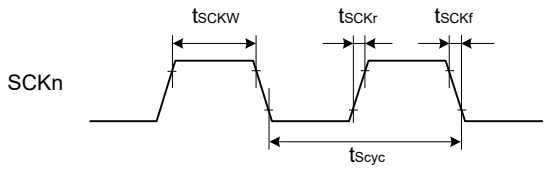
46.3.9 SCI Timing

Table 46.23 SCI timing (Asynchronous mode)

Conditions: High drive output is selected in the Port Drive Capability bit in the PmnPFS register.

Parameter	Symbol	Min	Max	Unit	Note
Input clock cycle	t_{Scyc}	4	—	t_{Tcyc}	
Input clock pulse width	t_{SCKW}	0.4	0.6	t_{Scyc}	
Input clock rise time	t_{SCKr}	—	5	ns	
Input clock fall time	t_{SCKf}	—	5	ns	
Output clock cycle	t_{Scyc}	6	—	t_{Tcyc}	
Output clock pulse width	t_{SCKW}	0.4	0.6	t_{Scyc}	
Output clock rise time	t_{SCKr}	—	5	ns	
Output clock fall time	t_{SCKf}	—	5	ns	

Note: t_{Tcyc}: SCITCLK cycle.



Note: n = 0 to 4, 9

Figure 46.32 SCK clock input/output timing

Table 46.24 SCI timing (Simple SPI) (1 of 2)

Conditions: High drive output is selected in the Port Drive Capability bit in the PmnPFS register.

Parameter		High Speed/ Default	Symbol	Min	Max	Unit	Note
SCK clock cycle output	Master		t_{SPcyc}	2	65536	t_{Tcyc}	
SCK clock cycle input	Slave			2	—		

Table 46.24 SCI timing (Simple SPI) (2 of 2)

Conditions: High drive output is selected in the Port Drive Capability bit in the PmnPFS register.

Parameter		High Speed/ Default	Symbol	Min	Max	Unit	Note
SCK clock high pulse width	Master		t_{SPCKWH}	0.4	0.6	t_{SPcyc}	
	Slave						
SCK clock low pulse width	Master		t_{SPCKWL}	0.4	0.6	t_{SPcyc}	
	Slave						
SCK clock rise and fall time	Output		t_{SPCKr}, t_{SPCKf}	—	5	ns	
	Input			—	1	us	
Data input setup time	Master	High Speed*1	t_{SU}	1.7	—	ns	
		Default*2		3	—	ns	
	Slave			3.3	—	ns	
Data input hold time	Master	High Speed*1	t_H	12	—	ns	
		Default*2		14	—	ns	
	Slave			3	—	ns	
Data output delay	Master	High Speed*1	t_{OD}	—	5	ns	
		Default*2		—	7.3	ns	
	Slave	High Speed*1		—	15	ns	
		Default*2		—	21	ns	
Data output hold time	Master		t_{OH}	0	—	ns	
	Slave			0	—	ns	
Data rise and fall time	Output		t_{Dr}, t_{Df}	—	5	ns	
	Input			—	1	ns	
Slave access time			t_{SA}	—	5	t_{Tcyc}	
Slave output release time			t_{REL}	—	5	t_{Tcyc}	

Note: t_{Tcyc}: SCITCLK cycle.

Note 1. Must use pins that have a letter appended to their name, for instance _A, _B, _C, to indicate group membership. SCI0 is instance _A, SCI2 and SCI3 are instance _B, SCI1 and SCI9 are instance _C, SCI4 is instance _C and RXD is only PD14.

Note 2. All pins of group membership can be used.

Table 46.25 SCI timing (Simple SPI mode)

Conditions: High drive output is selected in the Port Drive Capability bit in the PmnPFS register.

Parameter	Symbol	Min	Max	Unit	Note
SS input setup time	t_{LEAD}	1	—	t_{SPcyc}	
SS input hold time	t_{LAG}	1	—	t_{SPcyc}	
SS input rise and fall time	t_{SSLr}, t_{SSLf}	—	1	us	

Table 46.26 SCI timing (Clock synchronous mode) (1 of 2)

Conditions: High drive output is selected in the Port Drive Capability bit in the PmnPFS register.

Parameter		High Speed/ Default	Symbol	Min	Max	Unit	Note
SCK clock cycle output	Master		t_{SPcyc}	2	—	t_{Tcyc}	
SCK clock cycle input	Slave			2	—		
SCK clock high pulse width	Master		t_{SPCKWH}	0.4	0.6	t_{SPcyc}	
	Slave						

Table 46.26 SCI timing (Clock synchronous mode) (2 of 2)

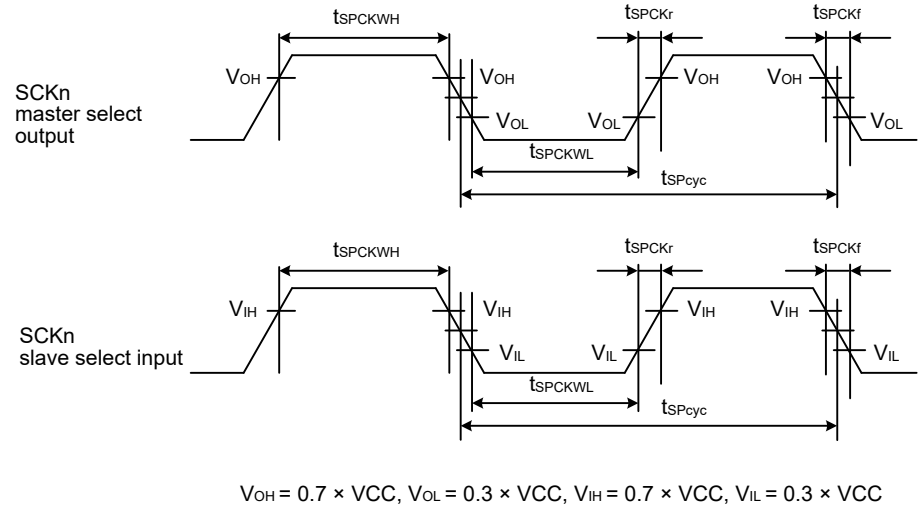
Conditions: High drive output is selected in the Port Drive Capability bit in the PmnPFS register.

Parameter		High Speed/ Default	Symbol	Min	Max	Unit	Note
SCK clock low pulse width	Master		t_{SPCKWL}	0.4	0.6	t_{SPcyc}	
	Slave						
SCK clock rise and fall time	Output		t_{SPCKr}, t_{SPCKf}	—	5	ns	
	Input						
Data input setup time	Master	High Speed*1	t_{SU}	2.6	—	ns	
		Default*2		2.8	—	ns	
	Slave			3.3	—	ns	
Data input hold time	Master	High Speed*1	t_H	12	—	ns	
		Default*2		14	—	ns	
	Slave			3	—	ns	
Data output delay	Master	High Speed*1	t_{OD}	—	5	ns	
		Default*2		—	7.3	ns	
	Slave	High Speed*1		—	15	ns	
		Default*2		—	21	ns	
Data output hold time	Master		t_{OH}	0	—	ns	
	Slave			0	—	ns	
Data rise and fall time	Output		t_{Dr}, t_{Df}	—	5	ns	
	Input			—	5	ns	

Note: t_{Tcyc}: SCITCLK cycle.

Note 1. Must use pins that have a letter appended to their name, for instance _A, _B, _C, to indicate group membership. SCI0 is instance _A, SCI2 and SCI3 are instance _B, SCI1 and SCI9 are instance _C, SCI4 is instance _C and RXD is only PD14.

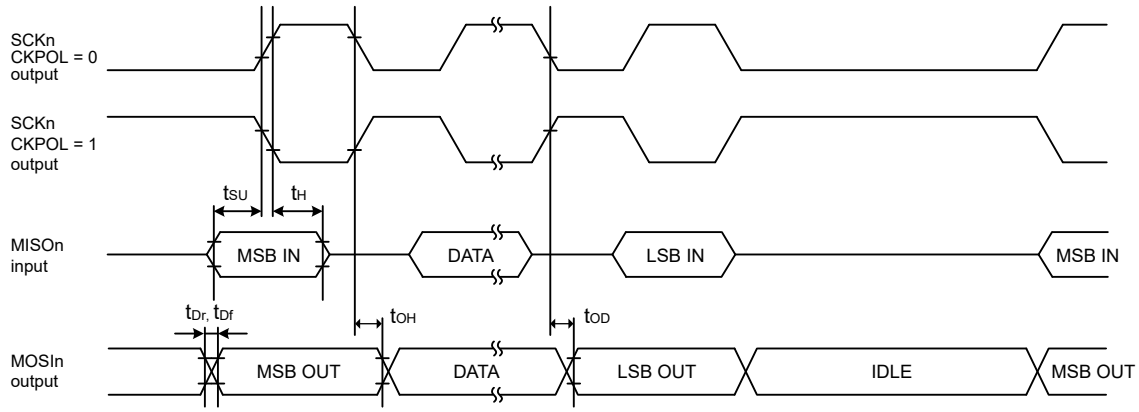
Note 2. All pins of group membership can be used.



V_{OH} = 0.7 × VCC, V_{OL} = 0.3 × VCC, V_{IH} = 0.7 × VCC, V_{IL} = 0.3 × VCC

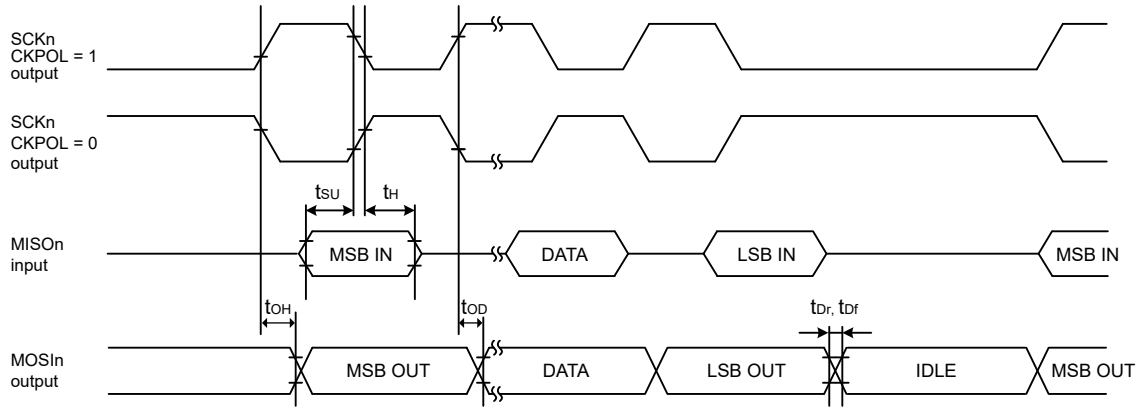
Note: n = 0 to 4, 9

Figure 46.33 SCI simple SPI mode clock timing



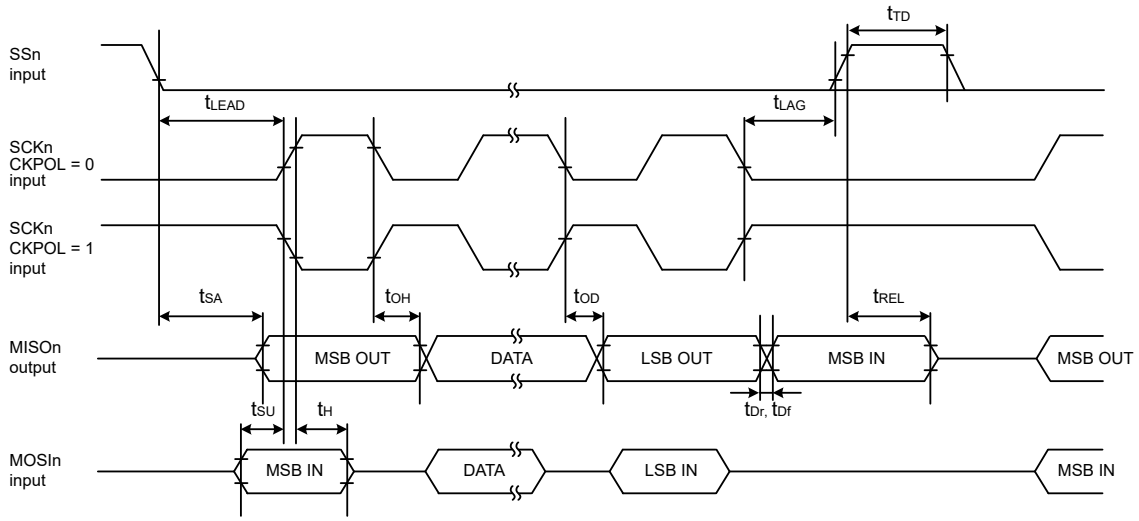
Note: n = 0 to 4, 9

Figure 46.34 SCI simple SPI mode timing for master when CKPH = 1



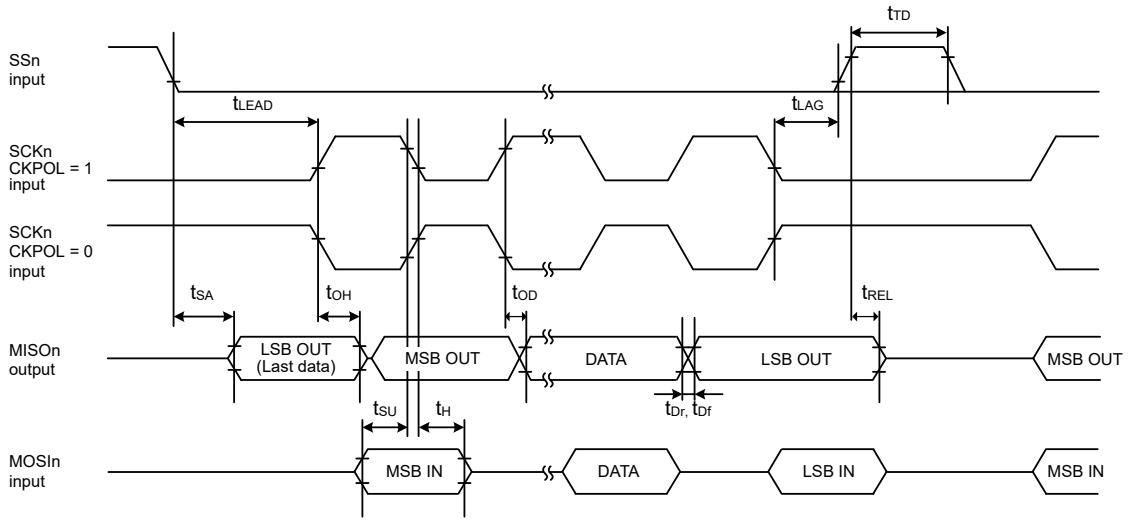
Note: n = 0 to 4, 9

Figure 46.35 SCI simple SPI mode timing for master when CKPH = 0



Note: n = 0 to 4, 9

Figure 46.36 SCI simple SPI mode timing for slave when CKPH = 1



Note: n = 0 to 4, 9

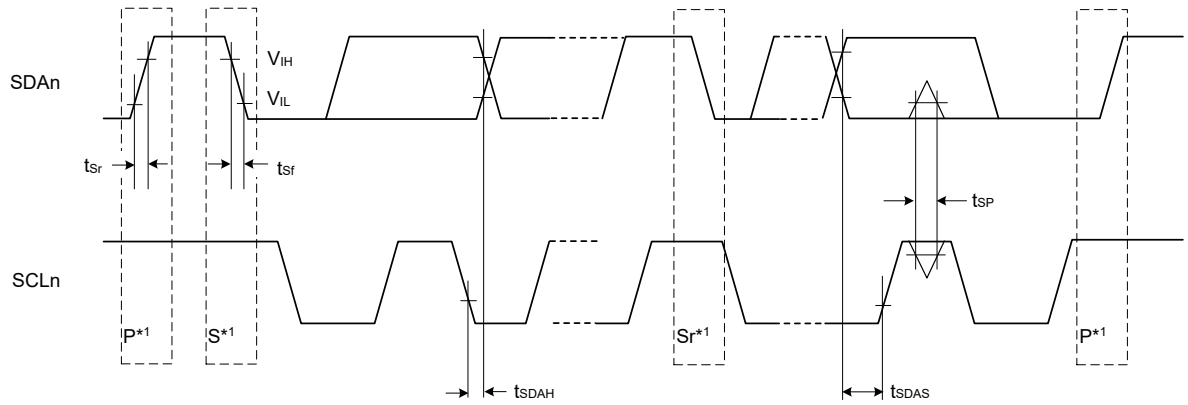
Figure 46.37 SCI simple SPI mode timing for slave when CKPH = 0

Table 46.27 SCI timing (Simple IIC mode)

Conditions: Middle drive output is selected in the Port Drive Capability bit in the PmnPFS register.

Parameter		Symbol	Min	Max	Unit	Note
Simple IIC (Standard mode)	SCL, SDA input rise time	t_{Sr}	—	1000	ns	
	SCL, SDA input fall time	t_{Sf}	—	300	ns	
	SCL, SDA input spike pulse removal time	t_{SP}	0	4 × t_{Tcyc}	ns	
	Data input setup time	t_{SDAS}	250	—	ns	
	Data input hold time	t_{SDAH}	0	—	ns	
	SCL, SDA capacitive load	C_b*1	—	400	pF	
Simple IIC (Fast mode)	SCL, SDA input rise time	t_{Sr}	—	300	ns	
	SCL, SDA input fall time	t_{Sf}	—	300	ns	
	SCL, SDA input spike pulse removal time	t_{SP}	0	4 × t_{Tcyc}	ns	
	Data input setup time	t_{SDAS}	100	—	ns	
	Data input hold time	t_{SDAH}	0	—	ns	
	SCL, SDA capacitive load	C_b*1	—	400	pF	

Note: t_{Tcyc}: SCITCLK cycle.
Note 1. Cb indicates the total capacity of the bus line.



Test conditions:
$V_{IH} = V_{CC} \times 0.7$, $V_{IL} = V_{CC} \times 0.3$
$V_{OL} = 0.6$ V, $I_{OL} = 6$ mA (ICFER.FMPE = 0)
$V_{OL} = 0.4$ V, $I_{OL} = 20$ mA (ICFER.FMPE = 1)

Note: n = 0 to 4, 9
Note 1. S, P, and Sr indicate the following conditions:
S: Start condition
P: Stop condition
Sr: Restart condition

Figure 46.38 SCI simple IIC mode timing

46.3.10 SPI Timing

Table 46.28 SPI timing (1 of 2)

Conditions: High drive output is selected in the Port Drive Capability bit in the PmnPFS register.

Parameter		High Speed/ Default	Symbol	VCC = 3.0 to 3.6 V, C = 15 pF		VCC = 2.7 to 3.6 V, C = 30 pF		Unit	Note
				Min	Max	Min	Max		
RSPCK clock cycle	Master		t_{SPcyc}	2	4096	2	4096	t_{Tcyc}	
	Slave			2	—	2	—		
RSPCK clock high pulse width	Master		t_{SPCKWH}	$(t_{SPcyc} - t_{SPCKr} - t_{SPCKf})/2 - 3$	—	$(t_{SPcyc} - t_{SPCKr} - t_{SPCKf})/2 - 3$	—	ns	
	Slave			0.4	0.6	0.4	0.6	t_{SPcyc}	
RSPCK clock low pulse width	Master		t_{SPCKWL}	$(t_{SPcyc} - t_{SPCKr} - t_{SPCKf})/2 - 3$	—	$(t_{SPcyc} - t_{SPCKr} - t_{SPCKf})/2 - 3$	—	ns	
	Slave			0.4	0.6	0.4	0.6	t_{SPcyc}	
RSPCK clock rise and fall time	Output		t_{SPCKr}, t_{SPCKf}	—	5	—	5	ns	
	Input			—	1	—	1	µs	
Data input setup time	Master	High Speed*1	t_{SU}	0	—	—	—	ns	
		Default*2		—	—	1.3	—	ns	
	Slave			2.5	—	2.7	—	ns	
Data input hold time	Master	High Speed*1	t_H	6.2	—	—	—	ns	
		Default*2		—	—	8	—	ns	
	Slave			2.5	—	2.5	—	ns	
SSL setup time	Master		t_{LEAD}	1	8	1	8	t_{SPcyc}	
	Slave			6	—	6	—	t_{Tcyc}	
SSL hold time	Master		t_{LAG}	1	8	1	8	t_{SPcyc}	
	Slave			6	—	6	—	t_{Tcyc}	
TI SSP SS input setup time	Slave		t_{TISS}	2.5	—	2.8	—	ns	
TI SSP SS input hold time	Slave		t_{TISH}	2.5	—	2.5	—	ns	
TI SSP next-access time	Slave		t_{TIND}	$2 \times t_{Tcyc}$ + SLNDL × t_{Tcyc}	—	$2 \times t_{Tcyc}$ + SLNDL × t_{Tcyc}	—	ns	
TI SSP master SS output delay	Master		t_{TISSOD}	—	8.9	—	8.9	ns	

Table 46.28 SPI timing (2 of 2)

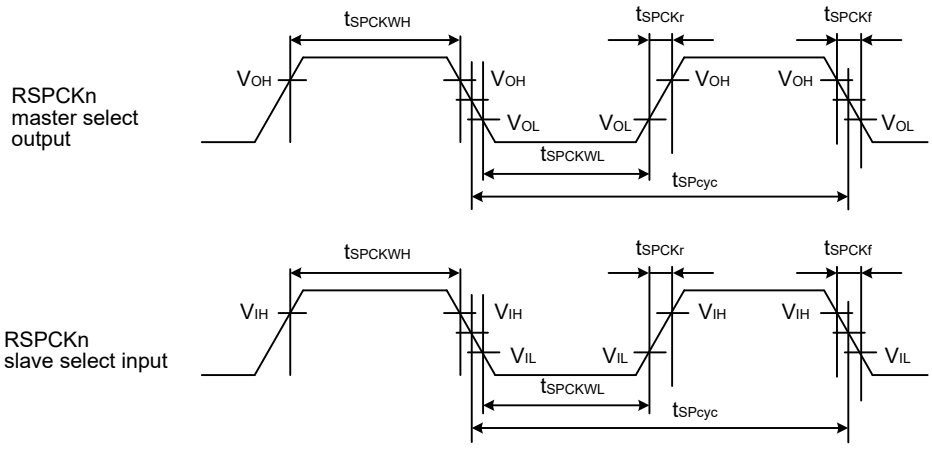
Conditions: High drive output is selected in the Port Drive Capability bit in the PmnPFS register.

Parameter		High Speed/ Default	Symbol	VCC = 3.0 to 3.6 V, C = 15 pF		VCC = 2.7 to 3.6 V, C = 30 pF		Unit	Note
				Min	Max	Min	Max		
Data output delay time	Master	High Speed*1	t_{OD}	—	4.6	—	—	ns	
		Default*2		—	—	—	7	ns	
	Slave	High Speed*1		—	14	—	—	ns	
		Default*2		—	—	—	21	ns	
Data output hold time	Master		t_{OH}	0	—	0	—	ns	
	Slave			0	—	0	—	ns	
Successive transmission delay time	Master		t_{TD}	t_{SPcyc} + 2 × t_{Tcyc}	8 × t_{SPcyc} + 2 × t_{Pcyc}	t_{SPcyc} + 2 × t_{Tcyc}	8 × t_{SPcyc} + 2 × t_{Pcyc}	ns	
	Slave			t_{Tcyc}	—	t_{Tcyc}	—	ns	
MOSI and MISO rise and fall time	Output		t_{Dr}	—	5	—	5	ns	
	Input		t_{Df}	—	1	—	1	µs	
SSL rise and fall time	Output		t_{SSLr}	—	5	—	5	ns	
	Input		t_{SSLf}	—	1	—	1	µs	
Slave access time	Slave		t_{SA}	—	20	—	20	ns	
Slave output release time	Slave		t_{REL}	—	20	—	20	ns	

Note: t_{Tcyc}: PCLKA or SCISPICLK cycle.

Note 1. Must use pins that have a letter appended to their name, for instance _A, _B, _C, to indicate group membership. SPIA is instance _B, SPIB is instance _A.

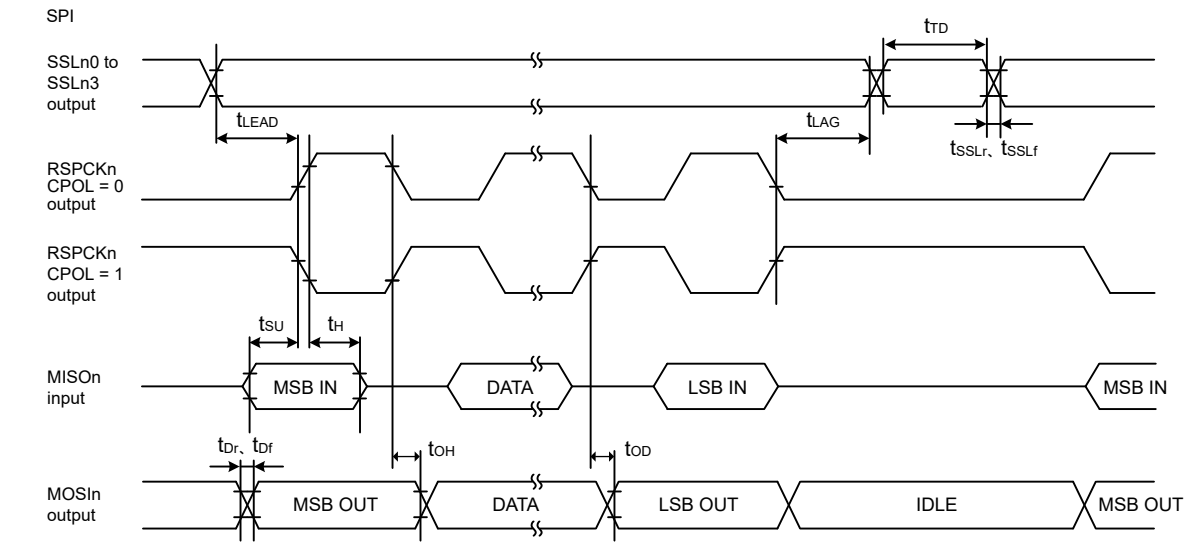
Note 2. All pins of group membership can be used.



V_{OH} = 0.7 × VCC, V_{OL} = 0.3 × VCC, V_{IH} = 0.7 × VCC, V_{IL} = 0.3 × VCC

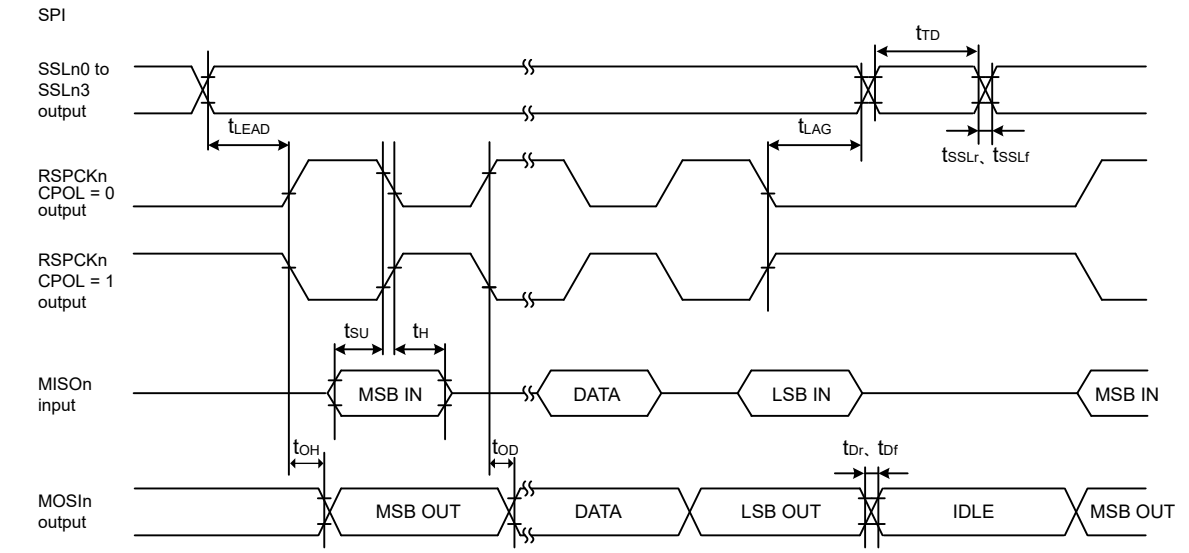
Note: n = A or B

Figure 46.39 SPI clock timing



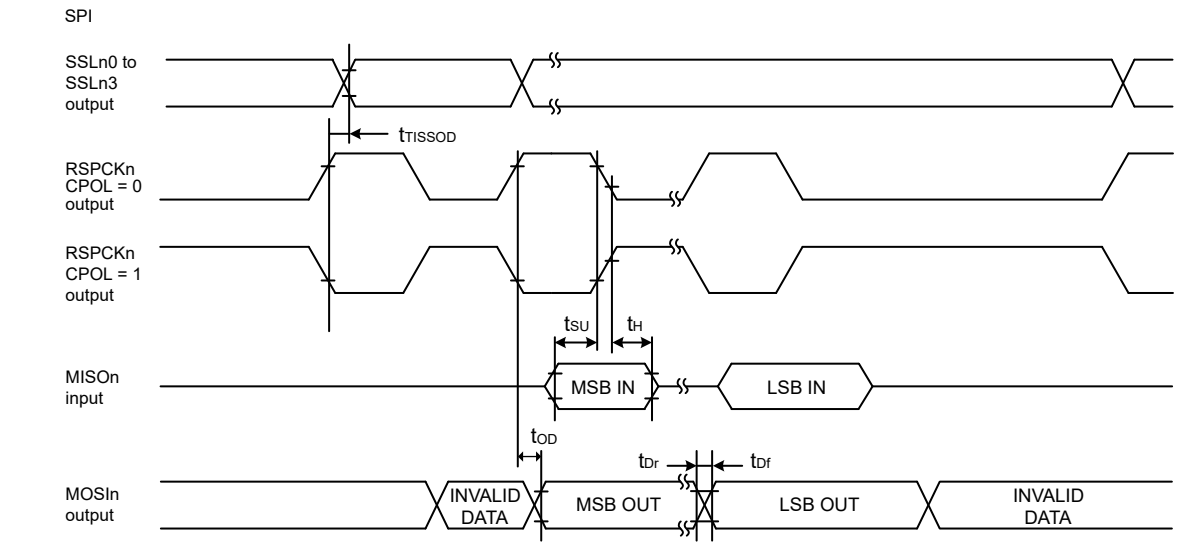
Note: n = A or B

Figure 46.40 SPI timing for Motorola SPI master when CPHA = 0



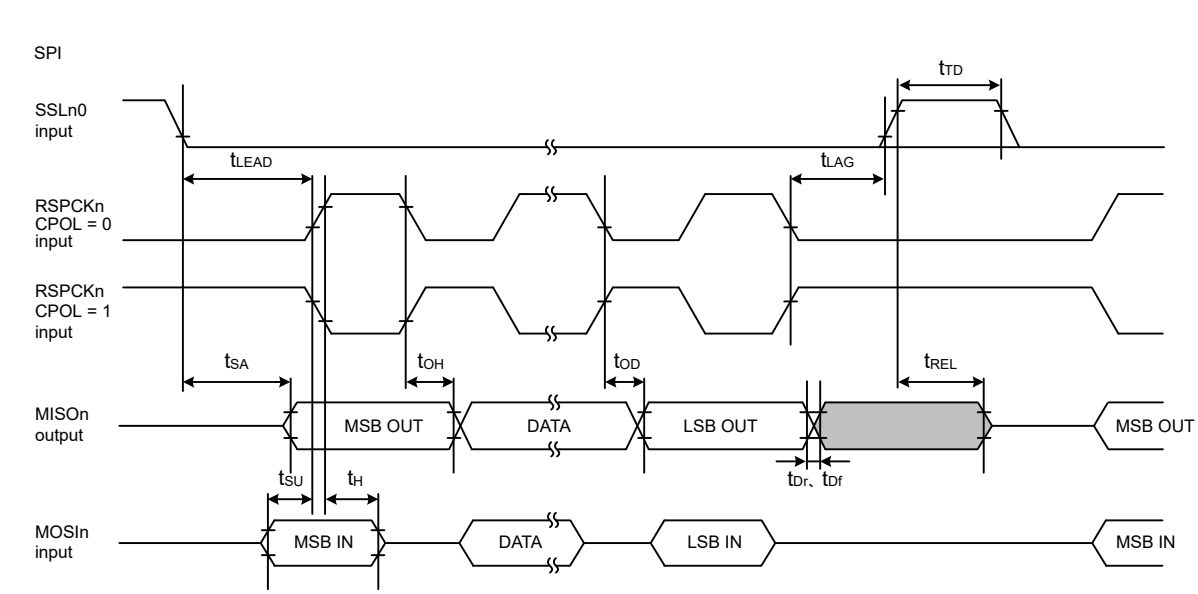
Note: n = A or B

Figure 46.41 SPI timing for Motorola SPI master when CPHA = 1



Note: n = A or B

Figure 46.42 SPI timing for TI SSP master



Note: n = A or B

Figure 46.43 SPI timing for Motorola SPI slave when CPHA = 0

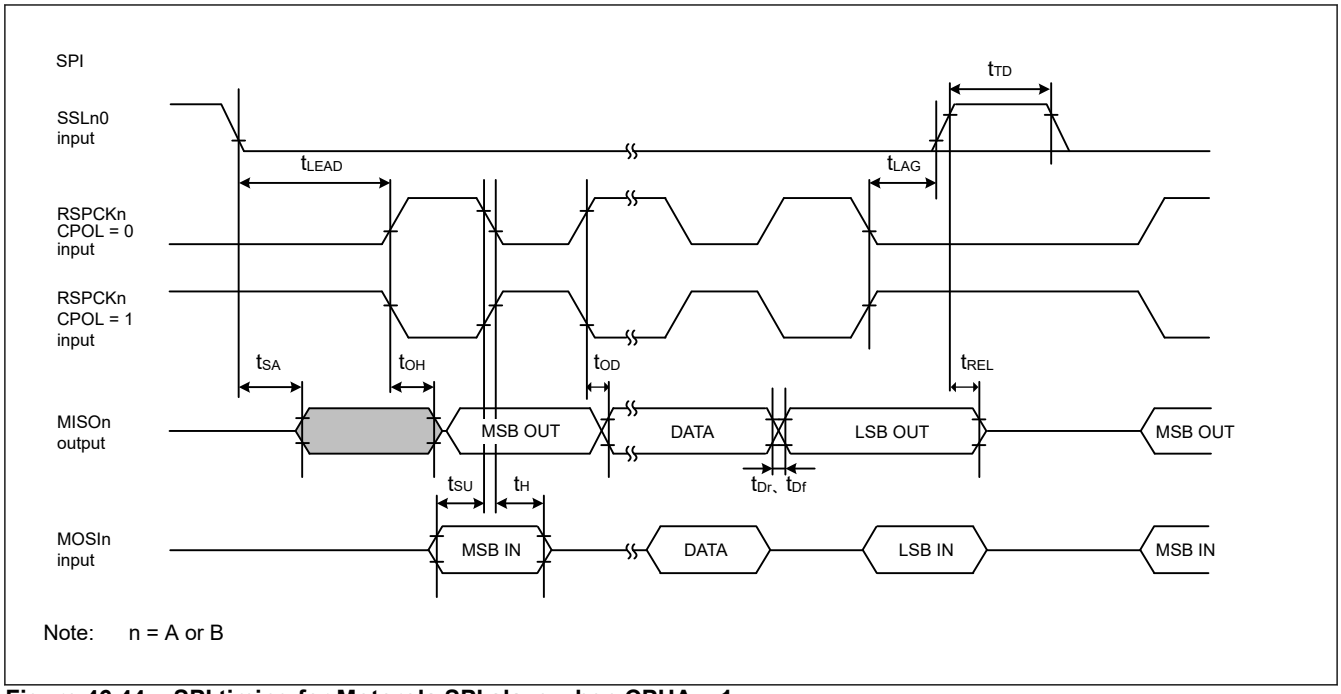


Figure 46.44 SPI timing for Motorola SPI slave when CPHA = 1

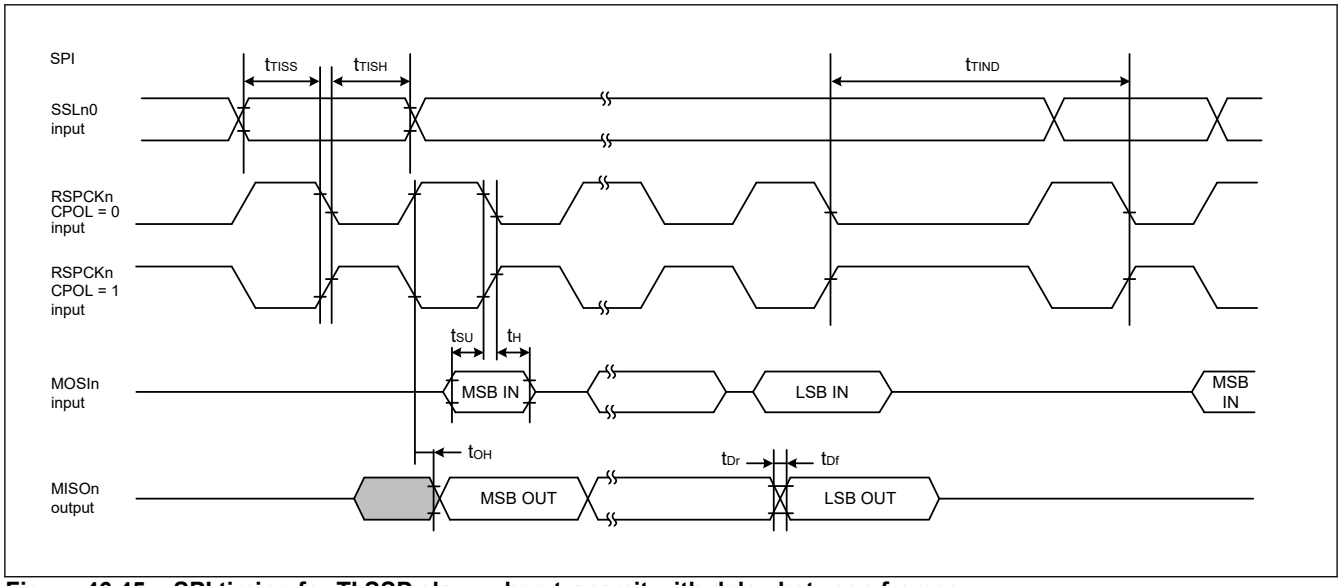


Figure 46.45 SPI timing for TI SSP slave when transmit with delay between frames

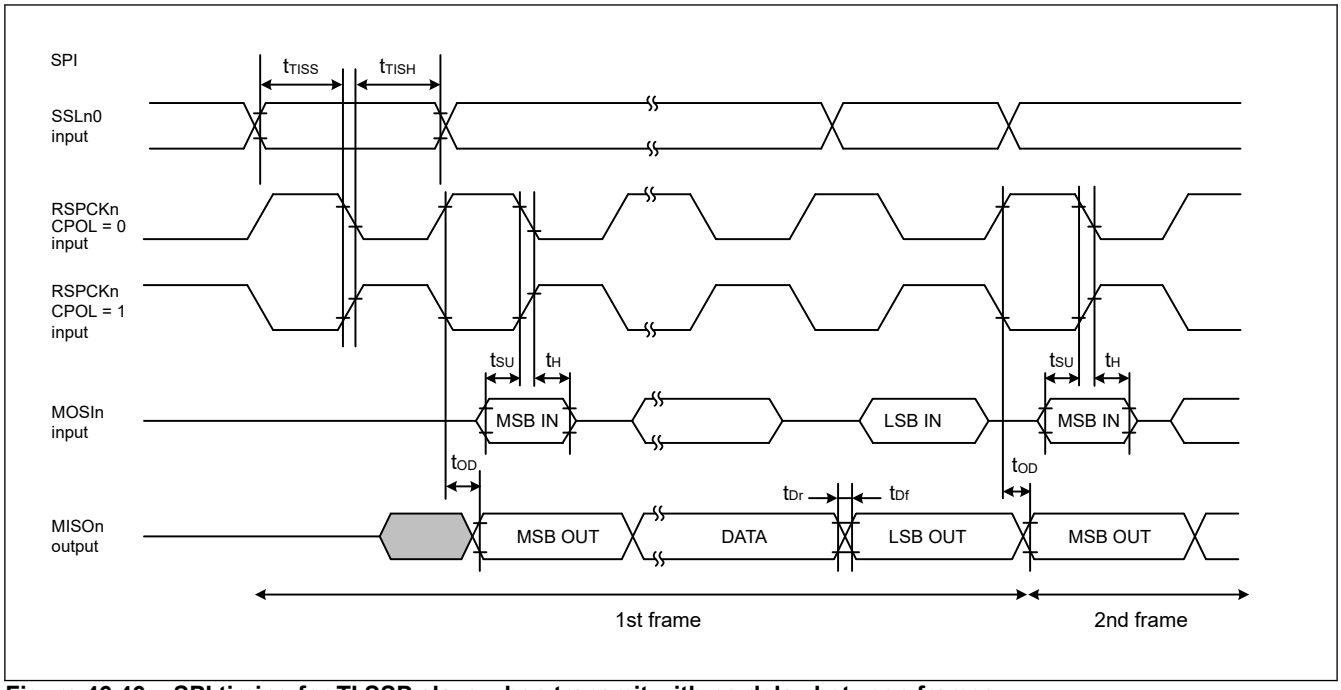


Figure 46.46 SPI timing for TI SSP slave when transmit with no delay between frames

46.3.11 IIC Timing

Table 46.29 IIC timing (1)-1

(1) Conditions: Middle drive output is selected in the Port Drive Capability bit in the PmnPFS register for the following pins: SDA0_B, SCL0_B, SDA1_B, SCL1_B, SCL0_C, SDA0_C, SCL0_D, SDA0_D, SCL0_E, SDA0_E, SCL0_F, SDA0_F, SCL1_C, SDA1_C, SCL1_D, SDA1_D, SCL1_E, SDA1_E.
(2) The following pins do not require setting: SCL0_A, SDA0_A, SCL1_A, SDA1_A.
(3) Use pins that have a letter appended to their names, for instance _A or _B or _C or _D or _E or _F, to indicate group membership. For the IIC interface, the AC portion of the electrical characteristics is measured for each group.

Parameter		Symbol	Min	Max	Unit	Test conditions
IIC (Standard mode, SMBus) BFCTL.FMPE = 0	SCL input cycle time	t_{SCL}	10 (18) × t_{IICcyc} + 1300	—	ns	Figure 46.47
	SCL input high pulse width	t_{SCLH}	5 (9) × t_{IICcyc}	—	ns	
	SCL input low pulse width	t_{SCLL}	5 (9) × t_{IICcyc}	—	ns	
	SCL, SDA rise time	t_{Sr}	—	1000	ns	
	SCL, SDA fall time	t_{Sf}	—	300	ns	
	SCL, SDA input spike pulse removal time	t_{SP}	0	1 (4) × t_{IICcyc}	ns	
	SDA input bus free time	t_{BUF}	5 (9) × t_{IICcyc} + 300	—	ns	
	START condition input hold time	t_{STAH}	t_{IICcyc} + 300	—	ns	
	Repeated START condition input setup time	t_{STAS}	1000	—	ns	
	STOP condition input setup time	t_{STOS}	1000	—	ns	
	Data input setup time	t_{SDAS}	t_{IICcyc} + 50	—	ns	
	Data input hold time	t_{SDAH}	0	—	ns	
	SCL, SDA capacitive load	C_b *1	—	400	pF	

Note: t_{IICcyc}: IIC internal reference clock (IICφ) cycle.
Note: Values in parentheses apply when INCTL.DNFS[3:0] is set to 0011b while the digital filter is enabled with INCTL.DNFE set to 1.
Note 1. C_b indicates the total capacity of the bus line.

Table 46.30 IIC timing (1)-2

(1) Conditions: Middle drive output is selected in the Port Drive Capability bit in the PmnPFS register for the following pins: SDA0_B, SCL0_B, SDA1_B, SCL1_B, SCL0_C, SDA0_C, SCL0_D, SDA0_D, SCL0_E, SDA0_E, SCL0_F, SDA0_F, SCL1_C, SDA1_C, SCL1_D, SDA1_D, SCL1_E, SDA1_E.
(2) The following pins do not require setting: SCL0_A, SDA0_A, SCL1_A, SDA1_A.
(3) Use pins that have a letter appended to their names, for instance _A or _B or _C or _D or _E or _F, to indicate group membership. For the IIC interface, the AC portion of the electrical characteristics is measured for each group.

Parameter		Symbol	Min	Max	Unit	Test conditions
IIC (Fast-mode)	SCL input cycle time	t_{SCL}	10 (18) × t_{IICcyc} + 600	—	ns	Figure 46.47
	SCL input high pulse width	t_{SCLH}	5 (9) × t_{IICcyc}	—	ns	
	SCL input low pulse width	t_{SCLL}	5 (9) × t_{IICcyc}	—	ns	
	SCL, SDA rise time	t_{Sr}	20 × (external pullup voltage/ 5.5V)*1	300	ns	
	SCL, SDA fall time	t_{Sf}	20 × (external pullup voltage/ 5.5V)*1	300	ns	
	SCL, SDA input spike pulse removal time	t_{SP}	0	1 (4) × t_{IICcyc}	ns	
	SDA input bus free time	t_{BUF}	5 (9) × t_{IICcyc} + 300	—	ns	
	START condition input hold time	t_{STAH}	t_{IICcyc} + 300	—	ns	
	Repeated START condition input setup time	t_{STAS}	300	—	ns	
	STOP condition input setup time	t_{STOS}	300	—	ns	
	Data input setup time	t_{SDAS}	t_{IICcyc} + 50	—	ns	
	Data input hold time	t_{SDAH}	0	—	ns	
	SCL, SDA capacitive load	C_b*2	—	400	pF	

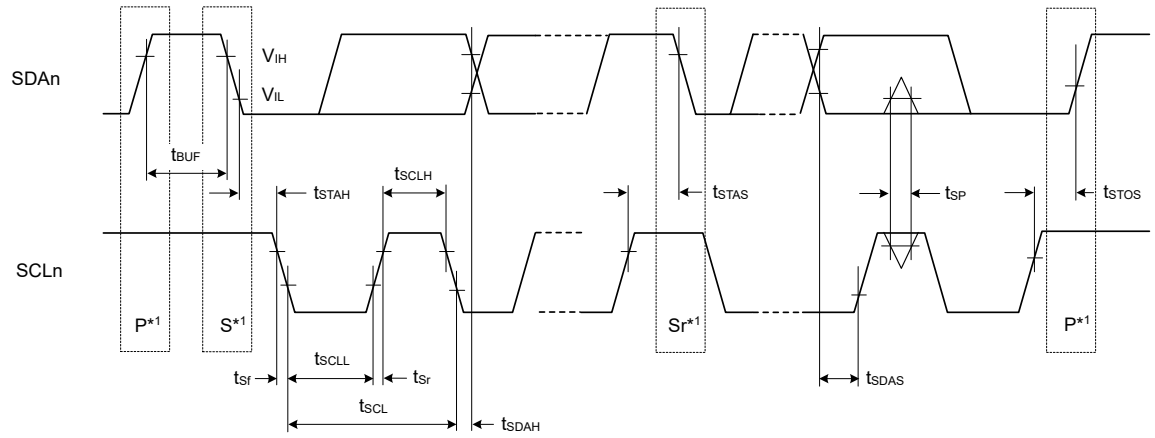
Note: t_{IICcyc}: IIC internal reference clock (IICφ) cycle.
Note: Values in parentheses apply when INCTL.DNFS[3:0] is set to 0011b while the digital filter is enabled with INCTL.DNFE set to 1.
Note: Must use pins that have a letter appended to their name, for instance _A, _B, to indicate group membership. For the IIC interface, the AC portion of the electrical characteristics is measured for each group.
Note 1. Only supported for SCL0_A, SDA0_A, SCL1_A, and SDA1_A. Other ports are depend on DC characteristics.
Note 2. C_b indicates the total capacity of the bus line.

Table 46.31 IIC timing (1)-3

Setting of the SCL0_A, SDA0_A pins are not required with the Port Drive Capability bit in the PmnPFS register.

Parameter		Symbol	Min	Max	Unit	Test conditions
IIC (Fast-mode+) BFCTL.FMPE = 1	SCL input cycle time	t_{SCL}	10 (18) × t_{IICcyc} + 240	—	ns	Figure 46.47
	SCL input high pulse width	t_{SCLH}	5 (9) × t_{IICcyc}	—	ns	
	SCL input low pulse width	t_{SCLL}	5 (9) × t_{IICcyc}	—	ns	
	SCL, SDA rise time	t_{Sr}	—	120	ns	
	SCL, SDA fall time	t_{Sf}	—	120	ns	
	SCL, SDA input spike pulse removal time	t_{SP}	0	1 (4) × t_{IICcyc}	ns	
	SDA input bus free time	t_{BUF}	5 (9) × t_{IICcyc} + 120	—	ns	
	START condition input hold time	t_{STAH}	t_{IICcyc} + 120	—	ns	
	Repeated START condition input setup time	t_{STAS}	120	—	ns	
	STOP condition input setup time	t_{STOS}	120	—	ns	
	Data input setup time	t_{SDAS}	t_{IICcyc} + 30	—	ns	
	Data input hold time	t_{SDAH}	0	—	ns	
	SCL, SDA capacitive load	C_b *1	—	550	pF	

Note: t_{IICcyc}: IIC internal reference clock (IICφ) cycle.
Note: Values in parentheses apply when INCTL.DNFS[3:0] is set to 0011b while the digital filter is enabled with INCTL.DNFE set to 1.
Note: Targets are SCL0_A and SDA0_A.
Note 1. C_b indicates the total capacity of the bus line.



Note 1. S, P, and Sr indicate the following conditions:
S: Start condition
P: Stop condition
Sr: Restart condition

Figure 46.47 I²C bus interface input/output timing

Table 46.32 IIC timing (2)

Setting of the SCL0_A, SDA0_A pins are not required with the Port Drive Capability bit in the PmnPFS register.

Parameter			Symbol	Min	Max	Unit	Test conditions
IIC (Hs-mode) BFCTL.HSME = 1	SCL input cycle time		t_{SCL}	10 (12) × t_{IICcyc} + 80	—	ns	Figure 46.48
	SCL input high pulse width		t_{SCLH}	5 (6) × t_{IICcyc}	—	ns	
	SCL input low pulse width		t_{SCLL}	5 (6) × t_{IICcyc}	—	ns	
	SCL rise time	C_b = 400pF	t_{SrCL}	—	80	ns	
		C_b = 100pF		—	40	ns	
	SDA rise time	C_b = 400pF	t_{SrDA}	—	160	ns	
		C_b = 100pF		—	80	ns	
	SCL fall time	C_b = 400pF	t_{SfCL}	—	80	ns	
		C_b = 100pF		—	40	ns	
	SDA fall time	C_b = 400pF	t_{SfDA}	—	160	ns	
		C_b = 100pF		—	80	ns	
	SCL, SDA input spike pulse removal time		t_{SP}	0	1 (1) × t_{IICcyc}	ns	
	Repeated START condition input setup time		t_{STAS}	40	—	ns	
	STOP condition input setup time		t_{STOS}	40	—	ns	
	Data input setup time		t_{SDAS}	10	—	ns	
	Data input hold time	C_b = 400pF	t_{SDAH}	0	150	ns	
		C_b = 100pF		0	70	ns	
	SCL, SDA capacitive load		C_b*1	—	400	pF	

Note: t_{IICcyc}: IIC internal reference clock (IICφ) cycle.
Note: Values in parentheses apply when INCTL.DNFS[3:0] is set to 0011b while the digital filter is enabled with INCTL.DNFE set to 1.
Note: Targets are SCL0_A and SDA0_A.
Note 1. C_b indicates the total capacity of the bus line.

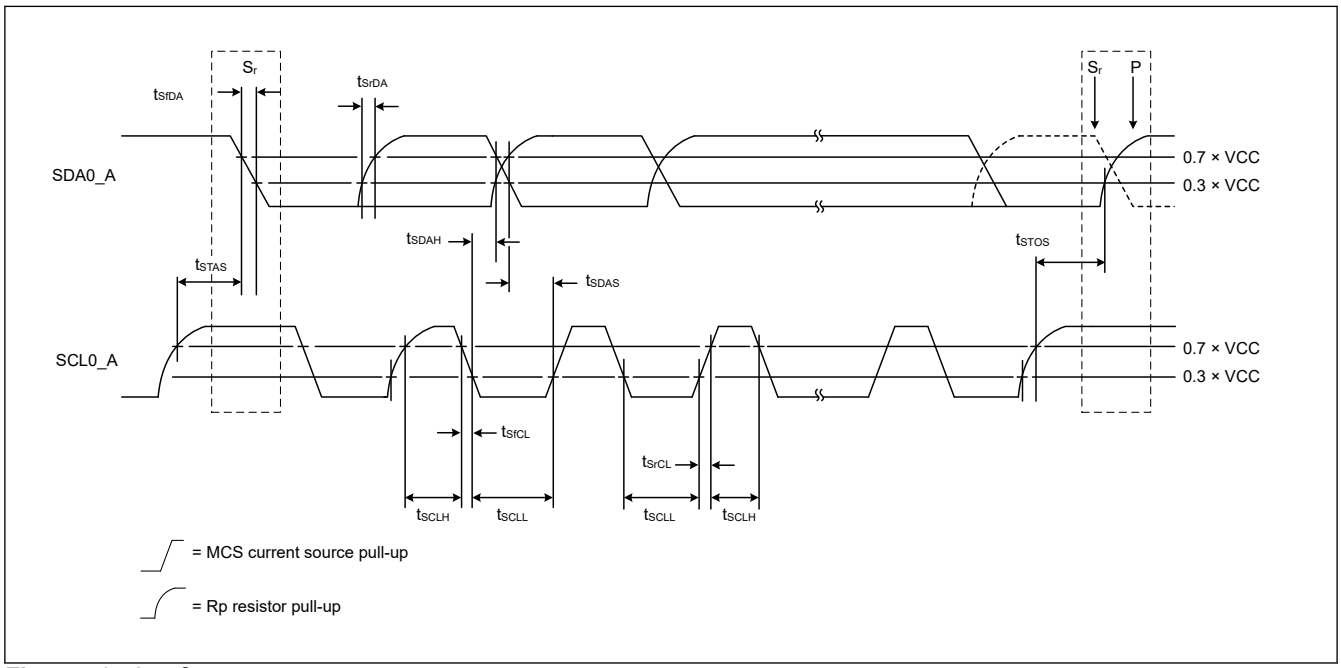


Figure 46.48 I²C bus interface input/output timing (Hs-mode)

46.3.12 CANFD Timing

Table 46.33 CANFD interface timing

Parameter	Symbol	CAN		CAN-FD		Unit	Test conditions
		Min	Max	Min	Max		
Internal delay time	t_{node}	—	100	—	75	ns	Figure 46.49
Transmission rate		—	1	—	5	Mbps	

Note: $t_{node} = t_{output} + t_{input}$

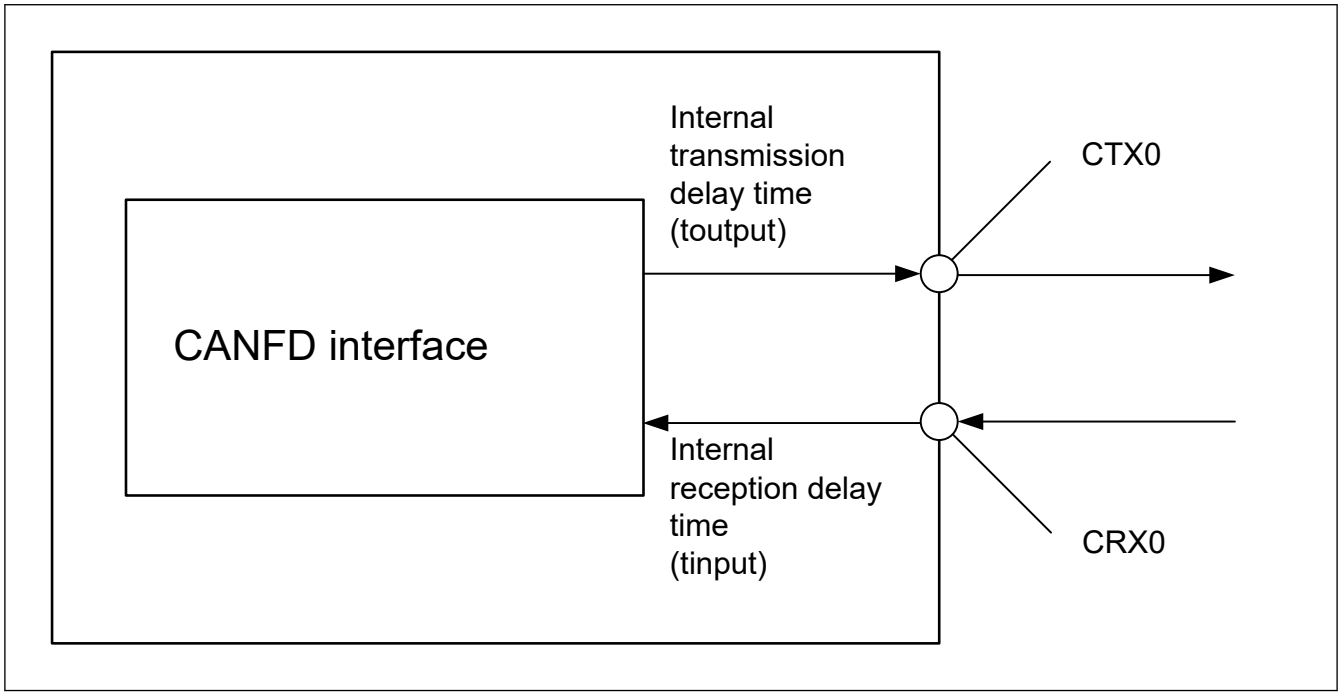


Figure 46.49 CANFD interface condition

46.4 A/D Converter Characteristics

Table 46.34 A/D conversion characteristics (Common) (1 of 2)

Parameter	Min	Typ	Max	Unit	Test conditions
A/D conversion clock frequency(ADCLK)	25	50	60	MHz	—
Successive approximation time	100	—	140	ns	—

Table 46.34 A/D conversion characteristics (Common) (2 of 2)

Parameter						Min	Typ	Max	Unit	Test conditions
A/D sampling time	Self-calibration				SAR mode	1 × t_{ADcyc} + 40	—	—	ns	—
					Oversampling mode	1 × t_{ADcyc} + 40	—	—	ns	—
					Hybrid mode	1 × t_{ADcyc} + 60	—	—	ns	—
	Self-diagnosis				SAR mode	1 × t_{ADcyc} + 40	—	—	ns	—
					Oversampling mode	1 × t_{ADcyc} + 40	—	—	ns	—
					Hybrid mode	1 × t_{ADcyc} + 60	—	—	ns	—
	A/D conversion	High-precision high-speed channels	Without channel-dedicated sample-and-hold circuits	(AN000 to AN005) (AN006 to AN011) (AN018 to AN019)	SAR mode	1 × t_{ADcyc} + 40	—	—	ns	—
					Oversampling mode	1 × t_{ADcyc} + 40	—	—	ns	—
					Hybrid mode	1 × t_{ADcyc} + 60	—	—	ns	—
			With channel-dedicated sample-and-hold circuits	(AN000 to AN005) (AN006 to AN011)	SAR mode	1 × t_{ADcyc} + 160	—	—	ns	—
					Hybrid mode	1 × t_{ADcyc} + 160	—	—	ns	—
		High-precision middle-speed channels		(AN012 to AN017)	SAR mode	180	—	—	ns	—
					Oversampling mode	180	—	—	ns	—
					Hybrid mode	180	—	—	ns	—
		Normal-precision low-speed channels		(AN020 to AN028)	SAR mode	400	—	—	ns	—
					Oversampling mode	400	—	—	ns	—
					Hybrid mode	400	—	—	ns	—
Channel-dedicated sample-and-hold circuits	Sampling time	Self-calibration				1 × t_{ADcyc} + 400	—	—	ns	—
		A/D conversion				400	—	—	ns	—
	Hold mode switching time					40	—	—	ns	—
	Hold time					—	—	5	µs	—
Operation stabilization time	A/D start-up time					2.0	—	—	µs	—
	Channel-dedicated sample-and-hold circuits start-up time					2.0	—	—	µs	—
	A/D shut-down time					1.0	—	—	µs	—

Note: t_{ADcyc} : ADCLK cycle

Table 46.35 A/D conversion characteristics (SAR mode) (1 of 2)

Parameter					Min	Typ	Max	Unit	Test conditions
SAR mode	Analog input voltage range				VREFL0	—	VREFH0	V	—
	Resolution				—	—	12	bit	—
	Quantization error				—	±0.5	—	LSB	—
	High-precision high-speed channels (AN000 to AN005) (AN006 to AN011) (AN018 to AN019)*3	Without channel-dedicated sample-and-hold circuits*3	Conversion time*1	Normal conversion	0.16	—	—	µs	• ADCLK: 50 MHz • Sampling time: 3 ADCLK • Successive approximation time: 5 ADCLK • Signal source impedance: 50 Ω or less
				With Averaging mode (4-time conversion)	0.64	—	—	µs	
			Offset error		—	±1.0	±3.0	LSB	—
			Full-scale error		—	±1.5	±2.5	LSB	—
			Absolute accuracy	Normal conversion	—	±5.5	±7.0	LSB	—
				With Averaging mode (4-time conversion)	—	±4.5	±5.5	LSB	—
			Total unadjusted error (TUE)*4		—	±3.5	±4.0	LSB	—
			DNL differential nonlinearity error		—	-1 to +1.5	-1 to +2.5	LSB	—
			INL integral nonlinearity error		—	±2.0	±3.0	LSB	—
		With channel-dedicated sample-and-hold circuits	Conversion time*2	Normal conversion	0.72	—	—	µs	• ADCLK: 50 MHz • Sampling time of channel-dedicated sample-and-hold circuits: 20 ADCLK • Hold mode switching time of channel-dedicated sample-and-hold circuits: 2 ADCLK • Sampling time: 9 ADCLK • Successive approximation time: 5 ADCLK • Signal source impedance: 50 Ω or less
				With Averaging mode (4-time conversion)	2.88	—	—	µs	
			Offset error		—	±0.5	±1.0	LSB	—
			Full-scale error		—	±1.5	±1.5	LSB	—
			Absolute accuracy	Normal conversion	—	±5.0	±7.0	LSB	—
				With Averaging mode (4-time conversion)	—	±4.0	±5.0	LSB	—
			Total unadjusted error (TUE)*4		—	±3.0	±3.4	LSB	—
			DNL differential nonlinearity error		—	-1 to +1.5	-1 to +2.5	LSB	—
			INL integral nonlinearity error		—	±2.0	±3.0	LSB	—

Table 46.35 A/D conversion characteristics (SAR mode) (2 of 2)

Parameter				Min	Typ	Max	Unit	Test conditions
SAR mode	High-precision middle-speed channels (AN012 to AN017)	Conversion time*1	Normal conversion	0.28	—	—	µs	• ADCLK: 50 MHz • Sampling time: 9 ADCLK • Successive approximation time: 5 ADCLK • Signal source impedance: 50 Ω or less
			With Averaging mode (4-time conversion)	1.12	—	—	µs	
		Offset error		—	±1.0	±1.5	LSB	—
		Full-scale error		—	±1.0	±2.5	LSB	—
		Absolute accuracy	Normal conversion	—	±4.0	±7.0	LSB	—
			With Averaging mode (4-time conversion)	—	±3.0	±5.5	LSB	—
		Total unadjusted error (TUE)*4		—	±3.4	±4.4	LSB	—
		DNL differential nonlinearity error		—	-1 to +1.5	-1 to +2.5	LSB	—
		INL integral nonlinearity error		—	±2.0	±3.0	LSB	—
	Normal-precision low-speed channels (AN020 to AN028)	Conversion time*1	Normal conversion	0.50	—	—	µs	• ADCLK: 50 MHz • Sampling time: 20 ADCLK • Successive approximation time: 5 ADCLK • Signal source impedance: 50 Ω or less
			With Averaging mode (4-time conversion)	2.00	—	—	µs	
		Offset error		—	±1.0	±2.5	LSB	—
		Full-scale error		—	±1.5	±2.5	LSB	—
		Absolute accuracy	Normal conversion	—	±5.5	±8.0	LSB	—
			With Averaging mode (4-time conversion)	—	±5.5	±7.0	LSB	—
		Total unadjusted error (TUE)*4		—	±4.2	±5.3	LSB	—
		DNL differential nonlinearity error		—	-1 to +1.5	-1 to +2.5	LSB	—
		INL integral nonlinearity error		—	±2.0	±4.0	LSB	—

Note 1. Without channel-dedicated sample-and-hold circuits; The conversion time is the sum of the sampling time and the successive approximation time. Each of the above state is indicated for the test conditions.
Note 2. With channel-dedicated sample-and-hold circuits; The conversion time is the sum of the sampling time of channel-dedicated sample-and-hold circuits, the hold mode switching time, the sampling time and the successive approximation time. Each of the above state is indicated for the test conditions.
Note 3. Channel-dedicated sample-and-hold circuits are not available in these channels.
Note 4. Excludes quantization error (±0.5 LSB).

Table 46.36 A/D conversion characteristics (Oversampling mode and Hybrid mode) (1)

Parameter				Min	Typ	Max	Unit	Test conditions
Oversampling mode and Hybrid mode	Analog input voltage range	Single-ended input voltage		VREFL0	—	VREFH0	V	—
		Differential input voltage*1		-VREFH0	—	+VREFH0	V	—
	Resolution			—	—	16	bit	—
	Oversampling period	Oversampling mode		0.16	—	—	µs	• ADCLK: 50 MHz • Sampling time: 3 ADCLK • Successive approximation time: 5 ADCLK • Without disconnection detection assist function
		Hybrid mode		0.18	—	—	µs	• ADCLK: 50 MHz • Sampling time: 4 ADCLK • Successive approximation time: 5 ADCLK • Without disconnection detection assist function
	Digital filter characteristics *2	Sinc filter	Initial delay	—	22	—	Fos	—
			Group delay	—	11	—		—
			Normalized Cutoff Frequency	—	0.033	—	Fin/Fos	—
		Minimum phase filter	Initial delay	—	22	—	Fos	—
			Group delay	—	2	—		—
			Normalized Cutoff Frequency	—	0.116	—	Fin/Fos	—
			Passband ripple	—	<± 0.01	—	dB	—

Note: Fos is oversampling frequency.
Note 1. Differential input voltage is (A_{INP} - A_{INN})
- A_{INP} is input voltage of ANx, and VREFL0 ≤ A_{INP} ≤ VREFH0.
- A_{INN} is input voltage of ANy, and VREFL0 ≤ A_{INN} ≤ VREFH0.
  (x = 2i, y = 2i +1, i = 0, 1, 2… (any integer))

Note 2. See Figure 46.50 and Figure 46.51.

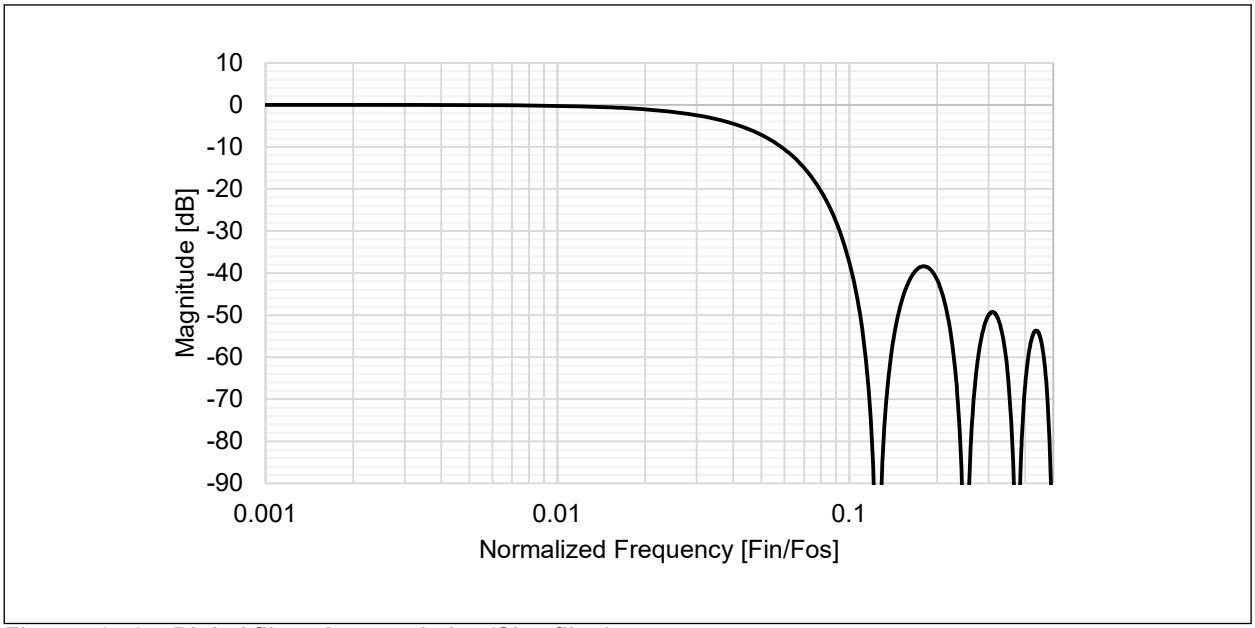


Figure 46.50 Digital filter characteristics (Sinc filter)

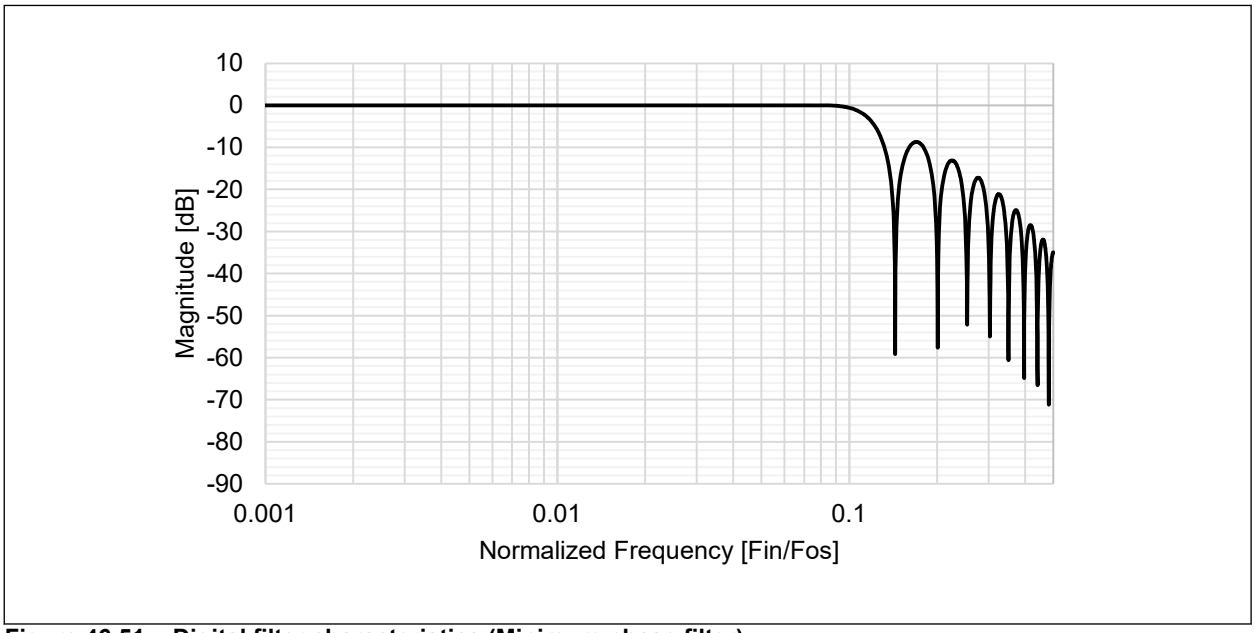


Figure 46.51 Digital filter characteristics (Minimum phase filter)

Table 46.37 A/D conversion characteristics (Oversampling mode and Hybrid mode) (2)

Parameter				Min	Typ	Max	Unit	Test conditions
Oversampling mode and Hybrid mode (AN000 to AN005) (AN006 to AN011) (AN018 to AN019) (AN012 to AN017)	Sinc filter	Single-ended input	SNDR signal-to-noise and distortion ratio	—	80	—	dB	• ADCLK: 50 MHz • Sampling time: High-precision high-speed channels (Oversampling mode): 3 ADCLK High-precision high-speed channels (Hybrid mode): 4 ADCLK High-precision middle-speed channels: 9 ADCLK • Successive approximation time: 5 ADCLK • Signal source impedance: 50 Ω or less • Input frequency: Oversampling mode: 10 kHz Hybrid mode: 4 kHz • Without channel-dedicated sample-and-hold circuits
			ENOB effective number of bits	—	13	—	bit	
		Differential input	SNDR signal-to-noise and distortion ratio	—	86	—	dB	
			ENOB effective number of bits	—	14	—	bit	
	Minimum phase filter	Single-ended input	SNDR signal-to-noise and distortion ratio	—	68	—	dB	
			ENOB effective number of bits	—	11	—	bit	
		Differential input	SNDR signal-to-noise and distortion ratio	—	74	—	dB	
			ENOB effective number of bits	—	12	—	bit	

Table 46.38 A/D conversion characteristics (Oversampling mode)

Parameter				Min	Typ	Max	Unit	Test conditions
Oversampling mode (AN000 to AN005) (AN006 to AN011) (AN018 to AN019) (AN012 to AN017) (AN020 to AN028)	Single-ended input	Unit0	Offset error	—	±8.0	—	LSB	• ADCLK: 50 MHz • Sampling time: High-precision high-speed channels: 3 ADCLK High-precision middle-speed channels: 9 ADCLK Normal-precision low-speed channels: 20 ADCLK • Successive approximation time: 5 ADCLK • Signal source impedance: 50 Ω or less • Digital filter: Sinc filter • Without channel-dedicated sample-and-hold circuits
			Gain error	—	±32.0	—		
			DNL differential nonlinearity error*1	—	±4.0	—		
			INL integral nonlinearity error*1	—	±8.0	—		
		Unit1	Offset error	—	±8.0	—		
			Gain error	—	±32.0	—		
			DNL differential nonlinearity error*1	—	±4.0	—		
			INL integral nonlinearity error*1	—	±8.0	—		
	Differential input	Unit0	Offset error	—	±4.0	—		
			Gain error	—	±14.0	—		
			DNL differential nonlinearity error*1	—	±2.0	—		
			INL integral nonlinearity error*1	—	±4.0	—		
		Unit1	Offset error	—	±4.0	—		
			Gain error	—	±14.0	—		
			DNL differential nonlinearity error*1	—	±2.0	—		
			INL integral nonlinearity error*1	—	±4.0	—		

Note 1. Test conditions: 0.2% to 99.8% of the analog input voltage range.

Table 46.39 A/D conversion characteristics (Hybrid mode) (1 of 2)

Parameter					Min	Typ	Max	Unit	Test conditions
Hybrid mode (AN000 to AN005) (AN006 to AN011) (AN018 to AN019)*1 (AN012 to AN017)*1 (AN020 to AN028)*1	Without channel-dedicated sample-and-hold circuits*1	Single-ended input	Unit0	Offset error	—	±8.0	—	LSB	• ADCLK: 50 MHz • Sampling time: High-precision high-speed channels: 4 ADCLK High-precision middle-speed channels: 9 ADCLK Normal-precision low-speed channels: 20 ADCLK • Successive approximation time: 5 ADCLK • Signal source impedance: 50 Ω or less • Digital filter: Sinc filter
				Gain error	—	±40.0	—		
				DNL differential nonlinearity error*2	—	±4.0	—		
				INL integral nonlinearity error*2	—	±8.0	—		
			Unit1	Offset error	—	±8.0	—		
				Gain error	—	±40.0	—		
				DNL differential nonlinearity error*2	—	±4.0	—		
				INL integral nonlinearity error*2	—	±8.0	—		
		Differential input	Unit0	Offset error	—	±4.0	—		
				Gain error	—	±20.0	—		
				DNL differential nonlinearity error*2	—	±2.0	—		
				INL integral nonlinearity error*2	—	±4.0	—		
			Unit1	Offset error	—	±4.0	—		
				Gain error	—	±20.0	—		
				DNL differential nonlinearity error*2	—	±2.0	—		
				INL integral nonlinearity error*2	—	±4.0	—		

Table 46.39 A/D conversion characteristics (Hybrid mode) (2 of 2)

Parameter					Min	Typ	Max	Unit	Test conditions
Hybrid mode (AN000 to AN005) (AN006 to AN011) (AN018 to AN019)*1 (AN012 to AN017)*1 (AN020 to AN028)*1	With channel-dedicated sample-and-hold circuits	Single-ended input	Unit0	Offset error	—	8 ± 72	—	LSB	• ADCLK: 50 MHz • Sampling time of channel-dedicated sample-and-hold circuits: 20 ADCLK • Hold mode switching time of channel-dedicated sample-and-hold circuits: 2 ADCLK • Sampling time: 9 ADCLK • Successive approximation time: 5 ADCLK • Signal source impedance: 50 Ω or less • Digital filter: Sinc filter
				Gain error	—	-23 ± 72	—		
				DNL differential nonlinearity error*2	—	±4.0	—		
				INL integral nonlinearity error*2	—	±8.0	—		
			Unit1	Offset error	—	36 ± 72	—		
				Gain error	—	-23 ± 72	—		
				DNL differential nonlinearity error*2	—	±4.0	—		
				INL integral nonlinearity error*2	—	±8.0	—		
		Differential input	Unit0	Offset error	—	8 ± 72	—		
				Gain error	—	-15 ± 36	—		
				DNL differential nonlinearity error*2	—	±4.0	—		
				INL integral nonlinearity error*2	—	±8.0	—		
			Unit1	Offset error	—	36 ± 72	—		
				Gain error	—	-15 ± 36	—		
				DNL differential nonlinearity error*2	—	±4.0	—		
				INL integral nonlinearity error*2	—	±8.0	—		

Note 1. Channel-dedicated sample-and-hold circuits are not available in these channels.
Note 2. Test conditions: 0.2% to 99.8% of the analog input voltage range.

Table 46.40 A/D internal reference voltage characteristics

Parameter	Min	Typ	Max	Unit	Test conditions
A/D internal reference voltage	1.13	1.18	1.23	V	
Sampling time	4.15	—	—	µs	

Table 46.41 A/D conversion characteristics of D/A output

Parameter	Min	Typ	Max	Unit	Test conditions
Sampling time	1	—	—	µs	

46.5 DAC12 Characteristics

Table 46.42 D/A conversion characteristics (1 of 2)

Parameter	Min	Typ	Max	Unit	Test conditions
Resolution	—	—	12	Bits	—
Without output amplifier					
Absolute accuracy	—	—	±24	LSB	Resistive load 2 MΩ
INL	—	±2.0	±8.0	LSB	Resistive load 2 MΩ
DNL	—	±1.0	±2.0	LSB	—
Output impedance	—	8.5	—	kΩ	—

Table 46.42 D/A conversion characteristics (2 of 2)

Parameter	Min	Typ	Max	Unit	Test conditions
Conversion time	—	—	3	µs	Resistive load 2 MΩ, Capacitive load 20 pF
Output voltage range	0	—	AVCC0	V	—
With output amplifier					
INL	—	±2.0	±4.0	LSB	—
DNL	—	±1.0	±2.0	LSB	—
Conversion time	—	—	4.0	µs	—
Resistive load	5	—	—	kΩ	—
Capacitive load	—	—	50	pF	—
Output voltage range	0.2	—	AVCC0 – 0.2	V	—

46.6 TSN Characteristics

Table 46.43 TSN characteristics

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
Relative accuracy	—	—	± 1.0	—	°C	—
Temperature slope	—	—	4.0	—	mV/°C	—
Output voltage (at 25 °C)	—	—	1.24	—	V	—
Temperature sensor start time	t_{START}	—	—	30	µs	—
Sampling time	—	4.15	—	—	µs	—

46.7 ACMPHS Characteristics

Table 46.44 ACMPHS characteristics

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
Input offset voltage	V_{IO}	—	—	40	mV	
Reference voltage range	V_{REF}	0	—	AVCC0	V	
Input voltage range	V_I	0	—	AVCC0	V	
Output delay	$t_{tot(r)}$	—	—	200	ns	VOD = 100 mV CMPCTL.CDFS = 0
	$t_{tot(f)}$	—	—	200	ns	
Waiting time for stabilization following switching of the input	t_{cwait}	300	—	—	ns	
Operation stabilization time	t_{cmp}	—	—	1	µs	

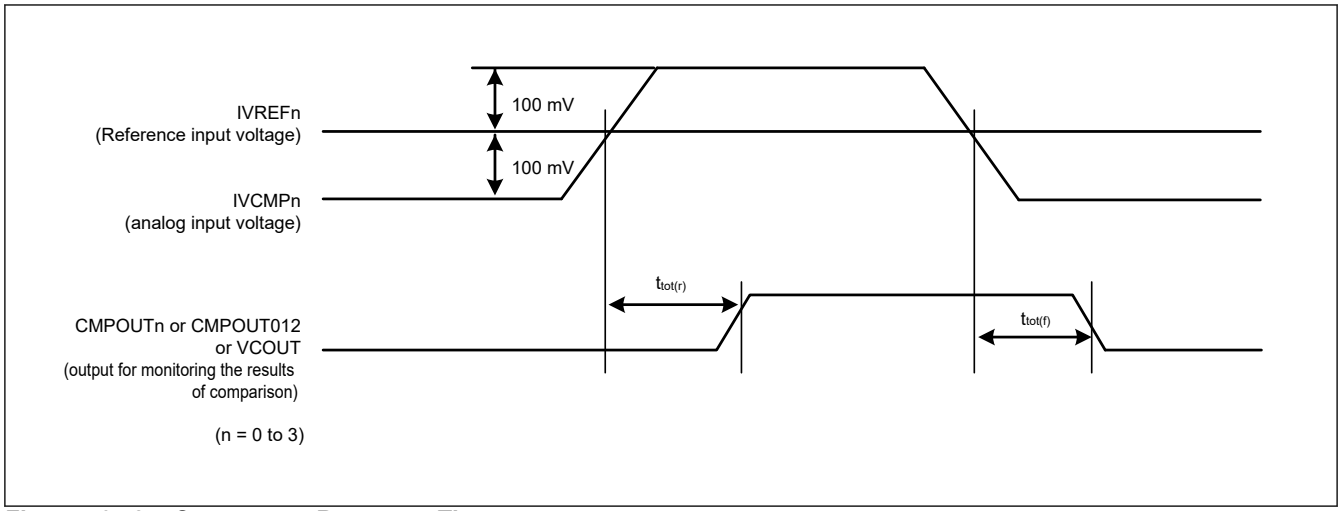


Figure 46.52 Comparator Response Time

46.8 PGA Characteristics

Table 46.45 PGA characteristics in Single-ended input mode (1 of 2)

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
Offset error	Voff	-8	—	8	mV	
PGAVSS input voltage range	PGAVSS	0	—	0	V	
Single-ended input voltage range	AIN0 (G = 2.000)	0.05 × AVCC0	—	0.45 × AVCC0	V	
	AIN1 (G = 2.500)	0.047 × AVCC0	—	0.36 × AVCC0	V	
	AIN2 (G = 2.667)	0.046 × AVCC0	—	0.337 × AVCC0	V	
	AIN3 (G = 2.857)	0.046 × AVCC0	—	0.32 × AVCC0	V	
	AIN4 (G = 3.077)	0.045 × AVCC0	—	0.292 × AVCC0	V	
	AIN5 (G = 3.333)	0.044 × AVCC0	—	0.265 × AVCC0	V	
	AIN6 (G = 3.636)	0.042 × AVCC0	—	0.247 × AVCC0	V	
	AIN7 (G = 4.000)	0.04 × AVCC0	—	0.212 × AVCC0	V	
	AIN8 (G = 4.444)	0.036 × AVCC0	—	0.191 × AVCC0	V	
	AIN9 (G = 5.000)	0.033 × AVCC0	—	0.17 × AVCC0	V	
	AIN10 (G = 5.714)	0.031 × AVCC0	—	0.148 × AVCC0	V	
	AIN11 (G = 6.667)	0.029 × AVCC0	—	0.127 × AVCC0	V	
	AIN12 (G = 8.000)	0.027 × AVCC0	—	0.09 × AVCC0	V	
	AIN13 (G = 10.000)	0.025 × AVCC0	—	0.08 × AVCC0	V	
	AIN14 (G = 13.333)	0.023 × AVCC0	—	0.06 × AVCC0	V	

Table 46.45 PGA characteristics in Single-ended input mode (2 of 2)

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
Output voltage range*1	PGAOUT0 (G = 2.000)	0.100 × AVCC0	—	0.900 × AVCC0	V	
	PGAOUT1 (G = 2.500)	0.118 × AVCC0	—	0.900 × AVCC0	V	
	PGAOUT2 (G = 2.667)	0.123 × AVCC0	—	0.899 × AVCC0	V	
	PGAOUT3 (G = 2.857)	0.131 × AVCC0	—	0.914 × AVCC0	V	
	PGAOUT4 (G = 3.077)	0.138 × AVCC0	—	0.898 × AVCC0	V	
	PGAOUT5 (G = 3.333)	0.147 × AVCC0	—	0.883 × AVCC0	V	
	PGAOUT6 (G = 3.636)	0.153 × AVCC0	—	0.898 × AVCC0	V	
	PGAOUT7 (G = 4.000)	0.160 × AVCC0	—	0.848 × AVCC0	V	
	PGAOUT8 (G = 4.444)	0.160 × AVCC0	—	0.849 × AVCC0	V	
	PGAOUT9 (G = 5.000)	0.165 × AVCC0	—	0.850 × AVCC0	V	
	PGAOUT10 (G = 5.714)	0.177 × AVCC0	—	0.846 × AVCC0	V	
	PGAOUT11 (G = 6.667)	0.193 × AVCC0	—	0.847 × AVCC0	V	
	PGAOUT12 (G = 8.000)	0.216 × AVCC0	—	0.720 × AVCC0	V	
	PGAOUT13 (G = 10.000)	0.250 × AVCC0	—	0.800 × AVCC0	V	
	PGAOUT14 (G = 13.333)	0.307 × AVCC0	—	0.800 × AVCC0	V	
Gain error	Gerr0 (G = 2.000)	-1.0	—	1.0	%	
	Gerr1 (G = 2.500)	-1.0	—	1.0	%	
	Gerr2 (G = 2.667)	-1.0	—	1.0	%	
	Gerr3 (G = 2.857)	-1.0	—	1.0	%	
	Gerr4 (G = 3.007)	-1.0	—	1.0	%	
	Gerr5 (G = 3.333)	-1.5	—	1.5	%	
	Gerr6 (G = 3.636)	-1.5	—	1.5	%	
	Gerr7 (G = 4.000)	-1.5	—	1.5	%	
	Gerr8 (G = 4.444)	-2.0	—	2.0	%	
	Gerr9 (G = 5.000)	-2.0	—	2.0	%	
	Gerr10 (G = 5.714)	-2.0	—	2.0	%	
	Gerr11 (G = 6.667)	-2.0	—	2.0	%	
	Gerr12 (G = 8.000)	-2.0	—	2.0	%	
	Gerr13 (G = 10.000)	-2.0	—	2.0	%	
	Gerr14 (G = 13.333)	-2.0	—	2.0	%	
Slew rate	SR	10	—	—	V/µs	
Operation stabilization time	t_{start}	—	—	5	µs	

Note 1. Calculate with the following formula. (n = 0 to 14)
PGAOUTn = AINn × G
Actual output range includes gain error.
PGAOUTn = (AINn × G) × (Gerr + 100%)

Table 46.46 PGA characteristics in Pseudo-differential input mode (1 of 2)

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
Offset error	Voff	-20	—	20	mV	
PGAVSS input voltage range	PGAVSS	-0.5	—	0.3	V	

Table 46.46 PGA characteristics in Pseudo-differential input mode (2 of 2)

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Differential input voltage range	G = 1.500	AIN-PGAVSS	-0.5	—	0.5	V	
	G = 2.333		-0.4	—	0.4	V	
	G = 4.000		-0.2	—	0.2	V	
	G = 5.667		-0.15	—	0.15	V	
Output voltage range*1	G = 1.500	V_{OR}	0.600	—	2.550	V	
	G = 2.333		0.417	—	2.733	V	
	G = 4.000		0.550	—	2.600	V	
	G = 5.667		0.500	—	2.650	V	
Gain error	G = 1.500	Gerr	-1.0	—	1.0	%	
	G = 2.333		-1.0	—	1.0	%	
	G = 4.000		-1.0	—	1.0	%	
	G = 5.667		-1.0	—	1.0	%	
Slew rate		SR	10	—	—	V/µs	
Operation stabilization time		t_{start}	—	—	5	µs	

Note 1. Calculate with the following formula.
V_{OR} = (AIN-PGAVSS) × G + (0.5 × AVCC0)
Actual output range includes gain error.
V_{OR} = (AIN-PGAVSS) × G × (Gerr +100%) + (0.5 × AVCC0)

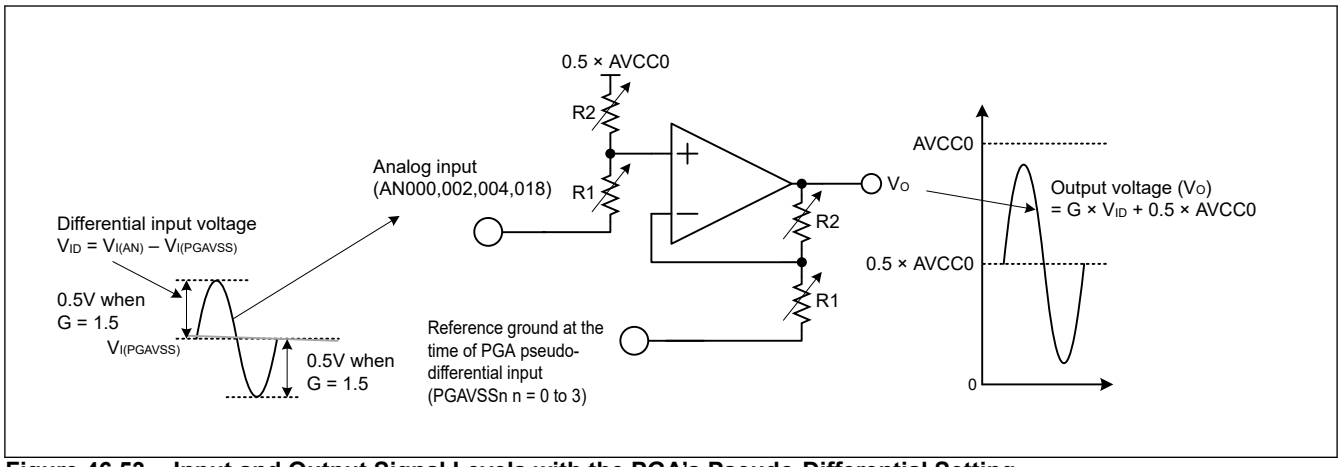


Figure 46.53 Input and Output Signal Levels with the PGA's Pseudo-Differential Setting

46.9 OSC Stop Detect Characteristics

Table 46.47 Oscillation stop detection circuit characteristics

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
Detection time	t_{dr}	—	—	1	ms	Figure 46.54

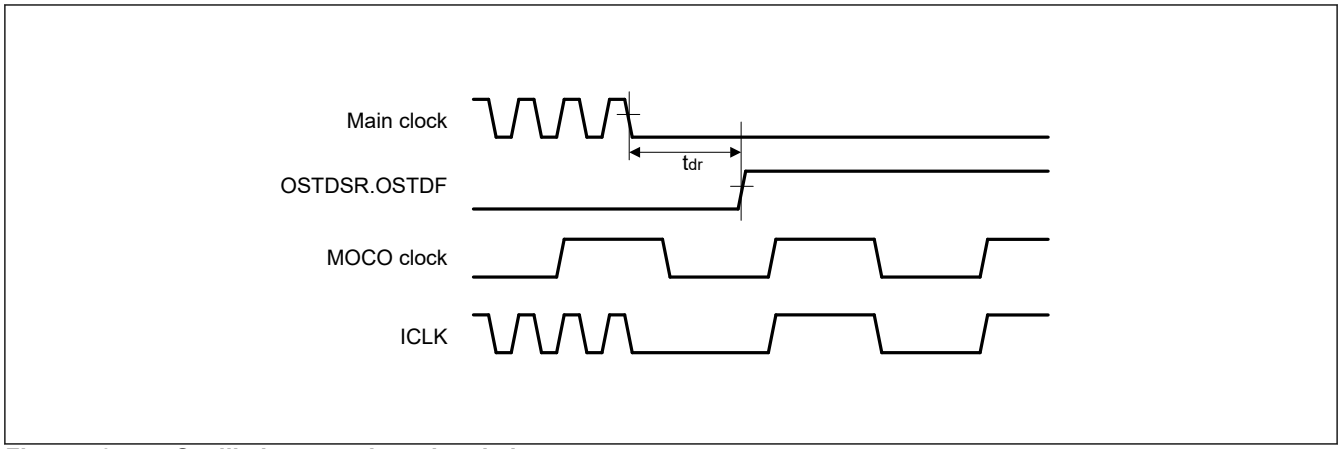


Figure 46.54 Oscillation stop detection timing

46.10 POR and LVD Characteristics

Table 46.48 Power-on reset circuit and voltage detection circuit characteristics (1)

Parameter			Symbol	Min	Typ	Max	Unit	Test conditions
Voltage detection level	Power-on reset (POR)	DPSBYCR.DEEPCUT[1:0] = 00b or 01b.	V_{POR}	2.5	2.6	2.7	V	Figure 46.55
		DPSBYCR.DEEPCUT[1:0] = 11b.		1.8	2.25	2.7		
	Voltage detection circuit (LVD0)		V_{det0_1}	2.84	2.94	3.04		Figure 46.56
			V_{det0_2}	2.77	2.87	2.97		
			V_{det0_3}	2.70	2.80	2.90		
	Voltage detection circuit (LVD1)		V_{det1_1}	2.89	2.99	3.09		Figure 46.57
			V_{det1_2}	2.82	2.92	3.02		
			V_{det1_3}	2.75	2.85	2.95		
	Voltage detection circuit (LVD2)		V_{det2_1}	2.89	2.99	3.09		Figure 46.58
			V_{det2_2}	2.82	2.92	3.02		
			V_{det2_3}	2.75	2.85	2.95		
Internal reset time	Power-on reset time		t_{POR}	—	4.5	—	ms	Figure 46.55
	LVD0 reset time		t_{LVD0}	—	0.51	—		Figure 46.56
	LVD1 reset time		t_{LVD1}	—	0.38	—		Figure 46.57
	LVD2 reset time		t_{LVD2}	—	0.38	—		Figure 46.58
Minimum VCC down time*1			t_{VOFF}	200	—	—	μs	Figure 46.55, Figure 46.56
Response delay			t_{det}	—	—	200	μs	Figure 46.56 to Figure 46.58
LVD operation stabilization time (after LVD is enabled)			$t_{d(E-A)}$	—	—	10	μs	Figure 46.57, Figure 46.58
Hysteresis width (LVD1 and LVD2)			V_{LVH}	—	70	—	mV	

Note 1. The minimum VCC down time indicates the time when VCC is below the minimum value of voltage detection levels V_{POR}, V_{det0}, V_{det1}, and V_{det2} for POR and LVD.

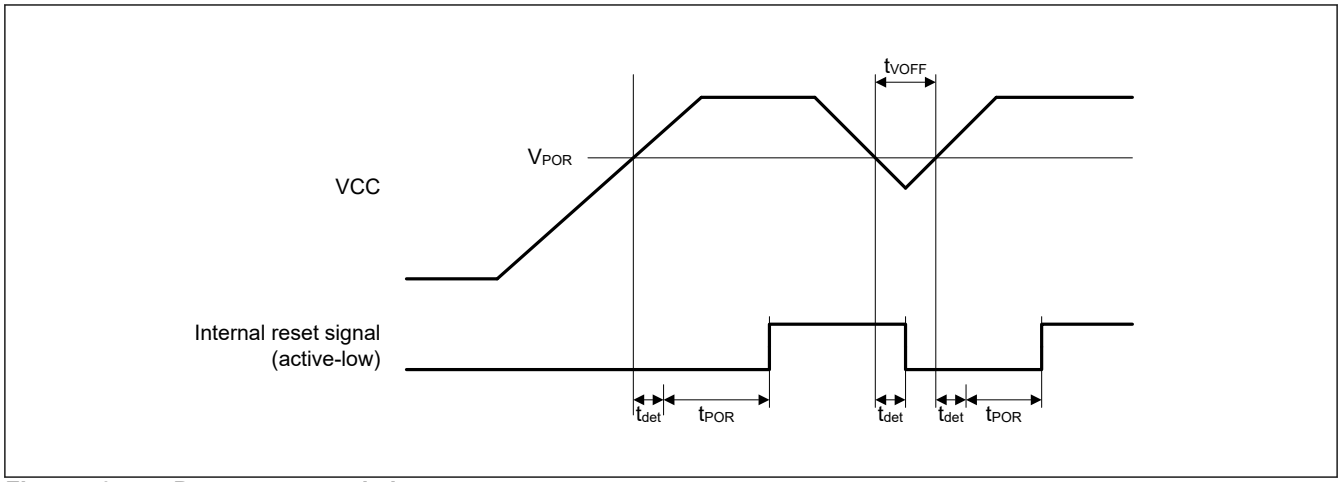


Figure 46.55 Power-on reset timing

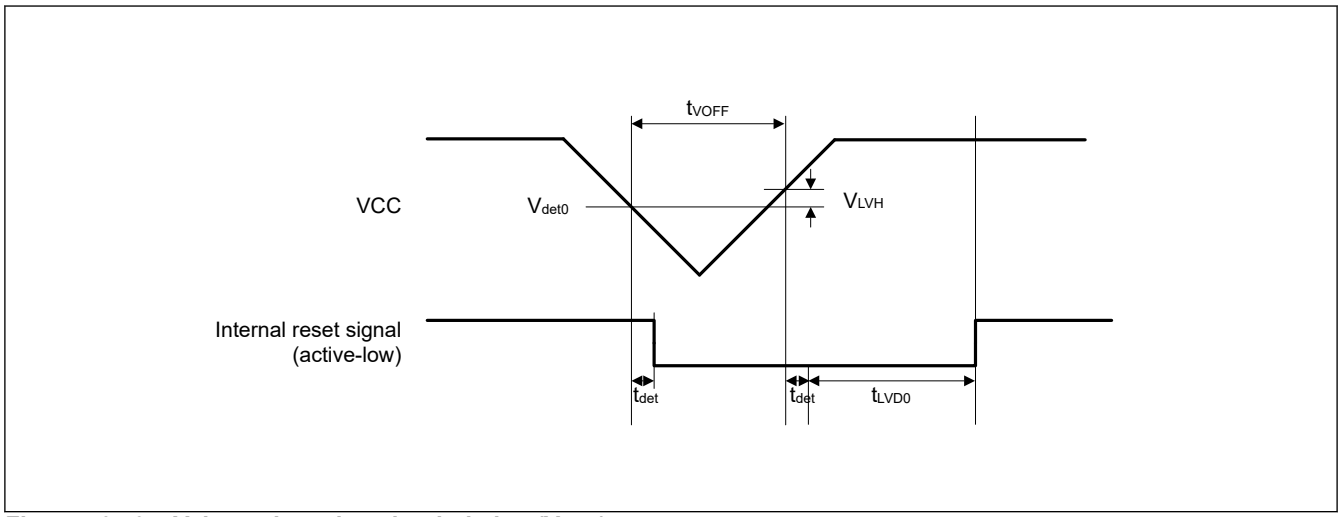


Figure 46.56 Voltage detection circuit timing (V_{det0})

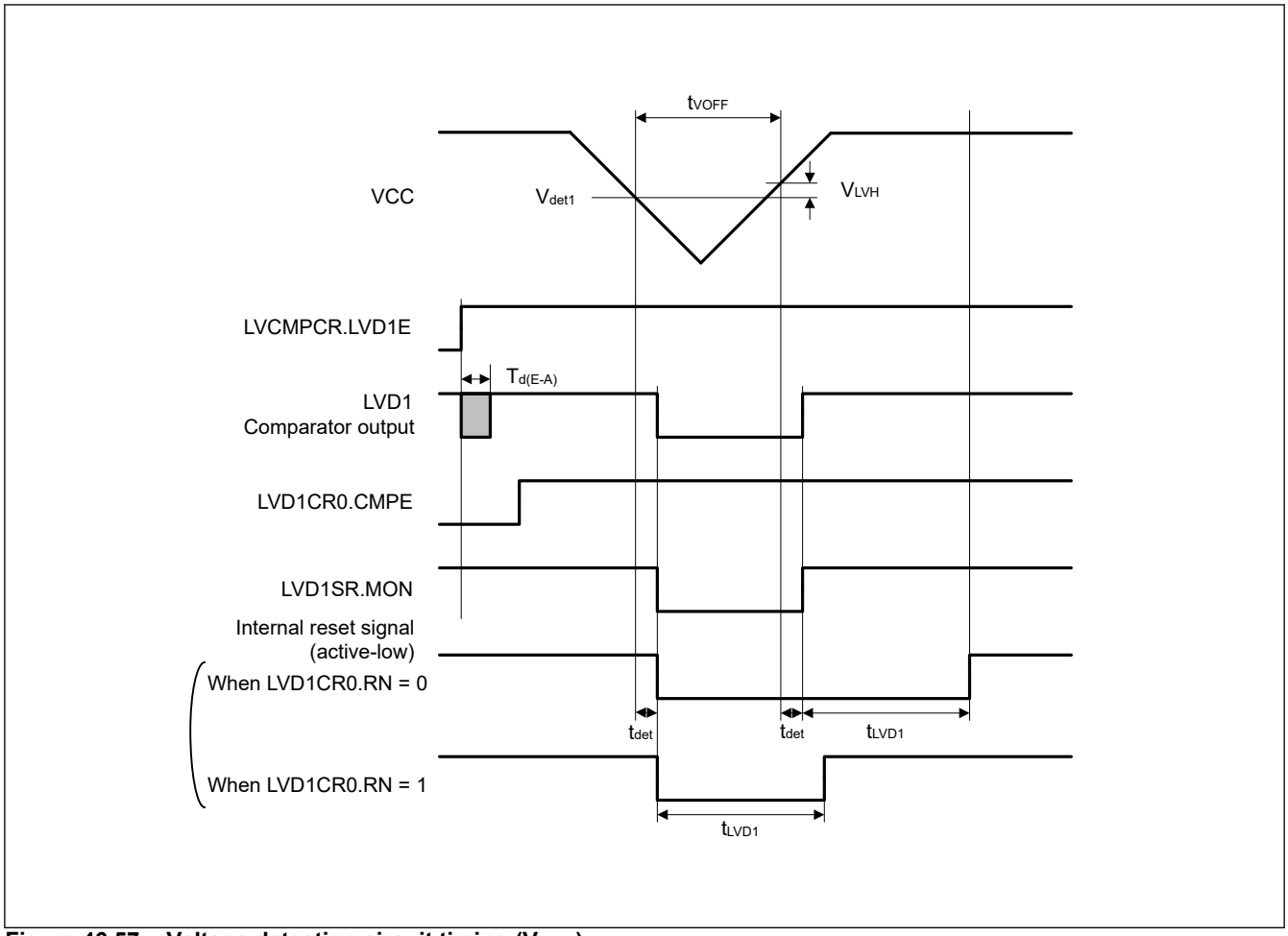


Figure 46.57 Voltage detection circuit timing (V_{det1})

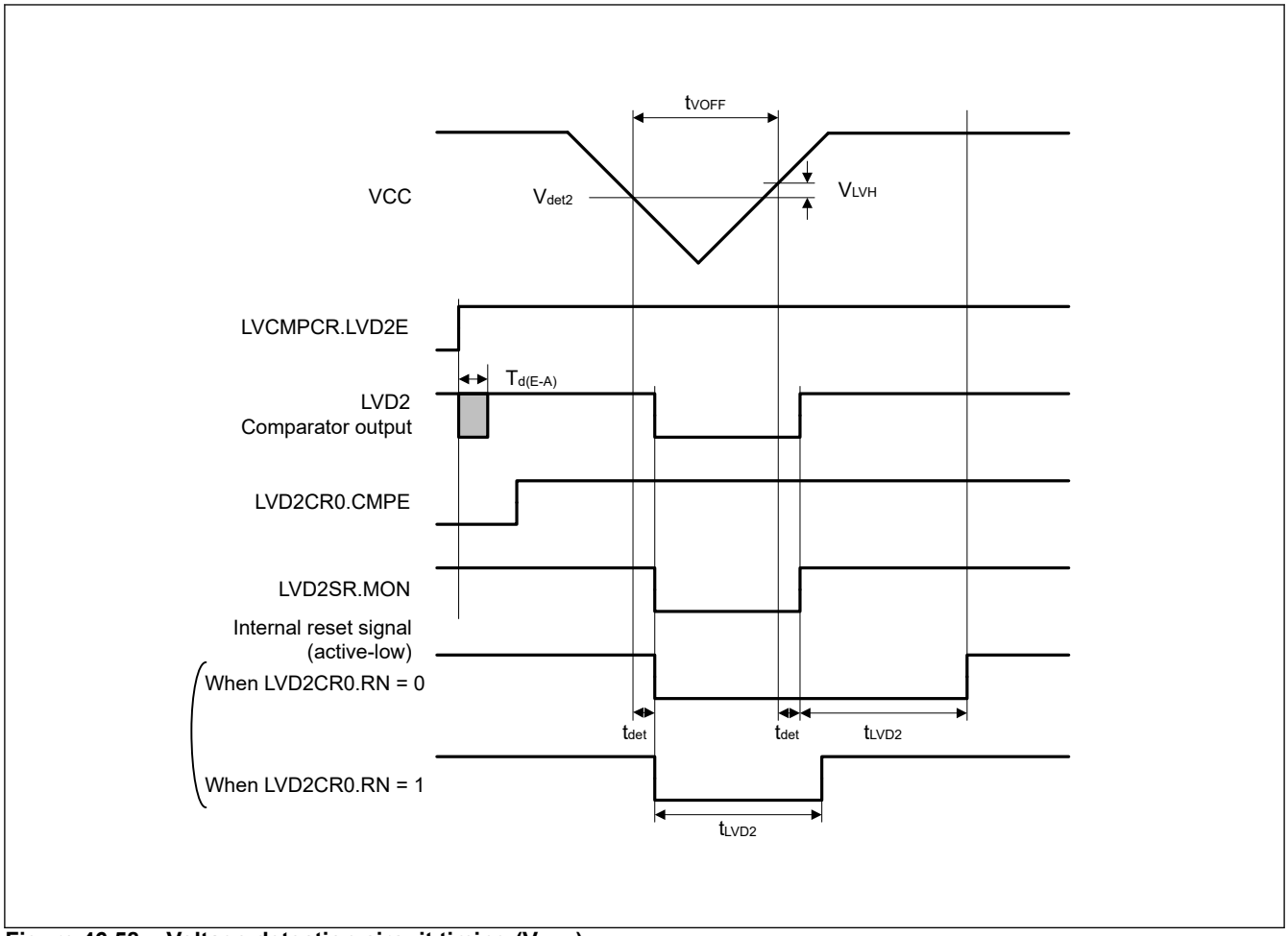


Figure 46.58 Voltage detection circuit timing (V_{det2})

46.11 Flash Memory Characteristics

46.11.1 Code Flash Memory Characteristics

Table 46.49 Code flash memory characteristics (1 of 2)

Conditions: Program or erase: FCLK = 4 to 60 MHz
Read: FCLK ≤ 60 MHz

Parameter		Symbol	FCLK = 4 MHz			20 MHz ≤ FCLK ≤ 60 MHz			Unit	Test conditions
			Min	Typ*6	Max	Min	Typ*6	Max		
Programming time N_{PEC} ≤ 100 times	128-byte	t_{P128}	—	0.75	13.2	—	0.34	6.0	ms	
	8-KB	t_{P8K}	—	49	176	—	22	80	ms	
	32-KB	t_{P32K}	—	194	704	—	88	320	ms	
Programming time N_{PEC} > 100 times	128-byte	t_{P128}	—	0.91	15.8	—	0.41	7.2	ms	
	8-KB	t_{P8K}	—	60	212	—	27	96	ms	
	32-KB	t_{P32K}	—	234	848	—	106	384	ms	
Erasure time N_{PEC} ≤ 100 times	8-KB	t_{E8K}	—	78	216	—	43	120	ms	
	32-KB	t_{E32K}	—	283	864	—	157	480	ms	
Erasure time N_{PEC} > 100 times	8-KB	t_{E8K}	—	94	260	—	52	144	ms	
	32-KB	t_{E32K}	—	341	1040	—	189	576	ms	
Reprogramming/erasure cycle*4		N_{PEC}	10000*1	—	—	10000*1	—	—	Times	

Table 46.49 Code flash memory characteristics (2 of 2)

Conditions: Program or erase: FCLK = 4 to 60 MHz
Read: FCLK ≤ 60 MHz

Parameter	Symbol	FCLK = 4 MHz			20 MHz ≤ FCLK ≤ 60 MHz			Unit	Test conditions
		Min	Typ*6	Max	Min	Typ*6	Max		
Suspend delay during programming	t_{SPD}	—	—	264	—	—	120	µs	
Programming resume time	t_{PRT}	—	—	110	—	—	50	µs	
First suspend delay during erasure in suspend priority mode	t_{SESD1}	—	—	216	—	—	120	µs	
Second suspend delay during erasure in suspend priority mode	t_{SESD2}	—	—	1.7	—	—	1.7	ms	
Suspend delay during erasure in erasure priority mode	t_{SEED}	—	—	1.7	—	—	1.7	ms	
First erasing resume time during erasure in suspend priority mode*5	t_{REST1}	—	—	1.7	—	—	1.7	ms	
Second erasing resume time during erasure in suspend priority mode	t_{REST2}	—	—	144	—	—	80	µs	
Erasing resume time during erasure in erasure priority mode	t_{REET}	—	—	144	—	—	80	µs	
Forced stop command	t_{FD}	—	—	32	—	—	20	µs	
Data hold time*2	t_{DRP}	10*2 *3	—	—	10*2 *3	—	—	Years	
		30*2 *3	—	—	30*2 *3	—	—		Ta = +85°C

Note 1. This is the minimum number of times to guarantee all the characteristics after reprogramming. The guaranteed range is from 1 to the minimum value.
Note 2. This indicates the minimum value of the characteristic when reprogramming is performed within the specified range.
Note 3. This result is obtained from reliability testing.
Note 4. The reprogram/erase cycle is the number of erasures for each block. When the reprogram/erase cycle is n times (n = 10,000), erasing can be performed n times for each block. For example, when 128-byte programming is performed 64 times for different addresses in 8-KB blocks, and then the entire block is erased, the reprogram/erase cycle is counted as one. However, programming the same address several times as one erasure is not enabled. Overwriting is prohibited.
Note 5. Time for resumption includes time for reapplying the erasing pulse (up to one full pulse) that was cut off at the time of suspension.
Note 6. The reference value at VCC = 3.3V and room temperature.

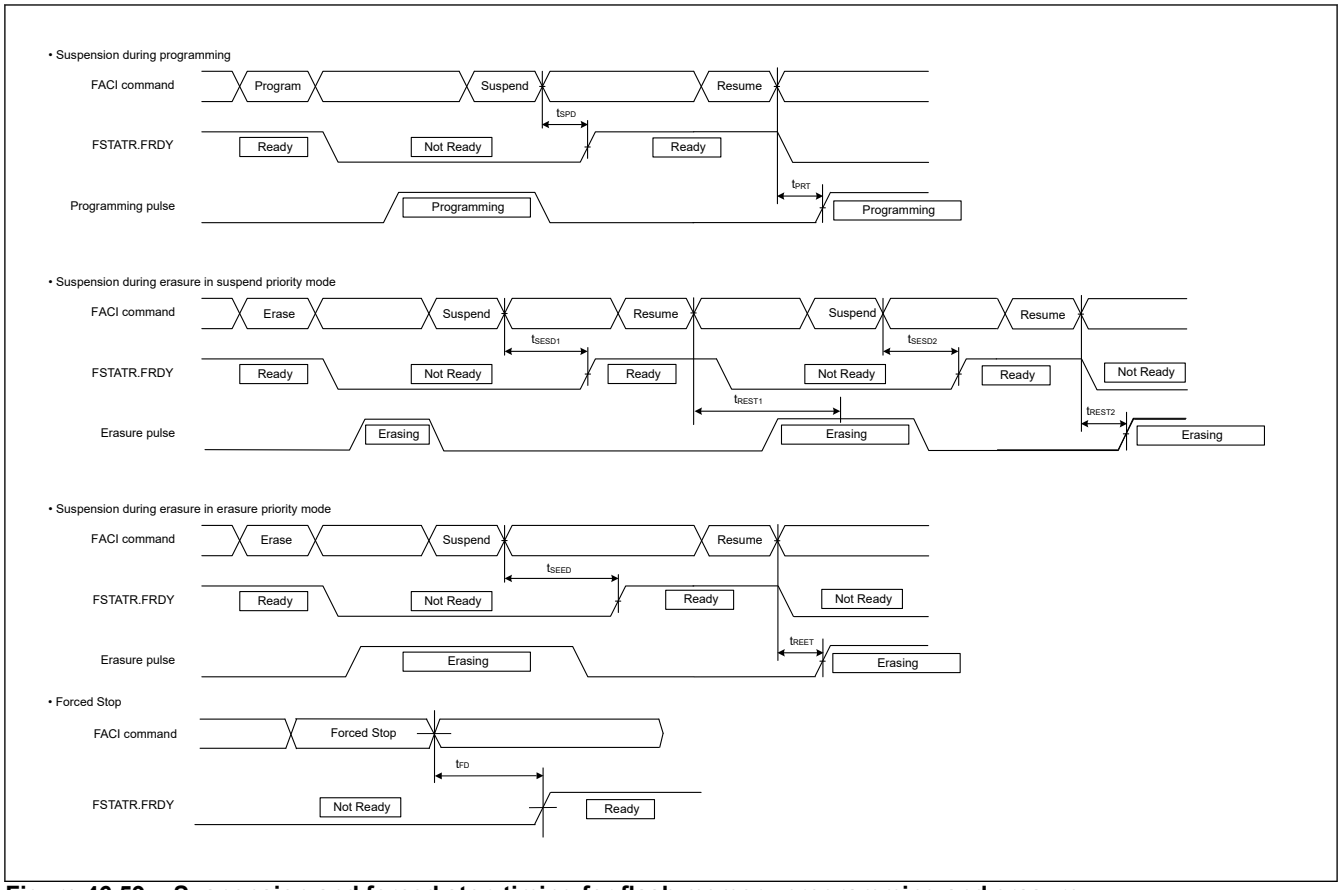


Figure 46.59 Suspension and forced stop timing for flash memory programming and erasure

46.11.2 Data Flash Memory Characteristics

Table 46.50 Data flash memory characteristics (1 of 2)

Conditions: Program or erase: FCLK = 4 to 60 MHz
Read: FCLK ≤ 60 MHz

Parameter		Symbol	FCLK = 4 MHz			20 MHz ≤ FCLK ≤ 60 MHz			Unit	Test conditions
			Min	Typ*6	Max	Min	Typ*6	Max		
Programming time	4-byte	t_{DP4}	—	0.36	3.8	—	0.16	1.7	ms	
	8-byte	t_{DP8}	—	0.38	4.0	—	0.17	1.8		
	16-byte	t_{DP16}	—	0.42	4.5	—	0.19	2.0		
Erasure time	64-byte	t_{DE64}	—	3.1	18	—	1.7	10	ms	
	128-byte	t_{DE128}	—	4.7	27	—	2.6	15		
	256-byte	t_{DE256}	—	8.9	50	—	4.9	28		
Blank check time	4-byte	t_{DBC4}	—	—	84	—	—	30	µs	
Reprogramming/erasure cycle*1		N_{DPEC}	125000*2	—	—	125000*2	—	—	—	
Suspend delay during programming	4-byte	t_{DSPD}	—	—	264	—	—	120	µs	
	8-byte		—	—	264	—	—	120		
	16-byte		—	—	264	—	—	120		
Programming resume time		t_{DPRT}	—	—	110	—	—	50	µs	
First suspend delay during erasure in suspend priority mode	64-byte	t_{DSESD1}	—	—	216	—	—	120	µs	
	128-byte		—	—	216	—	—	120		
	256-byte		—	—	216	—	—	120		

Table 46.50 Data flash memory characteristics (2 of 2)

Conditions: Program or erase: FCLK = 4 to 60 MHz
Read: FCLK ≤ 60 MHz

Parameter		Symbol	FCLK = 4 MHz			20 MHz ≤ FCLK ≤ 60 MHz			Unit	Test conditions
			Min	Typ*6	Max	Min	Typ*6	Max		
Second suspend delay during erasure in suspend priority mode	64-byte	t_{DSESD2}	—	—	300	—	—	300	μs	
	128-byte		—	—	390	—	—	390		
	256-byte		—	—	570	—	—	570		
Suspend delay during erasing in erasure priority mode	64-byte	t_{DSEED}	—	—	300	—	—	300	μs	
	128-byte		—	—	390	—	—	390		
	256-byte		—	—	570	—	—	570		
First erasing resume time during erasure in suspend priority mode*5		t_{DREST1}	—	—	300	—	—	300	μs	
Second erasing resume time during erasure in suspend priority mode		t_{DREST2}	—	—	126	—	—	70	μs	
Erasing resume time during erasure in erasure priority mode		t_{DREET}	—	—	126	—	—	70	μs	
Forced stop command		t_{FD}	—	—	32	—	—	20	μs	
Data hold time*3		t_{DRP}	10*3 *4	—	—	10*3 *4	—	—	Year	
			30*3 *4	—	—	30*3 *4	—	—		Ta = +85°C

Note 1. The reprogram/erase cycle is the number of erasures for each block. When the reprogram/erase cycle is n times (n = 125,000), erasing can be performed n times for each block. For example, when 4-byte programming is performed 16 times for different addresses in 64-byte blocks, and then the entire block is erased, the reprogram/erase cycle is counted as one. However, programming the same address several times as one erasure is not enabled. Overwriting is prohibited.
Note 2. This is the minimum number of times to guarantee all the characteristics after reprogramming. The guaranteed range is from 1 to the minimum value.
Note 3. This indicates the minimum value of the characteristic when reprogramming is performed within the specified range.
Note 4. This result is obtained from reliability testing.
Note 5. Time for resumption includes time for reapplying the erasing pulse (up to one full pulse) that was cut off at the time of suspension.
Note 6. The reference value at VCC = 3.3 V and room temperature.

46.11.3 Option Setting Memory Characteristics

Table 46.51 Option setting memory characteristics

Conditions: Program: FCLK = 4 to 60 MHz
Read: FCLK ≤ 60 MHz

Parameter	Symbol	FCLK = 4 MHz			20 MHz ≤ FCLK ≤ 60 MHz			Unit	Test conditions
		Min	Typ*4	Max	Min	Typ*4	Max		
Programming time N_{OPC} ≤ 100 times	t_{OP}	—	83	309	—	45	162	ms	
Programming time N_{OPC} > 100 times	t_{OP}	—	100	371	—	55	195	ms	
Reprogramming cycle	N_{OPC}	20000*1	—	—	20000*1	—	—	Times	
Data hold time*2	t_{DRP}	10*2 *3	—	—	10*2 *3	—	—	Years	
		30*2 *3	—	—	30*2 *3	—	—		Ta = +85°C

Note 1. This is the minimum number of times to guarantee all the characteristics after reprogramming. The guaranteed range is from 1 to the minimum value.
Note 2. This indicates the minimum value of the characteristic when reprogramming is performed within the specified range.
Note 3. This result is obtained from reliability testing.
Note 4. The reference value at VCC = 3.3 V and room temperature.

46.12 Boundary Scan

Table 46.52 Boundary scan characteristics

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
TCK clock cycle time	t_{TCKcyc}	100	—	—	ns	Figure 46.60
TCK clock high pulse width	t_{TCKH}	45	—	—	ns	
TCK clock low pulse width	t_{TCKL}	45	—	—	ns	
TCK clock rise time	t_{TCKr}	—	—	5	ns	
TCK clock fall time	t_{TCKf}	—	—	5	ns	
TMS setup time	t_{TMSS}	20	—	—	ns	Figure 46.61
TMS hold time	t_{TMSH}	20	—	—	ns	
TDI setup time	t_{TDIS}	20	—	—	ns	
TDI hold time	t_{TDIH}	20	—	—	ns	
TDO data delay	t_{TDOD}	—	—	40	ns	
Boundary scan circuit startup time*1	T_{BSSTUP}	t_{RESWP}	—	—	—	Figure 46.62

Note 1. Boundary scan does not function until the power-on reset becomes negative.

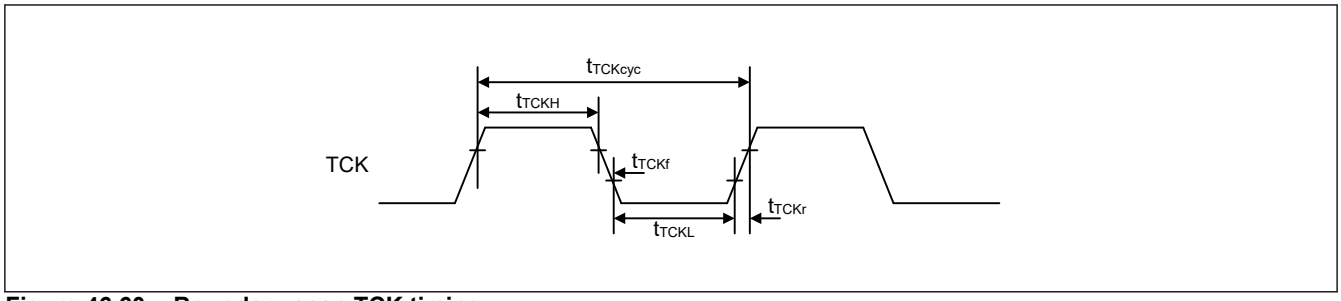


Figure 46.60 Boundary scan TCK timing

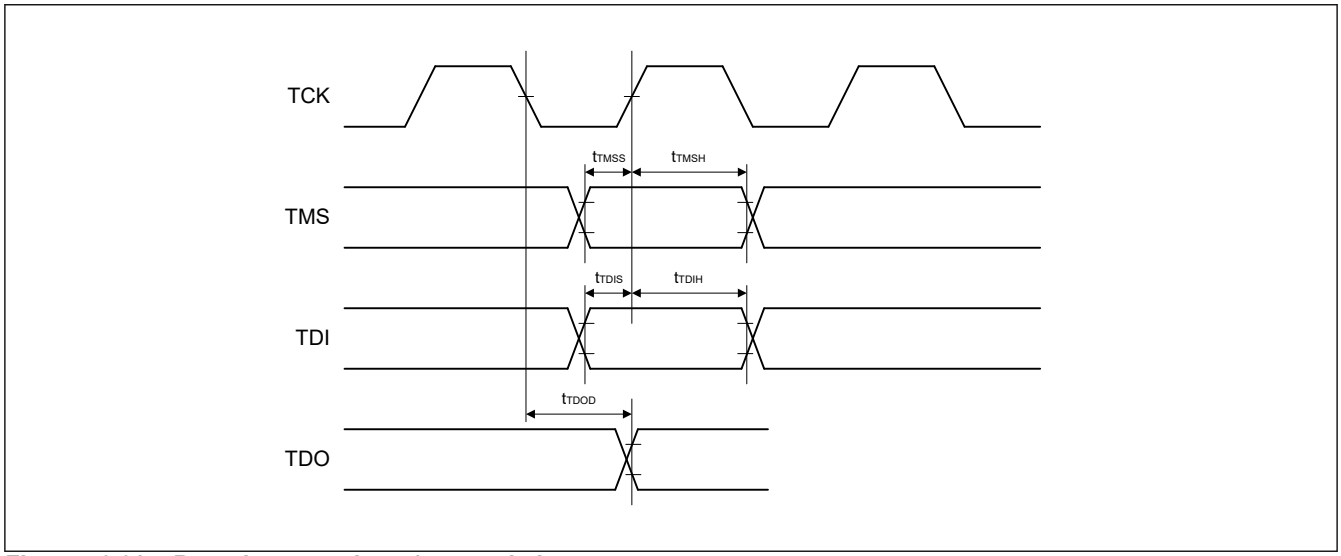


Figure 46.61 Boundary scan input/output timing

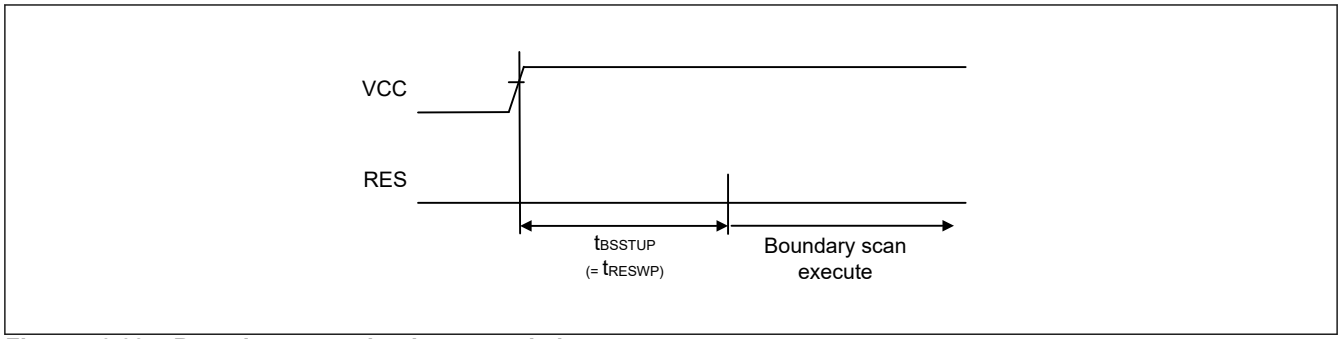


Figure 46.62 Boundary scan circuit startup timing

46.13 Joint Test Action Group (JTAG)

Table 46.53 JTAG

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
TCK clock cycle time	t_{TCKcyc}	40	—	—	ns	Figure 46.63
TCK clock high pulse width	t_{TCKH}	15	—	—	ns	
TCK clock low pulse width	t_{TCKL}	15	—	—	ns	
TCK clock rise time	t_{TCKr}	—	—	5	ns	
TCK clock fall time	t_{TCKf}	—	—	5	ns	
TMS setup time	t_{TMSS}	8	—	—	ns	Figure 46.64
TMS hold time	t_{TMSH}	8	—	—	ns	
TDI setup time	t_{TDIS}	8	—	—	ns	
TDI hold time	t_{TDIH}	8	—	—	ns	
TDO data delay time	t_{TDOD}	—	—	20	ns	

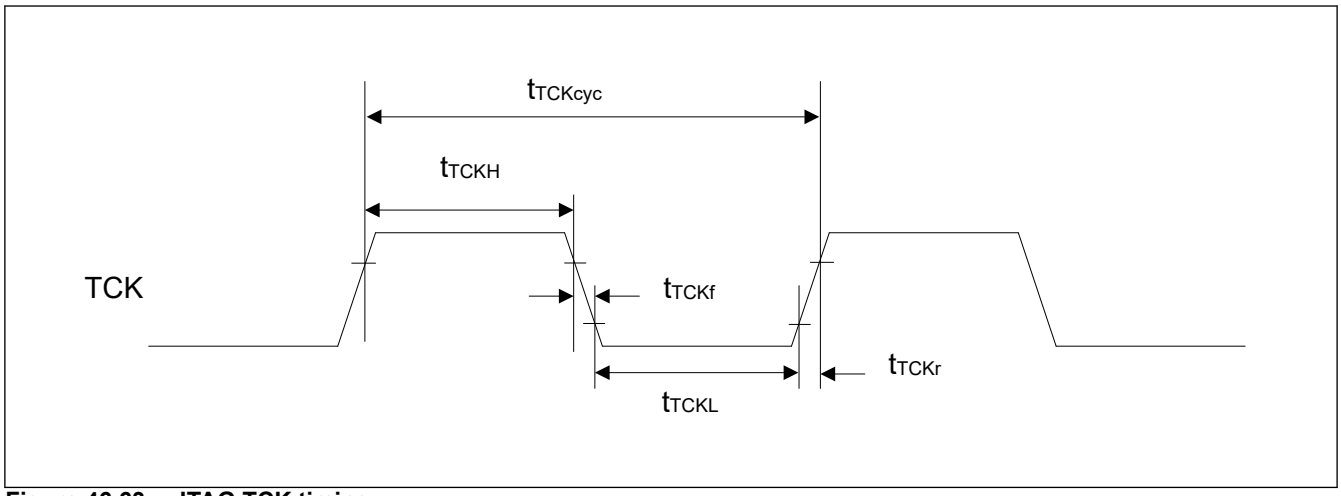


Figure 46.63 JTAG TCK timing

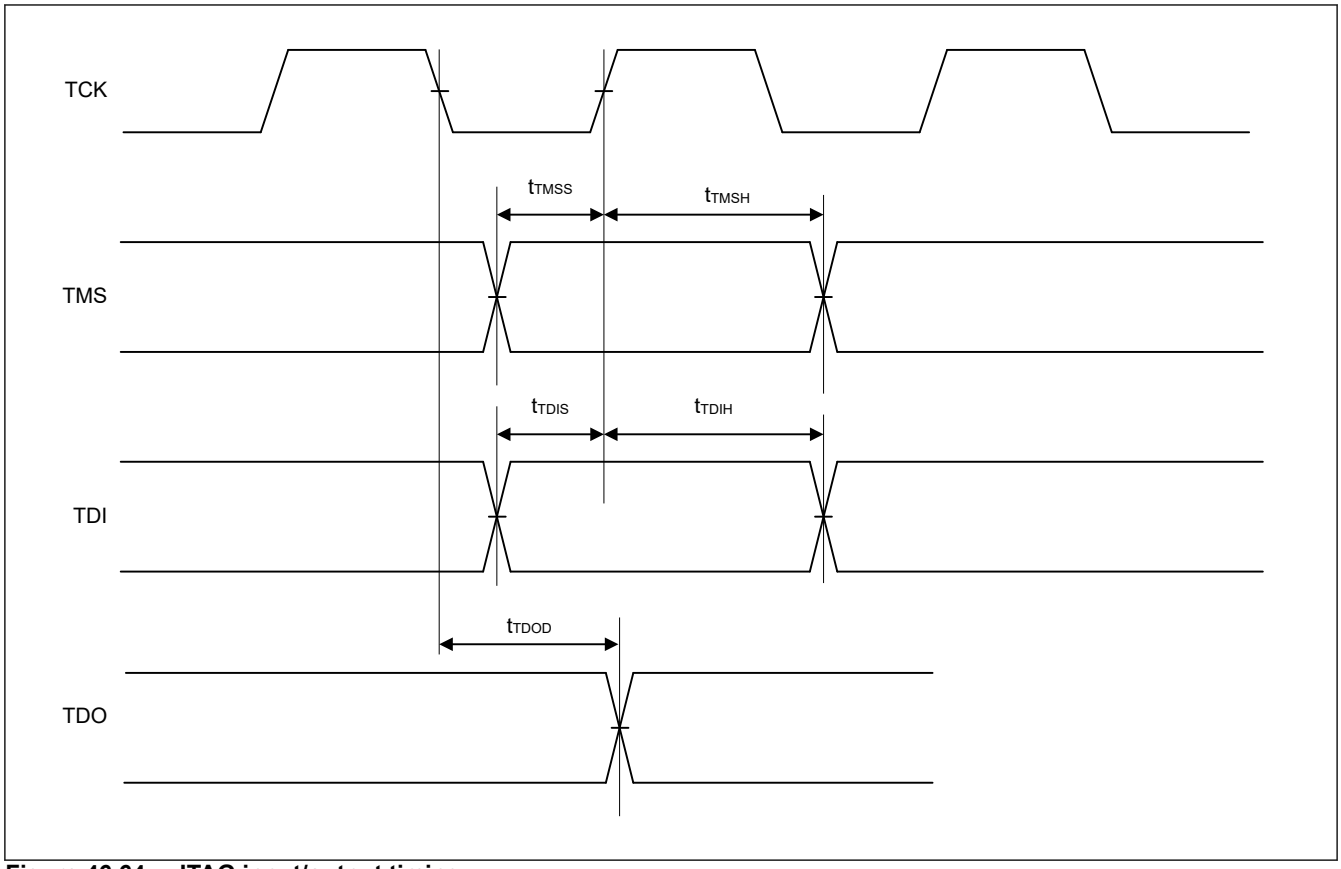


Figure 46.64 JTAG input/output timing

46.14 Serial Wire Debug (SWD)

Table 46.54 SWD

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
SWCLK clock cycle time	$t_{SWCKcyc}$	40	—	—	ns	Figure 46.65
SWCLK clock high pulse width	t_{SWCKH}	15	—	—	ns	
SWCLK clock low pulse width	t_{SWCKL}	15	—	—	ns	
SWCLK clock rise time	t_{SWCKr}	—	—	5	ns	
SWCLK clock fall time	t_{SWCKf}	—	—	5	ns	
SWDIO setup time	t_{SWDS}	8	—	—	ns	Figure 46.66
SWDIO hold time	t_{SWDH}	8	—	—	ns	
SWDIO data delay time	t_{SWDD}	2	—	28	ns	

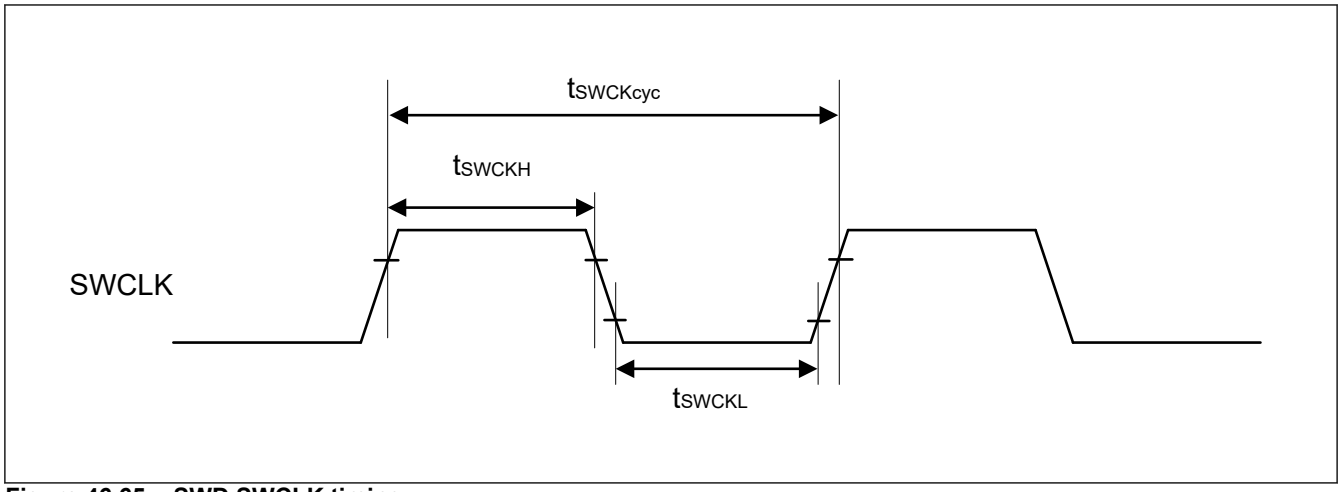


Figure 46.65 SWD SWCLK timing

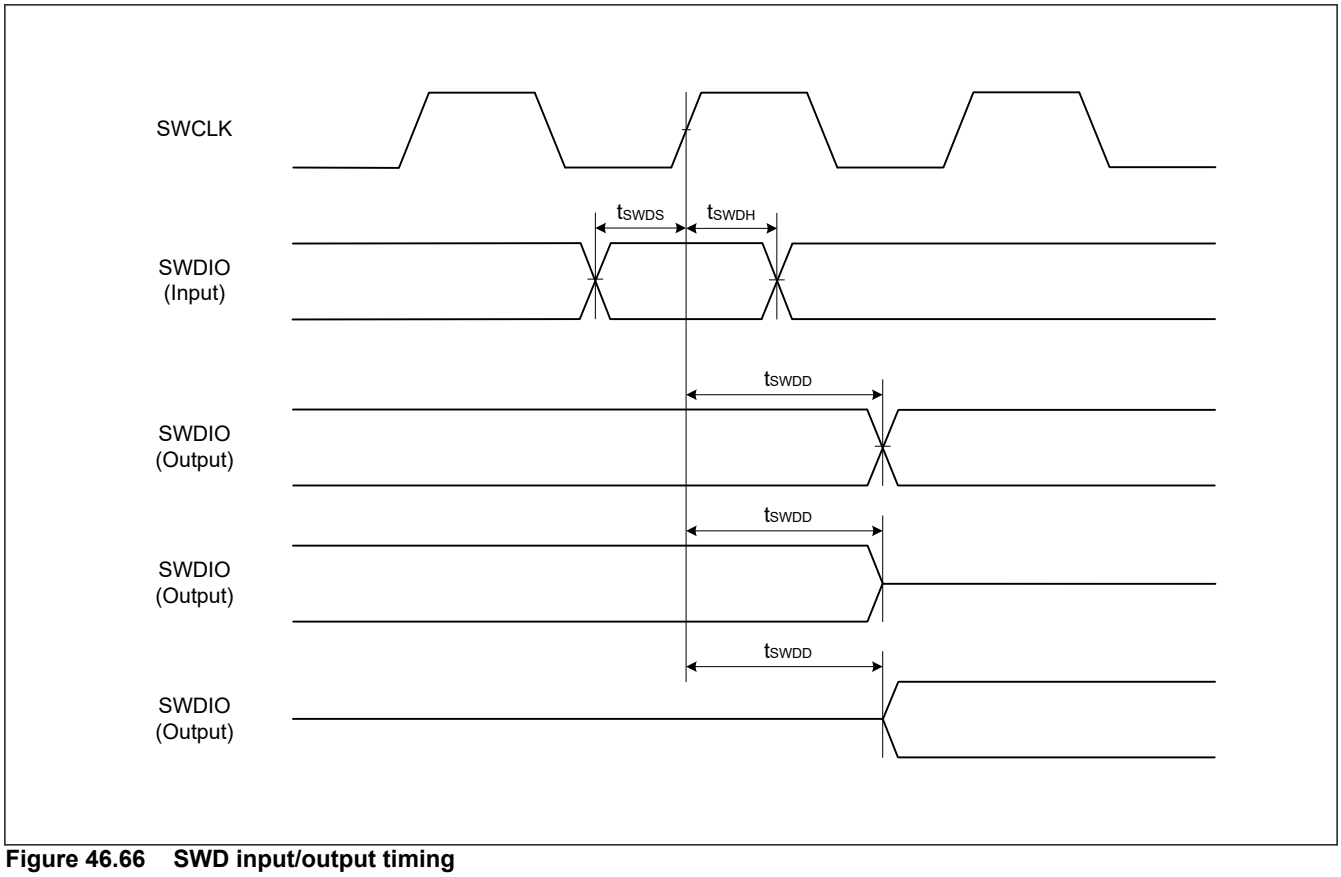


Figure 46.66 SWD input/output timing

46.15 Embedded Trace Macro Interface (ETM)

Table 46.55 ETM (1 of 2)

Conditions: High speed high drive output is selected in the Port Drive Capability bit in the PmnPFS register.

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
TCLK clock cycle time	$t_{TCLKcyc}$	16.7	—	—	ns	Figure 46.67
TCLK clock high pulse width	t_{TCLKH}	7.35	—	—	ns	
TCLK clock low pulse width	t_{TCLKL}	7.35	—	—	ns	
TCLK clock rise time	t_{TCLKr}	—	—	1	ns	
TCLK clock fall time	t_{TCLKf}	—	—	1	ns	

Table 46.55 ETM (2 of 2)

Conditions: High speed high drive output is selected in the Port Drive Capability bit in the PmnPFS register.

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
TDATA[3:0] output setup time	t_{TRDS}	2.5	—	—	ns	Figure 46.68
TDATA[3:0] output hold time	t_{TRDH}	1.5	—	—	ns	

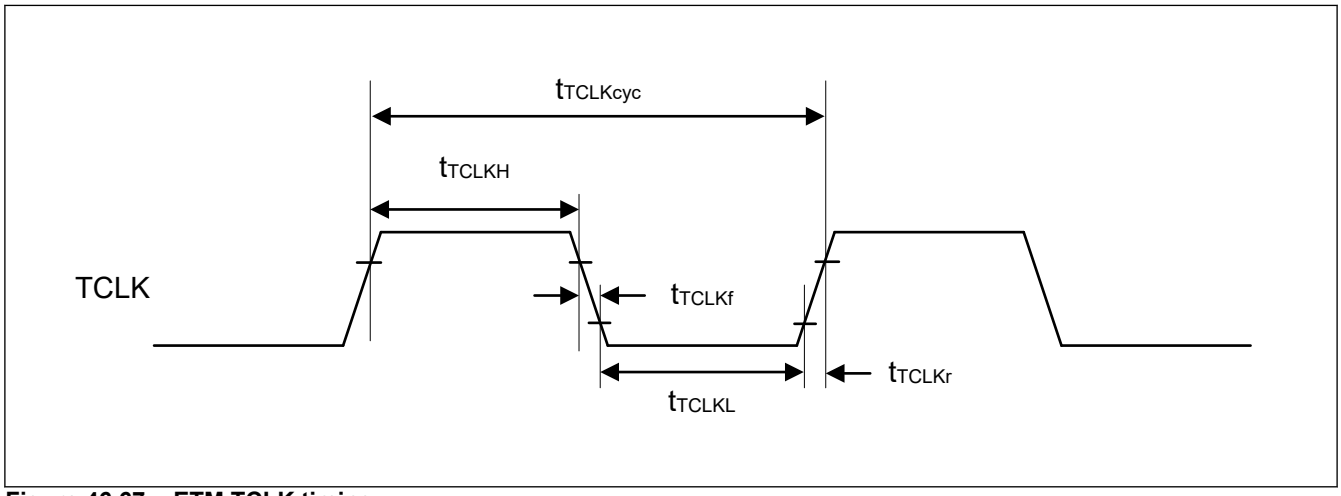


Figure 46.67 ETM TCLK timing

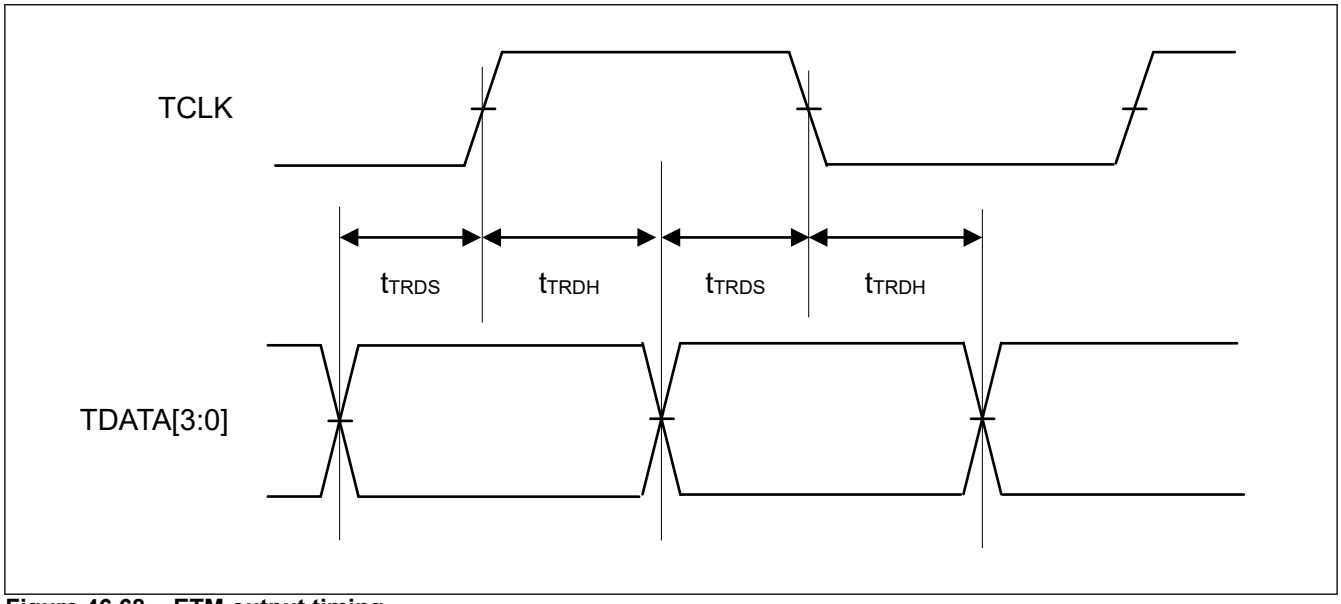


Figure 46.68 ETM output timing

Appendix 1. Port States in Each Processing Mode

Function	Pin function	Reset	Software Standby mode	Deep Software Standby mode	After Deep Software Standby mode is canceled (return to startup mode)	
					IOKEEP = 0	IOKEEP = 1[*1]
Mode	MD	Pull-up	Keep-O	Keep	Hi-Z	Keep
JTAG	TCK/TMS/TDI	Pull-up	Keep-O	Keep	Hi-Z	Keep
	TDO	TDO output	Keep-O	Keep	TDO output	Keep
Trace	TCLK/TDATAx	Hi-Z	Keep-O	Keep	Hi-Z	Keep
IRQ	IRQx	Hi-Z	Keep-O[*2]	Keep	Hi-Z	Keep
	IRQx-DS	Hi-Z	Keep-O[*2]	Keep[*3]	Hi-Z	Keep
KINT	KRxx	Hi-Z	Keep-O[*2]	Keep	Hi-Z	Keep
AGT	AGTIOn	Hi-Z	Keep-O[*2]	Keep	Hi-Z	Keep
IIC	SCLn/SDAn	Hi-Z	Keep-O[*2]	Keep	Hi-Z	Keep
CLKOUT	CLKOUT	Hi-Z	[CLKOUT selected] CLKOUT output	Keep	Hi-Z	Keep
ACMPHS	VCOUT, CMPOUTm, CMPOUT012	Hi-Z	Hi-Z (Keep-O)	Hi-Z (Keep-O)	Hi-Z	Keep
	IVREFn	Hi-Z	Hi-Z (Keep-O)	Hi-Z (Keep-O)	Hi-Z	Hi-Z
	IVCMPmn	Hi-Z	Hi-Z (Keep-O)	Hi-Z (Keep-O)	Hi-Z	Hi-Z
DAC12	DAn	Hi-Z	[DAn output (DAOE = 1)] D/A output retained	Keep	Hi-Z	Keep
ADC	ANxxx	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z
	PGAINn	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z
	PGAVSSn	Pull-up[*4]	Pull-up[*5] / Keep	Pull-up[*5] / Keep	Pull-up[*5] / Keep	Pull-up[*5] / Keep
	PGAOUTn	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z
Others	—	Hi-Z	Keep-O	Keep	Hi-Z	Keep

Note: H: High-level
L: Low-level
Hi-Z: High-impedance
Keep-O: Output pins retain their previous values. Input pins go to high-impedance.
Keep: Pin states are retained during periods in Software Standby mode.

Note 1. Retains the I/O port state until the DPSBYCR.IOKEEP bit is cleared to 0.

Note 2. Input is enabled if the pin is specified as the Software Standby canceling source while it is used as an external interrupt pin.

Note 3. Input is enabled if the pin is specified as the Deep Software Standby canceling source.

Note 4. The built-in pull-up is turned on to protect the circuit from negative potential inputs.

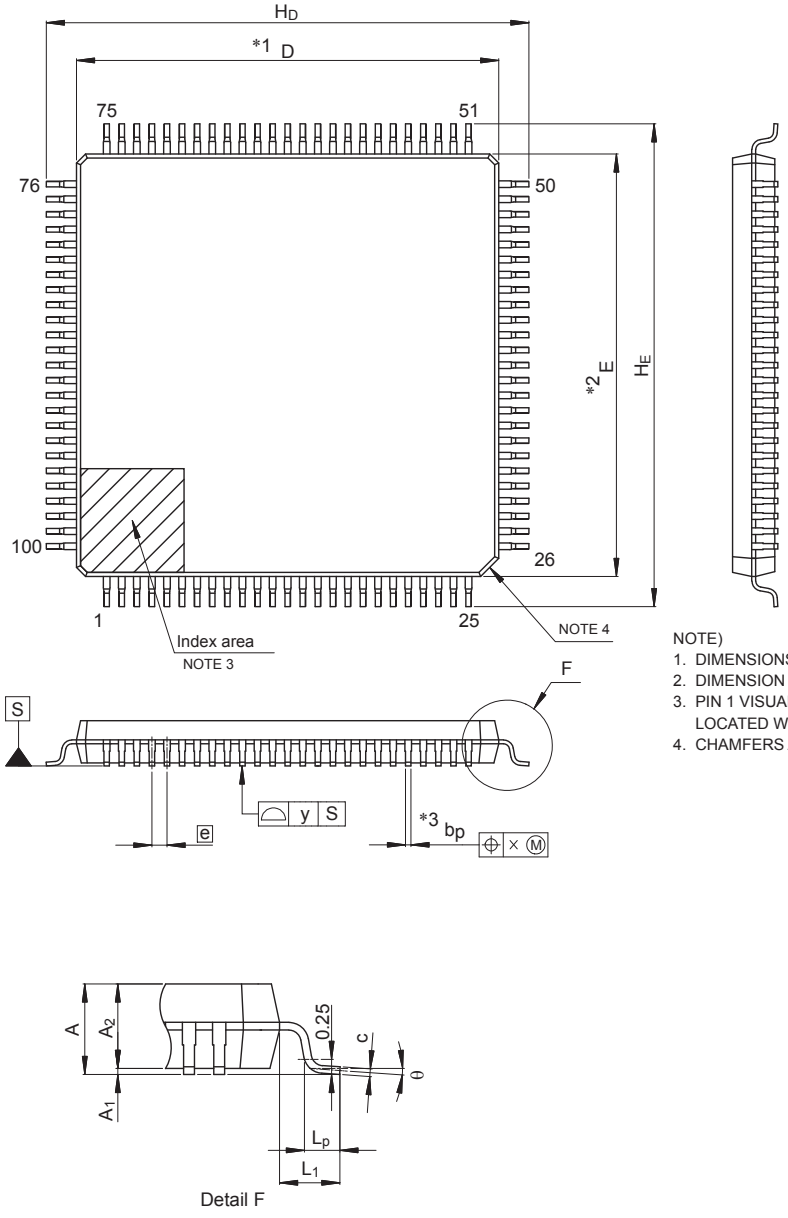
Note 5. Regardless of whether the PGA is enabled or disabled, when the PGA is set to pseudo-differential mode, the built in pull-up is turned on to protect the circuit from negative potential inputs. To turn off the built-in pull-up, turn off the PGA's pseudo-differential mode and set it to single mode.

Appendix 2. Package Dimensions

Information on the latest version of the package dimensions or mountings is displayed in Packages on the Renesas Electronics Corporation website.

JEITA Package Code	RENESAS Code	Previous Code	MASS (Typ) [g]
P-LFQFP100-14x14-0.50	PLQP0100KB-B	—	0.6

Unit: mm



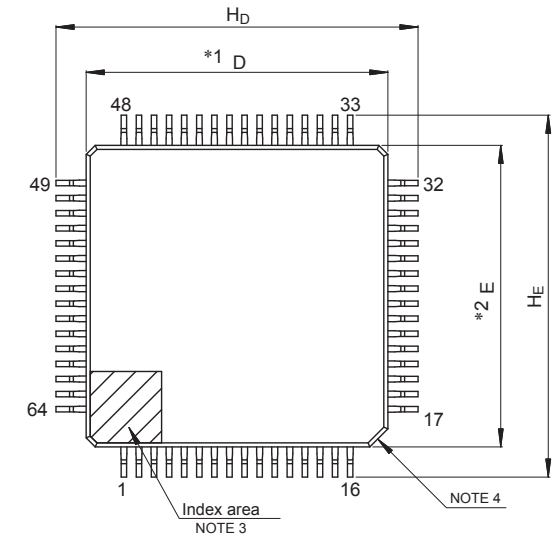
NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.
3. PIN 1 VISUAL INDEX FEATURE MAY VARY, BUT MUST BE LOCATED WITHIN THE HATCHED AREA.
4. CHAMFERS AT CORNERS ARE OPTIONAL, SIZE MAY VARY.

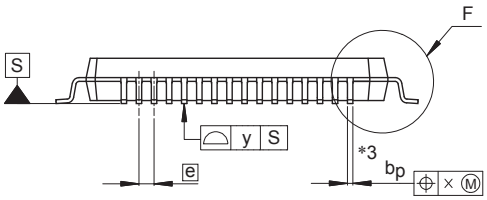
Reference Symbol	Dimensions in millimeters		
	Min	Nom	Max
D	13.9	14.0	14.1
E	13.9	14.0	14.1
A_2	—	1.4	—
H_D	15.8	16.0	16.2
H_E	15.8	16.0	16.2
A	—	—	1.7
A_1	0.05	—	0.15
b_p	0.15	0.20	0.27
c	0.09	—	0.20
θ	0°	3.5°	8°
e	—	0.5	—
x	—	—	0.08
y	—	—	0.08
L_p	0.45	0.6	0.75
L_1	—	1.0	—

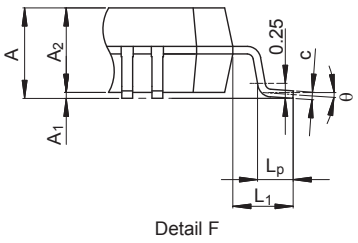
Figure 2.1 LQFP 100-pin

JEITA Package Code	RENESAS Code	Previous Code	MASS (Typ) [g]
P-LFQFP64-10x10-0.50	PLQP0064KB-C	—	0.3

Unit: mm







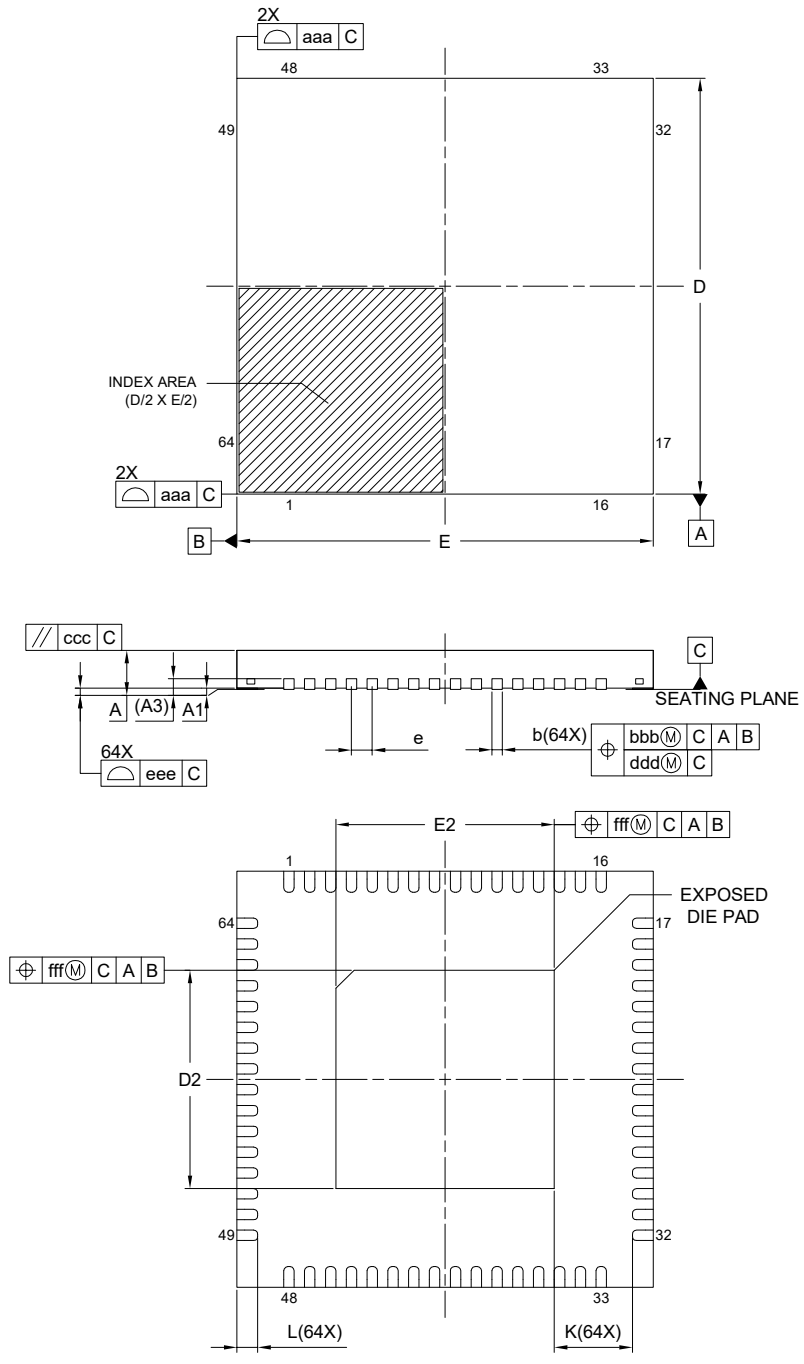
Detail F

NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.
3. PIN 1 VISUAL INDEX FEATURE MAY VARY, BUT MUST BE LOCATED WITHIN THE HATCHED AREA.
4. CHAMFERS AT CORNERS ARE OPTIONAL, SIZE MAY VARY.

Reference Symbol	Dimensions in millimeters		
	Min	Nom	Max
D	9.9	10.0	10.1
E	9.9	10.0	10.1
A_2	—	1.4	—
H_D	11.8	12.0	12.2
H_E	11.8	12.0	12.2
A	—	—	1.7
A_1	0.05	—	0.15
b_p	0.15	0.20	0.27
c	0.09	—	0.20
θ	0°	3.5°	8°
e	—	0.5	—
x	—	—	0.08
y	—	—	0.08
L_p	0.45	0.6	0.75
L_1	—	1.0	—

Figure 2.2 LQFP 64-pin

JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-HWQFN064-8x8-0.40	PWQN0064LB-A	0.18

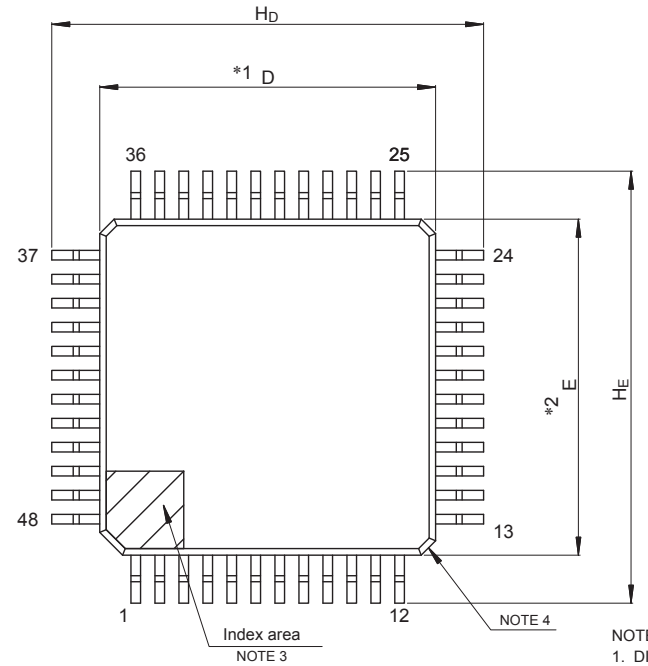


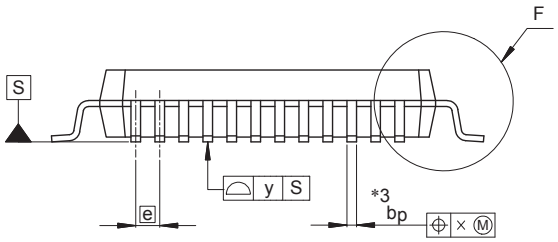
Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
A	—	—	0.80
A_1	0.00	0.02	0.05
A_3		0.203 REF.	
b	0.15	0.20	0.25
D		8.00 BSC	
E		8.00 BSC	
e		0.40 BSC	
L	0.35	0.40	0.45
K	0.20	—	—
D_2	4.15	4.20	4.25
E_2	4.15	4.20	4.25
aaa		0.10	
bbb		0.07	
ccc		0.10	
ddd		0.05	
eee		0.08	
fff		0.10	

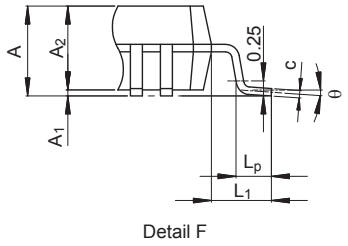
Figure 2.3 QFN 64-pin

JEITA Package Code	RENESAS Code	Previous Code	MASS (Typ) [g]
P-LFQFP48-7x7-0.50	PLQP0048KB-B	—	0.2

Unit: mm







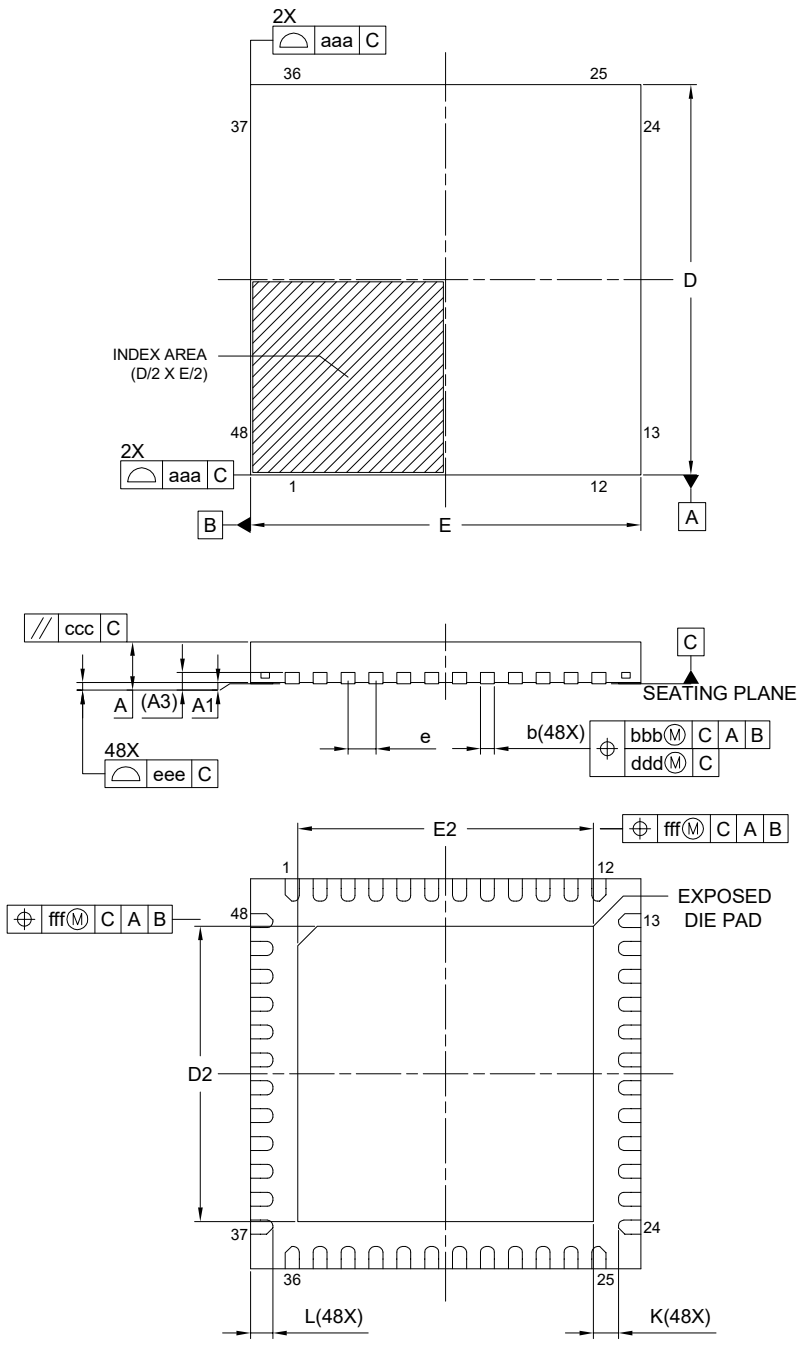
Detail F

NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.
3. PIN 1 VISUAL INDEX FEATURE MAY VARY, BUT MUST BE LOCATED WITHIN THE HATCHED AREA.
4. CHAMFERS AT CORNERS ARE OPTIONAL, SIZE MAY VARY.

Reference Symbol	Dimensions in millimeters		
	Min	Nom	Max
D	6.9	7.0	7.1
E	6.9	7.0	7.1
A_2	—	1.4	—
H_D	8.8	9.0	9.2
H_E	8.8	9.0	9.2
A	—	—	1.7
A_1	0.05	—	0.15
b_p	0.17	0.20	0.27
c	0.09	—	0.20
θ	0°	3.5°	8°
e	—	0.5	—
x	—	—	0.08
y	—	—	0.08
L_p	0.45	0.6	0.75
L_1	—	1.0	—

Figure 2.4 LQFP 48-pin

JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-HWQFN048-7x7-0.50	PWQN0048KC-A	0.13 g



Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
A	—	—	0.80
A_1	0.00	0.02	0.05
A_3		0.203 REF.	
b	0.20	0.25	0.30
D		7.00 BSC	
E		7.00 BSC	
e		0.50 BSC	
L	0.30	0.40	0.50
K	0.20	—	—
D_2	5.25	5.30	5.35
E_2	5.25	5.30	5.35
aaa		0.15	
bbb		0.10	
ccc		0.10	
ddd		0.05	
eee		0.08	
fff		0.10	

Figure 2.5 QFN 48-pin

Appendix 3. I/O Registers

This appendix describes I/O register address and access cycles by function.

3.1 Peripheral Base Addresses

This section provides the base addresses for peripherals described in this manual. Table 3.1 shows the name, description, and the base address of each peripheral.

Table 3.1 Peripheral base address (1 of 3)

Name	Description	Base address
RMPU	Renesas Memory Protection Unit	0x4000_0000
TZF	TrustZone Filter	0x4000_0E00
SRAM	SRAM Control	0x4000_2000
BUS	BUS Control	0x4000_3000
DMAC0	Direct memory access controller 0	0x4000_5000
DMAC1	Direct memory access controller 1	0x4000_5040
DMAC2	Direct memory access controller 2	0x4000_5080
DMAC3	Direct memory access controller 3	0x4000_50C0
DMAC4	Direct memory access controller 4	0x4000_5100
DMAC5	Direct memory access controller 5	0x4000_5140
DMAC6	Direct memory access controller 6	0x4000_5180
DMAC7	Direct memory access controller 7	0x4000_51C0
DMA	DMAC Module Activation	0x4000_5200
DTC	Data Transfer Controller	0x4000_5400
ICU	Interrupt Controller	0x4000_6000
CACHE	CACHE	0x4000_7000
CPSCU	CPU System Security Control Unit	0x4000_8000
DBG	Debug Function	0x4001_B000
FCACHE	Flash Cache	0x4001_C100
SYSC	System Control	0x4001_E000
PORT0	Port 0 Control Registers	0x4001_F000
PORT2	Port 2 Control Registers	0x4001_F040
PORTA	Port A Control Registers	0x4001_F140
PORTB	Port B Control Registers	0x4001_F160
PORTC	Port C Control Registers	0x4001_F180
PORTD	Port D Control Registers	0x4001_F1A0
PORTE	Port E Control Registers	0x4001_F1C0
PFS_B	Pmn Pin Function Control Register	0x4001_F800
IIRFA	IIR Filter Accelerator	0x4002_0000
TFU	Trigonometric Function Unit	0x4002_1000
ELC_B	Event Link Controller	0x4008_2000
IWDT	Independent Watchdog Timer	0x4008_3200
WDT	Watchdog Timer	0x4008_3400
CAC	Clock Frequency Accuracy Measurement Circuit	0x4008_3600
MSTP	Module Stop Control A, B, C, D, E	0x4008_4000

Table 3.1 Peripheral base address (2 of 3)

Name	Description	Base address
KINT	Key Interrupt Function	0x4008_5000
POEG	Port Output Enable for GPT	0x4008_A000
CANFD_B	CANFD Module Control	0x400B_0000
PSCU	Peripheral Security Control Unit	0x400E_0000
AGTW_B0	Low Power Asynchronous General purpose Timer 0	0x400E_8000
AGTW_B1	Low Power Asynchronous General purpose Timer 1	0x400E_8100
TSN	Temperature Sensor	0x400F_3000
ACMPHS0	High-Speed Analog Comparator	0x400F_4000
ACMPHS1	High-Speed Analog Comparator	0x400F_4100
ACMPHS2	High-Speed Analog Comparator	0x400F_4200
ACMPHS3	High-Speed Analog Comparator	0x400F_4300
CRC	Cyclic Redundancy Check	0x4010_8000
DOC_B	Data Operation Circuit	0x4010_9000
SCI_B0	Serial Communication Interface 0	0x4011_8000
SCI_B1	Serial Communication Interface 1	0x4011_8100
SCI_B2	Serial Communication Interface 2	0x4011_8200
SCI_B3	Serial Communication Interface 3	0x4011_8300
SCI_B4	Serial Communication Interface 4	0x4011_8400
SCI_B9	Serial Communication Interface 9	0x4011_8900
SPI_B0	Serial Peripheral Interface 0	0x4011_A000
SPI_B1	Serial Peripheral Interface 1	0x4011_A100
IIC_B0	Inter-Integrated Circuit 0	0x4011_F000
IIC0WU_B	Inter-Integrated Circuit 0 Wake-up Unit	0x4011_F098
IIC_B1	Inter-Integrated Circuit 1	0x4011_F400
ECCMB	CANFD ECC Module	0x4012_F200
SCE5_B	Secure Cryptographic Engine	0x4016_1000
GPT320	General PWM Timer 0	0x4016_9000
GPT321	General PWM Timer 1	0x4016_9100
GPT322	General PWM Timer 2	0x4016_9200
GPT323	General PWM Timer 3	0x4016_9300
GPT324	General PWM Timer 4	0x4016_9400
GPT325	General PWM Timer 5	0x4016_9500
GPT326	General PWM Timer 6	0x4016_9600
GPT327	General PWM Timer 7	0x4016_9700
GPT328	General PWM Timer 8	0x4016_9800
GPT329	General PWM Timer 9	0x4016_9900
GPT_OPS	Output Phase Switching Controller	0x4016_9A00
GPT_GTCLK	General PWM Timer	0x4016_9B00
PDG	PWM Delay Generation	0x4016_A000
ADC_B	A/D Converter	0x4017_0000
DAC120	12-bit D/A converter	0x4017_2000
DAC121	12-bit D/A converter	0x4017_2100

Table 3.1 Peripheral base address (3 of 3)

Name	Description	Base address
FLAD	Data Flash	0x407F_C000
FACI	Flash Application Command Interface	0x407F_E000

Note: Name = Peripheral name
Description = Peripheral functionality
Base address = Lowest reserved address or address used by the peripheral

3.2 Access Cycles

This section provides access cycle information for the I/O registers described in this manual.

- Registers are grouped by associated module.
- The number of access cycles indicates the number of cycles based on the specified reference clock.
- In the internal I/O area, reserved addresses that are not allocated to registers must not be accessed, otherwise operations cannot be guaranteed.
- The number of I/O access cycles depends on bus cycles of the internal peripheral bus, divided clock synchronization cycles, and wait cycles of each module. Divided clock synchronization cycles differ depending on the frequency ratio between ICLK and PCLK.
- When the frequency of ICLK is equal to that of PCLK, the number of divided clock synchronization cycles is always constant.
- When the frequency of ICLK is greater than that of PCLK, at least 1 PCLK cycle is added to the number of divided clock synchronization cycles.
- The number of write access cycles indicates the number of cycles obtained by non-bufferable write access.

Note: This applies to the number of cycles when access from the CPU does not conflict with bus access from other bus masters such as DTC or DMAC.

Table 3.2 Access cycles (1 of 3)

Peripherals	Address		Number of access cycles				Cycle Unit	Related function
			ICLK = PCLK		ICLK > PCLK*1			
	From	To	Read	Write	Read	Write		
RMPU, TZF, SRAM, BUS, DMACn, DMA, DTC, ICU	0x4000_0000	0x4000_6FFF	4	3	4	3	ICLK	Renesas Memory Protection Unit, TrustZone Filter, SRAM Control, BUS Control, Direct memory access controller n, DMAC Module Activation, DTC Control Register, Interrupt Controller
CACHE	0x4000_7000	0x4000_7FFF	4	5	4	5	ICLK	CACHE
CPSCU, DBG, FCACHE	0x4000_8000	0x4001_CFFF	4	3	4	3	ICLK	CPU System Security Control Unit, Debug Function, Flash Cache
SYSC	0x4001_E000	0x4001_E3FF	5	4	5	4	ICLK	System Control
SYSC	0x4001_E400	0x4001_E5FF	9	8	5 to 8	5 to 8	PCLKB	System Control
PORTn	0x4001_F000	0x4001_F7FF	5	3	5	3	ICLK	PORTn Control Register 1/3/4
PORTn (PCNTR2)	0x4001_F000	0x4001_F7FF	8	3	8	3	ICLK	PORTn Control Register 2
PFS	0x4001_F800	0x4001_FFFF	8	3	8	3	ICLK	Pmn Pin Function Control Register
IIRFA	0x4002_0000	0x4002_03FF	4	3	4	3	ICLK	IIR Filter Accelerator
IIRFA	0x4002_0400	0x4002_0FFF	6	3	6	3	ICLK	IIR Filter Accelerator

Table 3.2 Access cycles (2 of 3)

Peripherals	Address		Number of access cycles				Cycle Unit	Related function
			ICLK = PCLK		ICLK > PCLK*1			
	From	To	Read	Write	Read	Write		
TFU	0x4002_1000	0x4002_1FFF	4	3	4	3	ICLK	Trigonometric Function Unit
ELC	0x4008_2000	0x4008_2FFF	5	4	3 to 5	2 to 4	PCLKB	Event Link Controller
IWDT, WDT, CAC	0x4008_3000	0x4008_3FFF	5	4	3 to 5	2 to 4	PCLKB	Independent Watchdog Timer, Watchdog Timer, Clock Frequency Accuracy Measurement Circuit
MSTP	0x4008_4000	0x4008_4FFF	5	4	2 to 4	2 to 4	PCLKB	Module Stop Control
KINT	0x4008_5000	0x4008_5FFF	4	3	1 to 4	1 to 3	PCLKB	Key Interrupt Function
POEG	0x4008_A000	0x4008_AFFF	5	4	3 to 5	2 to 4	PCLKB	Port Output Enable for GPT
CANFD	0x400B_0000	0x400C_1FFF	5	4	2 to 5	2 to 4	PCLKB	CANFD Module
PSCU	0x400E_0000	0x400E_0FFF	5	4	2 to 5	2 to 4	PCLKB	Peripheral Security Control Unit
AGTn	0x400E_8000	0x400E_8FFF	7	4	4 to 7	2 to 4	PCLKB	Low Power Asynchronous General Purpose Timer n
TSN	0x400F_3000	0x400F_3FFF	5	4	2 to 5	2 to 4	PCLKB	Temperature Sensor
ACMPHSn	0x400F_4000	0x400F_4FFF	4	3	1 to 3	1 to 3	PCLKB	High-Speed Analog Comparator
CRC, DOC	0x4010_8000	0x4010_9FFF	5	4	2 to 5	2 to 4	PCLKA	Cyclic Redundancy Check, Data Operation Circuit
SCIn	0x4011_8000	0x4011_8FFF	5	4	2 to 4	2 to 4	PCLKA	Serial Communication Interface n
SPIn	0x4011_A000	0x4011_AFFF	5	4	2 to 5	2 to 4	PCLKA	Serial Peripheral Interface n
IICn	0x4011_F000	0x4011_FFFF	5	4	2 to 4	2 to 4	PCLKA	Inter-Integrated Circuit n
CANFD ECC	0x4012_F200	0x4012_FFFF	5	4	2 to 5	2 to 4	PCLKA	CANFD ECC Module
SCE5	0x4016_1000	0x4016_1FFF	6	4	3 to 6	2 to 4	PCLKA	Secure Cryptographic Engine
GPT32n, GPT_OPS (core clock = PCLKD)	0x4016_9000	0x4016_9FFF	8	5	5 to 8	3 to 5	PCLKA	General PWM Timer n, Output Phase Switching Controller
GPT32n, GPT_OPS (core clock = GPTCLK)	0x4016_9000	0x4016_9FFF	10	7	7 to 10	5 to 7	PCLKA	General PWM Timer n, Output Phase Switching Controller
GPT (GTCKCR)	0x4016_9B00	0x4016_9B00	5	4	2 to 4	2 to 4	PCLKA	GPT Clock Control Register
PDG	0x4016_A000	0x4016_AFFF	4	3	1 to 3	1 to 3	PCLKA	PWM Delay Generation
ADC	0x4017_0000	0x4017_0FFF	5	4	2 to 5	2 to 4	PCLKA	A/D Converter
ADC	0x4017_1000	0x4017_1FFF	4	3	1 to 3	1 to 3	PCLKA	A/D Converter
DAC12n	0x4017_2000	0x4017_2FFF	5	4	2 to 4	2 to 4	PCLKA	12-bit D/A Converter

Table 3.2 Access cycles (3 of 3)

Peripherals	Address		Number of access cycles				Cycle Unit	Related function
			ICLK = FCLK		ICLK > FCLK*1			
	From	To	Read	Write	Read	Write		
FLAD, FACI	0x407F_C000	0x407F_EFFF	5	4	2 to 5	2 to 4	FCLK	Data Flash, Flash Application Command Interface

Note 1. If the number of PCLK or FCLK cycles is non-integer (for example 1.5), the minimum value is without the decimal point, and the maximum value is rounded up to the decimal point. For example, 1.5 to 2. 5 is 1 to 3.

Appendix 4. Peripheral Variant

Table 4.1 shows the correspondence between the module neme used in this manual and the Peripheral Variant.

Table 4.1 Module name vs Peripheral Variant

Module name	Peripheral Variant
ELC	ELC_B
AGTW	AGTW_B
SCI	SCI_B
IIC	IIC_B
CANFD	CANFD_B
SPI	SPI_B
SCE5	SCE5_B
ADC	ADC_B
DOC	DOC_B

Revision History

Revision 1.10 — Dec 9, 2021

First edition, issued

Revision 1.20 — Mar 31, 2022

1. Overview:
- Changed Package code in Table 1.12.
- Changed number of ACMPHS in Table 1.13.
- Changed description of KINT in Table 1.14.

8. Clock Generation Circuit:
- Changed function of CANFDCKDIVCR.CANFDCKDIV[2:0] bit.
- Changed description of ICLK in 8.6.1 System Clock (ICLK).

10. Low Power Modes:
- Changed specification of Power control modes in Table 10.1.
- Changed function of MSTPCRD.MSTPD11 to MSTPD14 in 10.2.7 MSTPCRD.

12. Interrupt Controller Unit (ICU):
- Removed unnecessary explanation in 12.2.17 WUPEN0.
- Changed IIRFA event number in Table 12.4.

13. Buses:
- Changed PSBIU related information in Figure 13.3.
- Changed Slave-TZF Error description in 13.4.2 Operations When a Bus Error Occurs.
- Added secure access description in 13.6.2.1 CSAR: Cache Security Attribution Register.

15. DMA Controller (DMAC):
- Changed bit name from SM to DM in Table 15.4.

19. Key Interrupt Function (KINT):
- Changed description in 19.3.2 Operation When Using the Key Interrupt Flags (KRCTL.KRMD = 1).

21. General PWM Timer (GPT):
- Changed description of GTIOR.OADF[1:0] and OBDF[1:0] in 21.2.14 GTIOR.

23. Low Power Asynchronous General Purpose Timer (AGTW):
- Changed descriptoin of AGTMR2.LPM bit from write to read in 23.2.6 AGTMR2.

26. Serial Communications Interface (SCI):
- Changed description of TDR.TDAT[8:0] bits from reception to transmission in 26.2.3 TDR.
- Changed bit name of description [7] in Figure 26.48 SCI Initialization Flow in Manchester Mode.
- Changed regiseter name SCR to CCR0 in 26.20.11 Notes on Transmit Enable bit (CCR0.TE).
- Removed unnecessary phrase "input port or" in 26.20.11 Notes on Transmit Enable bit (CCR0.TE).

27. I^2C Bus Interface (IIC):
- Changed base address.
- Added DTC/DMAC Activation column in Table 27.10.
- Changed referenced figure numbers from Figure 1.41 and 1.42 to Figure 27.8 and 27.9.

28. CAN with Flexible Data-rate (CANFD):
- Changed mode name from Classical-only mode to Classical CAN mode.
- Changed operation frequency peripheral clock in Table 28.1.
- Changed value after reset of CFDC0FDCFG.CLOE bit in 28.2.8 CFDC0FDCFG.

30. Serial Peripheral Interface (SPI):
- Changed phrase RSPI to SPI.
- Changed image in Figure 30.66.
- Changed interrupt signal name SPRI and SPEI to SPIi_SPRI and SPIi_SPEI.
- Changed bit name SPSCLR.SPRFC to SPSRC.SPRFC in Figure 30.65.

32. Trigonometric Function Unit (TFU):
- Added description for hypot_k, atan and atanhypot_k.

33. IIR Filter Accelerator (IIRFA):
- Removed unnecessary description of output data reading procedure in 33.3.5.2 Procedure for Channel Processing Execution.

Revision 1.20 — Mar 31, 2022

36. 12-Bit A/D Converter (ADC):
- Changed description of Interrupt sources from "less than" to "less than or equal to" in Table 36.1.
- Changed description of ADDOPCRAn.OFSETSEL[3:0] bits from offset to gain in 36.2.3.2 ADDOPCRAn.
- Changed description of ADPGACRn.PGASEL1 bit from Transit to Amplifier Output in 36.2.5.5 ADPGACRn.
- Added description of PGAGAIN[3:0] in 36.2.5.5 ADPGACRn.
- Changed description of bit table in 36.2.8.5 ADCMPMDR1 from more than, less than to or more, or less.
- Changed description of bit function from falls to less than or equal to in 36.2.11.3 ADFIFOINTLR0 to 36.2.11.7 ADFIFOINTLR4.
- Removed unnecessary description in 36.3.3 Virtual Channel.
- Removed Scan end processing timing in Figure 36.7, 36.12, 36.13, 36.17, 36.34, 36.35.
- Changed description of internal reference voltage from and to or in 36.3.10 Internal Reference Voltage.
- Changed group number from 8 to n in 36.3.16 Group Priority Operation.
- Added group B to the description in 36.3.16.2 Group Priority Operation in Continuous Scan mode.
- Changed description of restrictions on the operation mode from more than 3 to three or more in 36.3.16.3 Restrictions on Group Priority Operation.
- Changed description from read to write in 36.4.9.1 Operation when A/D conversion data is read from FIFO.
- Changed bit name SCANENDFn to SCENDFn in Table 36.30, Table 36.33, 36.10.11 Notes on A/D Conversion Start Trigger.
- Removed tADCH in Figure 36.36.
- Added description of PGA setting in 36.10.7 Notes on using analog channels to which the PGA is connected.
- Added 36.10.15 Sampling Time Estimation.

39. High-Speed Analog Comparator (ACMPHS):
- Changed number of interrupt from three to four in 39.5 ACMPHS Interrupts.

43. Flash Memory:
- Changed R/W attribution of FSTATR.SECERR.
- Changed description in 43.16.3 Access Cycle.

46. Electrical Characteristics:
- Removed DAC12 from Reference power supply current in Table 46.7.
- Changed Package code in Table 46.11.
- Added parameter in Table 46.35.
- Changed full-scale error value in Table 46.35.
- Removed pseudo from DNL error in Table 46.35.

Appendix 2. Package Dimensions:
- Changed image in Figure 2.3 and Figure 2.5.

Appendix 3. I/O Registers:
- Changed Base address of IIC0WU_B.
- Changed module name of GPT320 to GPT329.

Revision 1.30 — Aug 05, 2022

Features:
- Updated Features.

1. Overview:
- Updated Table 1.3 System.
- Updated Table 1.8 Analog.
- Updated Table 1.10 Data processing accelerator.

7. Low Voltage Detection (LVD):
- Updated 7.2.2 LVD1CMPCR : Voltage Monitoring 1 Comparator Control Register and 7.2.3 LVD2CMPCR : Voltage Monitoring 2 Comparator Control Register.

10. Low Power Modes:
- Updated Table 10.2 Operating conditions of each low power mode and Table 10.3 Interrupt Source for canceling Snooze, Software Standby and Deep Software Standby Modes.
- Updated 10.2.11 SNZEDCR0 : Snooze End Control Register 0.
- Updated 10.8.3 Returning from Snooze Mode to Software Standby Mode.
- Removed 10.10.12 Conditions of A/D Conversion Start in Snooze Mode.
- Updated 10.10.13 ELC Events in Snooze Mode.

12. Interrupt Controller Unit (ICU):
- Updated Table 12.4 Event table.

15. DMA Controller (DMAC):
- Updated Figure 15.1 Block Diagram of DMAC.

16. Data Transfer Controller (DTC):
- Updated Figure 16.1 DTC block diagram.

17. Event Link Controller (ELC):
- Updated Table 17.3 Association between event signal names set in ELSRn.ELS[8:0] bits and signal numbers.

Revision 1.30 — Aug 05, 2022

21. General PWM Timer (GPT):
- Updated 21.2.46 OPSCR : Output Phase Switching Control Register.

23. Low Power Asynchronous General Purpose Timer (AGTW):
- Updated Table 23.9 Usable settings in Software Standby mode (AGTW0) and Table 23.10 Usable settings in Software Standby mode (AGTW1).

24. Watchdog Timer (WDT):
- Updated 24.2.5 WDTCSTPR : WDT Count Stop Control Register.
- Updated Table 24.5 Association between Option Function Select Register 0 (OFS0) and the WDT registers.

27. I^2C Bus Interface (IIC):
- Updated Table 27.1 I^2C specifications.
- Updated 27.2.27 BCST : Bus Condition Status Register.

30. Serial Peripheral Interface (SPI):
- Added Note in Figure 30.46 Example of Communication End Interrupt Operation (Enable control).
- Updated 30.3.10 Error Detection.
- Updated Figure 30.52 Operation example of the OVRF and PERF flags.
- Updated 30.3.10.3 Mode fault errors.
- Updated 30.4.3 Mode-Fault, Underrun, Overrun, Parity Error, or received data ready Event Output.

31. Cyclic Redundancy Check (CRC):
- Updated 31.3.2 CRC Snoop Function.

32. Trigonometric Function Unit (TFU):
- Updated 32.3.1 Arithmetic Processing.

35. Secure Cryptographic Engine (SCE5):
- Updated 35.3.1 Software Standby Mode.

Revision 1.30 — Aug 05, 2022

36. A/D Converter:
- Updated 36.1 Overview.
- Updated 36.2.1.8 ADMDR : A/D Converter Mode Selection Register.
- Updated 36.2.2.1 ADGSPCR : A/D Group scan Priority Control Register.
- Updated 36.2.3.1 ADCHCRn : A/D Conversion Channel Configuration Register n (n = 0 to 36), 36.2.3.2 ADDOPCRAn : A/D Conversion Data Operation Control A Register n (n = 0 to 36), 36.2.3.3 ADDOPCRBn : A/D Conversion Data Operation Control B Register n (n = 0 to 36), and 36.2.3.4 ADDOPCRCn : A/D Conversion Data Operation Control C Register n (n = 0 to 36).
- Updated 36.2.5.1 ADSHCR0 : Channel-dedicated Sample-and-hold Circuit Control Register 0 and 36.2.5.2 ADSHCR1 : Channel-dedicated Sample-and-hold Circuit Control Register 1.
- Added 36.2.6 Digital Filter and 36.2.6.1 ADDFSRn : A/D Converter Digital Filter Selection Register n (n = 0, 1).
- Updated 36.2.9.12 ADCMPEXSCR : Extended Analog Compare Match Status Clear Register.
- Updated 36.2.11.10 ADOVFCHSR0 : A/D Conversion Overflow Channel Status Register 0.
- Added 36.3.2 A/D Converter Operation Mode, 36.3.2.1 SAR Mode, 36.3.2.2 Oversampling Mode, and 36.3.2.3 Hybrid Mode.
- Added 36.3.3 Single-ended Input and Differential Input.
- Updated 36.3.6 Scan Group.
- Updated 36.3.7 Scanning Operation, 36.3.7.1 SAR Mode – Single Scan Mode, and 36.3.7.2 SAR Mode – Continuous Scan Mode.
- Added 36.3.7.3 Oversampling Mode – Single Scan Mode, 36.3.7.4 Oversampling Mode – Continuous Scan Mode, 36.3.7.5 Hybrid Mode – Single Scan Mode, 36.3.7.6 Hybrid Mode – Continuous Scan Mode, and 36.3.7.7 Hybrid Mode – Background Continuous Scan Mode.
- Updated Table 36.14 List of conditions under which Self-calibration should be performed.
- Updated Table 36.15 Procedure for self-calibration.
- Updated 36.3.8.3 Restrictions on Self-calibration.
- Updated 36.3.11.1 Self-diagnosis.
- Added 36.3.11.2 Self-diagnosis Mode.
- Updated 36.3.15.1 Configuration and Operation of PGA.
- Updated Table 36.19 PGA settings and available related functions.
- Updated 36.3.16 Channel-dedicated Sample-and-hold Circuit, 36.3.16.1 Configuration of Channel-dedicated Sample-and-hold Circuit, 36.3.16.2 Operation in SAR Mode with Channel-dedicated Sample-and-hold Circuit, and 36.3.16.4 Restrictions on Channel-dedicated Sample-and-hold Circuit.
- Added 36.3.16.3 Operation in Hybrid Mode with Channel-dedicated Sample-and-hold Circuit.
- Updated Table 36.32 Group priority operation usable conditions and corresponding register settings.
- Updated 36.3.18.1 Group Priority Operation in SAR Mode – Single Scan Mode, 36.3.18.2 Group Priority Operation in SAR Mode – Continuous Scan Mode, and 36.3.18.3 Restrictions on Group Priority Operation.
- Updated 36.3.19.1 Synchronous Operation Examples and 36.3.19.2 Restrictions on Synchronous Operation.
- Updated 36.4.1 Internal Data Processing Flow.
- Added 36.4.2 Digital Filter Function, 36.4.2.1 Configuration and Characteristics, and 36.4.2.2 Operation of Digital Filter Function.
- Updated 36.4.3 Calibration and Adjustment.
- Updated the figure title of Figure 36.33 User’s Gain Adjustment (16-bit data length format and Unsigned data format).
- Added Figure 36.34 User’s Gain Adjustment (16-bit data length format and Signed data format).
- Updated the figure title of Figure 36.35 User’s Offset Adjustment (16-bit data length format and Unsigned data format).
- Added Figure 36.36 User’s Offset Adjustment (16-bit data length format and Signed data format).
- Updated 36.4.4 A/D-converted Value Addition/Averaging Function, 36.4.5 Limiter Clip Function, and 36.4.6 Data Formatting process.
- Added 36.4.6.1 Signed/Unsigned Data Process and 36.4.6.2 Data Rounding Process.
- Updated 36.4.7 Data Format.
- Updated Table 36.47 Composite compare match and control register.
- Updated Figure 36.50 The block diagram of FIFO.
- Updated 36.4.10.3 FIFO Data Read Request and Overflow.
- Updated 36.6.2 A/D Conversion Overflow.
- Updated Table 36.52 Procedure for changing ADCLK setting.
- Updated Table 36.53 Procedure for changing the settings of the A/D converter.
- Updated 36.8 Interrupt Sources and ELC Events.
- Updated Table 36.56 A/D conversion processing time.
- Updated Figure 36.51 Scan start processing time.
- Updated Figure 36.52 A/D conversion processing time (SAR mode) and Figure 36.53 A/D conversion processing time (SAR mode with channel-dedicated sample-and-hold circuit).
- Added Figure 36.54 A/D conversion processing time (Oversampling mode), Figure 36.55 A/D conversion processing time (Hybrid mode), and Figure 36.56 A/D conversion processing time (Hybrid mode with channel-dedicated sample-and-hold circuit).
- Updated Figure 36.57 Scan end processing time (SAR/Oversampling/Hybrid mode – Single scan mode) and Figure 36.58 Forcibly stop processing time (SAR/Oversampling/Hybrid mode – Single scan mode).
- Updated 36.10.3 Usage Notes on A/D Data Registers.
- Added 36.10.15 Restrictions on SAR Mode, 36.10.16 Restrictions on Oversampling Mode, and 36.10.17 Restrictions on Hybrid Mode.
- Updated 36.10.18 Sampling Time Estimation.

37. 12-Bit D/A Converter (DAC12):
- Updated 37.2.5 DAASWCR : D/A Amplifier Stabilization Wait Control Register.

41. SRAM:
- Updated 41.2.2 PARIOAD : SRAM Parity Error Operation After Detection Register.

Revision 1.30 — Aug 05, 2022

45. Security Features:
- Updated 45.4 Key Injection.

46. Electrical Characteristics:
- Updated Note. 1 in Table 46.1 Absolute maximum ratings.
- Updated Table 46.7 Operating and standby current.
- Updated 46.4 A/D Converter Characteristics.
- Updated Table 46.45 PGA characteristics in Single-ended input mode.
- Updated Table 46.46 PGA characteristics in Pseudo-differential input mode.

RA6T2 Group User’s Manual: Hardware

Publication Date: Rev.1.30 Aug 05, 2022
Rev.1.20 Mar 31, 2022

Published by: Renesas Electronics Corporation

32-bit MCU
RA6T2 Group

Renesas Electronics Corporation

R01UH0951EJ0130