User's Manual

R9A02G021

User's Manual: Hardware

Renesas Electronics
www.renesas.com

Rev.1.10 Feb 2024

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

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

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

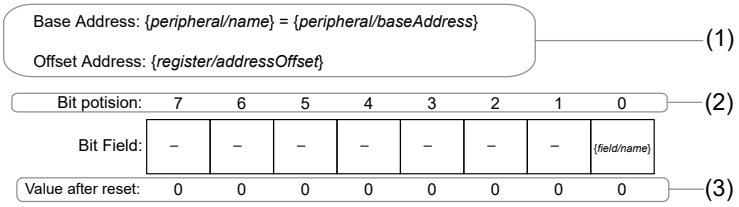
The following terms have special meanings.

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

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.

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



(4) (5) (6)

Bit	Symbol	Function	R/W
0	{*field/name*}	{*field/description*} 0: {*enumeratedValue/description*} 1: {*enumeratedValue/description*}	{*access*}
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, {*pheripheral/name*}, register symbol, {*register/name*}, and address assignment of this register are generally expressed. Base Address and Offset Address mean {*regiser/name*} : {*register/description*} of {*peripheral/name*} is assigned to address {*peripheral/baseAddress*} + {*register/addressOffset*}.

(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

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

(5) Function

Function indicates the full name of the bit field, {*field/description*}, 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
AHB	Advanced High-performance Bus
AHB-AP	AHB Access Port
APB	Advanced Peripheral Bus
ATB	Advanced Trace Bus
BCD	Binary Coded Decimal
BSDL	Boundary Scan Description Language
ETB	Embedded Trace Buffer
ETM	Embedded Trace Macrocell
HMI	Human Machine Interface
IrDA	Infrared Data Association
LSB	Least Significant Bit
MSB	Most Significant Bit
PC	Program Counter
PFS	Port Function Select
PLL	Phase Locked Loop
POR	Power-on reset
PWM	Pulse Width Modulation
S/H	Sample and Hold
SP	Stack Pointer
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

Ultra low power 48 MHz Renesas RISC-V core with 128-KB code flash memory, 16 KB SRAM, 12-bit A/D Converter, and Safety features.

Features

■ RISC-V Core

- Renesas RISC-V instruction-set architecture (RV32I [MACB])
- Maximum operating frequency: 48 MHz
- Debug and Trace: RISC-V External Debug Support
- Debug Port: cJTAG

■ Memory

- 128-KB code flash memory
- 4 KB data flash
- 16 KB SRAM
- 128-bit unique ID

■ Connectivity

- Serial Array Unit (SAU) × 2
  - Simplified SPI × 6
  - UART × 3
  - Simplified I^2C × 6
- I^2C Bus Interface (IICA) × 2
- Serial Interface UARTA (UARTA) × 2
- Remote Control Signal Receiver (REMC)

■ Analog

- 12-bit A/D Converter (ADC12)
- Comparator (CMP) × 2
- 8-bit D/A Converter (DAC8) × 2
- Temperature Sensor (TSN)

■ Timers

- Watchdog Timer (WDT)
- Realtime Clock (RTC)
- Timer Array Unit (TAU) × 8
- 32-bit Interval Timer (TML32)

■ Safety

- SRAM parity and ECC error check
- Flash area protection
- ADC test function
- Clock Frequency Accuracy Measurement Circuit (CAC)
- Cyclic Redundancy Check (CRC) calculator
- Data Operation Circuit (DOC)
- Independent Watchdog Timer (IWDT)
- GPIO readback level detection
- Register write protection
- Illegal memory access detection
- True Random Number Generator (TRNG)

■ System and Power Management

- Low power modes
- Event Link Controller (ELC)
- Data Transfer Controller (DTC)
- Key Interrupt Function (KINT)
- Power-on reset
- Low Voltage Detection (LVD) with voltage settings

■ Multiple Clock Sources

- External clock input (EXTAL) (1 to 20 MHz)
- Sub-clock oscillator (SOSC) (32.768 kHz)
- High-speed on-chip oscillator (HOCO) (24/32/48 MHz)
- Middle-speed on-chip oscillator (MOCO) (8 MHz)
- Low-speed on-chip oscillator (LOCO) (32.768 kHz)
- Clock trim function for HOCO/MOCO/LOCO
- IWDT-dedicated on-chip oscillator (15 kHz)
- Clock out support

■ Up to 42 pins for general I/O ports

- Open drain, input pull-up

■ Operating Voltage

- VCC: 1.6 to 5.5 V

■ Operating Temperature and Packages

- Ta = -40°C to +125°C
  - 48-pin HWQFN (7 mm × 7 mm, 0.5 mm pitch)
  - 32-pin HWQFN (5 mm × 5 mm, 0.5 mm pitch)
  - 24-pin HWQFN (4 mm × 4 mm, 0.5 mm pitch)
  - 16-pin WLCSP(1.99 mm × 1.99 mm, 0.4 mm pitch)

1. Overview

The MCU in this series incorporates an energy-efficient Renesas RISC-V 32-bit core, that is particularly well suited for cost-sensitive and low-power applications, with the following features:

- 128-KB code flash memory
- 4 KB data flash
- 16 KB SRAM
- 12-bit A/D Converter (ADC12)
- Analog peripherals

1.1 Function Outline

Table 1.1 RISC-V core

Feature	Functional description
RISC-V core	• Maximum operating frequency: up to 48 MHz • Instruction-set architecture (ISA) – RISC-V RV32I base integer instruction set – RISC-V C standard extension for compressed instructions – RISC-V M standard extension for integer multiplication and division – RISC-V A standard extension for atomic instructions – RISC-V Ziscr, Control and Status Register (CSR) instructions – RISC-V Zifencei Instruction-Fetch Fence – RISC-V B standard extension for bit manipulation (Zba, Zbb, Zbs) – Performance monitors, cycle and instruction count Control and Status Registers (CSRs) • Dynamic branch prediction • Privilege mode: Machine mode • Machine timer • RISC-V external debug support – Debug module (DM) • 4 hardware breakpoint/watchpoint registers – Debug transport module (DTM) – Debug port: cJTAG

Table 1.2 Memory

Feature	Functional description
Code flash memory	128-KB of code flash memory. See section 35, Flash Memory.
Data flash memory	4-KB of data 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	16-KB On-chip high-speed SRAM with either parity bit or Error Correction Code (ECC). See section 34, SRAM.

Table 1.3 System (1 of 2)

Feature	Functional description
Operating modes	Two operating modes: • Single-chip mode • UART (SAU) boot mode See section 3, Operating Modes.
Resets	The MCU provides 12 resets (RES pin reset, power-on reset, independent watchdog timer reset, watchdog timer reset, voltage monitor 0/1/2 resets, SRAM parity error reset, SRAM ECC error reset, bus error reset, debug reset, software reset). See section 5, Resets.

Table 1.3 System (2 of 2)

Feature	Functional description
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).
Clocks	• External clock input (EXTAL) • Sub-clock oscillator (SOSC) • High-speed on-chip oscillator (HOCO) • Middle-speed on-chip oscillator (MOCO) • Low-speed on-chip oscillator (LOCO) • IWDT-dedicated on-chip oscillator • 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 Core-Local Interrupt Controller (CLIC), 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 17, 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.
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 21, 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 22, Independent Watchdog Timer (IWDT).

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 15, 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 14, Data Transfer Controller (DTC).

Table 1.6 Timers

Feature	Functional description
Realtime Clock (RTC)	The realtime clock has the following features: ● Capable of counting years, months, days of the week, dates, hours, minutes, and seconds, for up to 99 years ● Fixed-cycle interrupt (with period selectable from among 0.5 of a second, 1 second, 1 minute, 1 hour, 1 day, or 1 month) ● Alarm interrupt (alarm set by day of week, hour, and minute) ● Pin output function of 1 Hz See section 20, Realtime Clock (RTC).
Timer Array Unit (TAU)	The timer array unit has eight 16-bit timers. Each 16-bit timer is called a channel and can be used as an independent timer. In addition, two or more channels can be used to create a high-accuracy timer. See section 18, Timer Array Unit (TAU).
32-bit Interval Timer (TML32)	The 32-bit interval timer is made up of four 8-bit interval timers (reference as channels 0 to 3). Each is capable of operating independently and in that case, they all have the same functions. Two 8-bit interval timer channels can be connected to operate as a 16-bit interval timer. Four 8-bit interval timer channels can be connected to operate as a 32-bit interval timer. See section 19, 32-bit Interval Timer (TML32).

Table 1.7 Communication interfaces

Feature	Functional description
Serial Array Unit (SAU)	A single serial array unit has up to four serial channels. Each channel can achieve 3-wire serial (simplified SPI), UART, and simplified I^2C communication. See section 23, Serial Array Unit (SAU).
I^2C Bus Interface (IICA)	The I^2C bus interface has the following three modes: ● Operation stop mode ● I^2C bus mode (multi-master supported) ● Wakeup mode See section 24, I^2C Bus Interface (IICA).
Serial Interface UARTA (UARTA)	The serial interface UARTA supports the following two modes: ● Operation stop mode ● UART mode See section 25, Serial Interface UARTA (UARTA).
Remote Control Signal Receiver (REMC)	The remote control signal receiver can receive data by checking the width and period of an external pulse input signal. See section 26, Remote Control Signal Receiver (REMC).

Table 1.8 Analog (1 of 2)

Feature	Functional description
12-bit A/D Converter (ADC12)	A 12-bit successive approximation A/D converter is provided. Up to 10 analog input channels are selectable. Temperature sensor output and internal reference voltage are selectable for conversion. See section 29, 12-bit A/D Converter (ADC12).
Comparator (CMP)	The Comparator (CMP) compares a test voltage with a reference voltage and provides a digital output based on the comparison result. The test voltages can be provided to the comparator from an external. The reference voltages can be provided to the comparator from internal DAC8 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 31, Comparator (CMP).

Table 1.8 Analog (2 of 2)

Feature	Functional description
8-bit D/A Converter (DAC8)	Two channels of 8-bit D/A Converter (DAC8) can be used as comparator reference voltage and can be output externally. See section 30, 8-Bit D/A Converter (DAC8).
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 ADC12 for conversion and can be further used by the end application. See section 32, Temperature Sensor (TSN).

Table 1.9 Data processing

Feature	Functional description
Cyclic Redundancy Check (CRC) calculator	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 27, 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 33, Data Operation Circuit (DOC).
True Random Number Generator (TRNG)	The true random number generator generates 32-bit random number seeds (which are true random numbers). See section 28, True Random Number Generator (TRNG).

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.

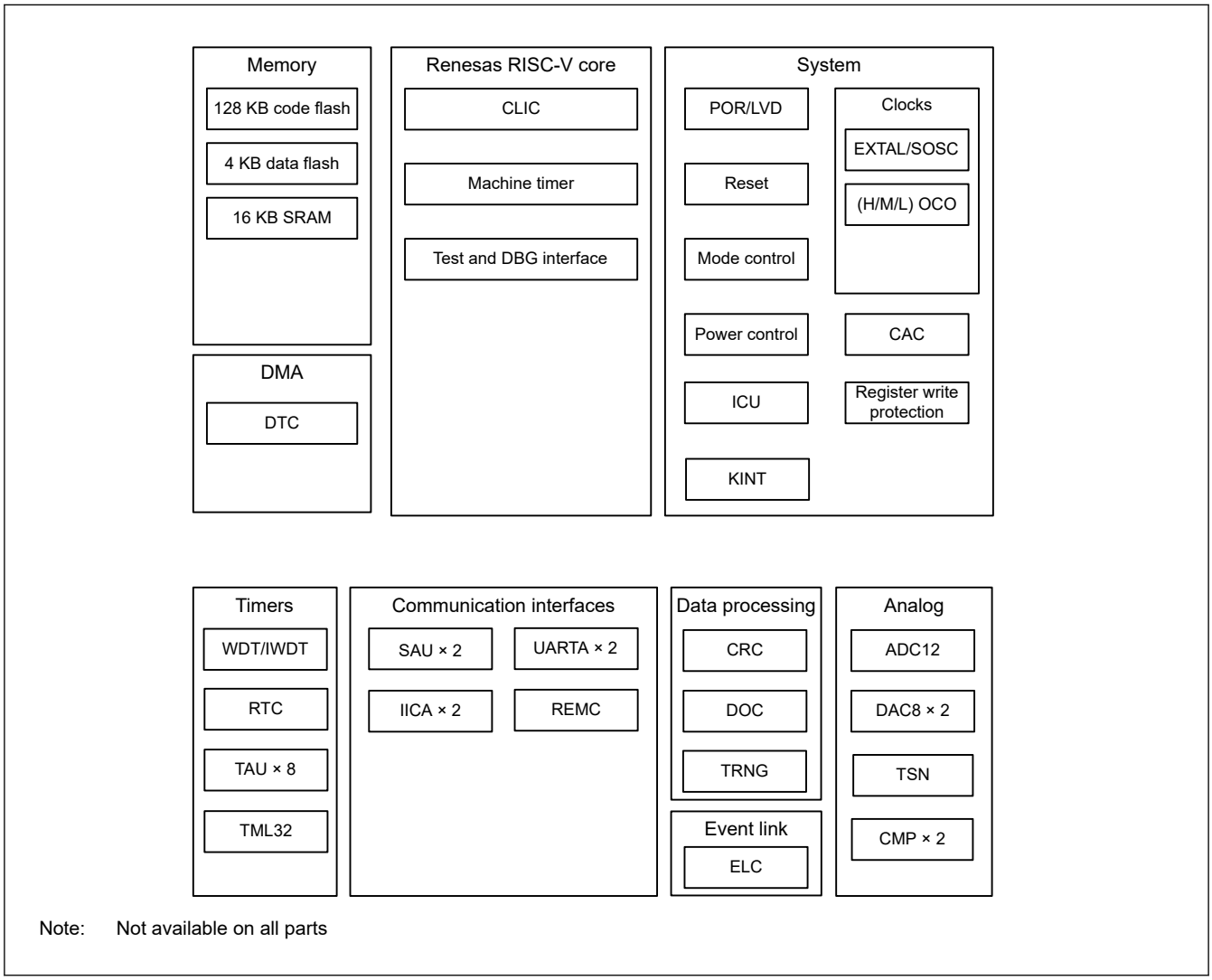


Figure 1.1 Block diagram

1.3 Part Numbering

Figure 1.2 shows the product part number information, including memory capacity and package type. Table 1.10 shows a list of products.

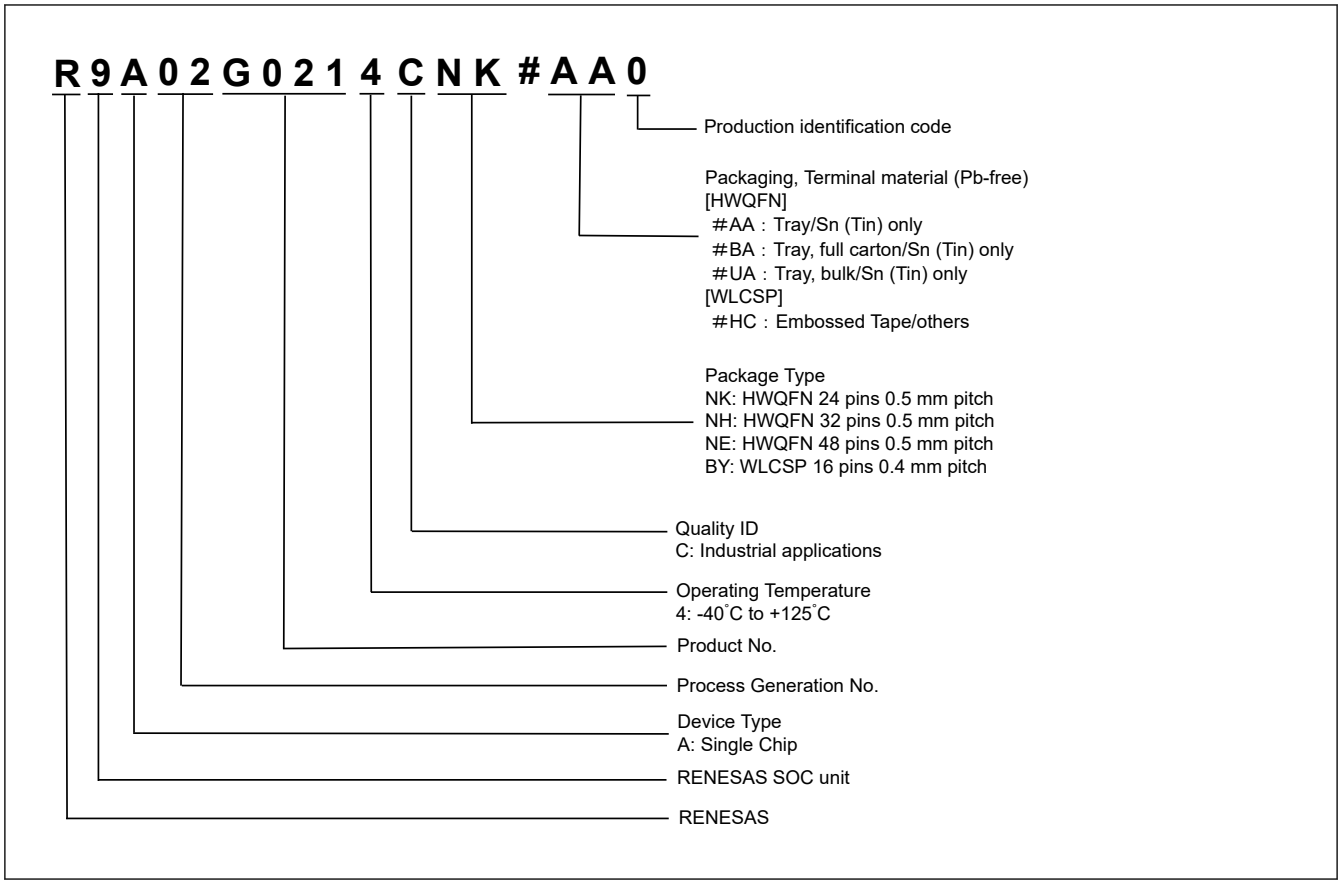


Figure 1.2 Part numbering scheme

Table 1.10 Product list

Product part number	Package code	Code flash	Data flash	SRAM	Operating temperature
R9A02G0214CNE	PWQN0048KC-A	128 KB	4 KB	16 KB	-40 to +125°C
R9A02G0214CNH	PWQN0032KE-A				
R9A02G0214CNK	PWQN0024KG-A				
R9A02G0214CBY	SUBG0016LC-A				

1.4 Function Comparison

Table 1.11 Function comparison

Part number		R9A02G0214CNE	R9A02G0214CNH	R9A02G0214CNK	R9A02G0214CBY
Pin count		48	32	24	16
Package		HWQFN	HWQFN	HWQFN	WLCSP
Code flash memory		128 KB	128 KB	128 KB	128 KB
Data flash memory		4 KB	4 KB	4 KB	4 KB
SRAM (Parity)		12 KB	12 KB	12 KB	12 KB
SRAM (ECC)		4 KB	4 KB	4 KB	4 KB
System	CPU clock	48 MHz	48 MHz	48 MHz	48 MHz
	Sub-clock oscillator	Yes	Yes	Yes	No
	ICU	Yes	Yes	Yes	Yes
	CAC	Yes	Yes	Yes	Yes
	KINT	6	2	No	No
ELC control	ELC	Yes	Yes	Yes	Yes
DMA	DTC	Yes	Yes	Yes	Yes
Timers	WDT/IWDT	Yes	Yes	Yes	Yes
	RTC	Yes	Yes	Yes	Yes
	TAU	8	8	8	6
	TML32	Yes	Yes	Yes	Yes
Communication	SAU	6 (Simplified SPI)	3 (Simplified SPI)	3 (Simplified SPI)	1 (Simplified SPI)
		3 (UART)	3 (UART)	3 (UART)	2 (UART)
		6 (Simplified I^2C)	3 (Simplified I^2C)	3 (Simplified I^2C)	1 (Simplified I^2C)
	IICA	2	1	1	1
	UARTA	2	No	No	No
	REMC	Yes	Yes	No	No
Analog	ADC12	10	8	6	4
	CMP	2	2	2	1
	DAC8	2	2	2	2
	TSN	Yes	Yes	Yes	Yes
Data processing	CRC	Yes	Yes	Yes	Yes
	DOC	Yes	Yes	Yes	Yes
	TRNG	Yes	Yes	Yes	Yes
I/O port	General-purpose I/O	42	26	18	12
	Output current control port	3	3	3	3

1.5 Pin Functions

Table 1.12 Pin functions (1 of 2)

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. Place the capacitor 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	An external clock signal can be input
	XT1	Input	Input/output pins for the sub-clock oscillator Connect a crystal resonator between XT1 and XT2
	XT2	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 debug	TMSC	I/O	On-chip emulator pins
	TCKC	Input	
Interrupt	NMI	Input	Non-maskable interrupt request pin
	IRQ0 to IRQ7	Input	Maskable interrupt request pins
KINT	KR00 to KR05	Input	A key interrupt can be generated by inputting a falling edge to the key interrupt input pins
RTC	RTC1HZ	Output	Realtime clock correction clock (1 Hz) output
TAU	TI00 to TI07	Input	The pins for inputting an external count clock/capture trigger to 16-bit timers 00 to 07
	TO00 to TO07	Output	Timer output pins of 16-bit timers 00 to 07
SAU	RxD0 to RxD2	Input	Serial data input pins of serial interfaces UART0, UART1, and UART2
	TxD0 to TxD2	Output	Serial data output pins of serial interfaces UART0, UART1, and UART2
	SCK00, SCK01, SCK10, SCK11,SCK20, SCK21	I/O	Serial clock I/O pins of serial interfaces SPI00, SPI01, SPI10, SPI11, SPI20, and SPI21
	SCL00, SCL01, SCL10, SCL11, SCL20, SCL21	Output	Serial clock output pins of serial interfaces IIC00, IIC01, IIC10, IIC11, IIC20, and IIC21
	SDA00, SDA01, SDA10, SDA11, SDA20, SDA21	I/O	Serial data I/O pins of serial interfaces IIC00, IIC01, IIC10, IIC11, IIC20, and IIC21
	SI00, SI01, SI10, SI11, SI20, SI21	Input	Serial data input pins of serial interfaces SPI00, SPI01, SPI10, SPI11, SPI20, and SPI21
	SO00, SO01, SO10, SO11, SO20, SO21	Output	Serial data output pins of serial interfaces SPI00, SPI01, SPI10, SPI11, SPI20, and SPI21
IICA	SCLA0, SCLA1	I/O	Clock I/O pins of I^2C bus interfaces IICA0 and IICA1
	SDAA0, SDAA1	I/O	Serial data I/O pins of I^2C bus interfaces IICA0 and IICA1
UARTA	RxDA0, RxDA1	Input	Serial data input pins of serial interfaces UARTA0 and UARTA1
	TxDA0, TxDA1	Output	Serial data output pins of serial interfaces UARTA0 and UARTA1
	CLKA0, CLKA1	Output	Clock output pins of serial interfaces UARTA0 and UARTA1

Table 1.12 Pin functions (2 of 2)

Function	Signal	I/O	Description
REMC	RIN0	Input	External pulse signal input pin for the remote control signal reception circuit
Analog power supply	AVREFP	Input	Analog reference voltage supply pin for the ADC12. Connect this pin to VCC when not using the ADC12.
	AVREFM	Input	Analog reference ground pin for the ADC12. Connect this pin to VSS when not using the ADC12.
ADC12	ANI0 to ANI5, ANI16 to ANI19	Input	Input pins for the analog signals to be processed by the ADC12.
CMP	IVREF0, IVREF1	Input	Reference voltage input pins for comparator
	IVCMP0, IVCMP1	Input	Analog voltage input pins for comparator
	VCOUT0, VCOUT1	Output	Comparator detection result output pins.
DAC8	DACOUT0, DACOUT1	Output	Output pins for the analog signals to be processed by the DAC8.
I/O ports	P000 to P003 , P006 to P011	I/O	General-purpose input/output pins
	P100 to P111	I/O	General-purpose input/output pins
	P200	Input	General-purpose input pin
	P201 to P207	I/O	General-purpose input/output pins
	P300 to P307	I/O	General-purpose input/output pins
	P400 to P403	I/O	General-purpose input/output pins

1.6 Pin Assignments

Figure 1.3 to Figure 1.6 show the pin assignments from the top view.

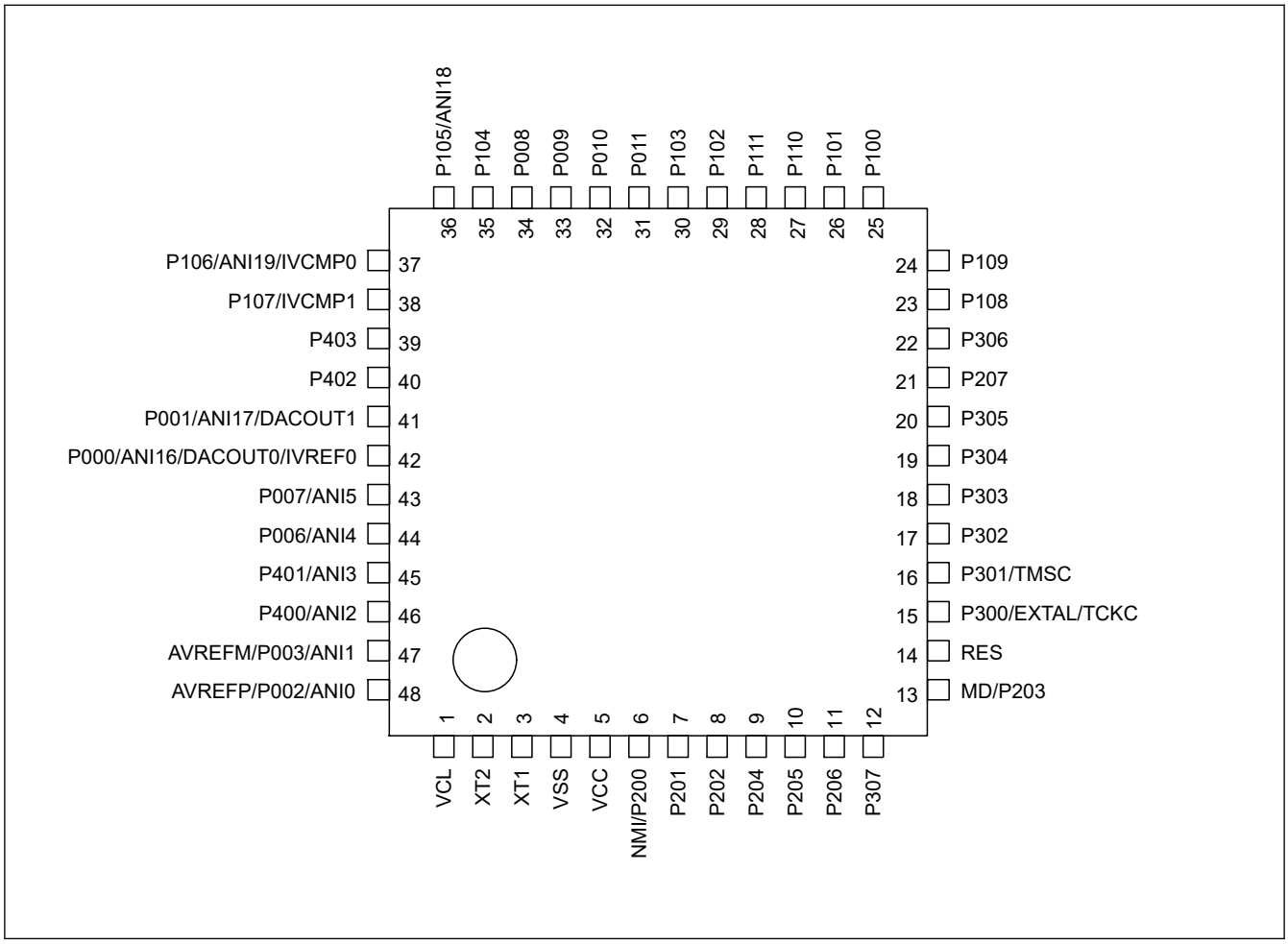


Figure 1.3 Pin assignment for HWQFN 48-pin (top view)

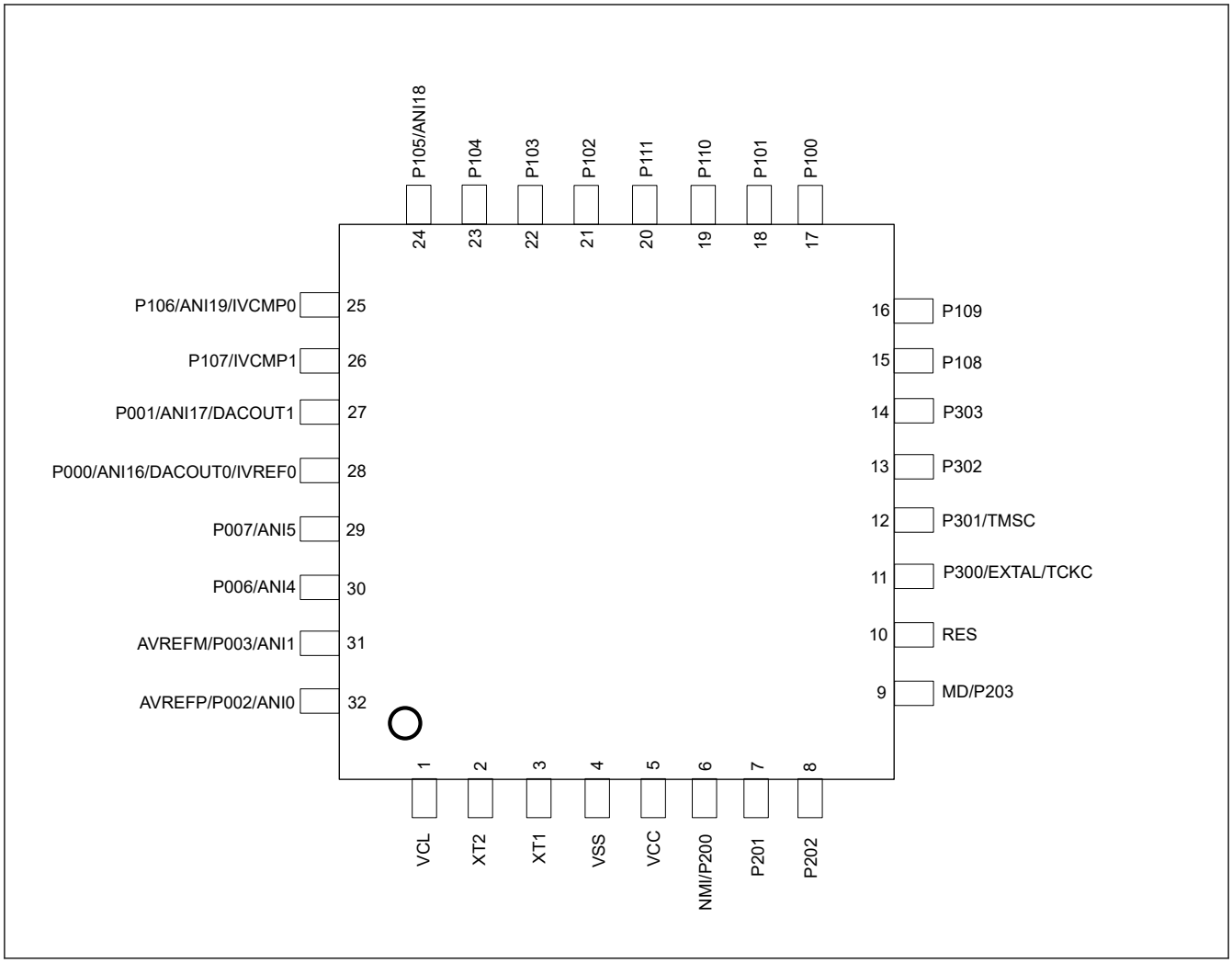


Figure 1.4 Pin assignment for HWQFN 32-pin (top view)

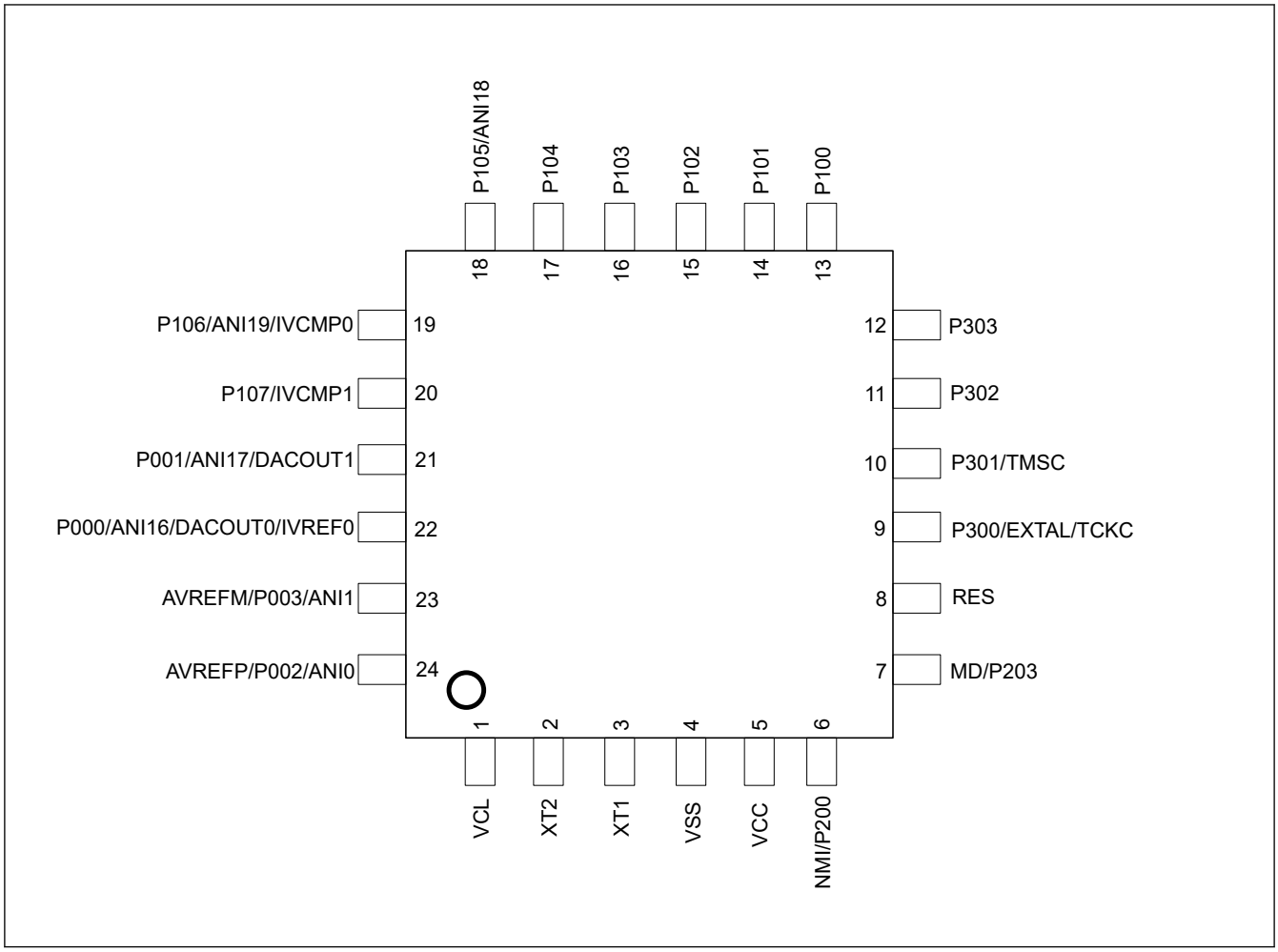


Figure 1.5 Pin assignment for HWQFN 24-pin (top view)

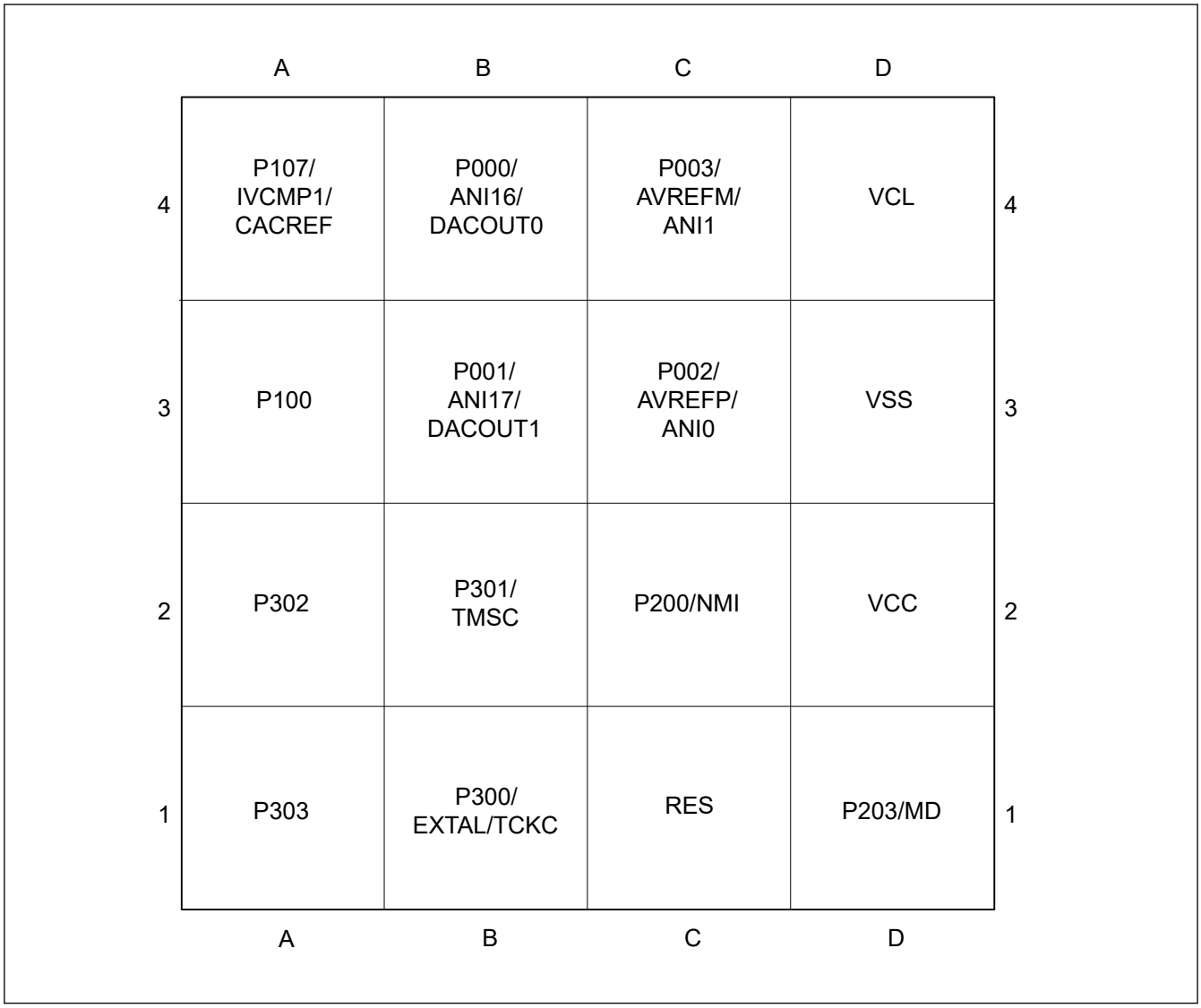


Figure 1.6 Pin assignment for WLCSP 16-pin (top view, pad side down)

1.7 Pin Lists

Table 1.13 Pin list (1 of 2)

Pin number				Power, System, Clock, Debug, CAC	I/O ports	Timers	Communication interfaces		Analogs	Interrupt, KINT
QFN 48-pin	QFN 32-pin	QFN 24-pin	WLCSP 16-pin			TAU, RTC	REMC, IICA, UARTA	SAU	ADC12, DAC8, CMP	
1	1	1	D4	VCL	—	—	—	—	—	—
2	2	2	—	XT2	—	—	—	—	—	—
3	3	3	—	XT1	—	—	—	—	—	—
4	4	4	D3	VSS/AVSS	—	—	—	—	—	—
5	5	5	D2	VCC/AVCC	—	—	—	—	—	—
6	6	6	C2	NMI	P200	—	—	—	—	NMI
7	7	—	—	—	P201	—	—	—	—	IRQ3_C
8	8	—	—	CLKOUT_B	P202	—	RIN0	—	—	IRQ2_C
9	—	—	—	—	P204	—	—	SCK21/SCL21	—	—
10	—	—	—	—	P205	—	—	SI21/SDA21	—	—
11	—	—	—	—	P206	—	—	SO21	—	—
12	—	—	—	—	P307	—	—	—	—	—
13	9	7	D1	MD	P203	—	—	—	—	—
14	10	8	C1	RES#	—	—	—	—	—	—
15	11	9	B1	EXTAL/TCKC	P300	TI07_A/TO07_A	—	SCK00/SCL00	—	IRQ0_A
16	12	10	B2	TMSC	P301	TI06/TO06	—	SI00/SDA00/ RxD0_A	—	IRQ1_A
17	13	11	A2	—	P302	TI03_B/TO03_B	SCLA0_A	TxD0_B	VCOUT1	IRQ3_B
18	14	12	A1	CLKOUT_A	P303	TI04/TO04	SDAA0_A	RxD0_B	—	IRQ2_B
19	—	—	—	—	P304	—	—	SO01	—	KR00
20	—	—	—	—	P305	—	—	SI01/SDA01	—	KR01
21	—	—	—	—	P207	—	—	SCK01/SCL01	—	KR02
22	—	—	—	—	P306	—	—	—	—	KR03
23	15	—	—	—	P108	—	—	—	—	IRQ4_B/KR04
24	16	—	—	—	P109	—	—	—	—	IRQ5_B/KR05
25	17	13	A3	—	P100	TI05/TO05	—	SO00/TxD0_A	—	IRQ6_C
26	18	14	—	—	P101	TI02_B/TO02_B	—	SCK20/SCL20	—	IRQ7_C
27	19	—	—	—	P110	—	—	—	—	IRQ7_B
28	20	—	—	—	P111	—	—	—	—	IRQ6_B
29	21	15	—	—	P102	TI01/TO01	SCLA0_B	SI20/SDA20/ RxD2	—	IRQ2_A
30	22	16	—	—	P103	TI02_A/TO02_A	SDAA0_B	SO20/TxD2	—	—
31	—	—	—	—	P011	TI07_B/TO07_B	SCLA1/CLKA0	—	—	—
32	—	—	—	—	P010	—	SDAA1/RxDA0	—	—	—
33	—	—	—	—	P009	—	TxDA0	SCK10/SCL10	—	—
34	—	—	—	—	P008	—	RxDA1	SI10/SDA10	—	—
35	23	17	—	—	P104	—	—	SCK11/SCL11	IVREF1	—
36	24	18	—	—	P105	RTC1HZ	—	SI11/SDA11	ANI18/VCOUT0	—
37	25	19	—	—	P106	—	—	SO11	ANI19/IVCMP0	—
38	26	20	A4	CACREF	P107	TI03_A/TO03_A	—	—	IVCMP1	—
39	—	—	—	—	P403	—	TxDA1	SO10	—	—
40	—	—	—	—	P402	—	CLKA1	—	—	—
41	27	21	B3	—	P001	TI00	—	TxD1	ANI17/DACOUT1	IRQ5_A
42	28	22	B4	—	P000	TO00	—	RxD1	ANI16/ DACOUT0/ IVREF0*1	IRQ6_A

Table 1.13 Pin list (2 of 2)

Pin number				Power, System, Clock, Debug, CAC	I/O ports	Timers	Communication interfaces		Analogs	Interrupt, KINT
QFN 48-pin	QFN 32-pin	QFN 24-pin	WLCSP 16-pin			TAU, RTC	REMC, IICA, UARTA	SAU	ADC12, DAC8, CMP	
43	29	—	—	—	P007	—	—	—	ANI5	IRQ3_A
44	30	—	—	—	P006	—	—	—	ANI4	IRQ4_A
45	—	—	—	—	P401	—	—	—	ANI3	—
46	—	—	—	—	P400	—	—	—	ANI2	—
47	31	23	C4	AVREFM	P003	—	—	—	ANI1	IRQ7_A
48	32	24	C3	AVREFP	P002	—	—	—	ANI0	—

Note: Several pin names have the added suffix of _A, _B, and _C. The suffix can be ignored when assigning functionality.
Note 1. IVREF0 is not supported in WLCSP 16-pin

2. CPU

2.1 Overview

2.1.1 CPU

- RISC-V instruction-set architecture (ISA)
  - RV32I base integer instruction set
  - "C" standard extension for compressed instructions
  - "M" standard extension for integer multiplication and division
  - "A" standard extension for atomic instructions
  - "Zicsr", Control and Status Register (CSR) instructions
  - "Zifencei" Instruction-Fetch Fence
  - "Zba_Zbb_Zbs" Bit manipulation instructions
- Performance monitors, cycle and instruction count Control and Status Registers (CSRs)
- RISC-V external debug support
- Debug Port: cJTAG

2.1.2 Interrupt Controller

- Core Local Interrupt Controller (CLIC)
  - 34 interrupts
  - 16 interrupt priority levels
  - Selective vectoring with priority preemption
  - Support for software-based tail chaining

2.1.3 Debug

- Debug support according to *RISC-V External Debug Support (Version 0.13.2)*
  - Debug Module (DM)
    - Follows the *RISC-V External Debug Support specification*
    - 4 hardware breakpoint registers
  - Debug Transport Module (DTM)

2.1.4 Operating Frequency

The operating frequencies for the MCU are as follows:

- CPU: Maximum 48 MHz
- cJTAG debug interface: Maximum 6.25 MHz

2.1.5 Block Diagram

Figure 2.1 shows a block diagram of RISC-V CPU core.

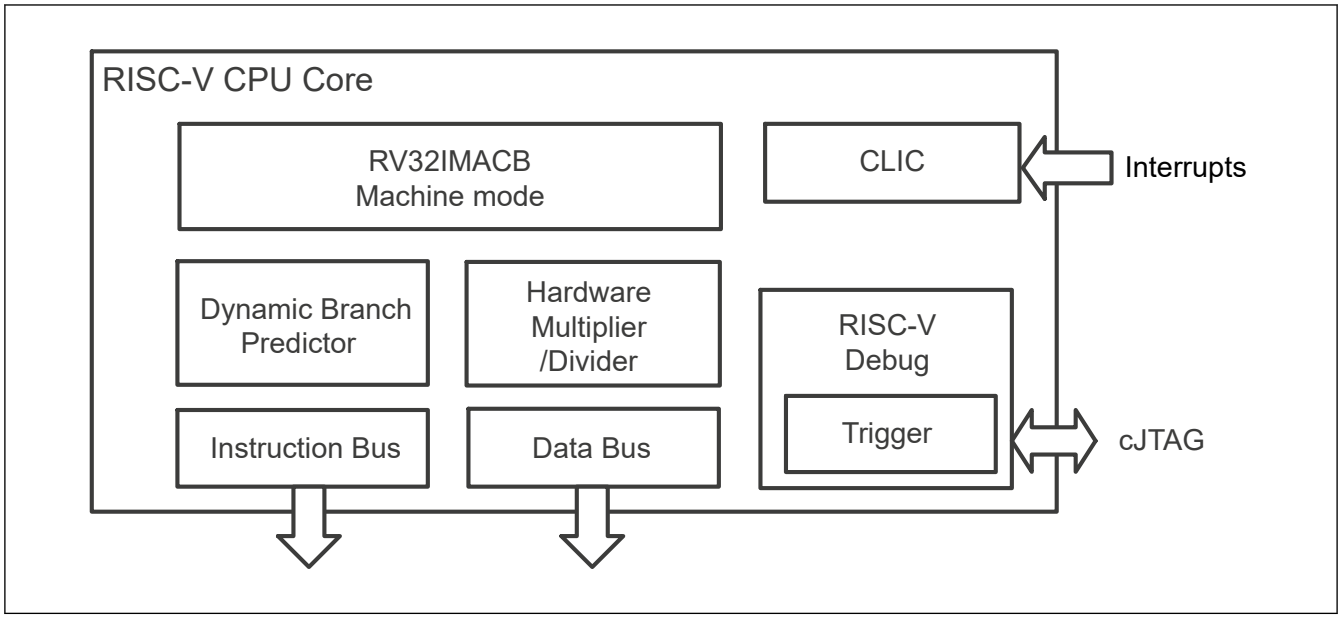


Figure 2.1 RISC-V CPU core block diagram

2.2 cJTAG Interface

The cJTAG interface is an IEEE Std 1149.7 style 2-wire interface.

Table 2.1 shows the cJTAG pins.

Table 2.1 cJTAG pins

Name	I/O	Function	When not in use
TCKC	Input	cJTAG debug port clock. This port should be driven by a pull-up pin.	Pull-up
TMSC	I/O	cJTAG bidirectional data signal. This port should be driven by a pull-up pin.	Pull-up

2.3 Debug Function

The debug subsystem implements RISC-V External Debug Support V0.13.2 and contains two components:

- the Debug Module (DM)
- the Debug Transport Module (DTM)

The debug module can be accessed through its two slave ports. One is the system interface port, which is connected to the processor. The other one is the Debug Memory Interface (DMI) port, which is accessed by the Debug Transport Module that converts debug commands in cJTAG interface of external debugger to bus read/write requests to the DMI port.

See *Core Local Interrupt Controller (CLIC) RISC-V Privileged Architecture Extension Version 0.9-draft, 5/10/2022* and *ARM® CoreSight™ Architecture Specification v3.0 (ARM IHI 0029E, February 2017)* for details.

2.3.1 Debug Mode Definition

In single chip mode, the debugger state of connection is defined as OCD mode, the debugger state of disconnection is defined as User mode. The debugger state of connection is determined by the dmactive bit output in the Debug Module Control register (dmcontrol). The bit can only be written by the OCD. The level of the bit can also be confirmed by reading the DBGSTR.DMACTIVE bit.

2.3.2 Debug Mode Effects

This section describes the effects of debug mode, which occur both internally and externally to the CPU.

2.3.2.1 Low Power Mode

All debug components can store the register settings even when the CPU enters Snooze or Software Standby mode. However, the Debug Module (DM) cannot respond to the Debug Module Interface (DMI) access from the On-Chip Debug (OCD) emulator in Snooze or Software Standby mode. The OCD emulator must first wake up the CPU from Snooze or Software Standby mode to access debug components.

The OCD emulator can use information in DTM Control and Status (dtmcs) register to determine whether the access from Debug Transport Module (DTM) to DM has completed successfully. When the access does not complete for a long time, there is a possibility that CPU is in Snooze or Software Standby mode. To wake up the MCU from Snooze or Software Standby mode, the DMI needs to be reset first by dtmcs.dmihardreset = 1, and the debug interrupt request bit (MCUCTRL.DBIRQ = 1) must be set to 1. Setting the MCUCTRL.DBIRQ bit to 1 works as a wakeup trigger and a halt request during the low power mode. The OCD emulator can resume communication with the CPU after the wakeup process. The MCUCTRL.DBIRQ bit must be cleared to 0 after the completion of DM access is confirmed.

2.3.2.2 Reset

In On-Chip Debug (OCD) mode, some resets depend on the CPU status and the Debug Stop Control Register (DBGSTOPCR) register setting. This is shown in Table 2.2.

Table 2.2 Reset or interrupt and mode setting

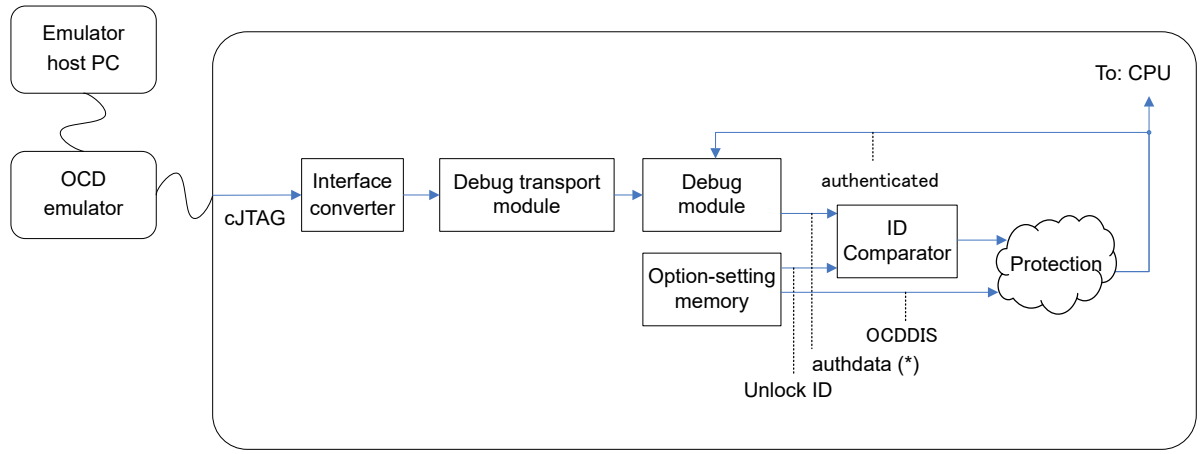
Reset or interrupt name	Control in On-Chip Debug (OCD) mode *1	
	OCD break mode	OCD run mode
RES pin reset	Same as user mode	
Power-on reset	Same as user mode	
Independent watchdog timer reset/interrupt	Does not occur *2	Depends on DBGSTOPCR setting
Watchdog timer reset/interrupt	Does not occur*2	Depends 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	
Bus error reset/interrupt	Same as user mode	
Software reset	Same as user mode	

Note 1. 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 2. The IWDT and WDT always stop in OCD break mode.

2.4 OCD Emulator Connection

A debugger authentication mechanism checks access permission for debug and MCU resources. To obtain full debug functionality, a pass result of the debugger authentication mechanism is required.

Figure 2.2 shows a block diagram of the debugger authentication mechanism.



Note 1. 128-bit authentication data written to the authdata register in four batches.

Figure 2.2 Debugger authentication mechanism block diagram

2.4.1 Debug Authentication Mechanism

An ID comparator is available in the MCU for authentication. The comparator compares the 128-bit authentication data (IAUTH[127:0]) by combining the last four 32-bit write accesses to AUTHDATA register and the 128-bit unlock ID code from the option-setting memory. The write access to AUTHDATA register is assigned in the order of IAUTH[127:96], IAUTH[95:64], IAUTH[63:32], and IAUTH[31:0]. The unlock ID code is written in the OCD/Serial Programmer ID Setting Register (OSIS) in the option-setting memory. The initial value of the unlock ID code is all 1s (0xFFFFFFFF_FFFFFFFF_FFFFFFFF_FFFFFFFF). See section 6, Option-Setting Memory for details.

The debug logic can be locked by setting the OCDDIS bit in the option-setting memory to 0, regardless of the ID comparison result. The default OCDDIS value is 1 and when the authentication data matches with the unlock ID, the debugger can have access to the system resources. Once the OCDDIS bit is cleared, debugging through cJTAG interface is disabled after the next system reset. The OCDDIS can be reverted by issuing an ALeRASE command.

The debug logic can be permanently locked, if the FAPR bit in the SECS register is cleared to 0. By default, the FAPR value is 1 which means that the debugger can be locked and unlocked with the OCDDIS bit and Unlock ID in the option-setting memory. Once the FAPR bit is cleared, the OCDDIS bit and Unlock ID cannot be changed, even if an ALeRASE command is performed, and debugging through cJTAG interface is disabled permanently.

2.4.2 Debug Enable

The OCD emulator must first set the DMACTIVE bit to 1 in the Debug Module Control (DMCONTROL) register to get access permission to the debug module, and then set the DBGEN bit in the System Control OCD Control Register (SYOCDCR) to 1. In addition, the OCD emulator must clear both DMACTIVE and DBGEN bit to 0 before disconnecting it. See section 10, Low Power Modes for details.

2.4.3 Connecting Sequence

Because the OCD emulator is protected by the debugger authentication mechanism, it might be required to unlock the debugger. The OCDDIS, unlock ID and FAPR bits determine whether this step is required and possible. The OCD connecting sequence is described in details in this section.

(1) When OCDDIS bit in flash memory is 1 and OSIS[127] = 0

The ID code is always a mismatch and connection to the debug module is prohibited.

(2) When OCDDIS bit in flash memory is 1 and OSIS[127:0] = All 1s (initial state)

ID authentication is not required, and the connection is enabled. Additionally, the OCD has full access to the debug modules and system resources. For details of the settings for using the Debug Transport Module (DTM) and Debug Module (DM), see *RISC-V External Debug Support (Version 0.13.2)* for details.

1. Connect the OCD emulator to the MCU through the cJTAG interface.
2. Set the DMACTIVE bit to 1 in Debug Module Control (dmcontrol) register.
3. Set the DBGEN bit to 1 in System Control OCD Control (SYOCDCR) register.
   - The debugger can start accessing the CPU debug resources using the following methods defined in RISC-V External Debug Support (see *RISC-V External Debug Support (Version 0.13.2)* for details):
     - Abstract commands to access registers and memory
     - Executing programs written in the program buffer
4. Clear SYOCDCR.DBGEN and dmcontrol.dmactive to 0 when disconnecting the emulator.

(3) When OCDDIS bit in flash memory is 1 and OSIS[127:126] = 10b

ID authentication is required, and the OCD emulator must write the unlock code to the AUTHDATA register in Debug module to unlock the debugger.

1. Connect the OCD emulator to the MCU through the cJTAG interface.
2. Set the DMACTIVE bit to 1 in Debug Module Control (dmcontrol) register.
3. Write the authentication data to AUTHDATA register from OCD emulator.
4. The authorization result can be confirmed by the dmstatus.authenticated bit
5. Set the DBGEN bit to 1 in System Control OCD Control (SYOCDCR) register.
   - The debugger can start accessing the CPU debug resources using the following methods defined in RISC-V External Debug Support (see *RISC-V External Debug Support (Version 0.13.2)* for details):
     - Abstract commands to access registers and memory
     - Executing programs written in the program buffer
6. Clear SYOCDCR.DBGEN and dmcontrol.dmactive to 0 when disconnecting the emulator.

(4) When OCDDIS bit in flash memory is 1 and OSIS[127:126] = 11b

ID authentication is required, and the OCD emulator must write the unlock code to the authdata register in Debug module to unlock the debugger. The connection sequence is the same when OSIS[127:126] is 10b except for "ALeRASE" capability.

When the 128-bit ID code written to the AUTHDATA register is "ALeRASE" in ASCII code, the contents of the code flash, data flash, and configuration area are erased at once. See section 35, Flash Memory for details. The ALeRASE sequence is as follows:

1. Connect the OCD emulator to the MCU through the cJTAG interface.
2. Set the dmactive bit to 1 in Debug Module Control (dmcontrol) register.
3. Write the authentication data to AUTHDATA register from OCD emulator.
4. If the 128-bit ID code is "ALeRASE" in ASCII code (0x414C_6552_4153_45FF_FFFF_FFFF_FFFF_FFFF), the contents of the code flash, data flash, and configuration area are erased. Thereafter, the MCU transitions to Sleep mode.

Note: ALeRASE command is not performed if the OSIS register is protected by the flash read protection function.

(5) When OCDDIS bit in flash memory is 0

Debugging operation through the cJTAG interface is disabled. If FAPR bit in the flash memory is of the default value 1, the debugger connection can be re-enabled by issuing an ALeRASE command so that OCDDIS bit reverts to its default value of 1. If FAPR bit is cleared (set to 0), the debugger is permanently locked (even if an ALeRASE command is performed).

2.4.4 Debug Reset

The debug subsystem modules can be reset using the signals as shown in Table 2.3.

Table 2.3 Debug reset options

	Power-on reset	DMCONTROL.DMACTIVE*1	DTMCS.DMIHARDRESET*2
Debug Module (DM)	✓	✓	—
Debug Transport Module (DTM)	✓	—	✓
DBGREG Module	✓	—	—

Note 1. Controlling reset signal for debug module itself. When writing 0 to this bit, the state of the debug module is set to its reset values. The bit can be set in the Debug Module Control (dmcontrol) register when OCD emulator is connected.
Note 2. Writing 1 to this bit does a hard reset of the DTM, causing the DTM to ignore any outstanding DMI transactions. In general, this bit should only be used when the debugger suspects that the outstanding DMI transaction will not complete. For example, a reset condition that causes an inflight DMI transaction to be canceled. This bit can be set in the DTM Control and Status (dtmcs) register when OCD emulator is connected.

2.4.5 Restrictions on Connecting an OCD Emulator

This section describes the restrictions on emulator access.

2.4.5.1 Starting Connection while in Low Power Mode

When starting a cJTAG connection from an OCD emulator, the MCU must be in Normal or Sleep mode. The Debug Module (DM) cannot respond to the debug module interface (dmi) access from the OCD emulator in Snooze or Software Standby mode. The OCD must wake up the CPU from Snooze or Software Standby mode to access debug components.

2.4.5.2 System Reset Request

The OCD emulator can trigger a system reset request by setting the ndmreset bit in the Debug Module Control (dmcontrol) register to 1. The reset only occurs when the value in dmcontrol.ndmreset transitions from 0 to 1 (edge is detected). Holding the system under reset while dmcontrol.ndmrest = 1 is not supported. After the reset period, the OCD emulator must clear the value in the register to 0.

2.4.5.3 Modify OCDDIS Bit and Unlock ID code in Option-Setting Memory

After modifying the OCDDIS bit and Unlock ID code in the option-setting memory, the OCD emulator must reset the MCU by asserting the RES pin or setting the ndmreset bit in the Debug Module Control (dmcontrol) register to 1. The modified values are reflected after reset. Also, be sure to clear dmcontrol.ndmreset to 0 after the reset period.

2.4.5.4 Connection in UART (SAU) Boot Mode

When a reset is released while the MD pin is low, the MCU starts in UART (SAU) boot mode. In this mode, the debug logic is locked and the OCD emulator cannot access the system resources.

2.5 Programmers Model

2.5.1 Address Spaces

The MCU debug system includes the Debug Module (DM) and the Debug Transport Module (DTM). DM can be accessed through its two slave ports. One is the system interface port, which is for the processor to access DM through the system bus. The other one is the Debug Memory Interface (DMI) port, which is accessed by DTM. Figure 2.3 shows a block diagram of the cJTAG connection.

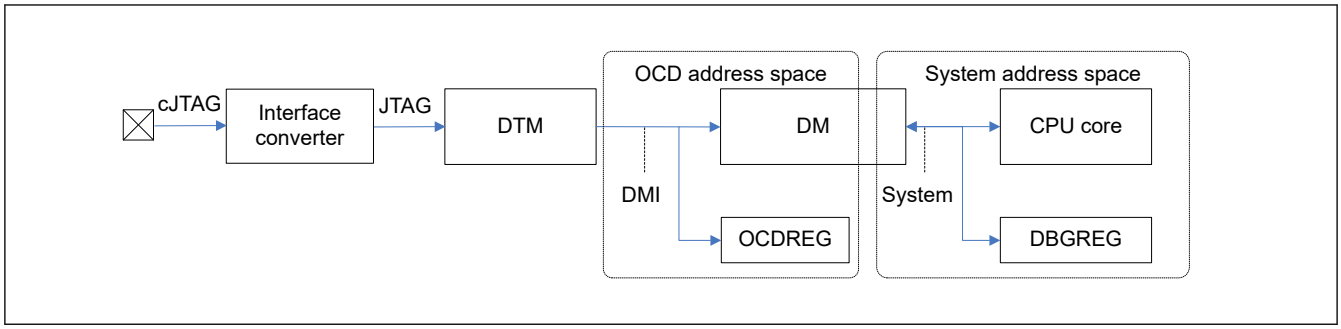


Figure 2.3 cJTAG connection block diagram

For debugging purposes, there are two register modules, DBGREG and OCDREG. DBGREG is in the system address space and can be accessed from the CPU and other bus masters in the MCU. As there is no direct system bus access by the debugger, it must issue commands or prepare instructions for the CPU to execute in debug mode. OCDREG is in the DMI address space and can only be accessed from the OCD tool. The CPU and other bus masters cannot access OCDREG.

2.5.2 CSRs

In the system address space, the processor has Control and Status Registers (CSRs) that can only be accessed from the CPU. Because there is no direct CSR access by the OCD emulator, the OCD emulator must issue commands or prepare instructions for the CPU to execute in debug mode.

See section 2.7.2.3.1. Supported Control and Status Registers.

2.5.3 Core Local Interrupt Controller

The Core Local Interrupt Controller (CLIC) complies with the Core-Local Interrupt Controller (CLIC) RISC-V Privileged Architecture Extension issued by RISC-V International.

2.5.3.1 Overview

The Core Local Interrupt Controller (CLIC) receives interrupt signals and presents the next interrupt to be processed by the Hart. CLIC supports the following functions:

- 51 interrupts including machine timer interrupt, machine software interrupt, 32 interrupt requests from peripherals through ICU, and 17 reserved interrupts
- 16 interrupt priority levels
- Selective hardware vectoring with priority preemption.

2.5.3.2 Register Descriptions

The registers corresponding to the reserved interrupts are not available. The write access is ignored and the read value is 0. For more information on the reserved interrupts, see Table 12.3 in section 12, Interrupt Controller Unit (ICU).

2.5.3.2.1 cliccfg : CLIC Configuration Register

Base address: CLIC = 0xE200_0000

Offset address: 0x0000

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

Bit	Symbol	Function	R/W
0	nvbits	Selective interrupt hardware vectoring feature is implemented. Read value is 1.	R
4:1	nlbits[3:0]	Number of bits in clicintctl[i] allocated for specifying interrupt levels	R/W

Bit	Symbol	Function	R/W
6:5	nmbits[1:0]	Read value are 00b. All interrupts are treated as Machine mode interrupt	R
7	—	This bit is read as 0.	R

2.5.3.2.2 clicinfo : CLIC Information Register

Base address: CLIC = 0xE200_0000

Offset address: 0x0004

Bit position:	31	30	29	28	27	26	25	24	21	20	13	12	0
Bit field:	—	—	—	—	—	—	—	CLICINTCTLBITS[3:0]		version[7:0]		num_interrupt[12:0]	
Value after reset:	0	0	0	0	0	0	0	0 1 0 0		0 0 0 0 0 0 0 1		0 0 0 0 0 0 0 1 1 0 0 1 1	

Bit	Symbol	Function	R/W
12:0	num_interrupt[12:0]	Number of total interrupts supported by CLIC, including all system reserved interrupts.	R
20:13	version[7:0]	Version of CLIC	R
24:21	CLICINTCTLBITS[3:0]	Number of implemented bits in clicintctl register.	R
31:25	—	These bits are read as 0.	R

2.5.3.2.3 clicintip[i] : CLIC Interrupt Pending Register i (i = 0 to 50)

Base address: CLIC = 0xE200_0000

Offset address: 0x1000 + 0x4 × i

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

Bit	Symbol	Function	R/W
0	ip	Indicates Interrupt Pending 0: No interrupt pending 1: Interrupt is pending	R/W
7:1	—	These bits are read as 0.	R

The clicintip[i] register is a read-write register that can be updated both by hardware interrupt inputs and by software. The bit is set by hardware after an edge is observed on the interrupt input. Hardware clears the associated interrupt pending bit when an interrupt is generated in vectored mode.

In contrast, when a non-vectored (common code) interrupt is selected, the hardware does not automatically clear the pending bit.

2.5.3.2.4 clicintie[i] : CLIC Interrupt Enable Register i (i = 0 to 50)

Base address: CLIC = 0xE200_0000

Offset address: 0x1001 + 0x4 × i

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

Bit	Symbol	Function	R/W
0	ie	Indicates Interrupt Enable 0: Interrupt disabled 1: Interrupt enabled	R/W
7:1	—	These bits are read as 0.	R

clicintie[i] is the individual enable bit while mstatus.mie is the global enable bit for the current privilege mode. Therefore, for an interrupt i to be enabled in the current privilege mode, both clicintie[i] and mstatus.mie must be set.

2.5.3.2.5 clicintattr[i] : CLIC Interrupt Attrtibute Register i (i = 0 to 50)

Base address: CLIC = 0xE200_0000

Offset address: 0x1002 + 0x4 × i

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

Bit	Symbol	Function	R/W
0	shv	Selective Hardware Vectoring 0: Non-vectored mode. Interrupts jump to the common code at mtvec CSR. 1: Vector mode. Interrupts jump to the trap-handler function pointer specified in mtvt CSR.	R/W
2:1	trig[1:0]	These bits are read as 01b. Indicate that this interrupt is rising edge trigger type.	R
5:3	—	These bits are read as 0.	R
7:6	mode[1:0]	Read values are 11b. Indicate that this interrupt operates in machine mode.	R

2.5.3.2.6 clicintctl[i] : CLIC Interrupt Input Control Register i (i = 0 to 50)

Base address: CLIC = 0xE200_0000

Offset address: 0x1003 + 0x4 × i

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

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 1.	R
7:4	lvl_prio[3:0]	Interrupt level and priority for the associated interrupt	R/W

These control bits are interpreted as level and priority according to the setting in the CLIC Configuration register (cliccfg.nlbits).

Table 2.4 Interrupt levels and priorities setting (1 of 2)

CLICINTCTLBITS	nlbits	clicintctl[7:0]	Interrupt levels	Interrupt priorities
4	0	p p p p	255	15, 31, 47, 63, 79, 95, 111, 127, 143, 159, 175, 191, 207, 223, 239, 255
4	1	| p p p	127, 255	31, 63, 95, 127, 159, 191, 223, 255
4	2	| | p p	63, 127, 191, 255	63, 127, 191, 255

Table 2.4 Interrupt levels and priorities setting (2 of 2)

CLICINTCTLBITS	nlbits	clicintctl[7:0]	Interrupt levels	Interrupt priorities
4	3	| | | p	31, 63, 95, 127, 159, 191, 223, 255	127, 255
4	≥ 4	| | | |	15, 31, 47, 63, 79, 95, 111, 127, 143, 159, 175, 191, 207, 223, 239, 255	255

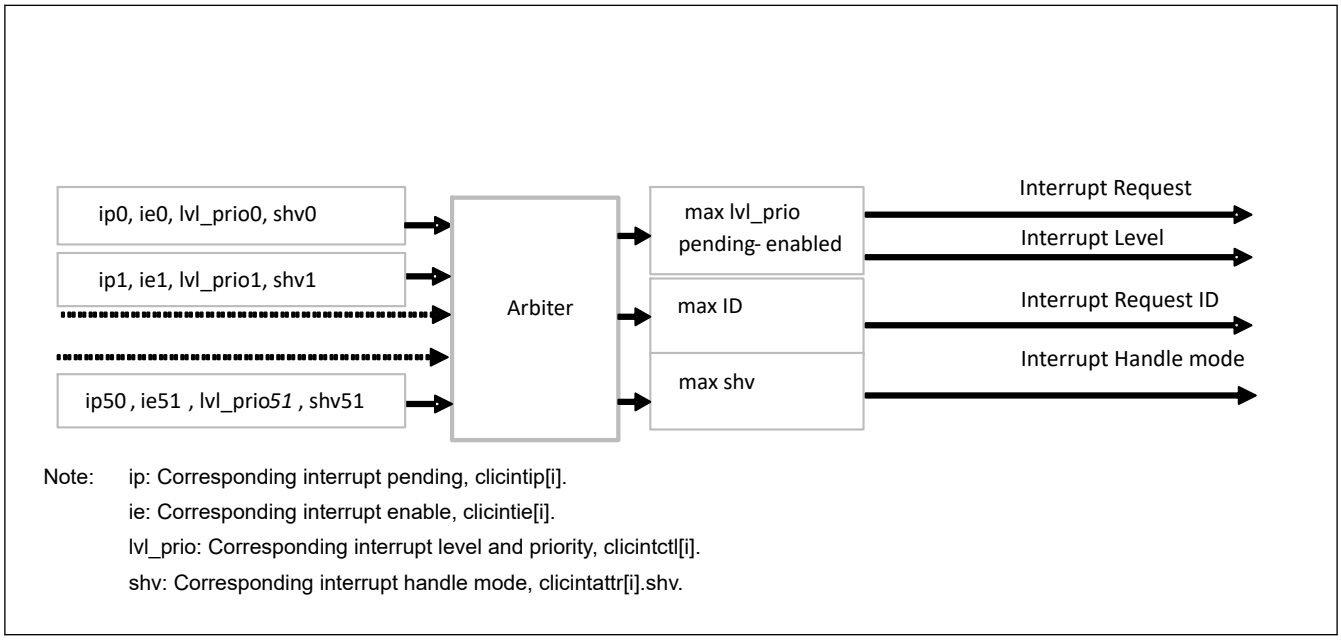
Note: "." bits are non-existent bits for level encoding, assumed to be 1.
"|" bits are available variable bits in level specification.
"p" bits are available variable bits in priority specification.

2.5.3.3 Operation

2.5.3.3.1 CLIC Interrupt Arbiter

To select an interrupt for the core, the CLIC hardware combines the valid bits in clicintctl to form an unsigned integer, then picks the global maximum across all pending-and-enabled interrupts based on this value. Next, the cliccfg.nlbits setting determines how to split the clicintctl value into interrupt level and interrupt priority. The highest-priority interrupt at highest interrupt level is requested first. In case there are multiple pending-and-enabled interrupts at the same highest level and highest priority, the highest-numbered interrupt is requested first.

Figure 2.4 shows a block diagram of the CLIC interrupt arbiter.



Note: ip: Corresponding interrupt pending, clicintip[i].
ie: Corresponding interrupt enable, clicintie[i].
lvl_prio: Corresponding interrupt level and priority, clicintctl[i].
shv: Corresponding interrupt handle mode, clicintattr[i].shv.

Figure 2.4 CLIC interrupt arbiter block diagram

2.5.4 Machine Timer

2.5.4.1 Overview

The machine timer has a set of 64-bit counter register, comparator register, and software interrupt register. It can generate machine timer interrupt (mtip) and machine software interrupt (msip) to CLIC.

2.5.4.2 Block Diagram

Figure 2.5 shows a block diagram of the machine timer.

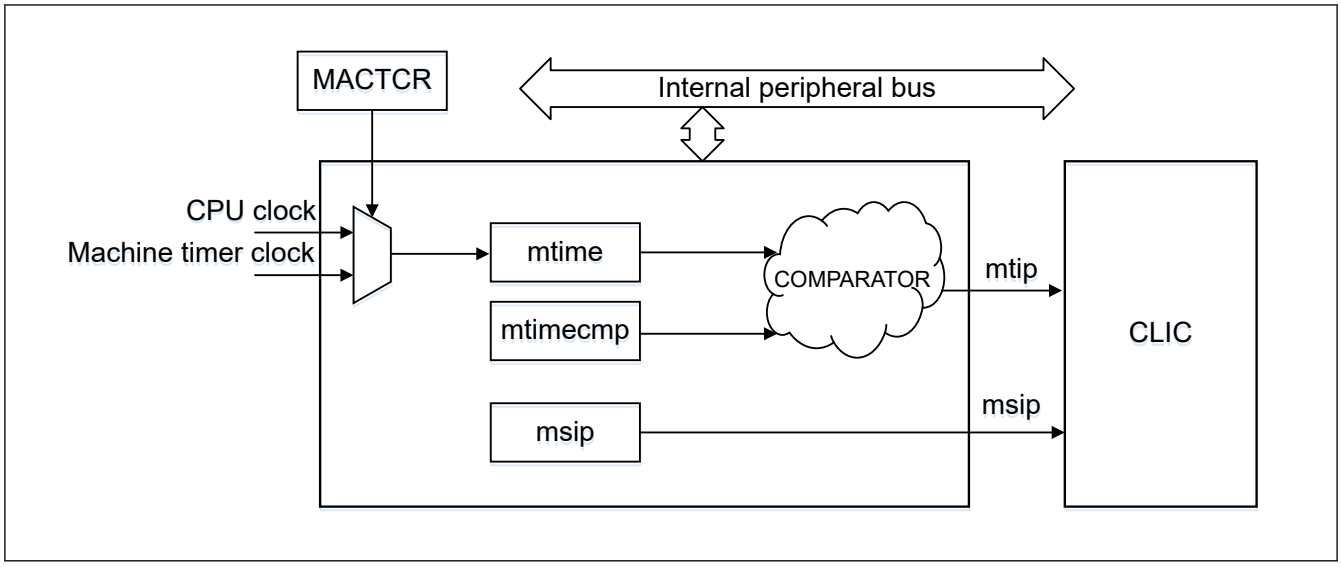


Figure 2.5 Machine timer block diagram

2.5.4.3 Register Descriptions

2.5.4.3.1 mtime_lo : Machine Timer Register Counter Low

Base address: IMT = 0xE600_0000

Offset address: 0x000

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	mtime_lo[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:	mtime_lo[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
31:0	mtime_lo[31:0]	Lower 32-bit of 64-bit counter	R/W

The mtime is a 64-bit counter (mtime_hi: mtime[63:32], mtime_lo: mtime[31:0]) that counts the number of cycles based on the clock selected by MACTCR.CLOCKSOURCE bit. The counter keeps counting while MACTCR.ENABLE bit is set to 1. The mtime can stop counting by setting dcsr.stoptime = 1 in debug mode.

2.5.4.3.2 mtime_hi : Machine Timer Register Counter High

Base address: IMT = 0xE600_0000

Offset address: 0x004

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	mtime_hi[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:	mtime_hi[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
31:0	mtime_hi[31:0]	Upper 32-bit of 64-bit counter	R/W

The mtime is a 64-bit counter (mtime_hi: mtime[63:32], mtime_lo: mtime[31:0]) that counts the number of cycles based on the clock selected by MACTCR.CLOCKSOURCE bit. The counter keeps counting cycles while MACTCR.ENABLE bit is set to 1. The mtime can stop counting by setting dcsr.stoptime = 1 in debug mode.

2.5.4.3.3 mtimecmp_lo : Machine Timer Comparator Register Low

Base address: IMT = 0xE600_0000

Offset address: 0x008

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	mtimecmp_lo[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:	mtimecmp_lo[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
31:0	mtimecmp_lo[31:0]	Lower 32-bit of 64-bit comparison value	R/W

The mtimecmp is a 64-bit register (mtimecmp_hi: mtimecmp[63:32], mtimecmp_lo: mtimecmp[31:0]). The timer interrupt sets the interrupt pending bit in the CLIC.

2.5.4.3.4 mtimecmp_hi : Machine Timer Comparator Register High

Base address: IMT = 0xE600_0000

Offset address: 0x00C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	mtimecmp_hi[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:	mtimecmp_hi[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
31:0	mtimecmp_hi[31:0]	Upper 32-bit of 64-bit comparison value	R/W

The mtimecmp is a 64-bit register (mtimecmp_hi: mtimecmp[63:32], mtimecmp_lo: mtimecmp[31:0]). The timer interrupt sets the interrupt pending bit in the CLIC.

2.5.4.3.5 msip : Triggering Software Interrupt Register

Base address: IMT = 0xE600_0000

Offset address: 0xFFC

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:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MSIP
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	MSIP	Triggering machine software interrupt 0: Trigger machine software interrupt 1: Clear machine software interrupt	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

The msip register is a 32-bit register where the upper 31 bits are tied to 0. The least significant bit can be used to trigger software interrupt and to set the interrupt pending bit corresponds to msip in the CLIC.

2.5.5 Debug Module

Table 2.5 describes the memory map within the Debug Module (DM) address space as viewed from the CPU. The MCU memory map assigns the DM base address to 0xE6800_0000.

The offset for the program buffer can be found by the external debugger through execution of the AUIPC instruction as the first instruction in the program buffer. The starting offset of abstract data is defined as hartinfo.DATAADDR. The offsets can be used as offsets of load/store instructions with the base register to access this memory space.

Table 2.5 Memory map of debug module from the CPU

Address offset	Description
0x0360 to 0x037F	Program buffer 0 to 7
0x0380 to 0x038F	Abstract data 0 to 3

Table 2.6 describes the memory map as viewed from the Debug Memory Interface (DMI). The address value in the table follows the address value assignment of the Debug Module Debug Bus registers as described in RISC-V External Debug Support (see *RISC-V External Debug Support (Version 0.13.2)* for details).

Table 2.6 Memory map of debug module from the DMI (1 of 2)

Address	Description
0x004 to 0x007	Abstract data 0 to 3
0x010	Debug module control
0x011	Debug module status
0x012	Hart info

Table 2.6 Memory map of debug module from the DMI (2 of 2)

Address	Description
0x016	Abstract control and status
0x017	Abstract command
0x018	Abstract command autoexec
0x01D	Next debug module
0x020 to 0x027	Program buffer 0 to 7
0x030	Authentication data
0x040	Halt summary 0

2.5.6 Debug Transport Module

The Debug Transport Module (DTM) provides access to the Debug Module (DM) over JTAG interface, as defined by the RISC-V External Debug Support (see *RISC-V External Debug Support (Version 0.13.2)* for details). The DTM implements the IEEE 1149.1 style Test Access Port (TAP) controller. The JTAG TAP allows access to arbitrary JTAG registers by first selecting one register using the JTAG Instruction Register (IR), and then accessing it through the JTAG Data Register (DR). Table 2.7 describes the supported instructions.

Table 2.7 Supported TAP instructions of DTM

Encoding	Instruction
0x1F	BYPASS
0x01	IDCODE
0x10	dtmcs register
0x11	dmi register

2.5.7 OCDREG

The OCDREG module is only accessible by the OCD emulator. OCDREG has a register that can be accessed through DMI interface, and it can wake up the MCU from low power mode. This register is not defined in RISC-V External Debug Support.

Table 2.8 Memory map of OCDREG through the DMI

Address*1	Description
0x0400	MCUCTRL

Note 1. Address used for DMI access in dmi.address.

2.5.7.1 MCUCTRL : MCU Control Register

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:	—	—	—	—	—	—	—	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

Bit	Symbol	Function	R/W
8	DBIRQ	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
31:9	—	These bits are read as 0. The write value should be 0.	R/W

2.5.8 AUXREG Module

Table 2.9 shows the AUXREG registers.

Table 2.9 AUXREG registers

Name		Address	Access size	R/W
Machine Timer Control Register	MACTCR	0x4001_A000	32 bits	R/W
Software Reset Control Register	SWRCR	0x4001_A100	32 bits	R/W
NMI Handler Address Register	NMIADDR	0x4001_A200	32 bits	R/W

2.5.8.1 MACTCR : Machine Timer Control Register

Base address: CPU_AUX = 0x4001_A000

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:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLOCKSOURCE	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	Machine Timer Clock Enable 0: No clock is supplied to machine timer 1: Clock is supplied to machine timer	R/W
1	CLOCKSOURCE	Machine Timer Clock Source Select 0: Machine timer clock 1: CPU clock	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

The MACTCR register controls the machine timer clock source and indicates the enabled status of the machine timer counter.

2.5.8.2 SWRCR : Software Reset Control Register

Base address: CPU_AUX = 0x4001_A000

Offset address: 0x100

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:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SYSRESETREQ
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	SYSRESETREQ	System Reset Request. The SYSRESETREQ bit is cleared to 0 after a system reset. 0: Do not request a system reset 1: Request a system reset	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

The SWRCR register allows software to request a system reset.

2.5.8.3 NMIADDR : NMI Handler Address Register

Base address: CPU_AUX = 0x4001_A000

Offset address: 0x200

Bit position:	31	0
Bit field:	n/a	
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	NMI handler address. Lower 6 bits must always be set to 0.	R/W

When the CPU receives an NMI, the CPU jumps to the address specified in this register.

2.5.9 DBGREG Module

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

Table 2.10 shows the DBGREG registers other than the CoreSight component registers.

Table 2.10 DBGREG registers

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

2.5.9.1 DBGSTR : Debug Status Register

Base address: CPU_DBG = 0x4001_B000

Offset address: 0x000

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	DMACTIVE	—	—	—	—	—	—	—	—	—	—	—	—
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	DMACTIVE	Debug active status from Debug Module	R
29	—	The read value is undefined.	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.

2.5.9.2 DBGSTOPCR : Debug Stop Control Register

Base address: CPU_DBG = 0x4001_B000

Offset address: 0x010

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	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_SIR	DBGSTOP_TIM	—	—	—	—	—	—	—	—	—	—	—	—	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 0: Enable IWDT reset/interrupt 1: Mask WDT reset/interrupt and stop WDT counter	R/W
1	DBGSTOP_WDT	Mask bit for WDT reset/interrupt in the OCD run mode 0: Enable WDT reset/interrupt 1: Mask WDT reset/interrupt and stop WDT counter	R/W
13:2	—	These bits are read as 0. The write value should be 0.	R/W
14	DBGSTOP_TIM	Control bit for RTC and TAU operation in the OCD break mode 0: Continue RTC and TAU operation 1: Stop RTC and TAU operation	R/W
15	DBGSTOP_SIR	Control bit for SAU and IICA operation in the OCD break mode 0: Continue SAU and IICA operation 1: Stop SAU and IICA operation	R/W

Bit	Symbol	Function	R/W
16	DBGSTOP_LVD0	Mask bit for LVD0 reset 0: Enable LVD0 reset 1: Enable 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 parity error reset/interrupt 1: Mask SRAM parity error reset/interrupt	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

The DBGSTOPCR register specifies the functional stop in OCD mode. All bits in the register are regarded as 0 when the MCU is not in OCD mode.

2.5.9.3 DBGREG CoreSight Component Registers

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

Table 2.11 shows the registers. See *ARM® CoreSight™ Architecture Specification v3.0 (ARM IHI 0029E, February 2017)* for details of each register.

Table 2.11 DBGREG CoreSight component registers

Name	Address	Access size	R/W	Initial value
PIDR4	0x4001_BFD0	32 bits	R	0x00000004
PIDR5	0x4001_BFD4	32 bits	R	0x00000000
PIDR6	0x4001_BFD8	32 bits	R	0x00000000
PIDR7	0x4001_BFDC	32 bits	R	0x00000000
PIDR0	0x4001_BFE0	32 bits	R	0x00000005
PIDR1	0x4001_BFE4	32 bits	R	0x00000030
PIDR2	0x4001_BFE8	32 bits	R	0x000000AA
PIDR3	0x4001_BFEC	32 bits	R	0x00000000
CIDR0	0x4001_BFF0	32 bits	R	0x0000000D
CIDR1	0x4001_BFF4	32 bits	R	0x000000F0
CIDR2	0x4001_BFF8	32 bits	R	0x00000005
CIDR3	0x4001_BFFC	32 bits	R	0x000000B1

2.6 Restrictions

This section describes the functional restrictions and limitations of the CPU.

- There is no bus locking mechanism associated with atomic operations. The software needs to guarantee that the same memory space used for atomic operation is not accessed by the DTC.
- The CPU clock frequency (ICLK) must be at least twice as fast as the MTCLK clock frequency to operate the machine timer.

2.7 RISC-V CPU Core Specification

2.7.1 ISA Overview

This section describes the RISC-V Instruction Set Architecture (ISA) of the CPU.

Table 2.12 Instruction set architecture specification

Function	Description
Basic instruction set	RV32I
Standard extension instruction set	MAC_Zicsr_Zifencei_Zba_Zbb_Zbs
Privileged mode	Machine mode (M mode) only
Misaligned access	Not supported
Core-Local Interrupt Controller (CLIC)	Support: • Interrupt level: 16 levels • Selective hardware vector support
Stack overflow detection function	Yes

(1) Conforming RISC-V Specification

This CPU conforms to the following RISC-V specifications. For version of the specification, see section 2.7.7. References.

- RISC-V ISA Specification Volume 1, Unprivileged spec v.20191213
- RISC-V ISA Specification Volume 2, Privileged Spec v.20190608
- RISC-V Bit-Manipulation ISA-extensions Version 1.0.0-38-g865e7a7
- Smclic Core-Local Interrupt Controller (CLIC) RISC-V Privileged Architecture Extension Version 0.9-draft, 5/10/2022

However, the following specifications do not conform to the above specifications:

- mcause.minhv is fixed to 0 because traps do not occur during hardware vectoring.
- Fetching of the trap vector table during hardware vectoring of CLIC is handled as a load, not an instruction execution. The "04/26/2022 issue #191" change in the "CLIC RISC-V Privileged Architecture Extension" is not supported.

2.7.2 Registers

2.7.2.1 General-purpose Registers

The RISC-V CPU has 32 general-purpose registers of 32-bit. x0 is a zero-register, and it always holds 0. The value after reset of general-purpose registers other than x0 is undefined. Figure 2.6 shows the general-purpose registers of the RISC-V CPU.

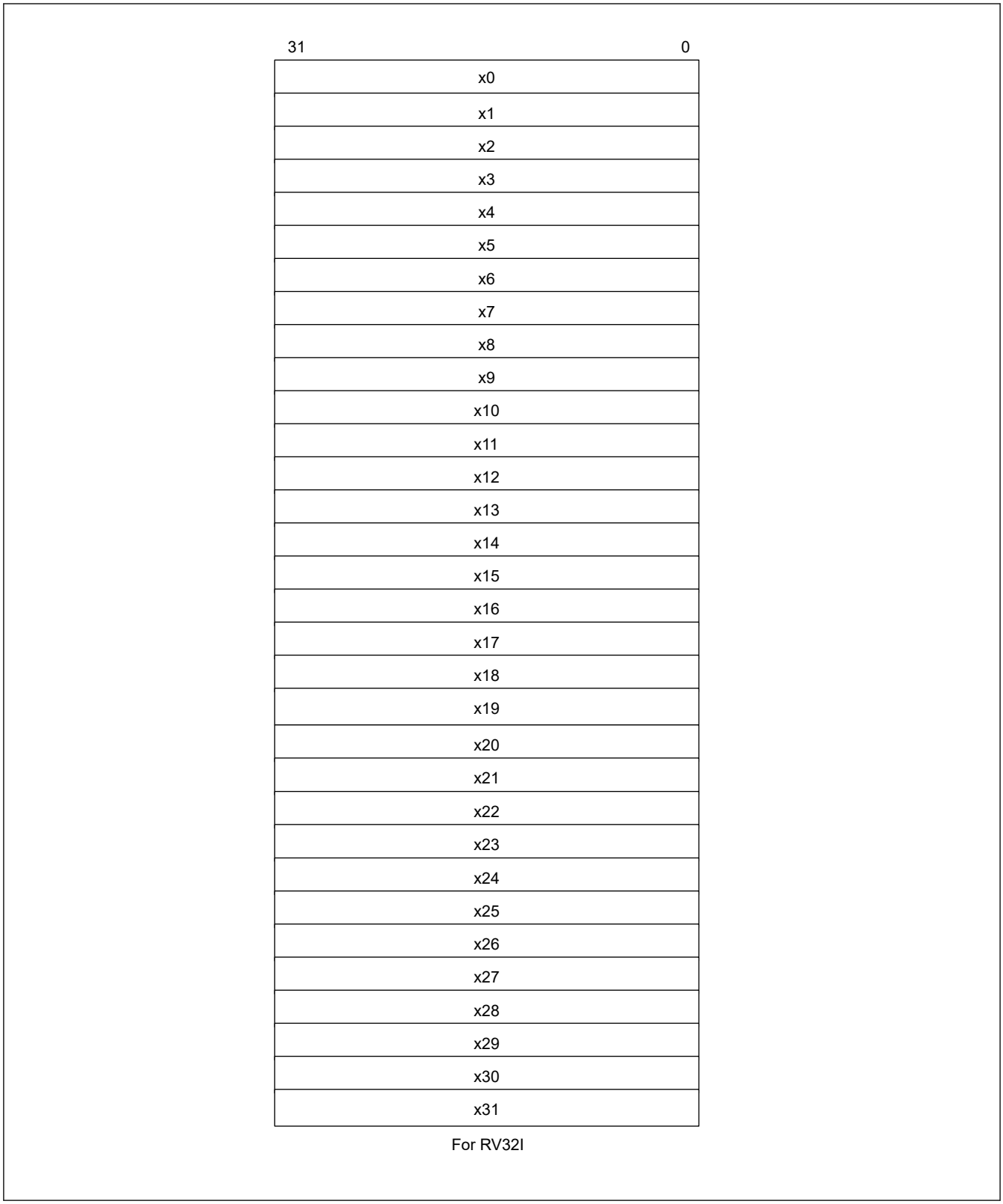


Figure 2.6 General-purpose registers of RISC-V CPU

2.7.2.2 Program Counter

The RISC-V CPU has a Program Counter (PC) of 32-bit. On reset, PC is cleared to 0.

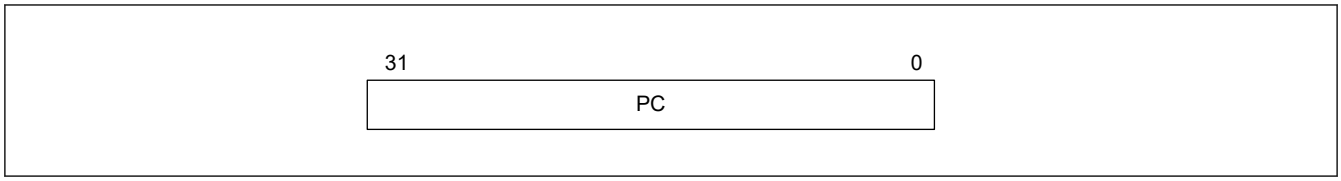


Figure 2.7 Program counter

2.7.2.3 CSRs (Control and Status Registers)

This section describes the supported Control and Status Registers.

Control and Status Registers that are not specified in RISC-V spec but must be clarified in the respective implementations are described in this section.

2.7.2.3.1 Supported Control and Status Registers

The supported Control and Status Registers (CSRs) are listed in Table 2.13. All CSRs are 32-bit wide.

In the table, the Privilege column describes the accessible privileged mode and read/write access.

- The first character of the 3-letter notation in the Privilege column represents privileged mode. Access is possible in this mode with privilege mode higher than the described privilege mode
- The second and third letters represent read/write attributes. RO indicates read-only. RW indicates read/write.

The following CSRs are custom CSRs:

- mstackctrl
- mstacklimit

Table 2.13 List of supported CSRs (1 of 2)

Number	Privilege	Name	Description
Machine Information Registers			
0xF11	MRO	mvendorid	Vendor ID
0xF12	MRO	marchid	Architecture ID
0xF13	MRO	mimpid	Implementation ID
0xF14	MRO	mhartid	Hardware thread ID
Machine Trap Setup			
0x300	MRW	mstatus	Machine status register
0x301	MRW	misa	ISA and extensions
0x304	MRW	mie	Machine interrupt-enable register
0x305	MRW	mtvec	Machine trap-handler base address
0x307	MRW	mtvt	Machine trap-handler vector table base address
Machine Trap Handling			
0x340	MRW	mscratch	Scratch register for machine trap handlers
0x341	MRW	mepc	Machine exception program counter
0x342	MRW	mcause	Machine trap cause
0x343	MRW	mtval	Machine bad address or instruction
0x344	MRW	mip	Machine interrupt pending

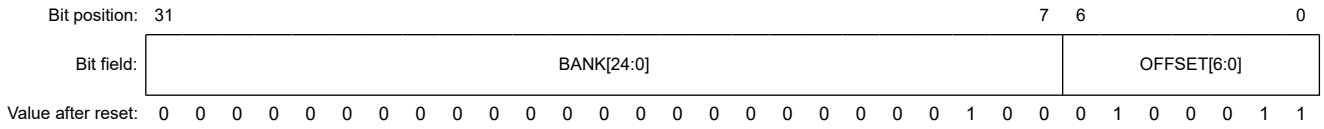
Table 2.13 List of supported CSRs (2 of 2)

Number	Privilege	Name	Description
0x345	MRW	mnxti	Machine interrupt handler address and enable modifier
0x346	MRW	mintstatus	Machine current interrupt levels
0x347	MRW	mintthresh	Machine interrupt-level threshold
0x349	MRW	mscratchcswl	Machine conditional scratch swap on level change
Machine Counter/Timers			
0xB00	MRW	mcycle	Machine cycle counter
0xB02	MRW	minstret	Machine instruction-retired counter
0xB80	MRW	mcycleh	Upper 32 bits of mcycle, RV32I only
0xB82	MRW	minstreth	Upper 32 bits of minstret, RV32I only
Machine Stack Overflow Detection			
0x7c0	MRW	mstackctrl	Machine stack overflow detection control
0x7c1	MRW	mstacklimit	machine stack limit

When accessing CSR, an illegal instruction exception occurs if one of the following conditions is true:

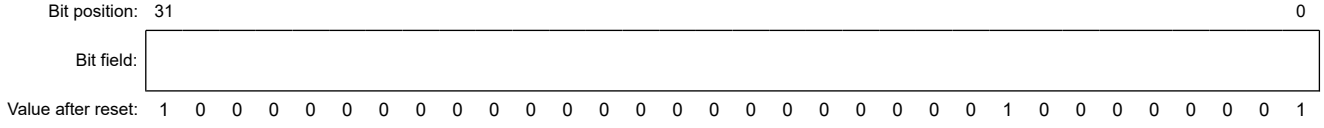
- Accessing an unsupported CSR number
- Accessing a CSR that does not have the required privilege attributes
- Writing to read-only registers.

2.7.2.3.2 mvendorid



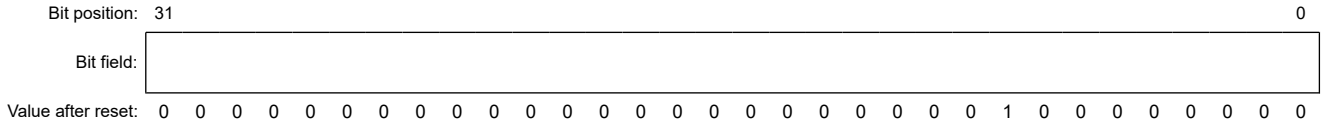
Bit	Symbol	Function	R/W
6:0	OFFSET[6:0]	The encoded version of JEDEC manufacturer ID. 0x00000223 is read.	R
31:7	BANK[24:0]		R

2.7.2.3.3 marchid



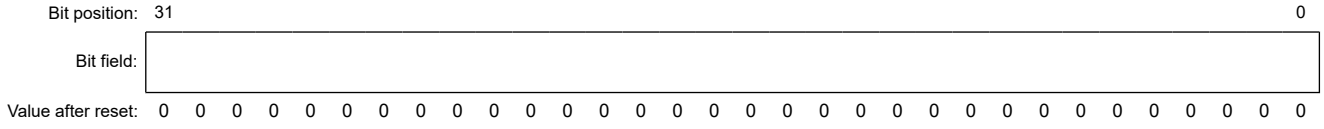
Bit	Symbol	Function	R/W
31:0	n/a	0x80000101 is read	R

2.7.2.3.4 mimpid



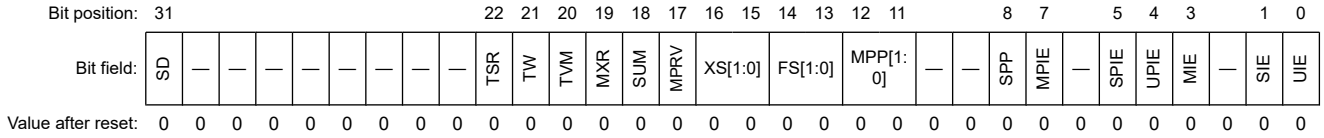
Bit	Symbol	Function	R/W
31:0	n/a	0x00000100 is read	R

2.7.2.3.5 mhartid



Bit	Symbol	Function	R/W
31:0	n/a	Represents a HART ID. 0 is read.	R

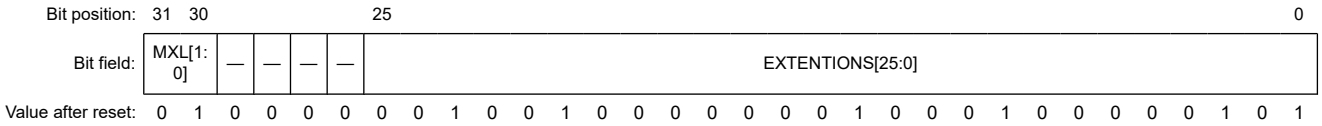
2.7.2.3.6 mstatus



Bit	Symbol	Function	R/W
0	UIE	U-mode Interrupt Enable Fixed to 0.	R/W
1	SIE	S-mode Interrupt Enable Fixed to 0.	R/W
2	—	This bit is read as 0. The write value should be 0.	R
3	MIE	M-mode Interrupt Enable Interrupts addressed to M mode when Hart is in M mode are globally disabled when MIE = 0 and globally enabled when MIE = 1. Interrupts addressed for M mode when Hart is in S or U mode are globally enabled regardless of MIE setting.	R/W
4	UPIE	U-mode Previous Interrupt Enable Fixed to 0.	R/W
5	SPIE	S-mode Previous Interrupt Enable Fixed to 0.	R/W
6	—	This bit is read as 0. The write value should be 0.	R
7	MPIE	M-mode Previous Interrupt Enable This bit indicates MIE prior to trapping into machine mode. When MRET instruction is executed, MPIE is copied from MPIE to MIE and set to 1.	R/W
8	SPP	S-mode Previous Privilege mode Fixed to 0.	R/W

Bit	Symbol	Function	R/W
10:9	—	These bits are read as 0. The write value should be 0.	R
12:11	MPP[1:0]	M-mode Previous Privilege mode This field indicates the privilege level prior to trapping into machine mode. When an MRET instruction is executed, MIE is set to MPIE and MPIE is set to 1. Writing any value other than 00 or 11 to this field has no effect. Do not write any value other than 11. When executing an MRET instruction other than returning from a trap handler, write 11 to this field before executing the instruction. 0 0: U mode 1 1: M mode	R/W
14:13	FS[1:0]	Fixed to 0.	R/W
16:15	XS[1:0]	Fixed to 0.	R/W
17	MPRV	Modify PRiVilege If MPRV = 0, the load and store behave normally using the current privileged-mode translation and protection mechanism. When MPRV = 1, the load and store memory addresses are translated and protected as if the present privileged mode is MPP one. Instruction address-translation and protection are not affected by MPRV setting.	R/W
18	SUM	Permit Supervisor User Memory Access Fixed to 0.	R/W
19	MXR	Make eXecutable Readable Fixed to 0.	R/W
20	TVM	Trap Virtual Memory Fixed to 0.	R/W
21	TW	Timeout Wait Fixed to 0.	R/W
22	TSR	Trap SRET Fixed to 0.	R/W
30:23	—	These bits are read as 0. The write value should be 0.	R
31	SD	Fixed to 0.	R/W

2.7.2.3.7 misa



Bit	Symbol	Function	R/W
25:0	EXTENTIONS[25:0]	EXTENTIONS Represents a supported ISA extension. Fixed to 0x0901105 (XUMICA). For more information on this field, see RISC-V ISA Specification.	R
29:26	—	These bits are read as 0. The write value should be 0.	R
31:30	MXL[1:0]	Machine XLEN Indicates the basic ISA length (RV32/RV64/RV128). It is fixed to 01 (RV32).	R

2.7.2.3.8 mie

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:	—	—	—	—	MEIE	—	SEIE	UEIE	MTIE	—	STIE	UTIE	MSIE	—	SSIE	USIE
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	USIE	Fixed to 0.	R/W
1	SSIE	Fixed to 0.	R/W
2	—	This bit is read as 0. The write value should be 0.	R
3	MSIE	Fixed to 0.	R/W
4	UTIE	Fixed to 0.	R/W
5	STIE	Fixed to 0.	R/W
6	—	This bit is read as 0. The write value should be 0.	R
7	MTIE	Fixed to 0.	R/W
8	UEIE	Fixed to 0.	R/W
9	SEIE	Fixed to 0.	R/W
10	—	This bit is read as 0. The write value should be 0.	R
11	MEIE	Fixed to 0.	R/W
31:12	—	These bits are read as 0. The write value should be 0.	R

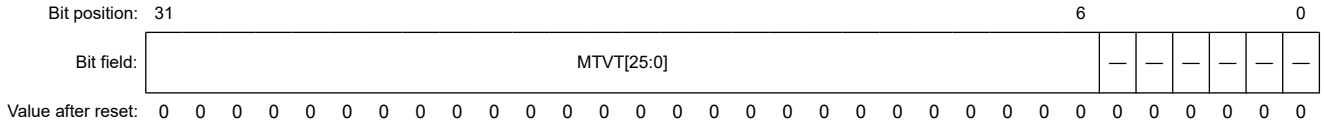
2.7.2.3.9 mtvec

Bit position:	31	6	5	0
Bit field:	BASE[25:0]		MODE[5:0]	
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		0 0 0 0 1 0	

Bit	Symbol	Function	R/W
5:0	MODE[5:0]	MODE Only b0 can be written to MODE with 0 or 1. However, the operation does not change depending on the value. xxxx0x: Non-CLIC (not supported) 00001x For interrupts, the following are set to PC: : mtvec[31:6] << 6 (non-vectored*2) M[mtvt[31:6] << 6 + 4 * excode*1] & ~ 1 (vectored*2) For exceptions, mtvec[31:6] << 6 is set to PC Others: Reserved	R/W
31:6	BASE[25:0]	BASE Indicates the base address that branches to the trap.	R/W

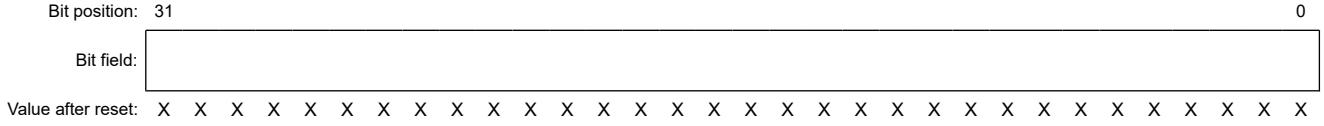
Note 1. excode: Exception code in Table 2.16.
Note 2. clicintattr[i]: Non-vectored when shv = 0 and vectored when shv = 1. For more information, see the *Core-Local Interrupt Controller (CLIC) RISC-V Privileged Architecture Extension* document.

2.7.2.3.10 mtvt



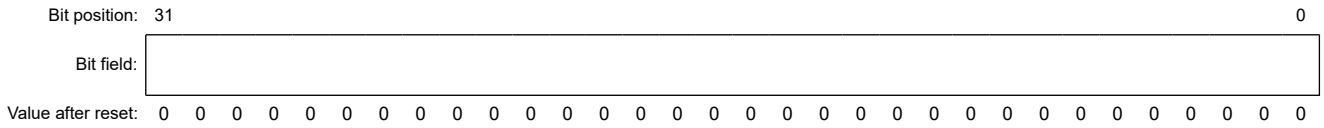
Bit	Symbol	Function	R/W
5:0	—	These bits are read as 0.	R
31:6	MTVT[25:0]	MTVT Holds the base address of the trap vector table. For more information, see section 2.7.2.3.9. mtvec.	R/W

2.7.2.3.11 mscratch



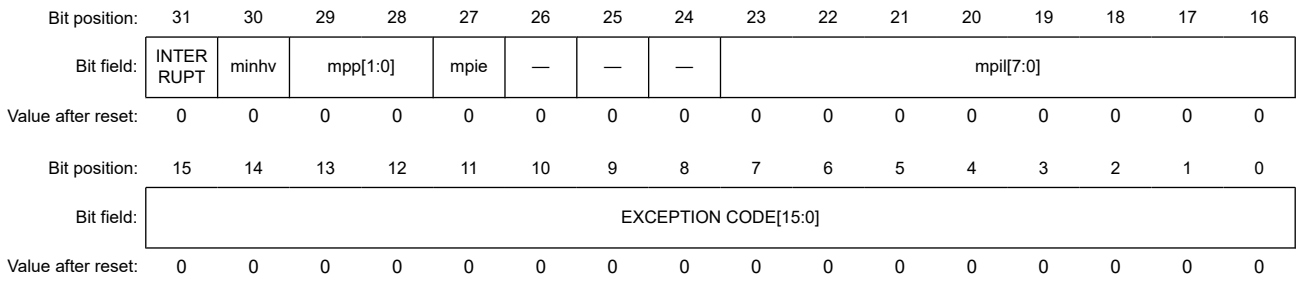
Bit	Symbol	Function	R/W
31:0	n/a	This register is a read/write register for M mode. It is typically used to hold a pointer to a local context space in M mode and is swapped with a general-purpose register when entering an M mode trap handler. The value after reset is undefined.	R/W

2.7.2.3.12 mepc



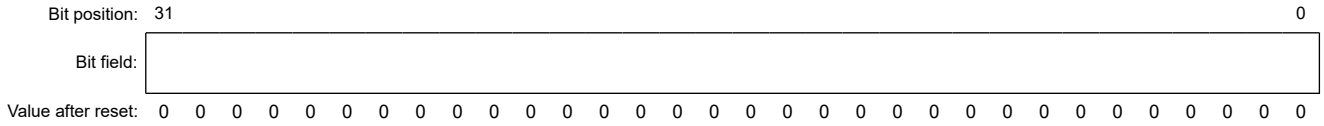
Bit	Symbol	Function	R/W
31:0	n/a	When a trap enters M mode, the address of the instruction that caused the exception or that was interrupted by the interrupt is written. Bit 0 is fixed to 0.	R/W

2.7.2.3.13 mcause



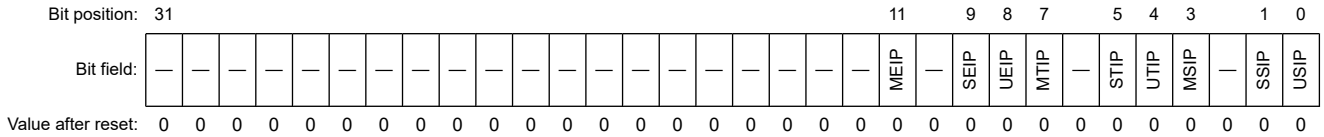
Bit	Symbol	Function	R/W
15:0	EXCEPTION CODE[15:0]	EXCEPTION CODE Represents an exception code for a trap. Updated when trap moves to M mode. See Table 2.16 for exception codes. For writes, only supported exception code writes are valid. If any other value is written, the operation is undefined.	R/W
23:16	mpil[7:0]	These bits indicate the interrupt level prior to trapping into machine mode.	R/W
26:24	—	These bits are read as 0. The write value should be 0.	R
27	mpie	mstatus.mpie mirror bit This bit is the same as mstatus.mpie. The operation when it is written is the same as mstatus.mpie.	R/W
29:28	mpp[1:0]	mstatus.mpp mirror bits These bits are the same as mstatus.mpp. The operation when they are written is the same as mstatus.mpp.	R/W
30	minhv	It is set at the start of hardware vectoring and cleared when hardware vectoring is successfully completed. This bit is fixed to 0. If an error occurs during hardware vectoring, an NMI occurs.	R/W
31	INTERRUPT	Indicates whether the trap is an exception or interrupt. Updated when a trap is taken into M-mode. 0: Exception 1: Interrupt	R/W

2.7.2.3.14 mtval



Bit	Symbol	Function	R/W
31:0	n/a	Represents exception-specific information for software to handle traps. Updated when a trap is taken into M-mode. For the value to be set in mtval when a trap occurs, see Table 2.19.	R/W

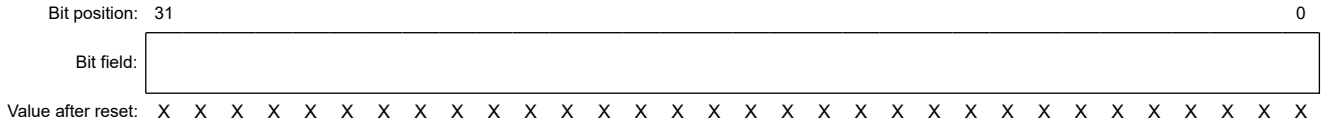
2.7.2.3.15 mip



Bit	Symbol	Function	R/W
0	USIP	Fixed to 0.	R/W
1	SSIP	Fixed to 0.	R/W
2	—	This bit is read as 0. The write value should be 0.	R
3	MSIP	Fixed to 0.	R
4	UTIP	Fixed to 0.	R/W
5	STIP	Fixed to 0.	R/W
6	—	This bit is read as 0. The write value should be 0.	R
7	MTIP	Fixed to 0.	R

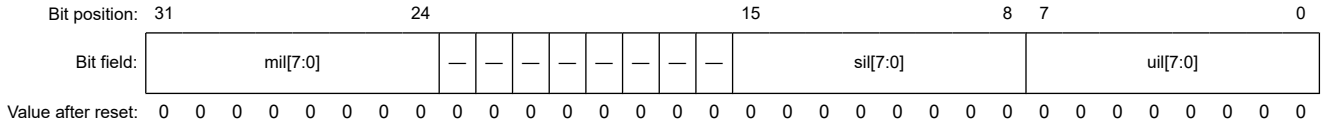
Bit	Symbol	Function	R/W
8	UEIP	Fixed to 0.	R/W
9	SEIP	Fixed to 0.	R/W
10	—	This bit is read as 0. The write value should be 0.	R
11	MEIP	Fixed to 0.	R
31:12	—	These bits are read as 0. The write value should be 0.	R

2.7.2.3.16 mnxti



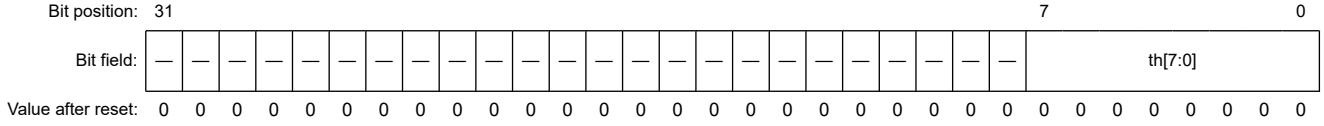
Bit	Symbol	Function	R/W
31:0	n/a	The operation differs depending on whether there is no acknowledgeable pending horizontal interrupt source or if vectored is set or not. When there are no pending interrupts, or when pending interrupts are vectored, the read value is 0. If written, the write value is ignored. Otherwise, the value read is a pointer to an entry in the trap handler table corresponding to the pending interrupt. If it is written, the operation is as follows: ● Writes are written to mstatus ● Accepts a pending horizontal interrupt ● For edge interrupts, the interrupt source is cleared For more information, see the *Smclic Core-Local Interrupt Controller (CLIC) RISC-V document*. Horizontal interrupt is a term that corresponds to *horizontal interrupt* in the *Smclic CLIC RISC-V specification*. Interrupts when the current privileged mode is M mode are called horizontal interrupts.	R/W

2.7.2.3.17 mintstatus



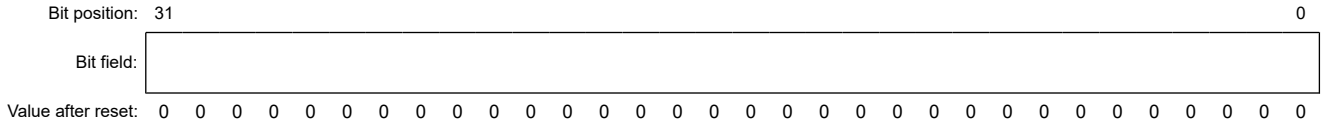
Bit	Symbol	Function	R/W
7:0	uil[7:0]	Fixed to 0.	R
15:8	sil[7:0]	Fixed to 0.	R
23:16	—	These bits are read as 0.	R
31:24	mil[7:0]	Indicates the current interrupt level.	R

2.7.2.3.18 mintthresh



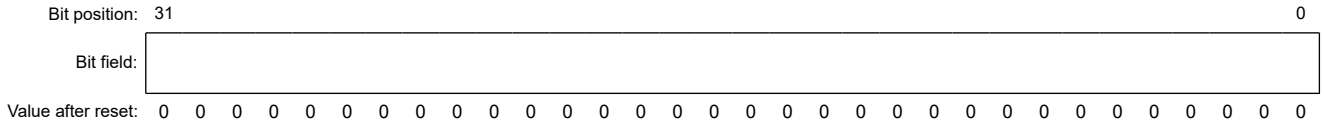
Bit	Symbol	Function	R/W
7:0	th[7:0]	This is the setting value of the interrupt level threshold.	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R

2.7.2.3.19 mscratchcswl



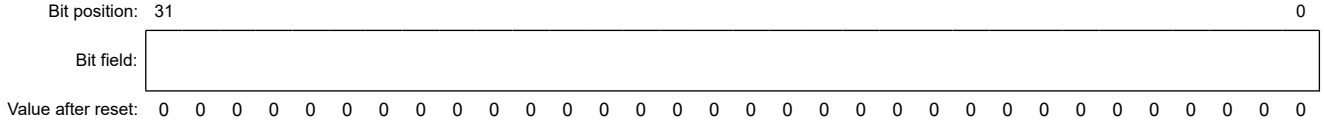
Bit	Symbol	Function	R/W
31:0	n/a	This register is used to help swap stack pointers between interrupt codes and non-interrupt codes. The operation of CSRRW instruction for this CSR varies depending on the conditions: ● if (mcause.pil = = 0) != (mintstatus.mil = = 0): – The write values are written to mscratch – The read value is read from mscratch ● if (mcause.pil = = 0) = = (mintstatus.mil = = 0): – The write values are ignored – The read value is read as a write value Operation is undefined when any of the following instructions are executed for this CSR: ● CSRRW instruction where rd or rs1 is x0 ● Instructions other than CSRRW (CSRRWI, CSRRC, CSRRCI, CSRRS, CSRRSI)	R/W

2.7.2.3.20 mcycle



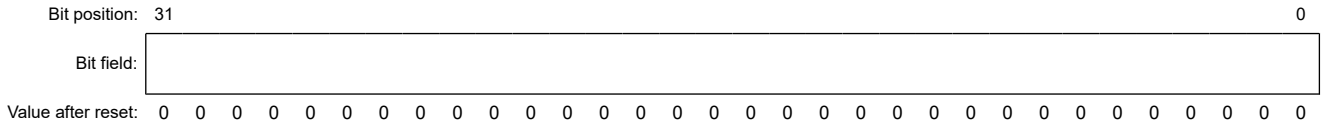
Bit	Symbol	Function	R/W
31:0	n/a	This register counts the elapsed cycles. The counters are 64 bits. The lower 32 bits can be accessed with the mcycle and upper 32 bits with the mcycleh register. Writing takes precedence over counting up. The value read by CSRRW instruction is read before counting up. For example, t0 is 0 in the following cases (depending on the state of the pipeline, t0 may be greater than 0 due to a stall): ● CSRWI mcycle, 0 ● CSRR t0, mcycle	R/W

2.7.2.3.21 minstret



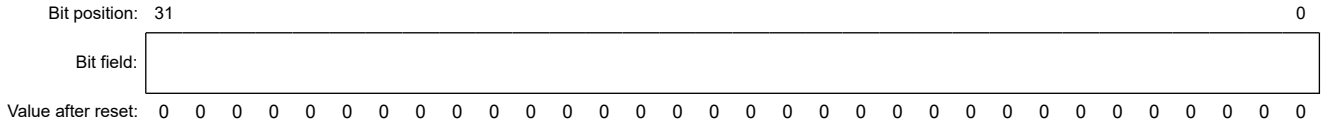
Bit	Symbol	Function	R/W
31:0	n/a	This register counts the number of instructions that have been executed. The counters are 64 bits. The lower 32 bits can be accessed with the minstret and upper 32 bits with the minstreth register. Writing takes precedence over counting up. The value read by CSRRW instruction is read before counting up. For example, t0 is 0 in the following cases: ● CSRWI minstret, 0 ● CSRR t0, minstret EBREAK and ECALL instructions are not counted up by the instruction. This is because an exception is always generated by these instructions and the instruction execution has not completed.	R/W

2.7.2.3.22 mcycleh



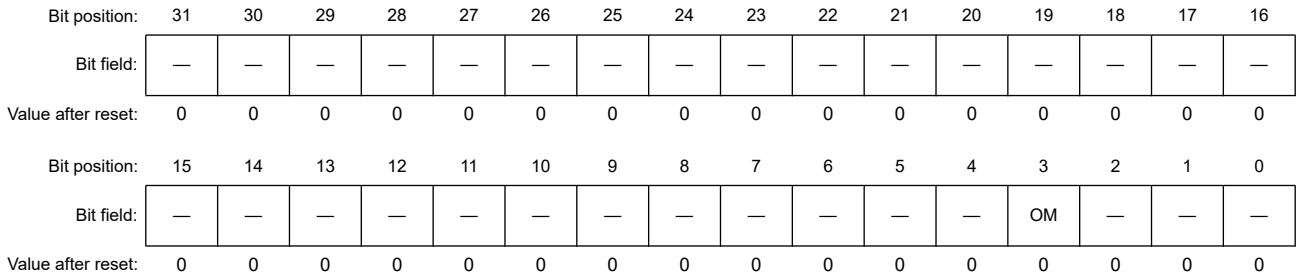
Bit	Symbol	Function	R/W
31:0	n/a	This register counts the elapsed cycles. See section 2.7.2.3.20. mcycle register for more information.	R/W

2.7.2.3.23 minstreth



Bit	Symbol	Function	R/W
31:0	n/a	This register counts the number of instructions that have been executed. See section 2.7.2.3.21. minstret for more information.	R/W

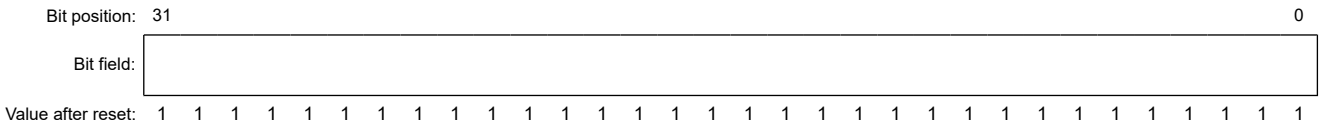
2.7.2.3.24 mstackctrl



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	OM	This is a setting for stack overflow detection in M mode. If a stack overflow exception occurs, it is cleared to 0 regardless of the privilege mode in which it occurred. 0: Detection disabled 1: Detection enabled	R/W
31:4	—	These bits are read as 0. The write value should be 0.	R/W

2.7.2.3.25 mstacklimit



Bit	Symbol	Function	R/W
31:0	n/a	This register sets the limit address of the stack. When overflow detection is enabled, an exception occurs if an attempt is made to save or restore the stack after exceeding the set value. Operation is performed when the LOAD/STORE/LR/SC/AMO instruction is executed with overflow detection enabled in the privileged mode. A stack overflow exception occurs when the instruction base register is "sp" and the load/store address is lower than Limit. If overflow detection is disabled in the current privileged mode, or if the executed instruction is other than a load/store/LR/SC/AMO instruction, this exception is not generated.	R/W

2.7.3 Instructions

2.7.3.1 Instruction List

Table 2.14 describes the instructions supported by the Instruction Set Architecture (ISA) of the CPU.

Table 2.14 List of ISA supported instructions for CPU (1 of 4)

ISA	Instruction name
RV32I	ADDI
	SLTI
	SLTIU
	ANDI
	ORI
	XORI
	SLLI
	SRLI
	SRAI
	LUI
	AUIPC
	ADD
	SLT
	SLTU
	AND
	OR
	XOR
	SLL
	SRL
	SUB
	SRA
	JAL
	JALR
	BEQ
	BNE
	BLT
	BLTU
	BGE
	BGEU
	LW
	LH
	LHU
	LB
	LBU
	SW
	SH
	SB
	FENCE
	ECALL
	EBREAK

Table 2.14 List of ISA supported instructions for CPU (2 of 4)

ISA	Instruction name
M	MUL
	MULH
	MULHU
	MULHSU
	DIV
	DIVU
	REM
	REMU
A	LR.W
	SC.W
	AMOSWAP.W
	AMOADD.W
	AMOAND.W
	AMOOR.W
	AMOXOR.W
	AMOMAX.W
	AMOMAXU.W
	AMOMIN.W
	AMOMINU.W

Table 2.14 List of ISA supported instructions for CPU (3 of 4)

ISA	Instruction name
C	C.LWSP
	C.SWSP
	C.LW
	C.SW
	C.J
	C.JAL
	C.JR
	C.JALR
	C.BEQZ
	C.BNEZ
	C.LI
	C.LUI
	C.ADDI
	C.ADDI16SP
	C.ADDI4SPN
	C.SLLI
	C.SRLI
	C.SRAI
	C.ANDI
	C.MV
	C.ADD
	C.AND
	C.OR
	C.XOR
	C.SUB
	C.NOP
	C.EBREAK
Zifencei	FENCE.I
Zicsr	CSRRW
	CSRRS
	CSRRC
	CSRRWI
	CSRRSI
	CSRRCI
Privileged	MRET
	WFI
Zba	SH1ADD
	SH2ADD
	SH3ADD

Table 2.14 List of ISA supported instructions for CPU (4 of 4)

ISA	Instruction name
Zbb	ANDN
	ORN
	XNOR
	CLZ
	CTZ
	CPOP
	MAX
	MAXU
	MIN
	MINU
	SEXT.B
	SEXT.H
	ZEXT.H
	ROL
	ROR
	RORI
	ORC.B
	REV8
Zbs	BCLR
	BCLRI
	BEXT
	BEXTI
	BINV
	BINVI
	BSET
	BSETI

ISA defined in RISC-V Standard is subject to RISC-V ISA specification.

2.7.3.2 Implementation-dependent Specifications

This section describes the implementation-dependent specification of the instructions specified in RISC-V Standard.

2.7.3.2.1 Load/Store/AMO Instruction Misaligned Access

Misaligned access is not supported. If an attempt is made to perform a misaligned access, a load address misaligned exception or a store/AMO address misaligned exception occurs.

2.7.4 Privilege Mode

Only M (Machine) mode is supported. The privilege mode after reset is M.

2.7.5 Trap

Table 2.15 describes terms related to trap. In this chapter, "trap (or interrupt exception) addressed to a privileged mode" means that the trap is delegated to that privileged mode. For example, a user software interrupt is an interrupt addressed to M mode if it is not delegated, an interrupt addressed to S mode if it is delegated to S mode, or an interrupt addressed to U mode if it is delegated to U mode.

Table 2.15 Description of trap-related terms

Term	Definition
Exception	An exception is an unusual situation that can occur in conjunction with the execution of an instruction.
Interrupt	An interrupt is an asynchronous event external to Hart that causes an unexpected transition of control.
Trap	A trap refers to the transition of control to a trap handler caused by an exception or interrupt.

2.7.5.1 Trap List

The CPU does not support S-mode and user-level interrupts and so it has no delegation capability. Therefore, all traps are traps addressed to M mode.

Table 2.16 Trap list

Exception/ interrupt	Exception code	Trap name	✓: Support —: Not supported
Interrupt	0	NMI	✓
Interrupt	*1	CLIC interrupt	✓
Exception	0	Instruction address misalignment exception	—
Exception	1	Instruction access fault exception	—
Exception	2	Illegal instruction exceptions	✓
Exception	3	Breakpoint exception (EBREAK)	✓
Exception	3	Breakpoint exceptions (Load/ Store/AMO)	✓
Exception	3	Breakpoint exception (instruction address)	✓
Exception	4	Load address misalignment exception	✓
Exception	5	Load access fault exception	—
Exception	6	Store/AMO address misalignment exception	✓
Exception	7	Store/AMO access fault exception	—
Exception	8	Environment call exception from U mode	—
Exception	9	Environment call exception from S mode	—
Exception	11	Environment call exception from M mode	✓
Exception	12	Instruction page fault exception	—
Exception	13	Load page fault exception	—
Exception	15	Store/AMO page fault exception	—
Exception	24	Stack overflow exception	✓

Note 1. The exception code is the CLIC interrupt ID.

2.7.5.2 Trap Priority

The priority of interrupts and the priority of synchronous exceptions that an instruction can generate are shown in Table 2.17 and Table 2.18. The synchronous exception has a lower priority than all interrupts.

Table 2.17 Interrupt priority

Priority	Exception code	Trap name	✓: Support —: Not supported
High	0	NMI	✓
Low	*1	CLIC interrupt	✓

Note 1. The exception code is the CLIC interrupt ID.

Table 2.18 Exception priority

Priority	Exception code	Trap name	✓: Support —: Not supported
High	3	Breakpoint exception (instruction address)	✓
	12	Instruction page fault exception	—
	1	Instruction access fault exception	—
	2	Illegal instruction exceptions	✓
	0	Instruction address misalignment exception	—
	11	Environment call exception from M mode	✓
	9	Environment call exception from S mode	—
	8	Environment call exception from U mode	—
	3	Breakpoint exception (EBREAK)	✓
	3	Breakpoint (load/store /AMO)	✓
	6	Store /AMO address misalignment exception	✓
	4	Load address misalignment exception	✓
	24	Stack overflow exception	✓
	15	Store/AMO page fault exception	—
	13	Load page fault exception	—
	7	Store/AMO access fault exception	—
Low	5	Load access fault exception	—

2.7.5.3 Trap Operation

The following are hardware operations when trap addressed to M mode occurs:

- Privileged mode ← M mode
- mepc ← PC (PC of the instruction that was interrupted by the interrupt or caused the exception)
- mstatus.MPIE ← mstatus.MIE
- mstatus.MIE ← 0
- mstatus.MPP ← Privilege mode before the trap occurred
- mtval ← See section 2.7.5.5. mtval after Trap
- mcause.EXCEPTION_CODE ← Values depend on the type of trap (see Table 2.16)
- mcause.INTERRUPT ← 1 if the trap source is an interrupt. Otherwise, 0
- mcause.MPIL ← mintstatus.mil

- mintstatus.mil ← When the trap source is a CLIC interrupt : Interrupt level of the acknowledged interrupt. When the trap source is other than a CLIC interrupt : The previous value is retained.
- pc ← Addresses determined by trap.

Note: For environmental call exceptions (ECALL instructions) and breakpoint exceptions (EBREAK), mepc contains the address of EBREAK or ECALL instruction. Therefore, to continue processing from the instruction following ECALL or EBREAK instruction after returning from the trap, mepc must be next address prior to returning from the trap.

2.7.5.4 Return from Trap Operation

The following are hardware operations when mret instruction is executed:

- Privileged mode ← msatus.MPP
- mstatus.MIE ← mstatus.MPIE
- mstatus.MPIE ← 1
- mstatus.MPP ← 0 (U mode)
- mintstatus.mil ← mcause.MPIL
- pc ← mepc

Reservation sets registered with LR instruction are cleared when mret is executed.

2.7.5.5 mtval after Trap

The mtval value after trap depends on the trap. This is shown in Table 2.19.

Table 2.19 mtval stored when a trap occurs

Exception interrupt	Exception code	Trap name	mtval values after trap
Interrupt	0	NMI	0
Interrupt	*1	CLIC interrupt	0
Exception	0	Instruction address misalignment exception	—
Exception	2	Illegal instruction exceptions	0
Exception	3	Breakpoint exception (EBREAK)	0
Exception	3	Breakpoint exception (instruction address)	PC of instructions that hit a breakpoint
Exception	3	Breakpoint (Load/Store/AMO)	Address of the Load/Store/AMO that hit the breakpoint
Exception	4	Load address misalignment exception	Address of the Load/Store/AMO that caused the exception
Exception	6	Store/AMO address misalignment exception	
Exception	8	Environment call exception from U mode	—
Exception	9	Environment call exception from S mode	—
Exception	11	Environment call exception from M mode	0
Exception	12	Instruction page fault exception	—
Exception	13	Load page fault exception	
Exception	15	Store/AMO page fault exception	
Exception	24	Stack overflow exception	Address of the Load/Store/AMO that caused the exception

Note: —: Traps that are not supported.
Note 1. CLIC interrupt ID is the exception code number.

2.7.6 Microarchitecture Specification

2.7.6.1 Features

The main features of this CPU are as follows.

- Minimum instruction execution time: 1 cycle
- Pipeline configuration: Single issue in-order 2-stage pipeline
- Branch predictor: Equipped with dynamic branch predictor
- Endian: Little endian for both instruction fetch and data access
- Address space: 4 GB
- Stack overflow detection: Detect stack pointer overflow during stack save/restore
- CLIC: 16 interrupt priority levels, selective hardware vectoring support
- NMI: Supported
- PMP: None
- Counters: Equipped with mcycle/mcycleh, minstret/minstreth CSR

2.7.6.2 Number of Cycles

Table 2.20 shows the number of cycles and latency for instruction execution. The throughput and latency of instructions with memory accesses are cycles with no-wait memory accesses from CPU.

Table 2.20 Instruction throughput and latency

Instruction type	Throughput (cycle/instruction)	Latency (Cycle)
ALU instruction	1	1
Multiplication instructions	1	1
Division instruction	35 (DIV, DIVU) 36 (REM, REMU)	35 (DIV, DIVU) 36 (REM, REMU)
Branch instruction	2*1	—
Load instruction	1	2
Store instruction	1	—
SC instruction	1	1
AMO instruction	3	3
CSR instruction	1	1
Trap instruction	2	—
Environment call/break instruction	3	—
FENCE instruction	1 (FENCE) 2 (FENCE.I)	—
WFI instruction	4 ~	—

Note 1. The throughput is 1 if the branch prediction succeeds.

Table 2.21 Correspondence between instruction types and instructions (1 of 2)

Instruction type	Corresponding instruction
ALU instruction	ADDI, SLTI, SLTIU, ANDI, ORI, XORI, SLLI, SRLI, SRAI, LUI, AUIPC, ADD, SLT, SLTU, AND, OR, XOR, SLL, SRL, SUB, SRA, SH1ADD, SH2ADD, SH3ADD, ANDN, ORN, XNOR, CLZ, CTZ, CPOP, MAX, MAXU, MIN, MINU, SEXT.B, SEXT.H, ZEXT.H, ROL, ROR, RORI, ORC.B, REV8, BCLR, BCLRI, BEXT, BEXTI, BINV, BINVI, BSET, BSETI
Multiplication instruction	MUL, MULH, MULHU, MULHSU

Table 2.21 Correspondence between instruction types and instructions (2 of 2)

Instruction type	Corresponding instruction
Division instruction	DIV, DIVU, REM, REMU
Branch instruction	JAL, JALR, BEQ, BNE, BLT, BLTU, BGE, BGEU
Load instruction	LW, LH, LHU, LB, LBU, LR.W
Store instruction	SW, SH, SB
SC instruction	SC.W
AMO instruction	AMOSWAP.W, AMOADD.W, AMOAND.W, AMOOR.W, AMOXOR.W, AMOMAX.W, AMOMAXU.W, AMOMIN.W, AMOMINU.W
CSR instruction	CSRRW, CSRRS, CSRRC, CSRRWI, CSRRSI, CSRRCI
Trap instruction	MRET
Environment call/break instruction	ECALL, EBREAK
FENCE instruction	FENCE, FENCE.I
WFI instruction	WFI

2.7.6.3 Number of Interrupt Response Cycles

Table 2.22 shows the number of interrupt response cycles. This is the number of cycles when the memory access from CPU is processed with no wait.

Table 2.22 Number of interrupt response cycles

Types of interrupt requests and processing details	Number of cycles
CPU Number of cycles from interrupt request notification to interrupt acknowledgment	N cycle (depending on the instruction being executed)
CPU Branch to exception handling routine	3 cycles (+2 cycles for selective hardware interrupt interrupts)

Table 2.23 shows the number of cycles N until interrupt acknowledgment.

Table 2.23 Number of interrupt acceptance cycles for each instruction type

Instruction type	Number of cycles until acceptance of interrupt
ALU instruction	0
Multiplication instruction	0
Division instruction	0 ~ 35
Branch instruction	0
Load instruction	0
Store instruction	0
SC instruction	0
AMO instruction	0 ~ 2
CSR instruction	0
Trap return instruction	0
Environment call/break instruction	0 ~ 1
FENCE instruction	0
WFI instruction	0 ~ 3

2.7.7 References

This section lists references and external references related to the RISC-V CPU Core Instruction Set Architecture Specification.

- RISC-V ISA Specification Volume 1, Unprivileged spec v.20191213
- RISC-V ISA Specification Volume 2, Privileged Spec v.20190608
- RISC-V Bit-Manipulation ISA-extensions Version 1.0.0-38-g865e7a7, 2021-06-28: Release candidate
- Smclic Core-Local Interrupt Controller (CLIC) RISC-V Privileged Architecture Extension Version 0.9-draft, 5/10/2022

2.8 RISC-V Debug Functional Specification

2.8.1 Overview

The debug function of this MCU complies with RISC-V External Debug Support Version 0.13.2. This section describes only the implementation-specific information.

Table 2.24 shows an overview of the debug system.

Table 2.24 Overview of debug system

Item	Description
Debug Transport Module (DTM)	This module supports interface with the external RISC-V platform. The MCU has only one DTM and JTAG is the interface protocol.
Debug Module Interface (DMI)	This module is the bus interface for access to DM registers.
Debug Module (DM)	This module supports the debug function of debug monitor program, the run control of hart, and the memory access. The MCU has only one DM and the number of harts this module supports is one.
Reset/halt/resume	This function supports the control for reset, halt, and resume.
Abstract commands	This function supports commands for access to GPR or CSR, the execution of debug monitor program and access to memory.
Program buffer	This function stores arbitrary instructions of debug monitor program. The buffer size is parameterized. This function supports 8 × 4 bytes.
Debug mode	This function supports debug mode, debug exception, debug return instruction, and single step. Each hart has this function.
Hardware trigger module	This function supports hardware breakpoint. The number of breakpoints for this MCU is 4. Each hart has this function.

2.8.1.1 Function List

Table 2.25 shows a list of functions the debug system supports.

Table 2.25 List of debug functions (1 of 2)

Function	Description
Abstract command	Argument width: 32 cmdtype = 0: GPRs, CSRs, PC (dpc) cmdtype = 1: Quick access support cmdtype = 2: Memory-mapped registers or memory Abstract command autoexec is supported
Program buffer	Support 8 × 4 bytes register
Version	Version is 2: The Debug Module (DM) is supported and it conforms to version 0.13 of this specification
Debug Module (DM)	All abstract command is supported. The number of DM is one.
CPU	Mprven is supported. Debug Scratch Register 0/1 is supported.

Table 2.25 List of debug functions (2 of 2)

Function	Description
Event trigger	Event trigger specification is defined in the Hart specification. The MCU supports the following: • 4 breakpoints • Only type 2 is supported • 32-bit size is the max

2.8.1.2 Address Map

Table 2.26 shows the system memory map of the debug module.

Table 2.26 System memory map of debug module

Address offset	Description	Definition
0x0360 to 0x037F	Program buffer	See Program Buffer i (progbufi) section
0x0380 to 0x038F	Abstract data 0 - 3	See Abstract Data n (datan) section

Table 2.27, Table 2.28, and Table 2.29 show the I/O registers of the debug module. Initial value is shown in I/O registers section.

Table 2.27 Registers of debug module

Name	Symbol	Address	Access size
Abstract Data 0	data0	0x004	32
Abstract Data 1	data1	0x005	32
Abstract Data 2	data2	0x006	32
Abstract Data 3	data3	0x007	32
Debug Module Control	dmcontrol	0x010	32
Debug Module Status	dmstatus	0x011	32
Halt Info	hartinfo	0x012	32
Abstract Control and Status	abstractcs	0x016	32
Abstract Command	command	0x017	32
Abstract Command Autoexec	abstractauto	0x018	32
Next Debug Module	nextdm	0x01D	32
Program Buffer 0-7	progbuf0-7	0x020-0x027	32
Authentication Data	authdata	0x030	32
Halt Summary 0	haltsum0	0x040	32

Table 2.28 Core registers of debug module

Name	Symbol	CSR addres	Access size
Debug Control and Status	dcsr	0x7B0	32
Debug PC	dpc	0x7B1	32
Debug Scratch Register 0	dscratch0	0x7B2	32
Debug Scratch Register 1	dscratch1	0x7B3	32

Table 2.29 Trigger registers of debug module (1 of 2)

Name	Symbol	CSR address	Access size
Trigger Select	tselect	0x7A0	32
Trigger Data 1	tdata1	0x7A1	32
Match Control	mcontrol	0x7A1 (type = 2)	32

Table 2.29 Trigger registers of debug module (2 of 2)

Name	Symbol	CSR address	Access size
Trigger Data 2	tdata2	0x7A2	32
Trigger Info	tinfo	0x7A4	32

2.8.2 cJTAG Interface

The MCU supports cJTAG incterface using cJTAG adapter. This section describes specification of the cJTAG adapter.

The cJTAG adapter in the MCU supports only the following functions:

- Oscan1 using short connect sequence
- Reset using escape sequence

The short connect sequence is described as follows:

```
EscapeSequence(10) // Escape sequence "Reset"
EscapeSequence(6) // Escape sequence "Selection"
TMS(0xC, 4) // OAC
TMS(0x8, 4) // EC
TMS(0x0, 4) // Check packet
```

If Escape sequence "Reset" is input, cJTAG adapter moves "Offline".

2.8.3 I/O Registers

2.8.3.1 Debug Module Registers

2.8.3.1.1 Abstract Data *n* (data*n*)

The MCU supports data0 to data3.

These registers can be accessed from hart. The memory map is shown in Table 2.26. These registers support read and write access.

2.8.3.1.2 dmcontrol : Debug Module Control

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	haltreq	resumereq	hartreset	ackhavereset	—	hasel	hartsello[9:0]									
Value after reset:	x	x	0	x	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:	hartselli[9:0]										—	—	setresethaltreq	clrresethaltreq	ndmreset	dmactive
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	x	x	0	0

Bit	Symbol	Function
0	dmactive	Supported
1	ndmreset	Supported
2	clrresethaltreq	Supported
3	setresethaltreq	Supported
5:4	—	These bits are read as 0. The write value should be 0.
15:6	hartselli[9:0]	Fixed to 0
25:16	hartsello[9:0]	Fixed to 0

Bit	Symbol	Function
26	hasel	Fixed to 0
27	—	This bit is read as 0. The write value should be 0.
28	ackhavereset	Supported
29	hartreset	Not supported
30	resumereq	Supported
31	haltreq	Supported

2.8.3.1.3 dmstatus : Debug Module Status

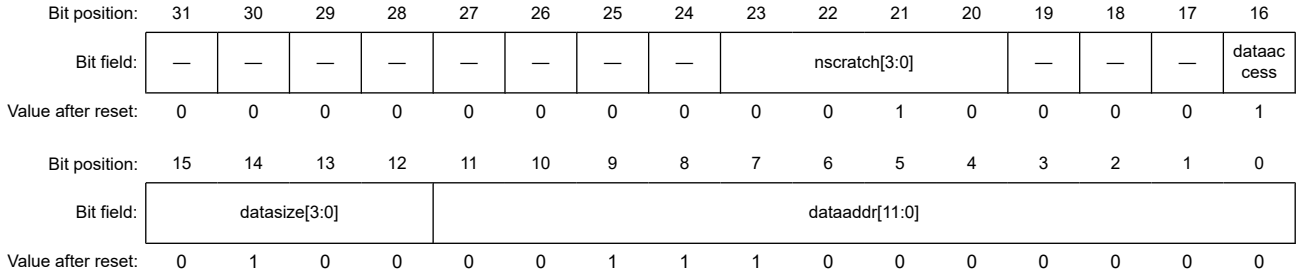
Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	impebreak	—	—	allhavereset	anyhavereset	allresumeack	anyresumeack
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	allnonexistent	anynonexistent	allunavail	anyunavail	allrunning	anyrunning	allhated	anyhalted	authenticated	authbusy	hasresethaltreq	confstrptrvalid	version[3:0]			
Value after reset:	x	x	x	x	x	x	x	x	User setting *1	0	1	0	0	0	1	0

Note 1. See section 2.4.1. Debug Authentication Mechanism and section 6, Option-Setting Memory.

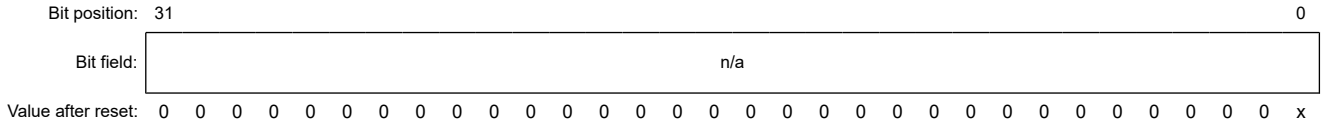
Bit	Symbol	Function
3:0	version[3:0]	Fixed to 2
4	confstrptrvalid	Fixed to 0
5	hasresethaltreq	Supported
6	authbusy	Supported
7	authenticated	Supported. For the initial value, see section 2.4.1. Debug Authentication Mechanism.
8	anyhalted	Supported
9	allhated	Supported
10	anyrunning	Supported
11	allrunning	Supported
12	anyunavail	Supported
13	allunavail	Supported
14	anynonexistent	Supported
15	allnonexistent	Supported
16	anyresumeack	Supported
17	allresumeack	Supported
18	anyhavereset	Supported
19	allhavereset	Supported
21:20	—	These bits are read as 0. The write value should be 0.
22	impebreak	Fixed to 0
31:23	—	These bits are read as 0. The write value should be 0.

2.8.3.1.4 hartinfo : Hart Info



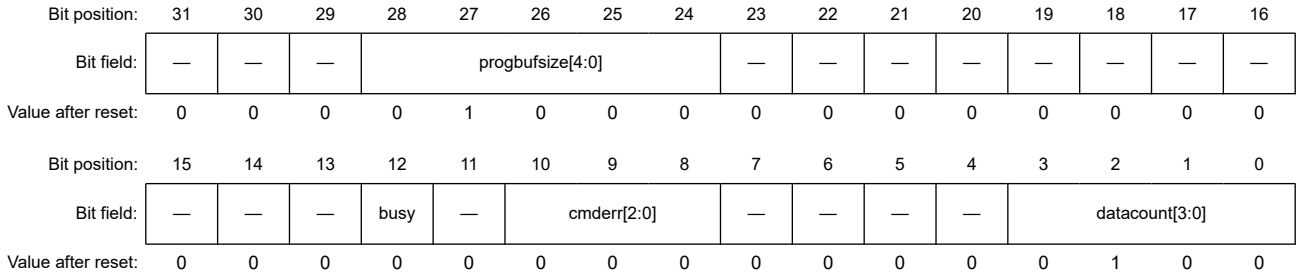
Bit	Symbol	Function
11:0	dataaddr[11:0]	Fixed to 0x380
15:12	datasize[3:0]	Fixed to 4
16	dataaccess	Fixed to 1
19:17	—	These bits are read as 0. The write value should be 0.
23:20	nscratch[3:0]	Fixed to 0x2
31:24	—	These bits are read as 0. The write value should be 0.

2.8.3.1.5 haltsum : Halt Summary 0



Bit	Symbol	Function
31:0	n/a	Only bit 0 is supported

2.8.3.1.6 abstractcs : Abstract Control and Status



Bit	Symbol	Function
3:0	datacount[3:0]	Fixed to 0x4
7:4	—	These bits are read as 0. The write value should be 0.
10:8	cmderr[2:0]	Supported
11	—	This bit is read as 0. The write value should be 0.
12	busy	Supported

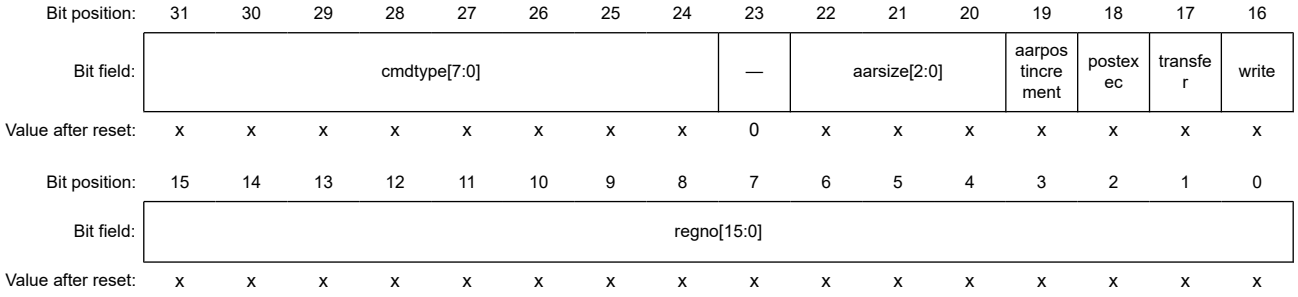
Bit	Symbol	Function
23:13	—	These bits are read as 0. The write value should be 0.
28:24	progbufsize[4:0]	Fixed to 8.
31:29	—	These bits are read as 0. The write value should be 0.

2.8.3.1.7 command : Abstract Command

Bit	Symbol	Function
23:0	control[23:0]	Refer to each command
31:24	cmdtype[7:0]	Refer to each command

If the hart becomes unavailable during the abstract command, the DM internal resource might be incompletely updated. When this situation occurs, the reset of DM is required through dmactive.

2.8.3.1.7.1 Access Register

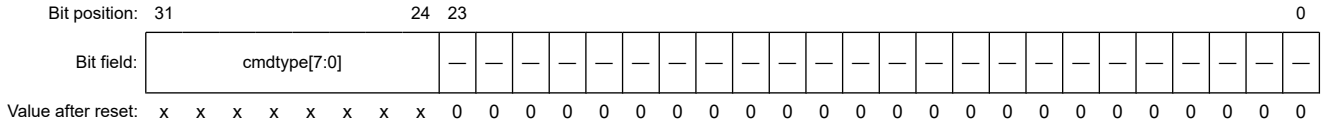


Bit	Symbol	Function
15:0	regno[15:0]	Supported
16	write	Supported
17	transfer	Supported
18	postexec	Supported
19	aarpostincrement	Supported
22:20	aarsize[2:0]	Only the value of 2 is supported.
23	—	The write value should be 0.
31:24	cmdtype[7:0]	This is 0 to indicate Access Register Command.

The relationship between regno and the registers is shown in the following table.

regno	register
0x0000-0x0fff	CSRs. The PC can be accessed here through dpc.
0x1000-0x101f	GPRs
0x1020-0xffff	Reserved

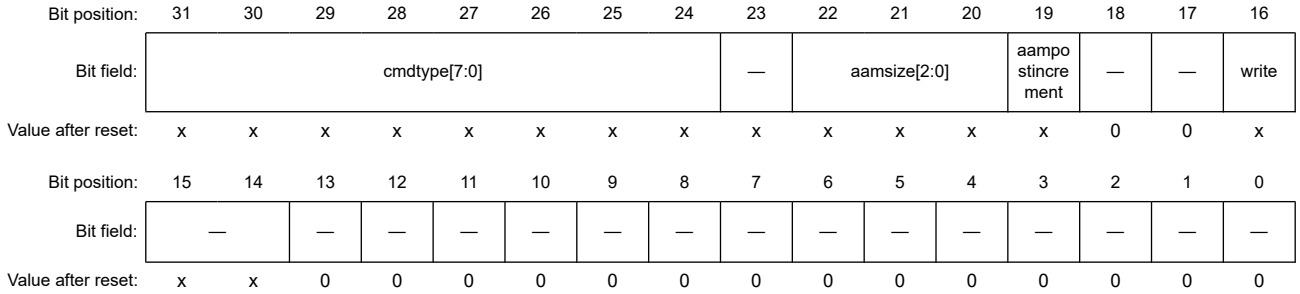
2.8.3.1.7.2 Quick Access



Bit	Symbol	Function
23:0	—	These bits are read as 0. The write value should be 0.
31:24	cmdtype[7:0]	This is 1 to indicate Quick Access Command.

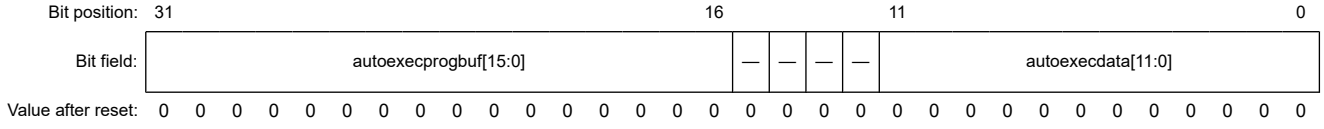
At the start of program buffer execution with QuickAccess, s0 and a0 are saved in dscratch0 and dscratch1, respectively. Therefore, s0 and a0 do not need to be saved in program buffer. If an exception occurs during program buffer execution in QuickAccess, s0 and a0 are restored from dscratch0 and dscratch 1, before Resume.

2.8.3.1.7.3 Access Memory



Bit	Symbol	Function
15:0	—	These bits are read as 0. The write value should be 0.
16	write	Supported
18:17	—	These bits are read as 0. The write value should be 0.
19	aampostincrement	Supported
22:20	aamsize[2:0]	Only 0, 1, and 2 are supported, 3 and 4 are not supported.
23	—	This bit is read as 0. The write value should be 0.
31:24	cmdtype[7:0]	This is 2 to indicate Access Memory Command.

2.8.3.1.8 abstractauto : Abstract Command Autoexec



Bit	Symbol	Function
11:0	autoexecdata[11:0]	Supported. The autoexecdata bit associated with the unsupported data bit is fixed to 0.
15:12	—	These bits are read as 0. The write value should be 0.

Bit	Symbol	Function
31:16	autoexecprogbuf[15:0]	Supported. The autoexecprogbuf bit associated with the unsupported progbuf bit is fixed to 0.

The autoexecprogbuf bit associated with the unsupported progbuf bit is fixed to 0.

The autoexecdata bit associated with the unsupported data bit is fixed to 0.

2.8.3.1.9 nextdm : Next Debug Module

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
31:0	n/a	Fixed to 0

2.8.3.1.10 progbuf*i* : Program Buffer *i*

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	x

Bit	Symbol	Function
31:0	n/a	Supported

The MCU has 8 program buffers. These registers can be read. The memory map is shown in Table 2.26. These registers support only read access. Write access is ignored.

2.8.3.1.11 Authentication Data (authdata)

The read data value of authdata is the value of written data.

2.8.3.2 Core Debug Registers

2.8.3.2.1 dcsr : Debug Control and Status

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	xdebugver[3:0]				—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	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:	ebreakm	—	ebreaks	ebreaku	stepie	stopcount	stoptime	cause[2:0]			—	mprven	nmip	step	prv[1:0]	
Value after reset:	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	1

Bit	Symbol	Function
1:0	prv[1:0]	0 or 3 can be set
2	step	Supported

Bit	Symbol	Function
3	nmip	Supported
4	mprven	Supported
5	—	This bit is read as 0. The write value should be 0.
8:6	cause[2:0]	Supported
9	stoptime	Supported. See section 2.5.9.2. DBGSTOPCR : Debug Stop Control Register.
10	stopcount	Supported
11	stepie	Supported
12	ebreaku	Supported
13	ebreaks	Fixed to 0
14	—	This bit is read as 0. The write value should be 0.
15	ebreakm	Supported
27:16	—	These bits are read as 0. The write value should be 0.
31:28	xdebugver[3:0]	Fixed to 4

2.8.3.2.2 dpc : Debug PC

Supported.

2.8.3.2.3 dscratch0, 1 : Debug Scratch Register 0, 1

This MCU has dscratch0 and dscratch1.

2.8.3.3 Trigger Registers

2.8.3.3.1 tselect : Trigger Select

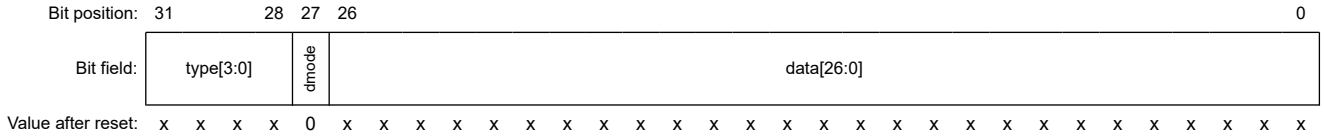
Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	Index[31:4]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	Index[31:4]												Index3	Index2	Index[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function
1:0	Index[1:0]	Supported
2	Index2	Fixed to 0
3	Index3	Fixed to 0
31:4	Index[31:4]	Fixed to 0

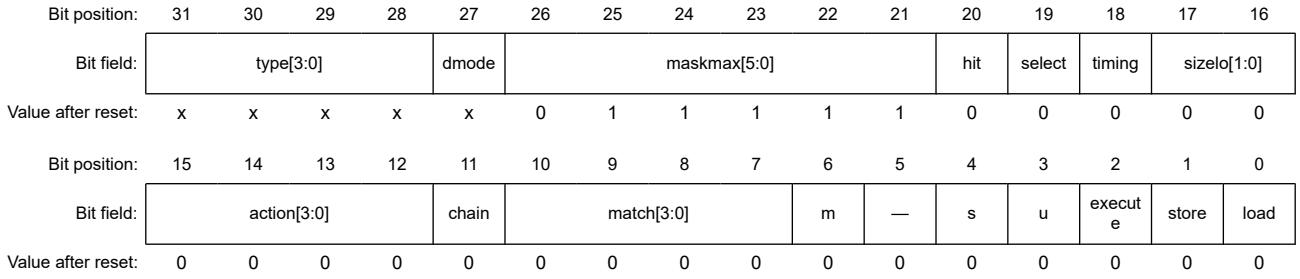
The MCU supports only 4 channels.

2.8.3.3.2 tdata1 : Trigger Data 1



Bit	Symbol	Function
26:0	data[26:0]	See each register according to its type
27	dmode	Supported
31:28	type[3:0]	MCU supports only type 2

2.8.3.3.3 mcontrol : Match Control



Bit	Symbol	Function
0	load	This bit must be set to 0 if execute is 1.
1	store	This bit must be set to 0 if execute is 1.
2	execute	This bit must be set to 0 if store or load is 1.
3	u	Supported
4	s	Fixed to 0
5	—	Fixed to 0
6	m	Supported
10:7	match[3:0]	0, 1, 2, 3, 4, and 5 are supported. Also, 8, 9, 12, 13 defined by Version 1.0.0-STABLE [see section 2.8.6. References] are supported.
11	chain	The MCU supports 2-chain only. So, this bit is supported in even trigger only.
15:12	action[3:0]	Supported. Must be set to 0 or 1.
17:16	sizelo[1:0]	The MCU supports 0, 1, 2, 3. Also, the bits are set to 0 when execute is set to 1.
18	timing	Fixed to 0. The MCU supports only timing of "before".
19	select	Fixed to 0. The MCU supports only instruction and access address trigger.
20	hit	Fixed to 0. Not supported.
26:21	maskmax[5:0]	The MCU supports NAPOT
27	dmode	See section 2.8.3.3.2. tdata1 : Trigger Data 1
31:28	type[3:0]	See section 2.8.3.3.2. tdata1 : Trigger Data 1

The supported chain combination condition is described in the following table.

Even trigger	Odd trigger
Even trigger	execute
store	store
load	load

2.8.3.3.4 tdata2 : Trigger Data 2

Supported.

2.8.3.3.5 tinfo : Trigger Info

Supported.

2.8.3.4 JTAG DTM Registers

2.8.3.4.1 IDCODE

Supported. The value is described in section 2.8.5. Product Information.

2.8.3.4.2 dtmcs : DTM Control and Status

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

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

Bit	Symbol	Function
3:0	version[3:0]	Fixed to 1
9:4	abits[5:0]	Fixed to 14
11:10	dmistat[1:0]	Supported
14:12	idle[2:0]	Fixed to 4
15	—	This bit is read as 0. The write value should be 0.
16	dmireset	Supported
17	dmihardreset	Supported
31:18	—	These bits are read as 0. The write value should be 0.

2.8.3.4.3 dmi : Debug Module Interface Access

Bit position:	47–34	33–24
Bit field:	address[13:0]	data[31:22]
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

Bit position:	23–2	1–0
Bit field:	data[21:0]	op[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

Bit	Symbol	Function
1:0	op[1:0]	Supported
33:2	data[31:0]	Supported
47:34	address[13:0]	Supported

Supported. To complete the request phase of DMI access, 2 additional cycles are required from Update-DR because DMI has asynchronous bridge. The additional cycle for access completion can be satisfied if the debugger reads the op bits to confirm the access result.

2.8.3.4.4 Bypass

Supported.

2.8.4 Restriction

Table 2.30 shows the restriction.

Table 2.30 Restriction

No.	Restriction
1	Input TMS = 1 for 5 cycles*1 after the start of Oscan1*2 to guarantee State = TestLogicReset.

Note 1. JTAG TCK cycle (Oscan1). Refer to IEEE1149.7.
Note 2. See section 2.2. cJTAG Interface.

2.8.5 Product Information

2.8.5.1 Product Information of Initial Value for Debug register

Register address	Section	Value
0x01 (IR address)	IDCODE	0x0E000447

2.8.6 References

1. *RISC-V External Debug Support Version 0.13.2 (d5029366d59e8563c08b6b9435f82573b603e48e)*
2. *RISC-V Debug Support Version 1.0.0-STABLE (c26287e8dde138de22ca26d7f214b457f90b380e)*

3. Operating Modes

3.1 Operating Mode Types and Selection

Table 3.1 shows the selection of operating modes by the mode-setting pin. For details on each of the operating modes, see section 3.2. Details of Operating Modes. Operation starts when the on-chip flash memory is 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
1	Single-chip mode
0	UART (SAU) boot mode

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 UART (SAU) Boot Mode

In this mode, the on-chip flash memory programming routine (UART (SAU) 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). For details, see section 35, Flash Memory. The MCU starts in UART (SAU) 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.

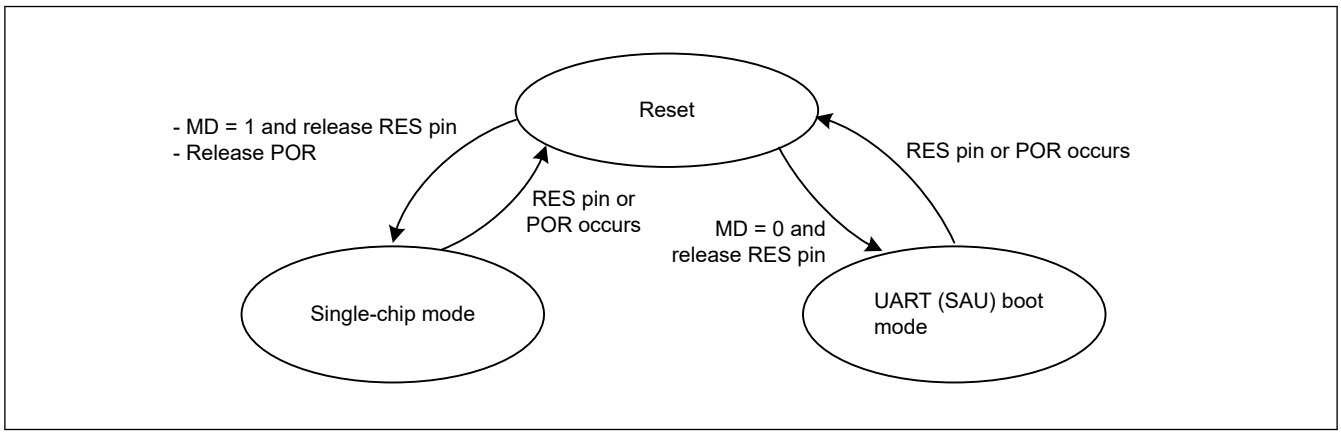
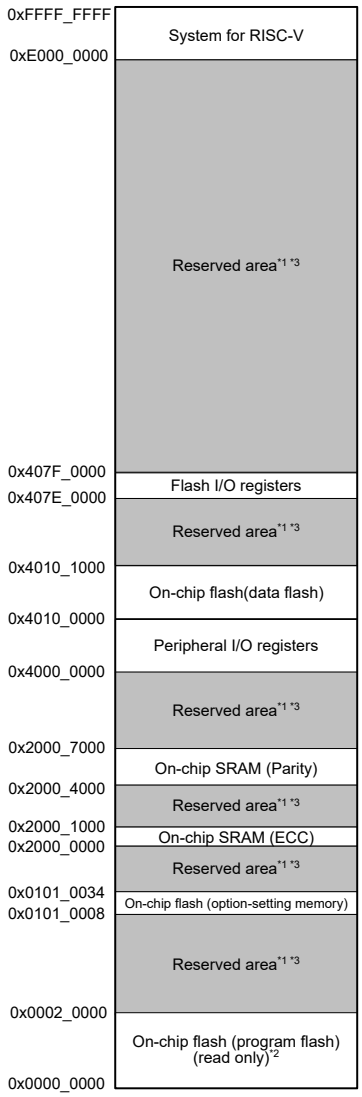


Figure 3.1 Mode-setting pin level and operating mode

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. Do not access reserved areas.
Note 2. Some regions are reserved for the option-setting memory. For details on the regions, see section 6, Option-Setting Memory.
Note 3. Access to some of the reserved areas will issue an illegal address access error. For more detail, see Table 13.3.

Figure 4.1 Memory map

5. Resets

5.1 Overview

The MCU provides 12 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 error reset	Bus error detection
Debug reset	On debug detection
Software reset	Register setting

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 37, 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 3)

Flags to be initialized	Reset source				
	RES pin reset	Power-on reset	Voltage monitor 0 reset	Independent watchdog timer reset	Watchdog timer 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)	✓	✓	✓	—	—
SRAM Parity Error Reset Detect Flag (RSTSR1.RPERF)	✓	✓	✓	—	—
SRAM ECC Error Reset Detect Flag (RSTSR1.REERF)	✓	✓	✓	—	—
Bus Error Reset Detect Flag (RSTSR1.BUSERF)	✓	✓	✓	—	—
Software Reset Detect Flag (RSTSR1.SWRF)	✓	✓	✓	—	—
Cold Start/Warm Start Determination Flag (RSTSR2.CWSF)	—	✓	—	—	—

Table 5.2 Reset detect flags initialized by each reset source (2 of 3)

Flags to be initialized	Reset source				
	Voltage monitor 1 reset	Voltage monitor 2 reset	Software reset	SRAM parity error reset	SRAM ECC error 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)	—	—	—	—	—
SRAM Parity Error Reset Detect Flag (RSTSR1.RPERF)	—	—	—	—	—
SRAM ECC Error Reset Detect Flag (RSTSR1.REERF)	—	—	—	—	—
Bus Error Reset Detect Flag (RSTSR1.BUSERF)	—	—	—	—	—
Software Reset Detect Flag (RSTSR1.SWRF)	—	—	—	—	—
Cold Start/Warm Start Determination Flag (RSTSR2.CWSF)	—	—	—	—	—

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

Flags to be initialized	Reset source	
	Bus error reset	Debug 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)	—	—
SRAM Parity Error Reset Detect Flag (RSTSR1.RPERF)	—	—
SRAM ECC Error Reset Detect Flag (RSTSR1.REERF)	—	—
Bus Error Reset Detect Flag (RSTSR1.BUSERF)	—	—
Software Reset Detect Flag (RSTSR1.SWRF)	—	—
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
Registers related to the IWDT	IWDTRR, IWDTSR	✓	✓	✓	✓	✓
Registers related to the WDT	WDTRR, WDTCR, WDTSR, WDTRCR, WDTCSTPR	✓	✓	✓	✓	✓

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
Registers related to the voltage monitor function 1	LVD1CR0, LVCMPCR.LVD1E, LVDLVLR.LVD1LVL	✓	✓	✓	✓	✓
	LVD1CR1/LVD1SR	✓	✓	✓	✓	✓
Registers related to the voltage monitor function 2	LVD2CR0, LVCMPCR.LVD2E, LVDLVLR.LVD2LVL	✓	✓	✓	✓	✓
	LVD2CR1/LVD2SR	✓	✓	✓	✓	✓
Register related to the LOCO	LOCOCR	✓	✓	✓	✓	✓
Register related to the BUS	BUSnERRADD (n = 1, 2, 3), BUSnERRSTAT (n = 1, 2, 3)	✓	✓	✓	✓	✓
Register related to the RTC	RTCC0, RTCC1, SUBCUD	✓	✓	✓	✓	✓
	Other than above	—	—	—	—	—
Register related to the TML32		—	✓	✓	—	—
Register related to the SOSC	SOSCCR	—	✓	—	—	—
	SOMCR	—	✓	—	—	—
	SOMRG	—	✓	—	—	—
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			
		Voltage monitor 1 reset	Voltage monitor 2 reset	Software reset	SRAM parity error reset
Registers related to the WDT	WDTRR, WDTCR, WDTSR, WDTRCR, WDTCSTPR	✓	✓	✓	✓
Registers related to the voltage monitor function 1	LVD1CR0, LVCMPCR.LVD1E, LVDLVLR.LVD1LVL	—	—	—	—
	LVD1CR1/LVD1SR	—	—	—	—
Registers related to the voltage monitor function 2	LVD2CR0, LVCMPCR.LVD2E, LVDLVLR.LVD2LVL	—	—	—	—
	LVD2CR1/LVD2SR	—	—	—	—
Register related to the LOCO	LOCOCR	✓	✓	✓	✓
Register related to the BUS	BUSnERRADD (n = 1, 2, 3), BUSnERRSTAT (n = 1, 2, 3)	✓	✓	✓	✓
Register related to the RTC	RTCC0, RTCC1, SUBCUD	✓	✓	✓	✓
	Other than above	—	—	—	—
Register related to the TML32		✓	✓	—	—
Register related to the SOSC	SOSCCR	—	—	—	—
	SOMCR	—	—	—	—
	SOMRG	—	—	—	—
Registers other than those shown, CPU, and internal state		✓	✓	✓	✓

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

Registers to be initialized		Reset source	
		Bus error reset	Debug reset
Registers related to the IWDT	IWDTRR, IWDTSR	✓	✓
Registers related to the WDT	WDTRR, WDTCR, WDTSR, WDTRCR, WDTCSTPR	✓	✓
Registers related to the voltage monitor function 1	LVD1CR0, LVCMPCR.LVD1E, LVDLVLR.LVD1LVL	—	—
	LVD1CR1/LVD1SR	—	—
Registers related to the voltage monitor function 2	LVD2CR0, LVCMPCR.LVD2E, LVDLVLR.LVD2LVL	—	—
	LVD2CR1/LVD2SR	—	—
Register related to the LOCO	LOCOCR	✓	✓
Register related to the BUS	BUSnERRADD (n = 1, 2, 3), BUSnERRSTAT (n = 1, 2, 3)	—	✓
Register related to the RTC	RTCC0, RTCC1, SUBCUD	✓	✓
	Other than above	—	—
Register related to the TML32		—	—
Register related to the SOSC	SOSCCR	—	—
	SOMCR	—	—
	SOMRG	—	—
Registers other than those shown, CPU, and internal state		✓	✓

Note: ✓ : Initialized
— : Not initialized

The RTC is not initialized by any reset source. SOSC and LOCO can be selected as the clock sources of the RTC. Table 5.4 and Table 5.5 show the states of SOSC and LOCO when a reset occurs.

Table 5.4 States of SOSC when a reset occurs

		Reset source	
		POR	Other
SOSC	Enable or disable	Initialized to disable	Continue with the state that was selected before the reset occurred
	Drive capability	Initialized to normal mode	Continue with the state that was selected before the reset occurred
	XT1/XT2	Initialized to the unselected state	Continue with the state that was selected before the reset occurred

Table 5.5 States of LOCO when a reset occurs

		Reset source	
		POR/LVD0/LVD1/LVD2	Other
LOCO	Enable or disable	Initialized to enable	

When a reset is canceled, reset exception handling starts.

Table 5.6 lists the pin related to the reset function.

Table 5.6 Pin related to reset

Pin name	I/O	Function
RES	Input	Reset pin

5.2 Register Descriptions

5.2.1 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:	—	—	—	—	LVD2RF	LVD1RF	LVD0RF	PORF
Value after reset:	0	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
7:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. The value after reset depends on the reset source.
Note 2. The register is cleared when a reset source listed in section 5.1. Overview 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

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

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

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

5.2.2 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:	—	—	—	BUSERF	—	—	REERF	RPERF	—	—	—	—	—	SWRF	WDTRF	IWDTRF
Value after reset:	0	0	0	x*1	0	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
11:10	—	These bits are read as 0. The write value should be 0.	R/W
12	BUSERF	Bus Error Reset Detect Flag 0: Bus error reset not detected 1: Bus error reset detected	R/W*2
15:13	—	These bits are read as 0. The write value should be 0.	R/W

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.

BUSERF flag (Bus Error Reset Detect Flag)

The BUSERF flag indicates that a bus error reset occurs.

[Setting condition]

- When a bus error reset occurs.

[Clearing conditions]

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

5.2.3 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
7:1	—	These bits are read as 0. The write value should be 0.	R/W

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 37, Electrical Characteristics.

5.3.2 Power-On Reset

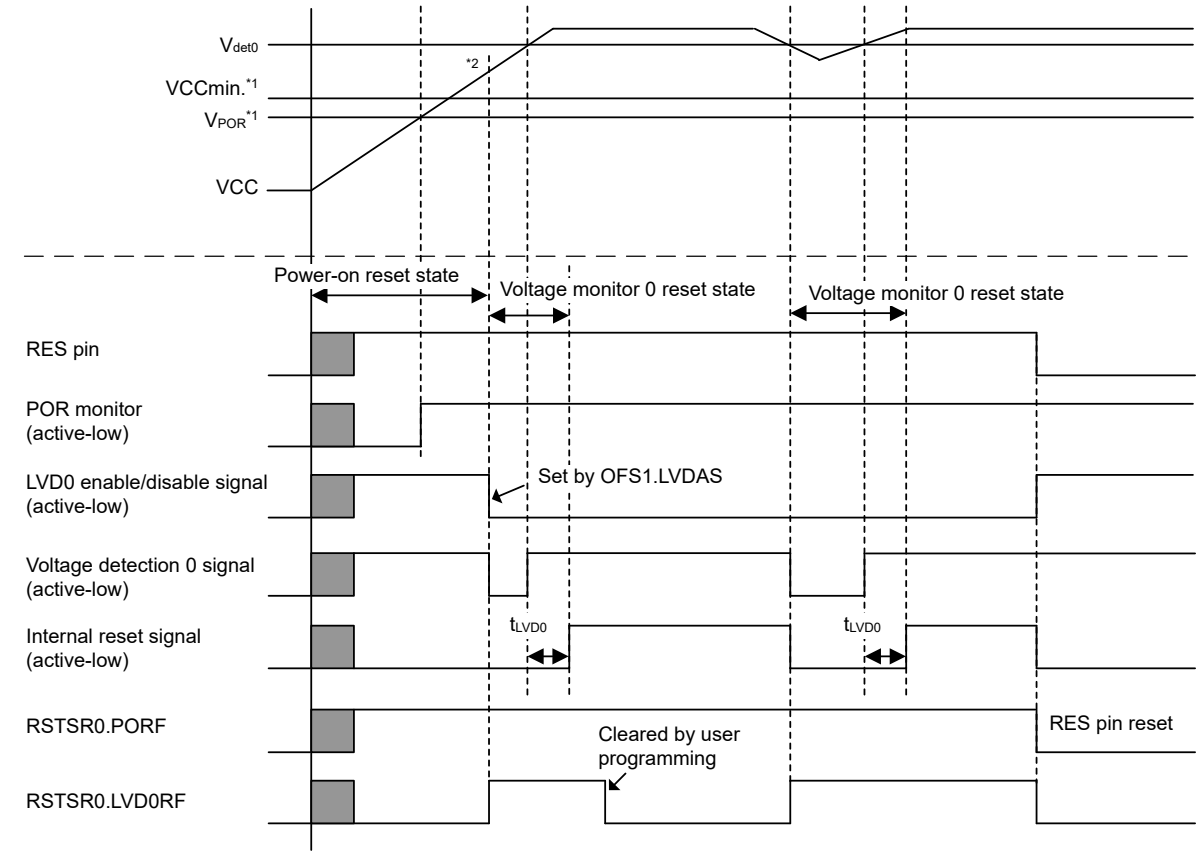
The power-on reset (POR) is an internal reset generated by the power-on reset circuit. A power-on reset is generated under the following conditions.

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 37, 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}.

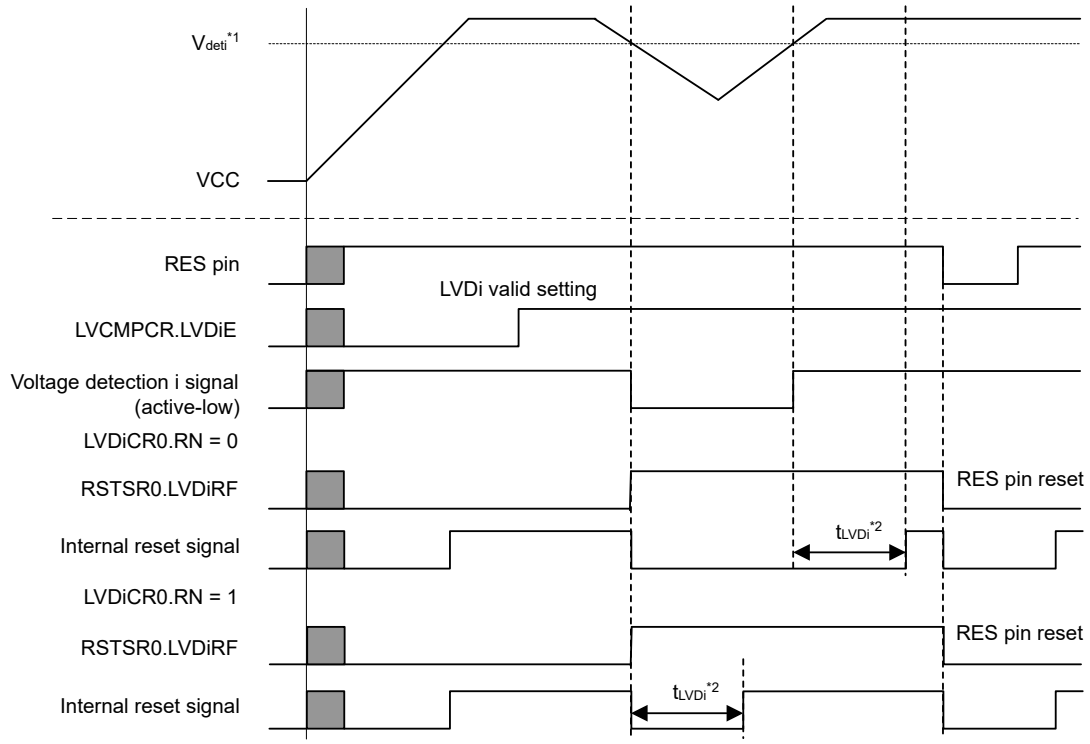
Likewise, when the Voltage Monitor 2 Interrupt/Reset Enable bit (RIE) is set to 1 (enabling generation of a reset or interrupt by the voltage detection circuit) and the Voltage Monitor 2 Circuit Mode Select bit (RI) is set to 1 (selecting generation of a reset in response to detection of a low voltage) 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 Detection Level Select Register (LVDLVLR).

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 37, 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 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_{RESWT3}) 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 22, Independent Watchdog Timer (IWDT).

5.3.5 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_{RESWT3}) 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 21, Watchdog Timer (WDT).

5.3.6 Software Reset

The software reset is an internal reset generated by a software setting of the SYSRESETREQ bit in the SWRCR register in the CPU core. When the SYSRESETREQ bit is set to 1, a software reset is generated. When the internal reset time (t_{RESWT3}) elapses after the software reset is generated, the internal reset is canceled and the CPU starts the reset exception handling.

5.3.7 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.

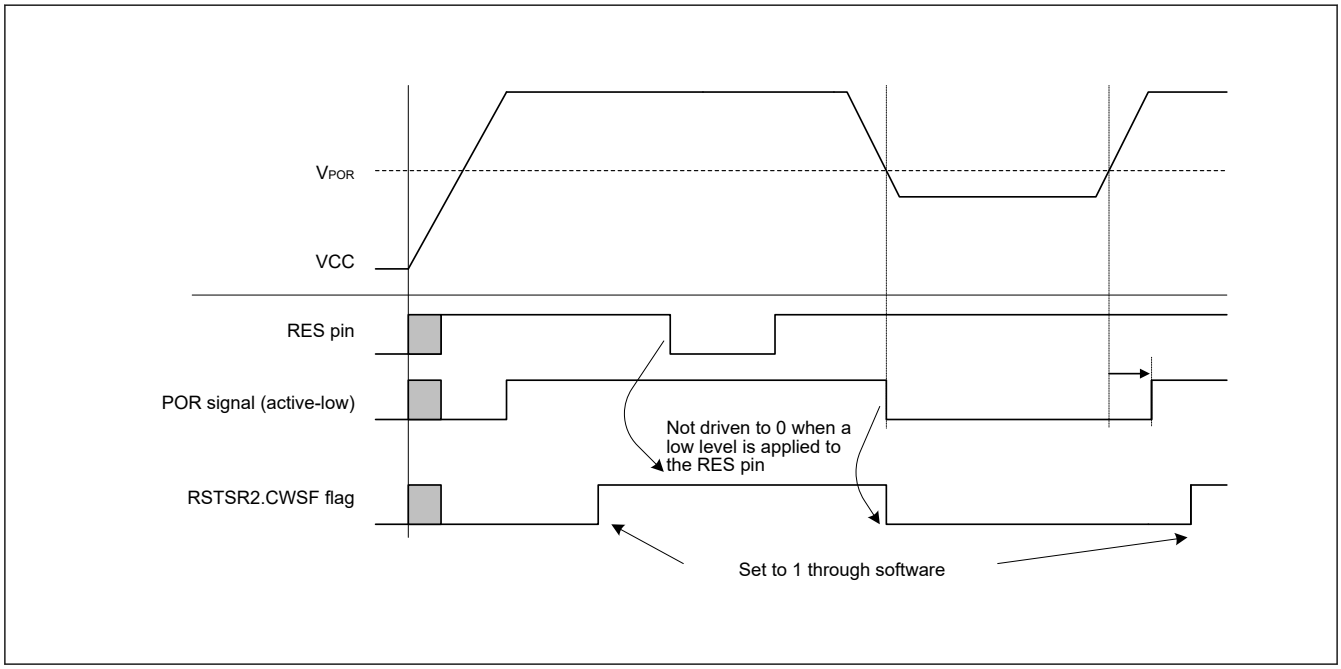
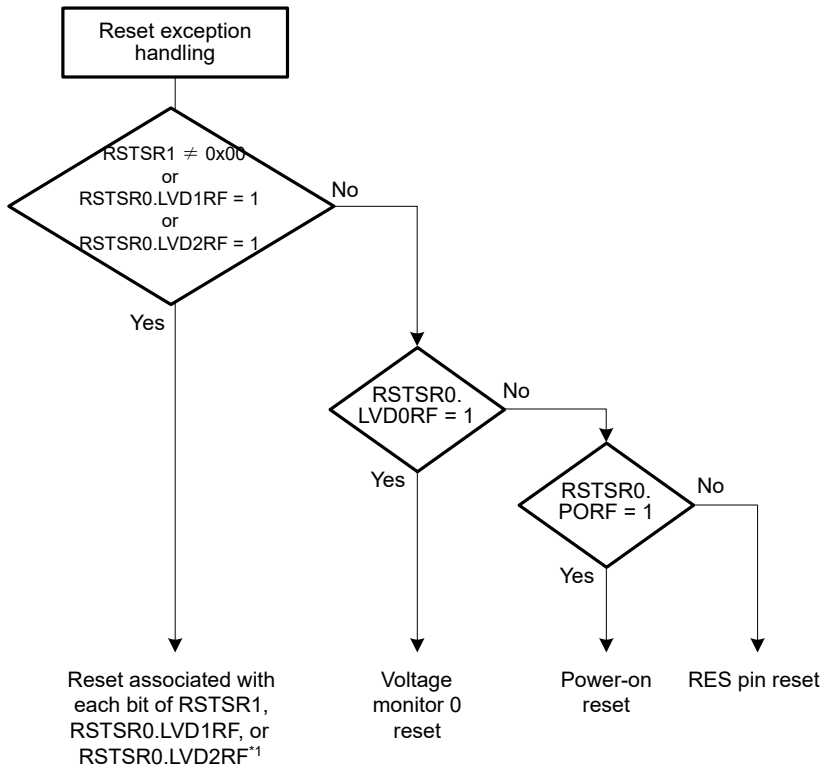


Figure 5.3 Example of cold/warm start determination operation

5.3.8 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 are set to 1.

Figure 5.4 Example of reset generation source determination flow

5.4 Usage Notes

5.4.1 Note on RES pin reset

A power-on reset may occur if RES pin reset is used with the following condition.

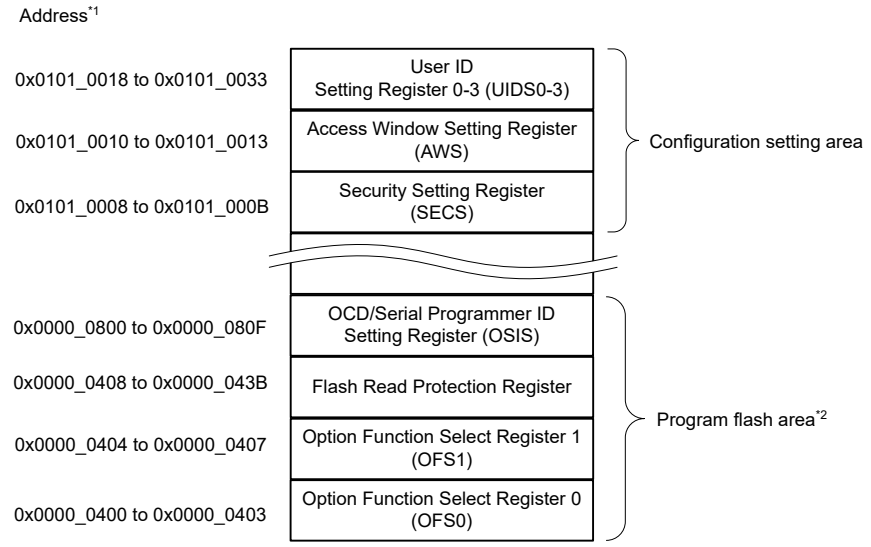
- When VCC ≤ 1.7V
- Voltage detection 0 circuit is enabled.

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 and the program flash area of the flash memory. The available methods of setting are different for the two areas.

Figure 6.1 shows the option-setting memory area.



Note 1. The option-setting memory must be allocated to the user area of the flash memory.
Note 2. The address of these registers are changed when the boot swap is set. See section 6.2.1. OFS0 : Option Function Select Register 0, section 6.2.2. OFS1 : Option Function Select Register 1, section 13.4.2. Setting of Flash Read Protection, and section 6.2.3. OSIS : OCD/Serial Programmer ID Setting Register for details.

Figure 6.1 Option-setting memory area

6.2 Register Descriptions

6.2.1 OFS0 : Option Function Select Register 0

Address: 0x0000_0400*1

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*2															

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*2															

Bit	Symbol	Function	R/W
0	—	When read, this bit returns the written value.	R

Bit	Symbol	Function	R/W
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
13	—	When read, this bit returns the written value.	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.	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: WDTCLK divided by 4 0x4: WDTCLK divided by 64 0xF: WDTCLK divided by 128 0x6: WDTCLK divided by 512 0x7: WDTCLK divided by 2048 0x8: WDTCLK 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

Bit	Symbol	Function	R/W
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.	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.	R

Note 1. When the boot swap is set, the address of this register changes. Therefore, set 0x0000_2400 and 0x0000_0400 to the same value if boot swap is used.
Note 2. 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 22, 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 22, 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 22, 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 22, 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 22, Independent Watchdog Timer (IWDT).

IWDTSTPCTL bit (IWDT Stop Control)

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

Table 6.1 shows the count stop control by the IWDTSTPCTL bit.

Table 6.1 Count stop control by the IWDTSTPCTL bit

IWDTSTPCTL	Mode	Counting of IWDT
0	Sleep/Software Standby mode	Continue counting
1	Sleep/Software Standby mode	Stop counting

For details, see section 22, 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 WDTCLK 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 21, 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 WDTCLK 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 WDTCLK cycles.

For details, see section 21, 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 21, 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 21, 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 21, Watchdog Timer (WDT).

WDTSTPCTL bit (WDT Stop Control)

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

For details, see section 21, Watchdog Timer (WDT).

6.2.2 OFS1 : Option Function Select Register 1

Address: 0x0000_0404*1

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	ICSATS	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	The value set by the user*2															

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	HOCOFRQ1[2:0]			—	—	—	HOCOEN	—	—	VDSEL0[2:0]			LVDAS	—	—
Value after reset:	The value set by the user*2															

Bit	Symbol	Function	R/W
1:0	—	When read, these bits return the written value.	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
5:3	VDSEL0[2:0]	Voltage Detection 0 Level Select*3 0 0 0: V_{det0_0} 0 0 1: V_{det0_1} 0 1 0: V_{det0_2} 0 1 1: V_{det0_3} 1 0 0: V_{det0_4} Others: Setting prohibited	R
7:6	—	When read, these bits return the written value.	R
8	HOCOEN	HOCO Oscillation Enable 0: Enable HOCO oscillation after a reset 1: Disable HOCO oscillation after a reset	R
11:9	—	When read, these bits return the written value.	R
14:12	HOCOFRQ1[2:0]	HOCO Frequency Setting 1 0 0 0: 24 MHz 0 1 0: 32 MHz 1 0 0: 48 MHz Others: Setting prohibited	R
30:15	—	When read, these bits return the written value.	R
31	ICSATS	Internal Clock Supply Architecture Type Select 0: Internal Clock Supply Architecture Type B 1: Internal Clock Supply Architecture Type A	R

Note 1. When the boot swap is set, the address of this register changes. Therefore, set 0x0000_2404 and 0x0000_0404 to the same value if boot swap is used.
Note 2. The value in a blank product is 0xFFFFFFFF. It is set to the value written by your application.
Note 3. See section 37, Electrical Characteristics for the voltage levels to be detected.

LVDAS bit (Voltage Detection 0 Circuit Start)

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

VDSEL0[2:0] bits (Voltage Detection 0 Level Select)

The VDSEL0[2:0] bits select the voltage detection level of the voltage detection 0 circuit.

HOCOEN bit (HOCO Oscillation Enable)

The HOCOEN bit selects whether the HOCO Oscillation Enable bit is valid 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, set the OFS1.HOCOFRQ1 bit*1 to an optimum value.

Note 1. The value of OFS1.HOCOFRQ1[2:0] bits is automatically transferred to HOCOCR2.HCFRQ1[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ1[2:0] bits when OFS1.HOCOEN=1.

HOCOFRQ1[2:0] bits (HOCO Frequency Setting 1)

The HOCOFRQ1[2:0] bits select the HOCO frequency after a reset as 24, 32, or 48 MHz.

ICSATS bit (Internal Clock Supply Architecture Type Select)

The ICSATS bit selects the internal clock supply architecture from Type A and Type B after a reset.

Internal Clock Supply Architecture Type A provides the clocks that the frequency of ICLK, PCLKB can be individually set in the System Clock Division Control Register (SCKDIVCR).

In Internal Clock Supply Architecture Type A, a fairly flexible operation frequency relationship between system and peripheral functions can be executed for various applications.

Internal Clock Supply Architecture Type B provides the clocks that the frequency of ICLK, PCLKB is fixed as ICLK = PCLKB regardless of the PCKB[2:0] settings in the System Clock Division Control Register (SCKDIVCR).

In Internal Clock Supply Architecture Type B, a simple operation frequency relationship between system and peripheral functions can be executed, and therefore, is a more advantageous type for power reduction.

For details of the System Clock Division Control Register (SCKDIVCR), see section 8.2.1. SCKDIVCR : System Clock Division Control Register.

For details of the clock generation circuit block diagram, see section 8.1. Overview.

Note: When Internal Clock Supply Architecture Type B is selected:

- 48 MHz HOCO frequency setting in OFS1.HOCOFRQ1[2:0] is not allowed. Set the HOCO frequency to 32 MHz or 24 MHz.
- Memory wait setting in the MEMWAIT.MEMWAIT and FLDWAITR.FLDWAIT1 is not allowed. Use the default.

6.2.3 OSIS : OCD/Serial Programmer ID Setting Register

The OSIS register stores the ID for ID code protection of the OCD/serial programmer. When connecting the OCD/serial programmer, write values so that the MCU can determine whether to permit the connection. Use this register to check whether a code transmitted from the OCD/serial programmer matches the ID code in the option-setting memory. When the ID codes match, connection with the OCD/serial programmer is permitted, if not, connection with the OCD/serial programmer is not possible. The OSIS register must be set in 32-bit words.

Address: 0x0000_0800, 0x0000_0804, 0x0000_0808, 0x0000_080C*1

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

Note 1. When the boot swap is set, the address of this register changes. Therefore, set 0x0000_280X and 0x0000_080X to the same value if boot swap is used.

These fields hold the ID for use in ID authentication for the OCD/serial programmer.

ID code bits [127] and [126] determine whether the ID code protection is enabled, and the authentication method to use with the host. See section 35.5.1. ID Code Protection for details.

Table 6.2 shows an example of the address assignment when writing the ID code 0xFFEE_DDCC_BBAA_9988_7766_5544_3322_1100 in b127 to b0.

Table 6.2 Example of address assignment

Address	b31 to b24	b23 to b16	b15 to b8	b7 to b0
0x0000_0800	33	22	11	00
0x0000_0804	77	66	55	44
0x0000_0808	BB	AA	99	88
0x0000_080C	FF	EE	DD	CC

6.2.4 AWS : Access Window Setting Register

Address: 0x0101_0010

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	FAWE[10:0]										
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:	—	—	—	—	—	FAWS[10:0]										
Value after reset:	User setting															

Bit	Symbol	Function	R/W
10:0	FAWS[10:0]	Access Window Start Block Address These bits specify the start block address for the access window. They do not represent the block number of the access window. The access window is only valid in the program flash area. The block address specifies the first address of the block and consists of the address bits [21:11].	R
15:11	—	When read, these bits return the written value.	R
26:16	FAWE[10:0]	Access Window End Block Address These bits specify the end block address for the access window. They do not represent the block number of the access window. The access window is only valid in the program flash area. The end block address for the access window is the next block to the acceptable programming and erasure region defined by the access window. The block address specifies the first address of the block and consists of the address bits [21:11].	R
31:27	—	When read, these bits return the written value.	R

Issuing the program or erase command to an area outside the access window causes a command-locked state. The access window is only valid in the program flash area. The access window provides protection in self-programming mode, serial programming mode, and on-chip debug mode. Issuing the read command to a flash memory area outside of the access window is protected in serial programming mode. The access window can be locked by the FAPR bit.

The access window is specified in both the FAWS[10:0] bits and the FAWE[10:0] bits. The settings for the FAWS[10:0] and FAWE[10:0] bits are as follows:

FAWE[10:0] = FAWS[10:0]: The P/E and read commands are allowed to execute in the full program flash area.

FAWE[10:0] > FAWS[10:0]: The P/E and read commands are only allowed to execute in the window from the block pointed to by the FAWS[10:0] bits to the block one lower than the block pointed to by the FAWE[10:0] bits.

FAWE[10:0] < FAWS[10:0]: The P/E and read commands are not allowed to execute in the program flash area.

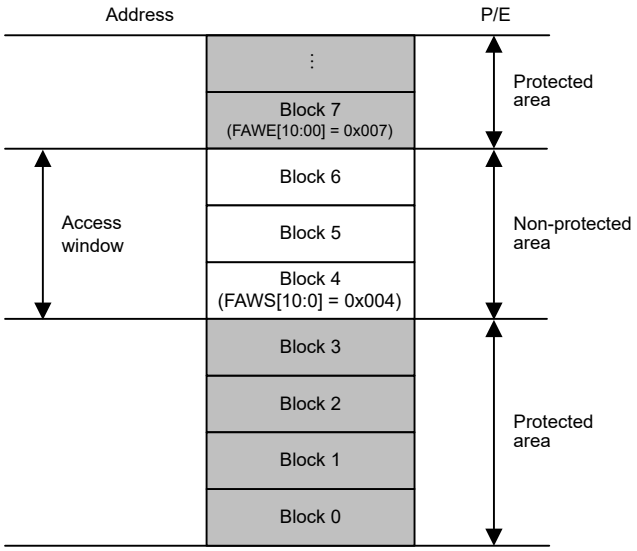


Figure 6.2 Access window overview

6.2.5 SECS : Security Setting Register

Address: 0x0101_0008

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

Bit	Symbol	Function	R/W
12:0	—	When read, these bits return the written value. Setting 0 is prohibited. Be sure to set 1.	R
13	OCDDIS	Protection of On-chip Debugger Connection This bit controls the connection to on-chip debugger. When this bit is set to 0, it cannot be changed to 1 except for ALeRASE sequence. 0: On-chip debugger is not connectable. 1: On-chip debugger is connectable.	R
14	FAPR	Protection of Access Window Function This bit controls the command for setting the access window. When this bit is set to 0, it cannot be changed to 1. 0: Executing the configuration setting command for programming Access Window (FAWE[10:0], FAWS[10:0]) is invalid. ALeRASE sequence is invalid. 1: Executing the configuration setting command for programming Access Window (FAWE[10:0], FAWS[10:0]) is valid. ALeRASE sequence is valid.	R
31:15	—	When read, these bits return the written value.	R

6.2.6 UIDSn : User ID Setting Register n (n = 0 to 3)

Base address: 0x0101_0018

Offset address: 0x08 × n

Bit position:	31															0
Bit field:	UIDSn[31:0]															
Value after reset:	User setting															

Bit	Symbol	Function	R/W
31:0	UIDSn[31:0]	Specifies the user ID setting	R

A user ID can be set through serial programming by using a flash memory programmer. Table 6.3 shows the settings and functions related to the user ID. See section 35.9.3. User ID Read Protection for details.

Table 6.3 Setting and functions related to the user ID

Register	Bit width	Function
UIDS0[31:0]	64	Any desired 64-bit value is specifiable.
UIDS1[31:0]		
UIDS2[31:0]	32 (UIDS2[31:21] are reserved)	Set a 21-bit value for the address in the code flash memory where the user ID key data is to be stored.
UIDS3[31:0]	32	Any desired 32-bit value is specifiable as the lock data. Set the same value as the user ID key data.

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

For details of the programming command, the configuration setting command, see section 35, Flash Memory.

(2) 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.

6.4 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.

6.4.2 Note on FAPR Bit

The SECS.FAPR bit cannot be changed to 1 once it is set to 0. At that time, access window cannot be set again.

6.4.3 Order of the FAPR Bit Setting

Set FAPR bit after setting of access window. When programming is using the hex file, programming is the ascending order of the address. In this case, the FAPR bit will be written in before setting of access window. Therefore, divide the hex file for FAPR into another file, and use it after setting of access window.

6.4.4 Note on OCDDIS Bit

The SECS.OCDDIS bit can be changed from 0 to 1 only ALeRASE sequence. When FAPR is 0, the SECS.OCDDIS bit cannot be changed from 0 to 1.

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 5 different levels in the OFS1.VDSEL0[2:0] bits	Selectable from 16 different levels in the LVDLVLR.LVD1LVL[4:0] bits	Selectable from 4 different levels in the LVDLVLR.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
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

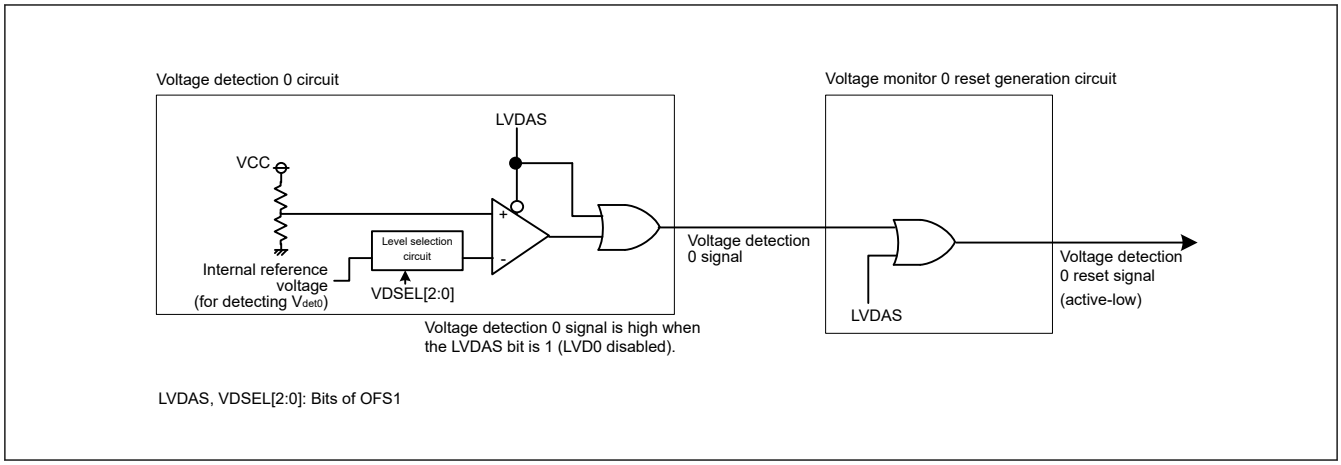


Figure 7.1 Block diagram of voltage monitor 0 reset generation circuit

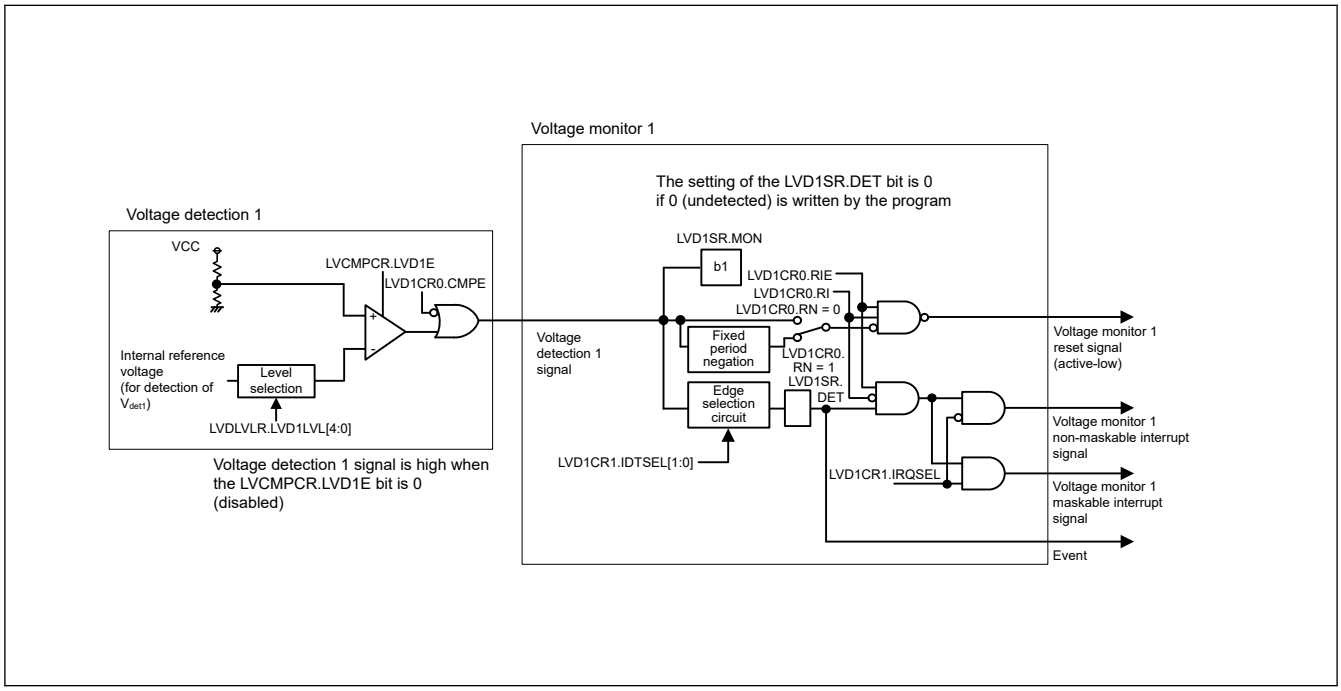


Figure 7.2 Block diagram of voltage monitor 1 interrupt and reset circuit

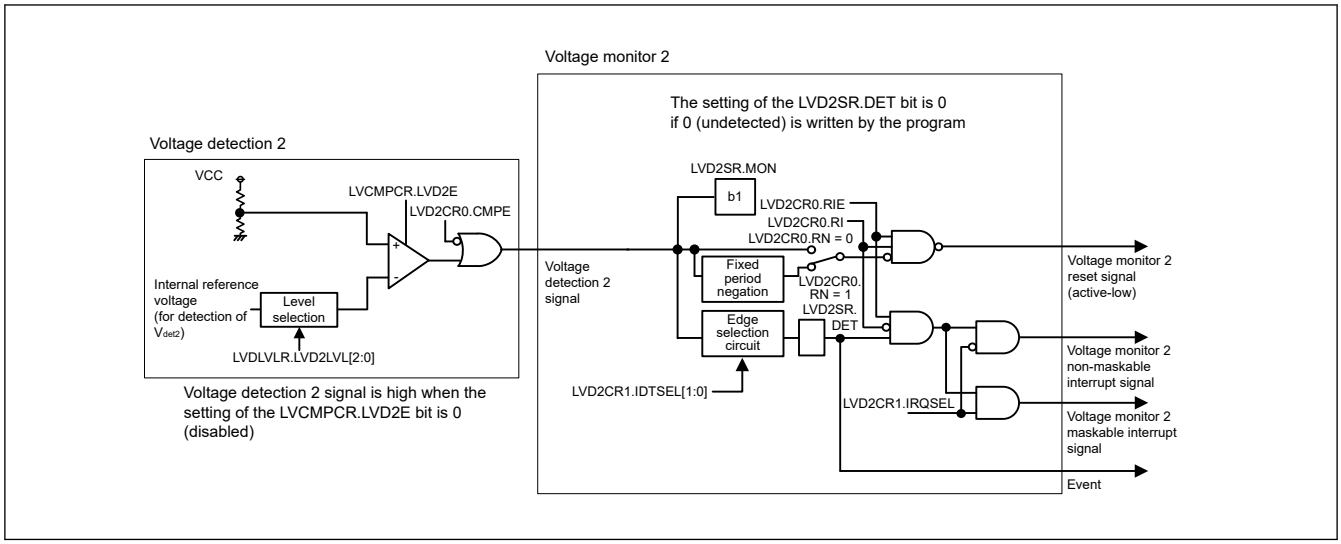


Figure 7.3 Block diagram of voltage monitor 2 interrupt and reset circuit

7.2 Register Descriptions

7.2.1 LVCMPCR : Voltage Monitor Circuit Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x417

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	LVD2E	LVD1E	—	—	—	—	—
Value after reset:	0	0	0	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
5	LVD1E	Voltage Detection 1 Enable 0: Voltage detection 1 circuit disabled 1: Voltage detection 1 circuit enabled	R/W
6	LVD2E	Voltage Detection 2 Enable 0: Voltage detection 2 circuit disabled 1: Voltage detection 2 circuit enabled	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

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

LVD1E bit (Voltage Detection 1 Enable)

When using voltage detection 1 interrupt/reset or the LVD1SR.MON flag, set the LVD1E bit to 1. The voltage detection 1 circuit starts when LVD1 operation stabilization time ($t_{d(E-A)}$) elapses after the LVD1E bit value is changed from 0 to 1. For details on $t_{d(E-A)}$, see section 37, Electrical Characteristics.

LVD2E bit (Voltage Detection 2 Enable)

When using voltage detection 2 interrupt/reset or the LVD2SR.MON flag, set the LVD2E bit to 1. The voltage detection 2 circuit starts when LVD2 operation stabilization time $t_{d(E-A)}$ elapses after the LVD2E bit value is changed from 0 to 1. For details on LVD2 operation stabilization time $t_{d(E-A)}$, see section 37, Electrical Characteristics.

7.2.2 LVDLVLR : Voltage Detection Level Select Register

Base address: SYSC = 0x4001_E000

Offset address: 0x418

Bit position:	7	6	5	4	3	2	1	0
Bit field:	LVD2LVL[2:0]			LVD1LVL[4:0]				
Value after reset:	0	0	0	0	0	1	1	1

Bit	Symbol	Function	R/W
4:0	LVD1LVL[4:0]	Voltage Detection 1 Level Select (Standard voltage during fall in voltage)*1 0x00: V_{det1_0} 0x01: V_{det1_1} 0x02: V_{det1_2} 0x03: V_{det1_3} 0x04: V_{det1_4} 0x05: V_{det1_5} 0x06: V_{det1_6} 0x07: V_{det1_7} 0x08: V_{det1_8} 0x09: V_{det1_9} 0x0A: V_{det1_A} 0x0B: V_{det1_B} 0x0C: V_{det1_C} 0x0D: V_{det1_D} 0x0E: V_{det1_E} 0x0F: V_{det1_F} Others: Setting prohibited	R/W
7:5	LVD2LVL[2:0]	Voltage Detection 2 Level Select (Standard voltage during fall in voltage)*1 0 0 0: V_{det2_0} 0 0 1: V_{det2_1} 0 1 0: V_{det2_2} 0 1 1: V_{det2_3} Others: Setting prohibited	R/W

Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.
Note 1. See section 37, Electrical Characteristics for the voltage levels to be detected. Keep the initial value if LVD1 is not used. When using LVD0, set the detection voltage of LVD1 higher than the detection voltage of LVD0. The LVD1LVL [4:0] bits can be rewritten only once after reset.

The contents of the LVDLVLR register can only be changed if the LVCMPCR.LVD1E and LVCMPCR.LVD2E bits (voltage detection n circuit disable, n = 1, 2) are both 0. Do not set LVD detectors 1 and 2 to the same voltage detection level.

7.2.3 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	—		—	CMPE	—	RIE
Value after reset:	1	0	0	0	x	0	0	0

Bit	Symbol	Function	R/W
0	RIE	Voltage Monitor 1 Interrupt/Reset Enable 0: Disable 1: Enable	R/W
1	—	This bit is read as 0. The write value should be 0.	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	—	These bits are read as 0. The write value should be 0.	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

Bit	Symbol	Function	R/W
7	RN	Voltage Monitor 1 Reset Negate Select 0: Negate after a stabilization time (t_{LVD1}) from the time that VCC > V_{det1} is detected 1: Negate after a stabilization time (t_{LVD1}) from the time that LVD1 reset is asserted	R/W

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.

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 circuit, disable the voltage detection 1 circuit after setting the CMPE bit is 0.

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 MOCOCR.MCSTP bit to 0 (the MOCO operates). In addition, for a transition to 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.

7.2.4 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	—		—	CMPE	—	RIE
Value after reset:	1	0	0	0	x	0	0	0

Bit	Symbol	Function	R/W
0	RIE	Voltage Monitor 2 Interrupt/Reset Enable 0: Disable 1: Enable	R/W
1	—	This bit is read as 0. The write value should be 0.	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	—	These bits are read as 0. The write value should be 0.	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}) from the time that VCC > V_{det2} is detected 1: Negate after a stabilization time (t_{LVD2}) from the time that LVD2 reset is asserted	R/W

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.

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 circuit, disable the voltage detection 2 circuit after setting the CMPE bit is 0.

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 MOCOCR.MCSTP bit to 0 (the MOCO operates). Additionally, for a transition to 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.

7.2.5 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: 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.

7.2.6 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: 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 LVCMPCR.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 LVCMPCR.LVD1E bit is 1 (voltage detection 1 circuit enabled) and the LVD1CR0.CMPE bit is 1 (voltage monitor 1 circuit comparison result output enabled).

7.2.7 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: 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.

7.2.8 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: 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 LVCMPCR.LVD2E bit is 1 (voltage detection 2 circuit enabled) and the LVD2CR0.CMPE bit is 1 (voltage monitor 2 circuit comparison result output enabled).

When detecting V_{det2}, 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 LVCMPCR.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 LVCMPCR.LVD1E = 0 to disable voltage detection 1 before writing to the LVDLVLR.LVD1LVL[4:0] bits.
	2	Select the detection voltage in the LVDLVLR.LVD1LVL[4:0] bits.
	3	Set LVCMPCR.LVD1E = 1 to enable the voltage detection 1 circuit.
	4	Wait for at least $t_{d\ (E\text{-}A)}$ for the LVD1 operation stabilization time after LVD1 is enabled.
Enabling output	5	Set LVD1CR0.CMPE = 1 to enable output of the comparison results from voltage monitor 1.

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.

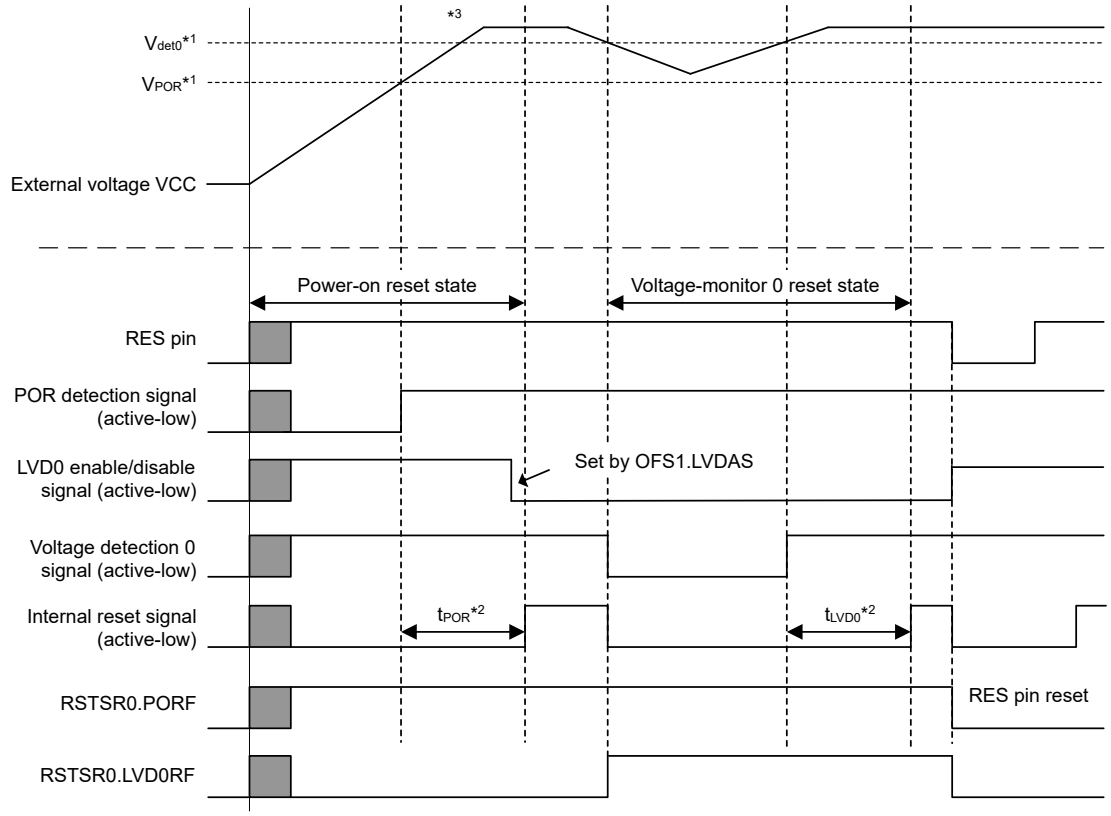
Table 7.3 Procedures to set up monitoring against V_{det2}

Step		Monitoring the results of comparison by voltage monitor 2
Setting up the voltage detection 2 circuit	1	Set LVCMPCR.LVD2E = 0 to disable voltage detection 2 before writing to the LVDLVLR.LVD2LVL[2:0] bits.
	2	Select the detection voltage in the LVDLVLR.LVD2LVL[2:0] bits.
	3	Set LVCMPCR.LVD2E = 1 to enable the voltage detection 2 circuit.
	4	Wait for at least $t_{d\ (E\text{-}A)}$ for the LVD2 operation stabilization time after LVD2 is enabled.
Enabling output	5	Set LVD2CR0.CMPE = 1 to enable output of the comparison results from voltage monitor 2.

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 37, 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} indicates the period of a power-on reset and t_{LVD0} indicates the period of a voltage monitor 0 reset.
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, set up the circuit using the procedures in this section.

(1) Setting in Software Standby mode

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

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 LVCMPCR.LVD1E = 0 to disable voltage detection 1 before writing to the LVDLVLR register.	
	2	Select the detection voltage in the LVDLVLR.LVD1LVL[4:0] bits.	
	3	Set LVCMPCR.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 up the voltage monitor 1 interrupt or reset	5	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.
	6	• Select the interrupt request condition in the LVD1CR1.IDTSEL[1:0] bits. • Select the interrupt type in the LVD1CR1.IRQSEL bit.	—
Enabling output	7	Set LVD1SR.DET = 0.	
	8	Set LVD1CR0.RIE = 1 to enable the voltage monitor 1 interrupt or reset. *2	
	9	Set LVD1CR0.CMPE = 1 to enable output of the comparison results from voltage monitor 1.	

Note 1. Steps 5 to 8 can be performed during the wait time in step 4. For details on $t_{d\ (E-A)}$, see section 37, Electrical Characteristics.
Note 2. Step 8 is not required if only the ELC event signal is to be output.

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

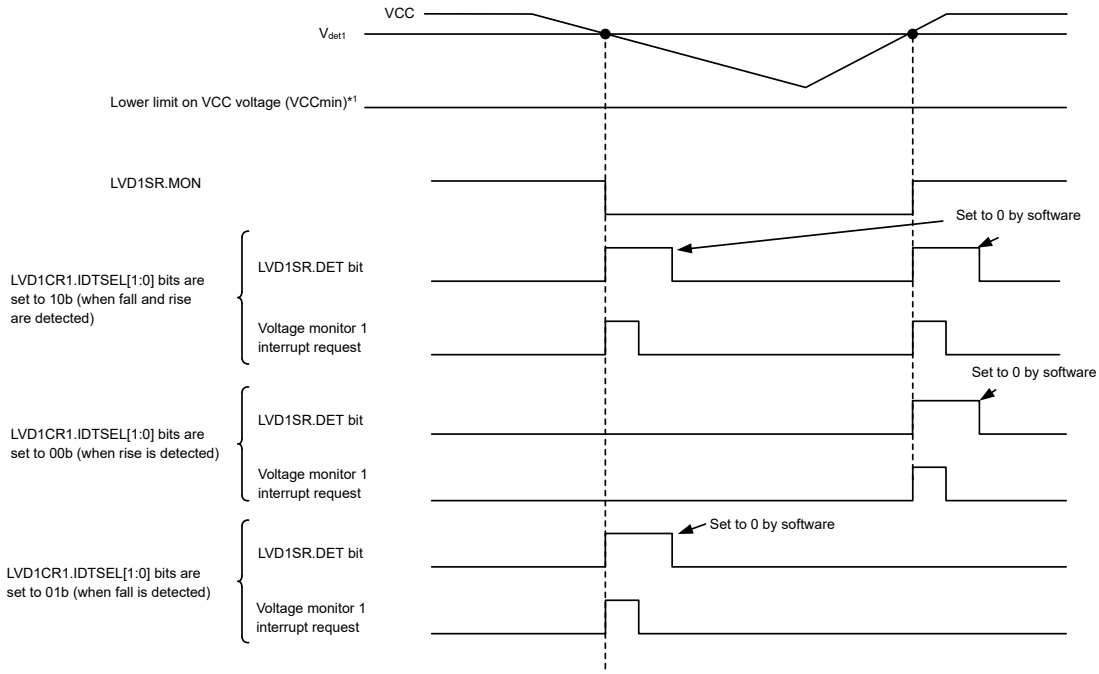
Step		Voltage monitor 1 interrupt (voltage monitor 1 ELC event output), voltage monitor 1 reset
Stopping the enabling output	1	Set LVD1CR0.CMPE = 0 to disable output of the comparison results from voltage monitor 1.
	2	Set LVD1CR0.RIE = 0 to disable the voltage monitor 1 interrupt or reset.*1
Stopping the voltage detection 1 circuit	3	Set LVCMPCR.LVD1E = 0 to disable the voltage detection 1 circuit.

Note 1. Step 2 is not required if only the ELC event signal is to be output.

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 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 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, set up the circuit with the following procedures.

(1) Setting in Software Standby mode

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

Table 7.6 Procedures for setting bits related to voltage monitor 2 interrupt and voltage monitor 2 reset so that voltage monitoring occurs (1 of 2)

Step		Voltage monitor 2 interrupt (voltage monitor 2 ELC event output)	Voltage monitor 2 reset
Setting up the voltage detection 2 circuit	1	Set LVCMPCR.LVD2E = 0 to disable voltage detection 2 before writing to the LVDLVLR register.	
	2	Select the detection voltage in the LVDLVLR.LVD2LVL[2:0] bits.	
	3	Set LVCMPCR.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	

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

Step		Voltage monitor 2 interrupt (voltage monitor 2 ELC event output)	Voltage monitor 2 reset
Setting up the voltage monitor 2 interrupt or reset	5	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.
	6	• Select the interrupt request condition in the LVD2CR1.IDTSEL[1:0] bits. • Select the interrupt type in the LVD2CR1.IRQSEL bit.	—
Enabling output	7	Set LVD2SR.DET = 0.	
	8	Set LVD2CR0.RIE = 1 to enable the voltage monitor 2 interrupt or reset.*2	
	9	Set LVD2CR0.CMPE = 1 to enable output of the comparison results from voltage monitor 2.	

Note 1. Steps 5 to 8 can be performed during the wait time in step 4. For details on $t_{d(E-A)}$, see section 37, Electrical Characteristics.
Note 2. Step 8 is not required if only the ELC event signal is to be output.

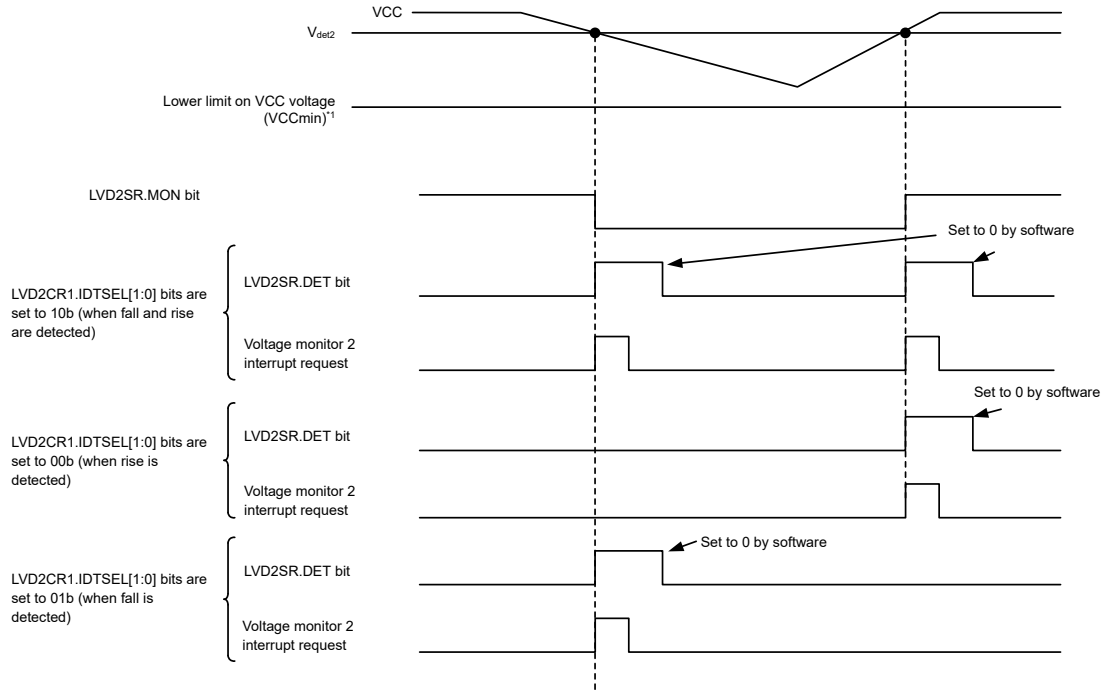
Table 7.7 Procedures for setting bits related to voltage monitor 2 interrupt and voltage monitor 2 reset so that voltage monitoring stops

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	Set LVD2CR0.RIE = 0 to disable the voltage monitor 2 interrupt or reset.*1
Stopping the voltage detection 2 circuit	3	Set LVCMPCR.LVD2E = 0 to disable the voltage detection 2 circuit.

Note 1. Step 2 is not required if only the ELC event signal is to be output.

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 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 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 mode.

- 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.

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 and Figure 8.2 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
External clock input (EXTAL)	External clock input frequency	Up to 20 MHz
	Connection pins	EXTAL
Sub-clock oscillator (SOSC)	Resonator frequency	32.768 kHz
	External resonator or additional circuit	crystal resonator
	Connection pins	XT1, XT2
	Drive capability switching	Available
High-speed on-chip oscillator (HOCO)	Oscillation frequency	24/32/48 MHz
	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	No
External clock input for cJTAG (TCKC)	Input clock frequency	Up to 6.25 MHz

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

Item	Clock source	Clock supply	Specification
System clock (ICLK)	HOCO/MOCO/LOCO/ SOSC*1/EXTAL	CPU, DTC, Flash, Flash-IF, SRAM	Up to 48 MHz Division ratios: 1/2/4/8/16/32/64 1 MHz to 48 MHz (P/E)
Peripheral module clock B (PCLKB)	HOCO/MOCO/LOCO/ SOSC*1/EXTAL	Peripheral modules (CAC, ELC, I/O Ports, KINT, RTC, WDT, IWDT, IICA, SAU, TAU, UARTA, REMC, TML32, CRC, ADC12, DAC8, CMP, DOC and TRNG)	Up to 48 MHz*2 Division ratios:1/2/4/8/16/32/64
CAC External clock input (CACMCLK)	EXTAL	CAC	Up to 20 MHz
CAC Sub clock (CACSCLK)	SOSC*1	CAC	32.768 kHz
CAC LOCO clock (CACLCLK)	LOCO	CAC	32.768 kHz
CAC MOCO clock (CACMOCLK)	MOCO	CAC	8 MHz
CAC HOCO clock (CACHCLK)	HOCO	CAC	24/32/48 MHz
CAC IWDTLOCO clock (CACILCLK)	IWDTLOCO	CAC	15 kHz
RTC clock (RTCSCLK*1/ RTCS128CLK/RTCLCLK)	LOCO/SOSC	RTC	32.768 kHz / 128 Hz
TML32 External clock (TML32MCLK)	EXTAL	TML32	Up to 20 MHz

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

Item	Clock source	Clock supply	Specification
TML32 HOCO clock (TML32HCLK)	HOCO	TML32	24/32/48 MHz
TML32 MOCO clock (TML32MOCLK)	MOCO	TML32	8 MHz
TML32 LOCO/SOSC clock (TML32LCLK/TML32SCLK)	LOCO/SOSC	TML32	32.768 kHz
UARTA External clock (UARTAMCLK)	EXTAL	UARTA	Up to 20 MHz
UARTA HOCO clock (UARTAHCLK)	HOCO	UARTA	24/32/48 MHz
UARTA MOCO clock (UARTAMOCLK)	MOCO	UARTA	8 MHz
UARTA LOCO/SOSC clock (UARTALCLK/UARTASCLK)	LOCO/SOSC	UARTA	32.768 kHz
REMC LOCO/SOSC clock (REMCLCLK/REMCSCLK)	LOCO/SOSC	REMC	32.768 kHz
IWDT clock (IWDTCLK)	IWDTLOCO	IWDT	15 kHz
Machine timer clock (MTCLK)	LOCO	Machine timer	32.768 kHz
Clock/buzzer output (CLKOUT)	LOCO/MOCO/HOCO/ SOSC*1/EXTAL	CLKOUT pin	Up to 16 MHz Division ratios: 1/2/4/8/16/32/64/128
cJTAG clock (TCKC)	TCKC pin	OCD	Up to 6.25 MHz

Note: Restrictions on setting clock frequency: ICLK ≥ PCLKB
PCLKB Restrictions on clock frequency ratio: (N: integer, and up to 64)
ICLK:PCLKB = N:1
Minimum ICLK frequency is 1 MHz in Programming/Erasure (P/E) mode.
Note 1. This clock is not supplied in Software Standby mode when the RTCC0.RTC128EN bit is 1.
Note 2. Maximum PCLKB frequency for IICA is 32 MHz.

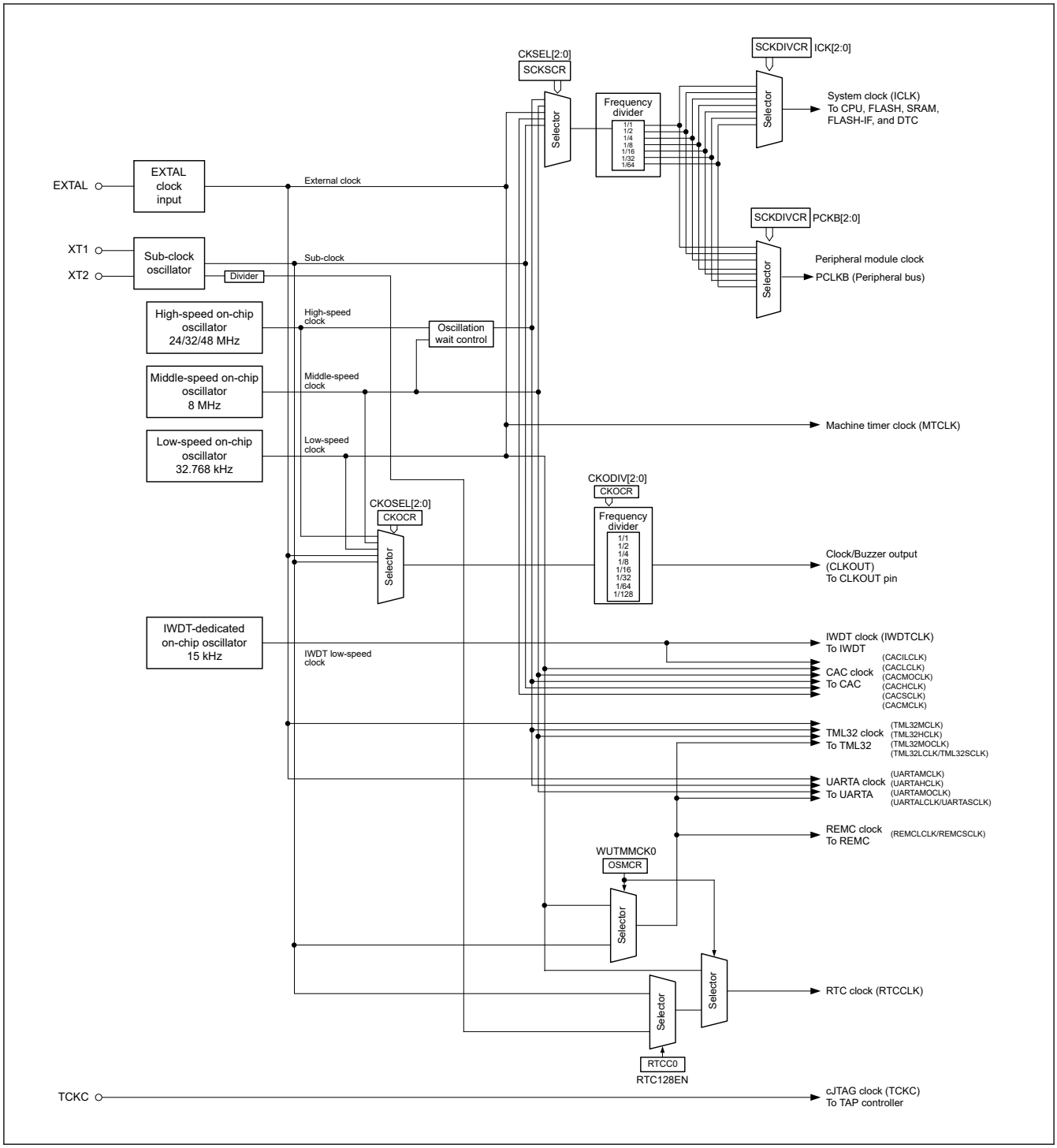


Figure 8.1 Clock generation circuit block diagram (Internal Clock Supply Architecture Type A)

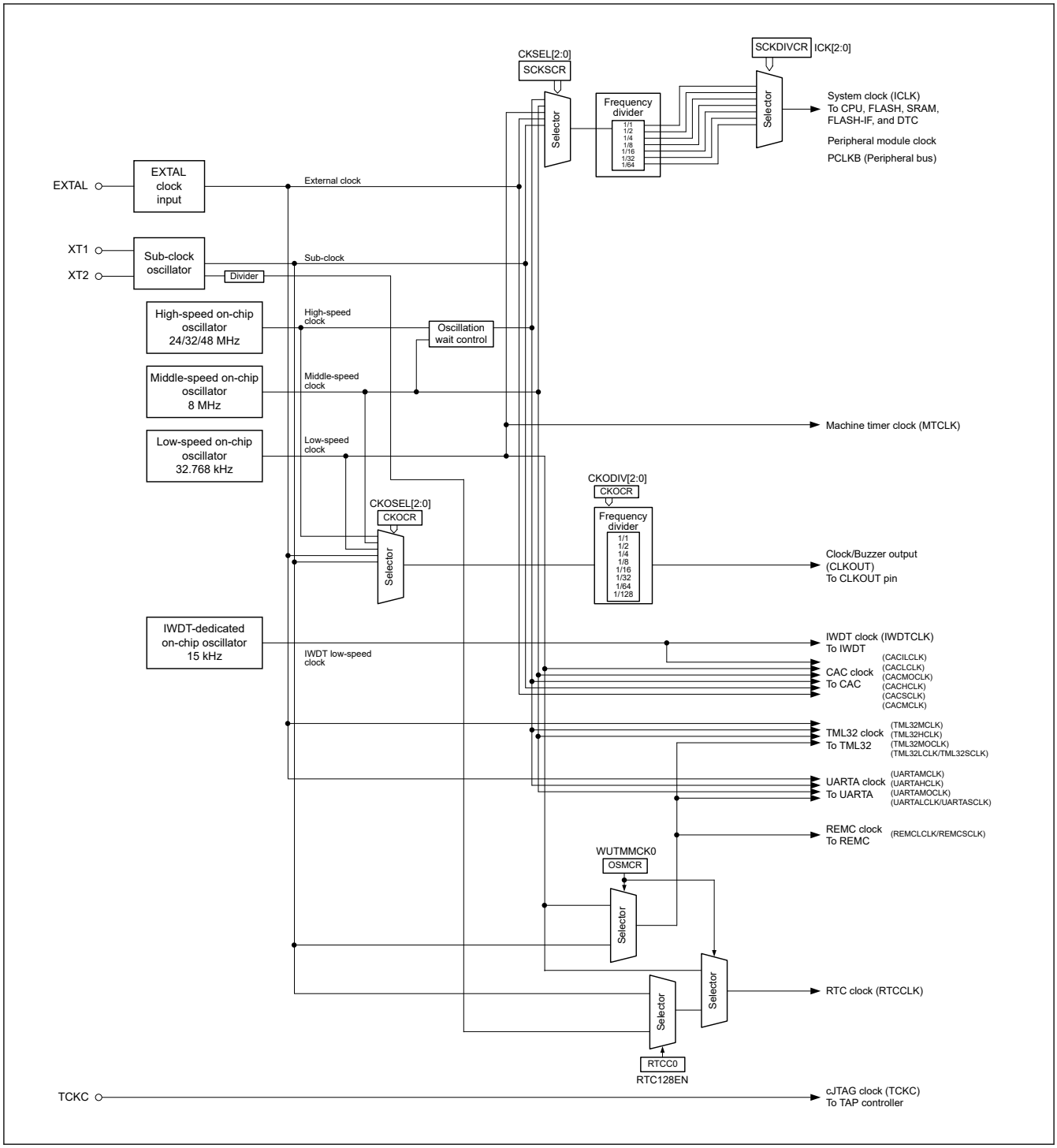


Figure 8.2 Clock generation circuit block diagram (Internal Clock Supply Architecture Type B)

Table 8.3 Clock generation circuit input/output pins

Pin name	I/O	Description
EXTAL	Input	The EXTAL pin can be used to input an external clock. For details, see section 8.3. External Clock Input.
XT1	Input	These pins are used to connect a 32.768-kHz crystal resonator
XT2	Output	
CLKOUT	Output	This pin is used to output the CLKOUT/BUZZER clock
TCKC	Input	This pin is used for input from the cJTAG

8.2 Register Descriptions

8.2.1 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:	—	—	—	—	—	ICK[2:0]			—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	1	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:	—	—	—	—	—	PCKB[2:0]			—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0

Bit	Symbol	Function	R/W
2:0	—	These bits are read as 100b. The write value should be 100b.	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
10:8	PCKB[2:0]	Peripheral Module Clock B (PCLKB) Select*1 *2 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
23:11	—	These bits are read as 0. The write value should be 0.	R/W
26:24	ICK[2:0]	System Clock (ICLK) Select*1 *2 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
31:27	—	These bits are read as 0. The write value should be 0.	R/W

Note: For Internal Clock Supply Architecture Type B, the association between the frequencies of the system clock (ICLK), the peripheral module clock (PCLKB) should be ICLK : PCLKB = 1 : 1.
Write the same value to ICK[2:0] and PCKB[2:0] when setting SCKDIVCR in Internal Clock Supply Architecture Type B.

Note 1. The association between the frequencies of the system clock (ICLK) and the peripheral module clock (PCLKB) should be ICLK:PCLKB = N:1 (N: integer).

Note 2. Selecting division by 1 to ICLK is prohibited when SCKSCR.CKSEL[2:0] bits select the system clock source that is faster than 32 MHz and MEMWAIT.MEMWAIT = 0.

The SCKDIVCR register selects the frequencies of the system clock (ICLK), and peripheral module clock (PCLKB).

8.2.2 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*1 0 0 0: HOCO 0 0 1: MOCO 0 1 0: LOCO 0 1 1: External clock input (EXTAL) 1 0 0: Sub-clock oscillator (SOSC) 1 0 1: Setting prohibited 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 1. Selecting a system clock source that is faster than 32 MHz (system clock source > 32 MHz) is prohibited when the SCKDIVCR.ICK[2:0] bits select division by 1 and MEMWAIT.MEMWAIT = 0.

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

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 (PCLKB)

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)
- Sub-clock oscillator (SOSC)
- External clock input (EXTAL)

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.

8.2.3 MEMWAIT : Memory Wait Cycle Control Register for Code Flash

Base address: SYSC = 0x4001_E000

Offset address: 0x031

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

Bit	Symbol	Function	R/W
0	MEMWAIT	Memory Wait Cycle Select for Code Flash 0: No wait 1: Wait	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/(W)

Note: Writing 0 to the MEMWAIT bit is prohibited when SCKDIVCR.ICK bit selects division by 1 and SCKSCR.CKSEL[2:0] bits select the system clock source that is faster than 32 MHz (ICLK > 32 MHz).
Note: For Internal Clock Supply Architecture Type B selected by OFS1.ICSATS bit, the MEMWAIT settings is prohibited.

The MEMWAIT register controls the wait cycle of code flash read access.

MEMWAIT bit (Memory Wait Cycle Select for Code Flash)

This bit selects the wait cycle of code flash read access. The wait cycle of code flash access is set to no wait (MEMWAIT = 0) after a reset is released.

Before writing to the MEMWAIT bit, check the ICLK frequency and operation power control mode. The following restrictions apply when setting the ICLK and operation power control mode, and the MEMWAIT bit:

- When setting the ICLK to faster than 32 MHz (ICLK > 32 MHz), set MEMWAIT to 1 while ICLK is 32 MHz or less (ICLK ≤ 32 MHz) and the operation power control mode is High-speed mode (OPCCR.OPCM[1:0] = 00b). Setting MEMWAIT to 1 is prohibited in operation modes other than High-speed mode. Setting the ICLK faster than 32 MHz is prohibited while MEMWAIT = 0.
- When setting the ICLK from 32 MHz or faster (ICLK > 32 MHz) to 32 MHz or less (ICLK ≤ 32 MHz), the ICLK frequency must be set to 32 MHz or less while MEMWAIT = 1. Setting MEMWAIT to 0 is prohibited while ICLK is faster than 32 MHz. Setting MEMWAIT to 1 is prohibited in operation modes other than High-speed mode. MEMWAIT can be set to 0 while the ICLK frequency is 32 MHz or less and operation power control mode is High-speed mode (OPCCR.OPCM[1:0] = 00b).

Table 8.4 MEMWAIT bit setting

MEMWAIT bit	MCU operation power control		
		High-speed mode	
	Mode: except High-speed mode	ICLK ≤ 32 MHz	ICLK > 32 MHz
0	✓	✓	—
1	—	✓	✓

Note: ✓ : Setting is possible.
— : Setting is not possible.

8.2.4 FLDWAITR : Memory Wait Cycle Control Register for Data Flash

Base address: FLCN = 0x407E_C000

Offset address: 0x3FC4

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	FLDWAIT1
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	FLDWAIT1	Memory Wait Cycle Select for Data Flash 0: 1 wait access (Default) 1: 2 wait access	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: Writing 0 to the FLDWAIT1 bit is prohibited when SCKDIVCR.ICK bit selects division by 1 and SCKSCR.CKSEL[2:0] bits select the system clock source that is faster than 32 MHz (ICLK > 32 MHz).
Note: For Internal Clock Supply Architecture Type B selected by OFS1.ICSATS bit, setting FLDWAIT1 is prohibited.
Note: There is no need to set FLDWAIT1 if data flash is not used.

The FLDWAITR register controls the wait cycle of data flash read access.

FLDWAIT1 bit (Memory Wait Cycle Select for Data Flash)

This bit selects the wait cycle of data flash read access. The wait cycle of data flash access is set to 1 wait (FLDWAIT1 = 0) after a reset is released.

The FLDWAIT1 settings for the data flash read access wait cycle are as follows:

- FLDWAIT1 = 0: 1 wait cycle
- FLDWAIT1 = 1: 2 wait cycles

Before writing to the FLDWAIT1 bit, check the ICLK frequency and operation power control mode. The following restrictions apply when setting the ICLK and operation power control mode, and the FLDWAIT1 bit:

- When setting the ICLK faster than 32 MHz (ICLK > 32 MHz), set FLDWAIT1 to 1 while ICLK is 32 MHz or less (ICLK ≤ 32 MHz) and the operation power control mode is High-speed mode (OPCCR.OPCM[1:0] = 00b). Setting FLDWAIT1 to 1 is prohibited in operation modes other than High-speed mode. Setting the ICLK faster than 32 MHz is prohibited while FLDWAIT1 = 0.
- When setting the ICLK from 32 MHz or faster (ICLK > 32 MHz) to 32 MHz or less (ICLK ≤ 32 MHz), the ICLK frequency must be set to 32 MHz or less while FLDWAIT1 = 1. Setting FLDWAIT1 to 0 is prohibited while ICLK is faster than 32 MHz. Setting FLDWAIT1 to 1 is prohibited in operation modes other than High-speed mode. FLDWAIT1 can be set to 0 while the ICLK frequency is 32 MHz or less and operation power control mode is High-speed mode (OPCCR.OPCM[1:0] = 00b).

Table 8.5 FLDWAIT1 bit setting

FLDWAIT1 bit	MCU operation power control		
	Mode: except High-speed mode	High-speed mode	
		ICLK ≤ 32 MHz	ICLK > 32 MHz
0	✓	✓	—
1	—	✓	✓

Note: ✓ : Setting is possible.
— : Setting is not possible.

Figure 8.3 shows an example flow when setting ICLK faster than 32 MHz.

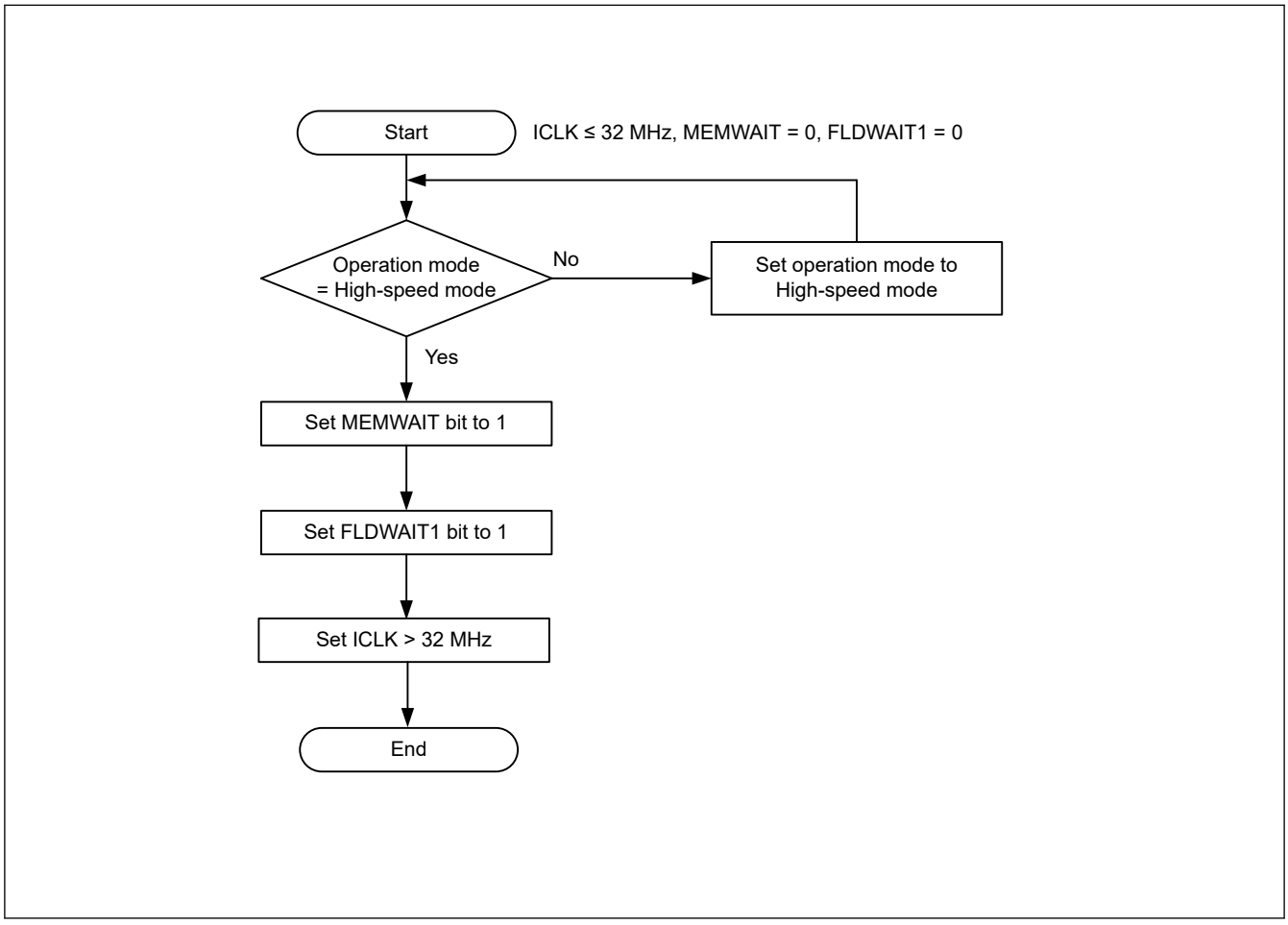


Figure 8.3 When setting ICLK > 32 MHz

Figure 8.4 shows an example flow when setting ICLK less than or equal to 32 MHz when ICLK is greater than 32 MHz.

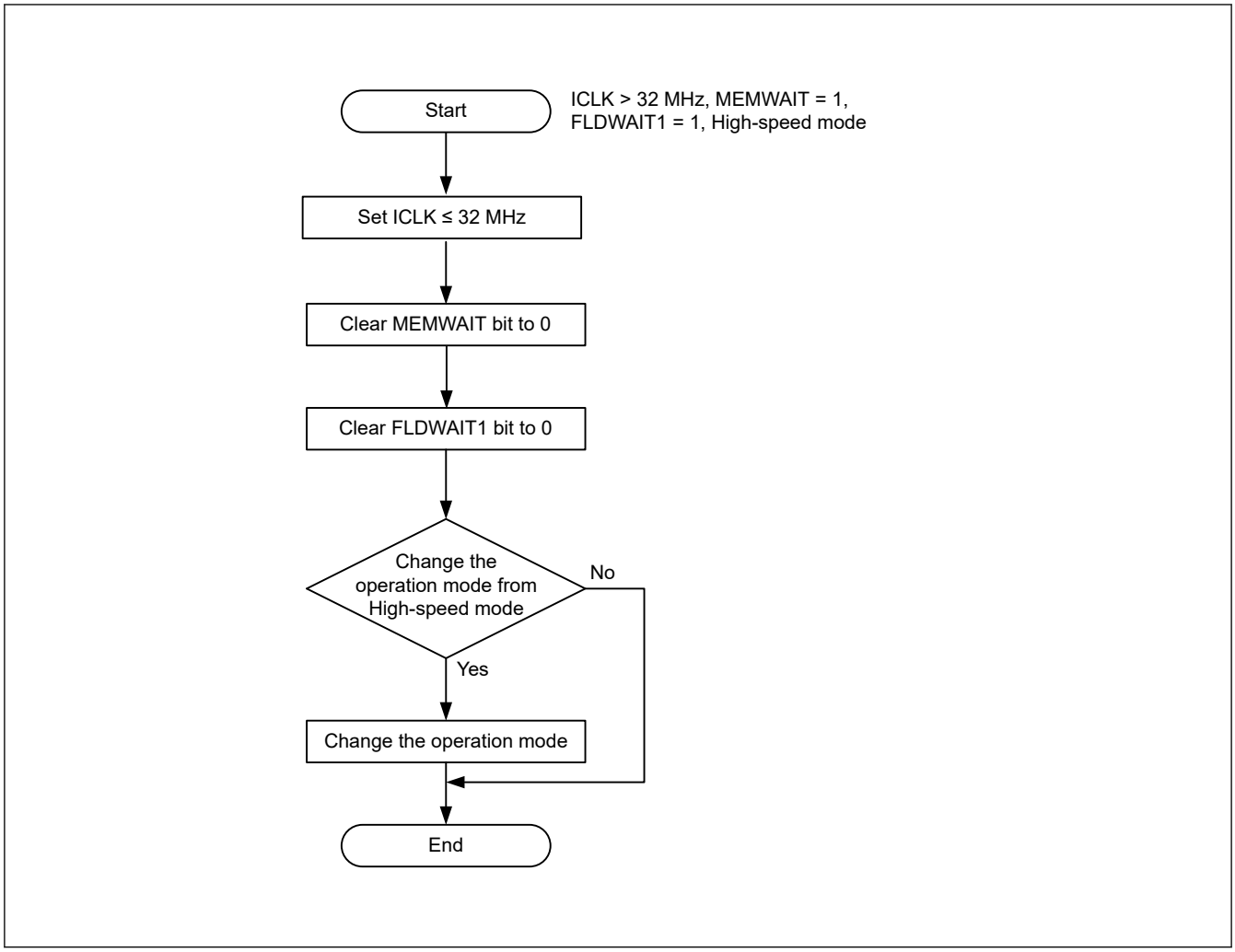


Figure 8.4 When setting ICLK ≤ 32 MHz from ICLK > 32 MHz

8.2.5 MOSCCR : External Clock Input 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	External Clock Input Stop 0: Operate the external clock input 1: Stop the external clock input	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.

The MOSCCR register controls the external clock input.

MOSTP bit (External Clock Input Stop)

The MOSTP bit starts or stops the external clock input.

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

Writing 1 to MOSTP is prohibited under the following condition:

- SCKSCR.CKSEL[2:0] = 011b (system clock source = EXTAL).

8.2.6 SOSCCR : Sub-Clock Oscillator Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x480

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

Bit	Symbol	Function	R/W
0	SOSTP	Sub Clock Oscillator Stop 0: Operate the sub-clock oscillator*1 1: Stop the sub-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 1. The SOMCR register must be set before setting SOSTP to 0.

The SOSCCR register controls the sub-clock oscillator.

SOSTP bit (Sub Clock Oscillator Stop)

The SOSTP bit starts or stops the sub-clock oscillator. When changing the value of the SOSTP bit, only execute subsequent instructions after reading the bit to check that the value is updated. Use the SOSTP bit when using the sub-clock oscillator as the source for a peripheral module, for example the RTC. When using the sub-clock oscillator, set the Sub-Clock Oscillator Mode Control Register (SOMCR) before setting SOSTP to 0.

The following restrictions apply when starting and stopping the operation:

- After stopping the sub-clock oscillator, allow a stop interval of at least 5 SOSC clock cycles before restarting it
- After setting the SOSTP bit to 0, use the sub-clock only after the sub-clock oscillation stabilization time (t_{SUBOSC}) has elapsed.
- Regardless of whether the sub-clock oscillator is selected as the system clock, confirm that the sub-clock oscillation is stable before executing a WFI instruction to place the MCU in Software Standby mode
- When a transition to Software Standby mode is to follow the setting to stop the sub-clock oscillator, wait for at least 3 SOSC clock cycles before executing the WFI instruction.

Writing 1 to SOSTP is prohibited under the following condition:

- SCKSCR.CKSEL[2:0] = 100b (system clock source = SOSC).

8.2.7 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

Bit	Symbol	Function	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.

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 mode
- When a transition to 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.

8.2.8 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, *4 1: Stop the HOCO clock	R/W*3
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: Writing to OPCCR.OPCM[1:0] is prohibited while HOCOCR.HCSTP = 0 and OSCSF.HOCOSF = 0 (HOCO is in stabilization wait counting).
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.HOCOFRQ1[2:0] bit to an optimum value.
Note 3. Writing HCSTP is prohibited while OPCCR.OPCMTSF = 1 or SOPCCR.SOPCMTSF = 1 (during transition of operating power control mode).
Note 4. The value of OFS1.HOCOFRQ1[2:0] bits is automatically transferred to HOCOCR2.HCFRQ1[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ1[2:0] even if OFS1.HOCOFRQ1[2:0] is not an appropriate 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 mode after setting HOCO operation with the HCSTP bit.
- When a transition to 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 condition:

- SCKSCR.CKSEL[2:0] = 000b (system clock source = HOCO).

8.2.9 HOCOCR2 : High-Speed On-Chip Oscillator Control Register 2

Base address: SYSC = 0x4001_E000

Offset address: 0x037

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	HCFRQ1[2:0]			—	—	—
Value after reset:	0	0	0/1*1	0/1*1	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
5:3	HCFRQ1[2:0]	HOCO Frequency Setting 1 0 0 0: 24 MHz 0 1 0: 32 MHz 1 0 0: 48 MHz Others: Setting prohibited	R/W
7:6	—	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 1. Value after reset of the HCFRQ1[2:0] bits depends on OFS1.HOCOFRQ1[2:0] bits.

The HOCOCR2 register controls the HOCO clock.

Writing to the HOCOCR2 is prohibited when the HOCOCR.HCSTP bit is 0 (the HOCO operates).

8.2.10 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

Bit	Symbol	Function	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.

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_{MOCO}) 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 mode
- When a transition to 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).

8.2.11 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:	—	—	—	—	—	—	—	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
7:1	—	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.

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 MOCO clock and supply of the HOCO clock within the MCU is started. For the HOCO oscillation stabilization time, see section 37, Electrical Characteristics.

[Clearing condition]

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

8.2.12 SOMCR : Sub-Clock Oscillator Mode Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x481

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

Bit	Symbol	Function	R/W
1:0	SODRV[1:0]	Sub-Clock Oscillator Drive Capability Switching 0 0: Normal Mode 0 1: Low Power Mode 1 1 0: Low Power Mode 2 1 1: Low Power Mode 3	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W

The SOMCR register must be modified when SOSCCR.SOSTP is 1 (SOSC is stopped).

SODRV[1:0] bits (Sub-Clock Oscillator Drive Capability Switching)

The SODRV[1:0] bits switch the drive capability of the sub-clock oscillator. The relationship between the drive capability and the setting value is as follows:

Normal Mode > Low Power Mode 1 > Low Power Mode 2 > Low Power Mode 3

8.2.13 SOMRG : Sub-Clock Oscillator Margin Check Register

Base address: SYSC = 0x4001_E000

Offset address: 0x482

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

Bit	Symbol	Function	R/W
1:0	SOSCMRG[1:0]	Sub Clock Oscillator Margin check Switching 0 0: Normal Current 0 1: Lower Margin check 1 0: Upper Margin check 1 1: Setting prohibited	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W

SOSCMRG[1:0] bits (Sub Clock Oscillator Margin check Switching)

The SOSCMRG[1:0] bits control amp current in the SOSC for oscillation margin check.

8.2.14 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: External clock input (EXTAL) 1 0 0: Sub-clock oscillator (SOSC) 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.

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. The division ratio of the output clock frequency must be set to a value no higher than the characteristics of the CLKOUT pin output frequency. For details on the characteristics of the CLKOUT pin, see section 37, Electrical Characteristics.

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 mode if the selecting clock out source clock is stopped in that mode.

8.2.15 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: 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.

8.2.16 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: 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.

8.2.17 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: 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.

8.2.18 OSMCR : Subsystem Clock Supply Mode Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x054

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	WUTMMCK0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	WUTMMCK0	Selection of the operating clock for the realtime clock, 32-bit interval timer, serial interfaces UARTA0 and UARTA1, remote control signal receiver 0: Subsystem clock (SOSC) 1: Low-speed on-chip oscillator clock (LOCO)*1 *2	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Do not set the WUTMMCK0 bit to 1 while the subsystem clock (SOSC) is oscillating.

Note 2. Switching between the subsystem clock (SOSC) and the low-speed on-chip oscillator clock can be enabled by the WUTMMCK0 bit only when all of the realtime clock, 32-bit interval timer, serial interfaces UARTA0 and UARTA1, and remote control signal receiver are stopped.

The OSMCR register can be used to select the operating clock for the realtime clock, 32-bit interval timer, serial interfaces UARTA0 and UARTA1, and remote control signal receiver.

8.3 External Clock Input

Figure 8.5 shows an example of connecting an external clock input.

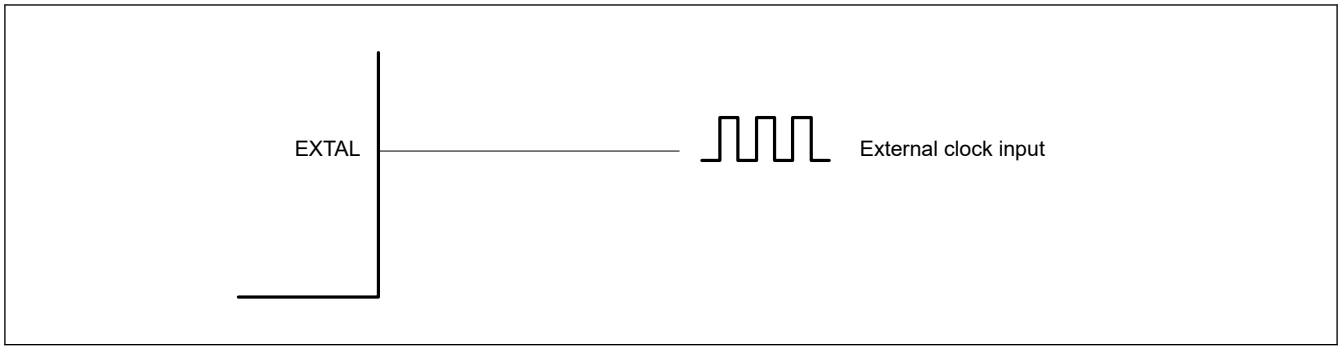


Figure 8.5 Equivalent circuit for external clock

8.3.1 Notes on External Clock Input

Do not change the frequency of the external clock input when the setting of the External Clock Input Stop bit (MOSCCR.MOSTP) is 0.

8.4 Sub-Clock Oscillator

The only way of supplying a clock signal to the sub-clock oscillator is by connecting a crystal oscillator.

8.4.1 Connecting a 32.768-kHz Crystal Resonator

To supply a clock to the sub-clock oscillator, connect a 32.768-kHz crystal resonator as shown in Figure 8.6. A damping resistor (Rd) can be added, if necessary. Because the resistor values vary according to the resonator and the oscillation drive capability, use values recommended by the resonator manufacturer. If the resonator manufacturer recommends the use of an external feedback resistor (Rf), insert an Rf between XT1 and XT2 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 sub-clock oscillator as described in Table 8.1.

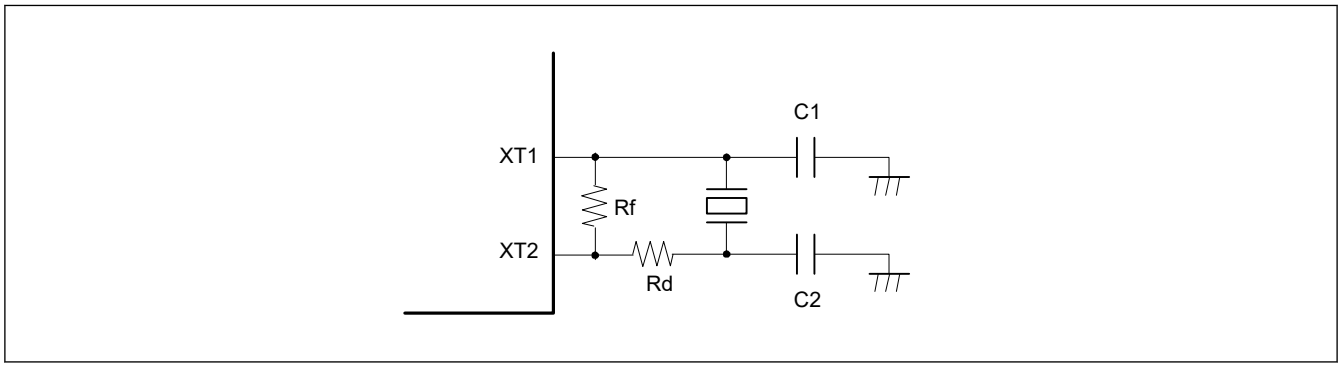


Figure 8.6 Connection example of 32.768-kHz crystal resonator

Figure 8.7 shows an equivalent circuit for the 32.768-kHz crystal resonator.

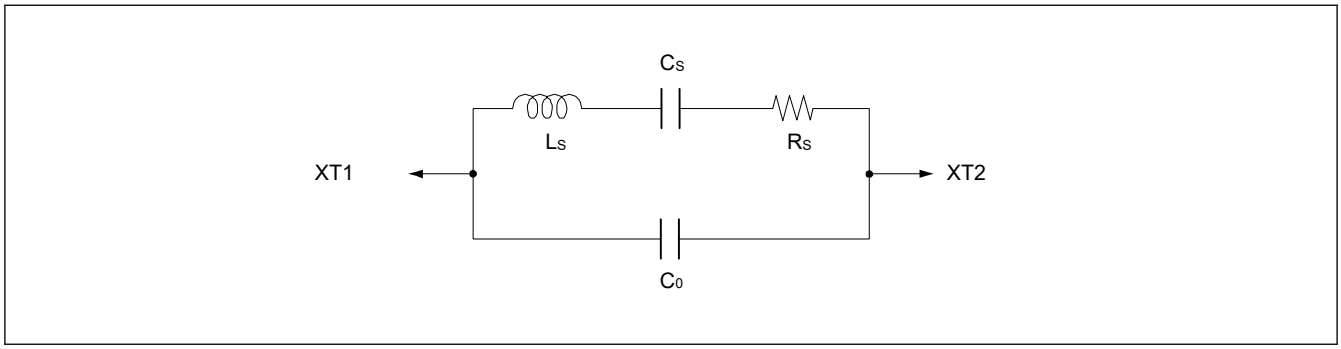


Figure 8.7 Equivalent circuit for the 32.768-kHz crystal resonator

8.4.2 Pin Handling When the Sub-Clock Oscillator Is Not Used

When the sub-clock oscillator is not in use, connect the XT1 pin to VSS through a resistor (to pull VSS down) and leave the XT2 pin open as shown in Figure 8.8. In addition, if an oscillator is not connected, set the Sub-Clock Oscillator Stop bit (SOSCCR.SOSTP) to 1 to stop the oscillator.

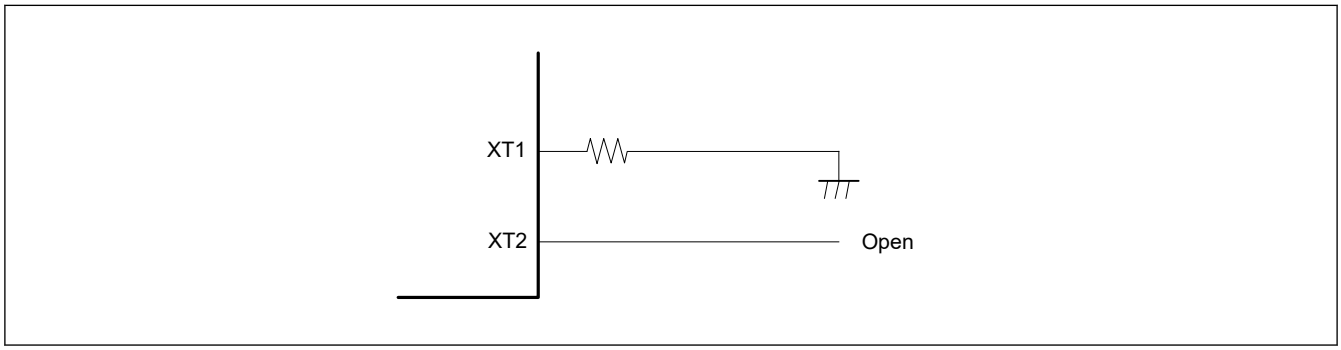


Figure 8.8 Pin handling when the sub-clock oscillator is not used

8.5 Internal Clock

Clock sources for the internal clock signals include:

- EXTAL clock
- Sub-clock
- HOCO clock
- MOCO clock
- LOCO clock
- IWDT-dedicated clock

The following internal clocks are produced from these sources.

- Operating clock of the CPU, DTC, Flash, Flash-IF, and SRAM — system clock (ICLK)
- Operating clocks of peripheral modules — PCLKB
- Operating clocks for the CAC — CACCLK
- Operating clock for the RTC sub clock — RTCSCLK
- Operating clock for the RTC sub 128 Hz clock — RTCS128CLK
- Operating clock for the RTC LOCO clock — RTCLCLK
- Operating clock for the TML32 External clock — TML32MCLK
- Operating clock for the TML32 HOCO clock — TML32HCLK
- Operating clock for the TML32 MOCO clock — TML32MOCLK
- Operating clock for the TML32 LOCO clock — TML32LCLK
- Operating clock for the TML32 sub clock — TML32SCLK
- Operating clock for the UARTA External clock — UARTAMCLK
- Operating clock for the UARTA HOCO clock — UARTAHCLK
- Operating clock for the UARTA MOCO clock — UARTAMOCLK
- Operating clock for the UARTA LOCO clock — UARTALCLK
- Operating clock for the UARTA sub clock — UARTASCLK
- Operating clock for the REMC LOCO clock — REMCLCLK
- Operating clock for the REMC sub clock — REMCSCLK
- Operating clock for the IWDT — IWDTCLK

- Operating clock for the Machine timer — MTCLK
- Clock for external pin output — CLKOUT.

For details of the registers used to set the frequencies of the internal clocks, see section 8.5.1. System Clock (ICLK) to section 8.5.10. External Pin Output Clock (CLKOUT).

If the value of any of these bits is changed, subsequent operation is at the frequency determined by the new value.

8.5.1 System Clock (ICLK)

The system clock (ICLK) is the operating clock for the CPU, DTC, Flash, Flash-IF, and SRAM.

The ICLK frequency is specified by the HOCOFRQ1[2:0] bits in OFS1, the ICK[2:0] bits in SCKDIVCR, and the CKSEL[2:0] bits in SCKSCR.

The value of OFS1.HOCOFRQ1[2:0] bits is automatically transferred to HOCOCR2.HCFRQ1[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ1[2:0] bits.

When the ICLK clock source is switched, the duration of the ICLK clock cycle becomes longer during the clock source transition period. See Figure 8.9 and Figure 8.10.

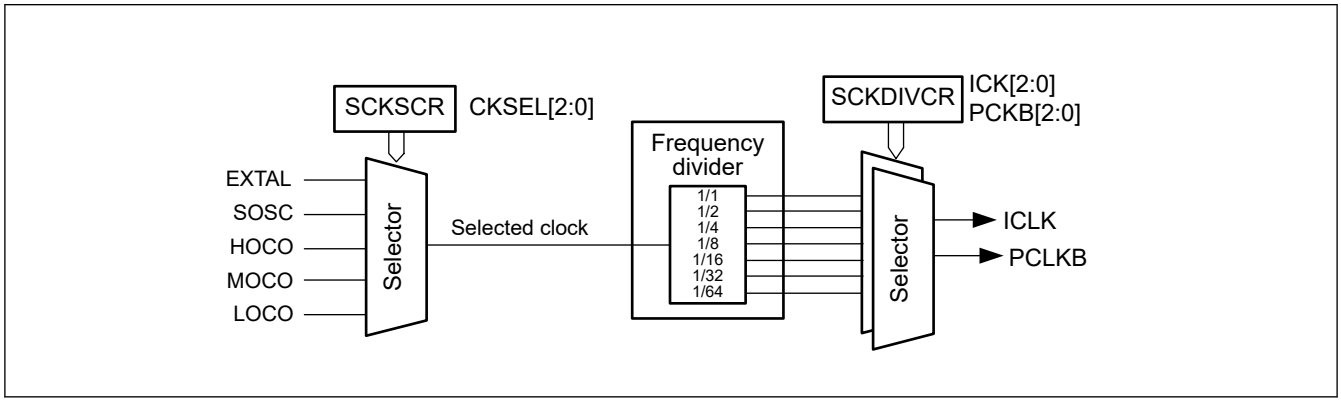


Figure 8.9 Block diagram of clock source selector

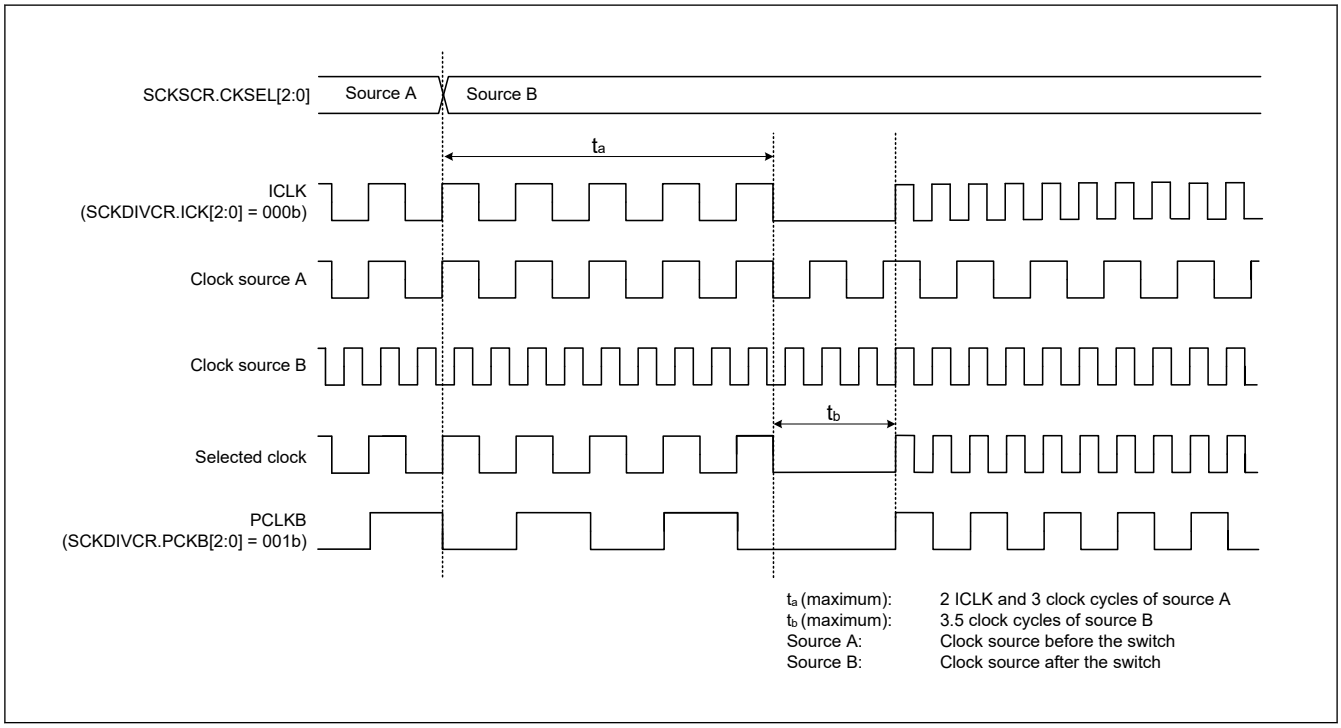


Figure 8.10 Timing of clock source switching

8.5.2 Peripheral Module Clock (PCLKB)

The peripheral module clock (PCLKB) is the operating clocks for the peripheral modules.

The frequency of the given clock is specified in the following bits:

- HOCOFRQ1[2:0] in OFS1*1.
- PCKB[2:0] in SCKDIVCR
- CKSEL[2:0] in SCKSCR

Note 1. The value of OFS1.HOCOFRQ1[2:0] bits is automatically transferred to HOCOCR2.HCFRQ1[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ1[2:0] bits.

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 8.9 and Figure 8.10.

8.5.3 CAC Clock (CACCLK)

The CAC clock (CACCLK) is the operating clock for the CAC. CACCLK is generated by the following oscillators:

- External clock input (EXTAL)
- Sub-clock oscillator (SOSC)
- High-speed clock oscillator (HOCO)
- Middle-speed clock oscillator (MOCO)
- Low-speed on-chip oscillator (LOCO)
- IWDT-dedicated on-chip oscillator. (IWDTLOCO)

8.5.4 RTC-Dedicated Clock (RTCSCLK, RTCS128CLK, RTCLCLK)

The RTC-dedicated clocks (RTCSCLK, RTCS128CLK, RTCLCLK) are the operating clocks for the RTC. RTCSCLK and RTCS128CLK are generated by the sub-clock oscillator, and RTCLCLK is generated by the LOCO clock.

8.5.5 TML32-Dedicated Clock

TML32-dedicated clocks (TML32MCLK, TML32HCLK, TML32MOCLK, TML32LCLK, TML32SCLK) are the operating clocks for the TML32. These clocks are generated by the following oscillators:

- External clock input (TML32MCLK)
- Sub-clock oscillator (TML32SCLK)
- High-speed clock oscillator (TML32HCLK)
- Middle-speed clock oscillator (TML32MOCLK)
- Low-speed clock oscillator (TML32LCLK)

Do not stop the clock or change the frequency when TML32 is operating with this dedicated clock. If this clock is changed during operation, operation cannot be guaranteed.

8.5.6 UARTA-Dedicated Clock

UARTA-dedicated clocks (UARTAMCLK, UARTAHCLK, UARTAMOCLK, UARTALCLK, UARTASCLK) are the operating clocks for the UARTA. These clocks are generated by the following oscillators:

- External clock input (UARTAMCLK)
- Sub-clock oscillator (UARTASCLK)
- High-speed clock oscillator (UARTAHCLK)
- Middle-speed clock oscillator (UARTAMOCLK)
- Low-speed clock oscillator (UARTALCLK)

Do not stop the clock or change the frequency when UARTA is operating with this dedicated clock. If this clock is changed during operation, operation cannot be guaranteed.

8.5.7 REMC-Dedicated Clock

The REMC-dedicated clock (REMCLCLK, REMCSCLK) are the operating clocks for the REMC. REMCSCLK is generated by the sub-clock oscillator, and REMCLCLK is generated by the LOCO clock.

Do not stop the clock or change the frequency when REMC is operating with this dedicated clock. If this clock is changed during operation, operation cannot be guaranteed.

8.5.8 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.

8.5.9 Machine Timer-Dedicated Clock (MTCLK)

The Machine Timer-Dedicated Clock (MTCLK) is the operating clock for the Machine timer. MTCLK is generated by the LOCO clock.

8.5.10 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
- HOCOFRQ1[2:0] bits in OFS1*1

Note 1. The value of OFS1.HOCOFRQ1[2:0] bits is automatically transferred to HOCOCR2.HCFRQ1[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ1[2:0] bits.

8.6 Usage Notes

8.6.1 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 (PCLKB)

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 37, 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.

8.6.2 Notes on Resonator

Because various resonator characteristics relate closely to your board design, adequate evaluation is required before use. See the resonator connection example in Figure 8.6. The circuit constants for the resonator depend on the resonator to be used and the stray capacitance of the mounting circuit. Therefore, consult the resonator manufacturer when determining the circuit constants. The voltage to be applied between the resonator pins must be within the absolute maximum rating.

8.6.3 Notes on Board Design

When using a crystal resonator, place the resonator and its load capacitors as close to the XT1 and XT2 pins as possible. Other signal lines should be routed away from the oscillation circuit as shown in Figure 8.11 to prevent electromagnetic induction from interfering with correct oscillation. Figure 8.11 shows the case which the sub-clock oscillator is used.

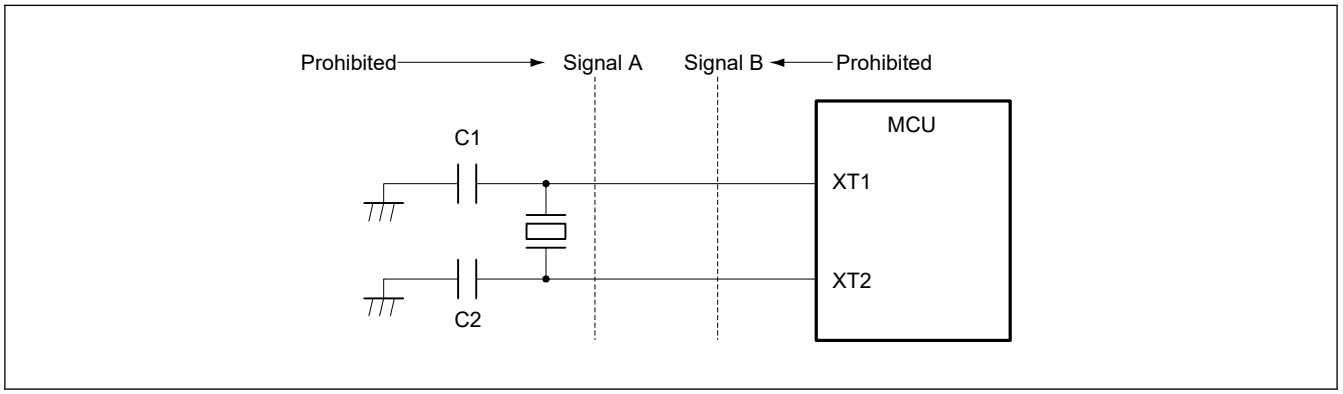


Figure 8.11 Signal routing in board design for oscillation circuit

8.6.4 Notes on External Clock Input Pin

The EXTAL pin can be used as general port. When this pin is used as general port, the external 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: • External clock input (EXTAL) • Sub-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 • External clock input (EXTAL) • Sub-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

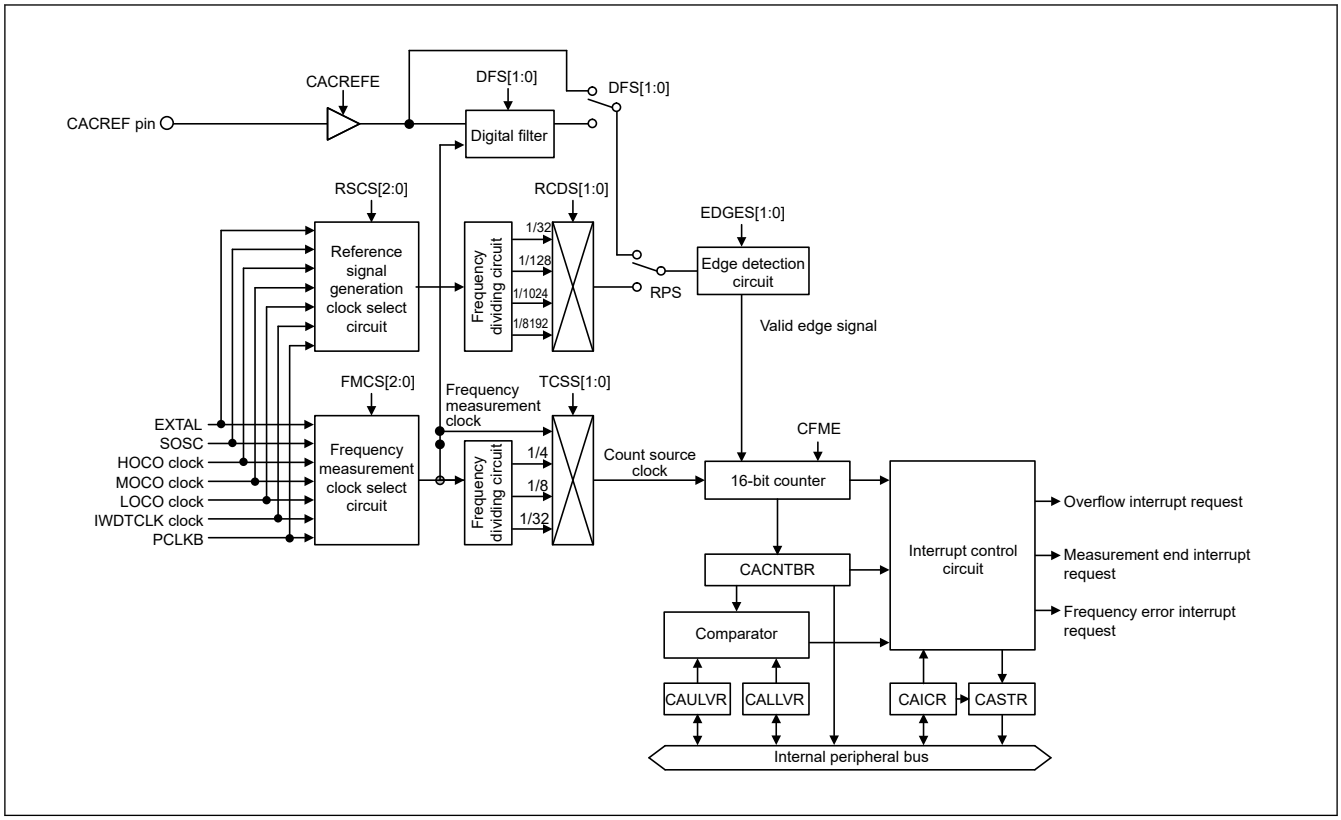


Figure 9.1 CAC block diagram

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 = 0x4004_4600

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 = 0x4004_4600

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: External clock input (EXTAL) 0 0 1: Sub-clock oscillator 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 = 0x4004_4600

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: External clock input (EXTAL) 0 0 1: Sub-clock oscillator 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 = 0x4004_4600

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 = 0x4004_4600

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 = 0x4004_4600

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 = 0x4004_4600

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 = 0x4004_4600

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.

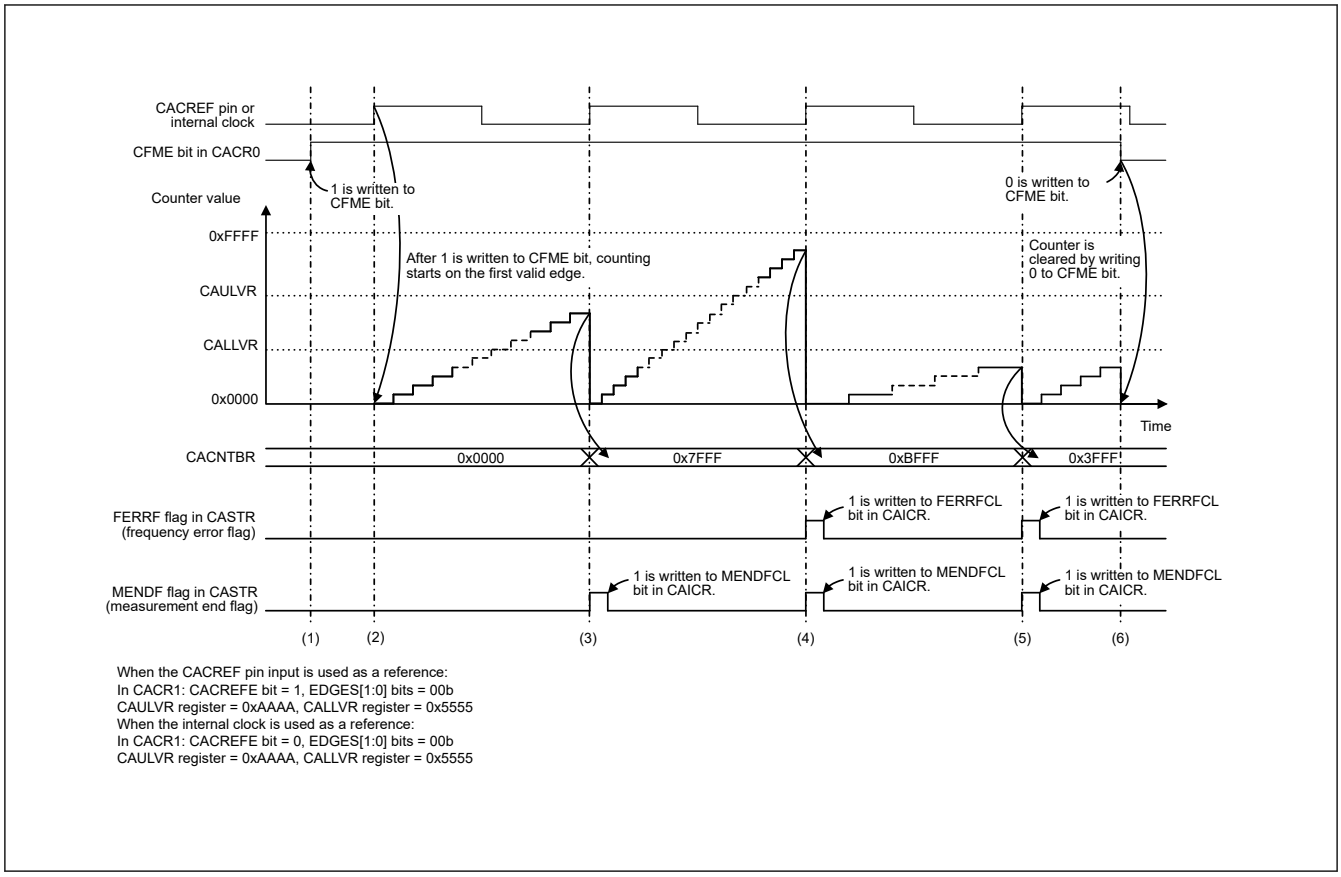


Figure 9.2 CAC operating example

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

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
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 provides 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 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 clock (PCLKB)*1
Module stop	Functions can be stopped independently for each peripheral module
Low power modes	• Sleep mode • Software Standby mode • Snooze mode
Power control modes	Power consumption can be reduced in Normal, Sleep, and Snooze mode by selecting an appropriate operating power control mode according to the operating frequency and voltage. Four operating power control modes are available: • High-speed mode • Middle-speed mode • Low-speed mode • Subosc-speed mode

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*1
Transition condition	WFI instruction while SBYCR.SSBY = 0	WFI instruction while SBYCR.SSBY = 1	Snooze request in Software Standby mode. SNZCR.SNZE = 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.
State after cancellation by an interrupt	Program execution state (interrupt processing)	Program execution state (interrupt processing)	Program execution state (interrupt processing)
State after cancellation by a reset	Reset state	Reset state	Reset state
External clock input	Selectable	Stop	Selectable*2
Sub-clock oscillator	Selectable	Selectable	Selectable
High-speed on-chip oscillator	Selectable	Stop	Selectable
Middle-speed on-chip oscillator	Selectable	Stop	Selectable
Low-speed on-chip oscillator	Selectable	Selectable	Selectable
IWDT-dedicated on-chip oscillator	Selectable*4	Selectable*4	Selectable*4
Clock/buzzer output function	Selectable	Selectable*3	Selectable
CPU	Stop (Retained)	Stop (Retained)	Stop (Retained)
SRAM	Selectable	Stop (Retained)	Selectable
Flash memory	Operating	Stop (Retained)	Stop (Retained)
Data Transfer Controller (DTC)	Selectable	Stop (Retained)	Selectable
Watchdog Timer (WDT)	Selectable*4	Stop (Retained)	Stop (Retained)
Independent Watchdog Timer (IWDT)	Selectable*4	Selectable*4	Selectable*4

Table 10.2 Operating conditions of each low power mode (2 of 2)

Item	Sleep mode	Software Standby mode	Snooze mode*1
Real Time Clock (RTC)	Selectable	Selectable	Selectable
Timer Array Unit (TAU)	Selectable	Stop (Retained)	Selectable
32-bit Interval Timer (TML32)	Selectable	Selectable*5	Selectable*5
12-bit A/D Converter (ADC12)	Selectable	Stop (Retained)	Selectable
8-bit D/A Converter (DAC8)	Selectable	Stop (Retained)	Selectable
Data Operation Circuit (DOC)	Selectable	Stop (Retained)	Selectable
Remote Control Signal Receiver (REMC)	Selectable	Stop (Retained)*11	Selectable
Serial Array Unit (SAU)	Selectable	Stop (Retained)*10	Selectable
I^2C Bus Interface (IICA)	Selectable	Stop (Retained)*9	Selectable*9
Serial Interface UARTA (UARTA)	Selectable	Selectable*8	Selectable
Event Link Controller (ELC)	Selectable	Stop (Retained)	Selectable*6
Comparator (CMP)	Selectable	Selectable*7	Selectable*7
NMI, IRQn (n = 0 to 7) pin interrupt	Selectable	Selectable	Selectable
Key Interrupt Function (KINT)	Selectable	Selectable	Selectable
Low Voltage Detection (LVD)	Selectable	Selectable	Selectable
Power-on reset circuit	Operating	Operating	Operating
Other peripheral modules	Selectable	Stop (Retained)	Operation prohibited
I/O ports	Operating	Retained	Operating

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.
Otherwise, proper operation is not guaranteed in Snooze mode.

Note 1. All modules whose module-stop bits are 0 start as soon as PCLKs are supplied after entering Snooze mode. To avoid an increasing power consumption in Snooze mode, set the module-stop bit of modules that are not required in Snooze mode to 1 before entering Software Standby mode.
Note 2. When using SAU (UART0/2, SPI00/20) in Snooze mode, MOSCCR.MOSTP bits must be 1.
Note 3. Stopped when the Clock Output Source Select bits (CKOCR.CKOSEL[2:0]) are set to a value other than 010b (LOCO) and 100b (SOSC).
Note 4. 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 or by setting the WDTCSTPR.SLCSTP in WDT register-start mode.
Note 5. TML32 operation is possible when LOCO or SOSC is selected.
Note 6. Events are limited to those described in section 10.9.11. ELC Events in Snooze Mode.
Note 7. Only VCOUT function is permitted. The VCOUT pin operates when CMP uses no digital filter.
Note 8. UARTA operation is possible when LOCO or SOSC is selected.
Note 9. Only wakeup interrupt is available.
Note 10. Able to change to Snooze mode when the value of RxD/SCK port changes.
Note 11. Able to change to Snooze mode when the value of RIN port changes.

Table 10.3 Available interrupt sources to transition to Normal mode from Snooze mode and Software Standby mode (1 of 2)

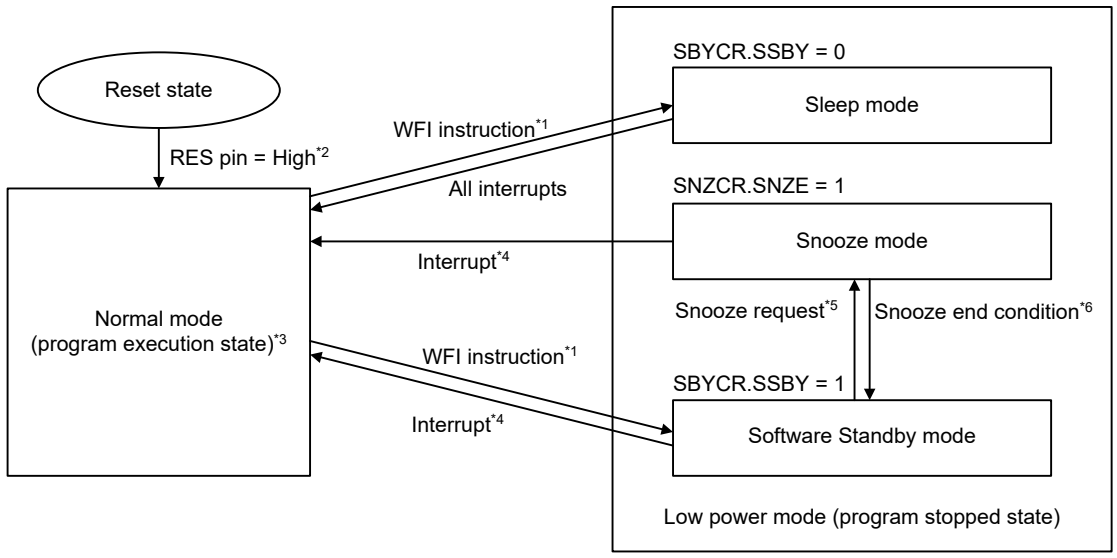
Interrupt source	Name	Software Standby mode	Snooze mode
NMI		Yes	Yes
Port	PORT_IRQn (n = 0 to 7)	Yes	Yes
LVD	LVD_LVD1	Yes	Yes
	LVD_LVD2	Yes	Yes
IWDT	IWDT_NMIUNDF	Yes	Yes
KINT	KEY_INTKR	Yes	Yes
TML32	TML32_OUTI	Yes	Yes*3

Table 10.3 Available interrupt sources to transition to Normal mode from Snooze mode and Software Standby mode (2 of 2)

Interrupt source	Name	Software Standby mode	Snooze mode
RTC	RTC_ALM_OR_PRD	Yes	Yes
CMP	COMP_DET0	Yes	Yes
	COMP_DET1	Yes	Yes
UARTA	UARTA_RX_ENDI0	Yes	Yes
	UARTA_RX_ERI0	Yes	Yes
	UARTA_RX_ENDI1	Yes	Yes
	UARTA_RX_ERI1	Yes	Yes
IICA	IIC0_ENDI/IIC0_WUI	Yes*2	Yes*2
	IIC1_ENDI/IIC1_WUI	Yes*2	Yes*2
REMC	REMC_OUTI	No	Yes with SELSR0 *1
SAU	SAU0_ENDI1	No	Yes with SELSR0*1
	SAU1_ENDI1	No	Yes with SELSR0 *1
	SAU0_INTSRE0	No	Yes with SELSR0 *1
	SAU1_INTSRE2	No	Yes with SELSR0 *1
DTC	DTC_COMPLETE	No	Yes with SELSR0 *1
DOC	DOC_DOPCI	No	Yes with SELSR0 *1
ADC12	ADC_ENDI	No	Yes with SELSR0 *1

Note 1. To use the interrupt request as a trigger for exiting Snooze mode, the request must be selected in SELSR0. See section 12, Interrupt Controller Unit (ICU). When a trigger selected in SELSR0 occurs after executing a WFI instruction, and during the transition from Normal mode to Software Standby mode, the request can be accepted depending on the timing of the occurrence.
Note 2. When response to address matching occurs.
Note 3. Event that is enabled by the SNZEDCR1 register must not be used.

Figure 10.1 shows the transition between Normal mode to low power mode.



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 a transition to low power mode.
Note 2. The MOCO 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 from an interrupt in Sleep mode, Software Standby mode, or Snooze mode. The clock source is the same as before entering the low power mode.
Note 4. See Table 10.3.
Note 5. See section 10.5.1. Setting Operating Power Control Mode.
Note 6. See section 10.8.3. Returning from Snooze Mode to Software Standby Mode.

Figure 10.1 Low power mode transitions

10.2 Register Descriptions

10.2.1 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	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 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: 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 FENTRYR.FENTRY0 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.

10.2.2 MSTPCRA : Module Stop Control Register A

Base address: SYSC = 0x4001_E000

Offset address: 0x01C

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:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
21:0	—	These bits are read as 1. The write value should be 1.	R/W
22	MSTPA22	DTC Module Stop*1 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 1. When rewriting the MSTPA22 bit from 0 to 1, disable the DTC before setting the MSTPA22 bit.

10.2.3 MSTPCRB : Module Stop Control Register B

Base address: MSTP = 0x4004_7000

Offset address: 0x000

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	MSTPB21	MSTPB20	—	—	MSTPB17	—
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	—	These bits are read as 1. The write value should be 1.	R/W
8	MSTPB8	I^2C Bus Interface 1 Module Stop Target module: IICA1 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
9	MSTPB9	I^2C Bus Interface 0 Module Stop Target module: IICA0 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
16:10	—	These bits are read as 1. The write value should be 1.	R/W
17	MSTPB17	Serial Interface UARTA0/UARTA1 Module Stop Target module: UARTA0/UARTA1 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
19:18	—	These bits are read as 1. The write value should be 1.	R/W

Bit	Symbol	Function	R/W
20	MSTPB20	Serial Array Unit 1 Module Stop Target module: SAU1 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
21	MSTPB21	Serial Array Unit 0 Module Stop Target module: SAU0 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
31:22	—	These bits are read as 1. The write value should be 1.	R/W

10.2.4 MSTPCRC : Module Stop Control Register C

Base address: MSTP = 0x4004_7000

Offset address: 0x004

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	MSTPC28	—	—	—	—	—	MSTPC22	—	—	MSTPC19	—	—	—
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 Calculator 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
18:15	—	These bits are read as 1. The write value should be 1.	R/W
19	MSTPC19	Remote Control Signal Receiver Module Stop Target module: REMC 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	MSTPC22	8-bit D/A Converter Module Stop Target module: DAC8 Set the same value as MSTPCRD.MSTPD20. 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
27:23	—	These bits are read as 1. The write value should be 1.	R/W

Bit	Symbol	Function	R/W
28	MSTPC28	True Random Number Generator Module Stop*2 Target module: TRNG 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 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. Set the MSTPC28 bit to 0 once at the beginning of the program, to initialize an unused circuit, even if the TRNG is not used in this MCU. See section 10.9.13. Module-Stop Function for an Unused Circuit.

10.2.5 MSTPCRD : Module Stop Control Register D

Base address: MSTP = 0x4004_7000

Offset address: 0x008

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	MSTPD28	—	—	—	—	—	—	—	MSTPD20	—	—	—	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:	—	—	—	—	—	MSTPD10	MSTPD9	—	—	—	—	—	—	—	—	—
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	—	These bits are read as 1. The write value should be 1.	R/W
9	MSTPD9	32-bit Interval Timer Module Stop Target module: TML32 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
10	MSTPD10	Timer Array Unit Module Stop Target module: TAU 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
15:11	—	These bits are read as 1. The write value should be 1.	R/W
16	MSTPD16	12-bit A/D Converter Module Stop Target module: ADC12 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
19:17	—	These bits are read as 1. The write value should be 1.	R/W
20	MSTPD20	8-bit D/A Converter Module Stop Target module: DAC8 Set the same value as MSTPCRC.MSTPC22. 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
27:21	—	These bits are read as 1. The write value should be 1.	R/W
28	MSTPD28	Comparator Module Stop Target module: CMP 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

10.2.6 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	1

Bit	Symbol	Function	R/W
1:0	OPCM[1:0]	Operating Power Control Mode Select 0 0: High-speed mode 0 1: Middle-speed mode 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

The OPCCR register is used to reduce power consumption in Normal mode, Sleep mode, and Snooze mode. Power consumption can be reduced according to the operating frequency and operating voltage used by the OPCCR setting. For the procedure to change the operating power control modes, see section 10.5. Function for Lower Operating Power Consumption.

OPCM[1:0] bits (Operating Power Control Mode Select)

The OPCM[1:0] bits select the operating power control mode in Normal mode, Sleep mode, and Snooze mode.

Table 10.4 shows the relationship between the operating power control modes, the OPCM[1:0], and SOPCM bits settings.

Writing to OPCCR.OPCM[1:0] is prohibited while MCU is under the following conditions:

1. HOCOCR.HCSTP and OSCSF.HOCOSF are 0 (the oscillation of the HOCO clock is not yet stable).
2. The MCU is in Sleep or Snooze mode, the MCU transitions to Normal mode from Sleep or Snooze mode, the MCU transitions to Sleep, Snooze, or Software Standby mode from Normal mode, or the MCU is in transfer state when in operating power mode.
3. Flash is in programming mode.
4. The MCU is in Subosc-speed mode (SOPCCR.SOPCM bit is 1).

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.

10.2.7 SOPCCR : Sub Operating Power Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x0AA

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

Bit	Symbol	Function	R/W
0	SOPCM	Sub Operating Power Control Mode Select 0: Other than Subosc-speed mode 1: Subosc-speed mode	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
4	SOPCMTSF	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

The SOPCCR register is used to reduce power consumption in Normal mode, Sleep mode, and Snooze mode. Setting this register initiates entry to and exit from Subosc-speed mode. Subosc-speed mode is available only when using the sub-clock oscillator or LOCO without dividing the frequency.

For the procedure to change operating power control modes, see section 10.5. Function for Lower Operating Power Consumption.

SOPCM bit (Sub Operating Power Control Mode Select)

The SOPCM bit selects the operating power control mode in Normal mode, Sleep mode, and Snooze mode. Setting this bit to 1 allows transition to Subosc-speed mode. Setting this bit to 0 allows a return to the operating mode (set in OPCCR.OPCM[1:0]) before the transition to Subosc-speed mode.

Writing to SOPCCR.SOPCM is prohibited while MCU is under the following conditions:

1. The MCU is in Sleep or Snooze mode, the MCU transitions to Normal mode from Sleep, Snooze, or Software Standby mode, the MCU transitions to Sleep, Snooze, or Software Standby mode from Normal mode, or the MCU is in transfer state when in operating power mode.
2. Flash is in programming mode.
3. EXTAL is operating (MOSCCR.MOSTP bit is 0), HOCO is operating (HOCOCR.HCSTP bit is 0), or MOCO is operating (MOCOCR.MCSTP bit is 0).
4. The value of SCKDIVCR register is not equal to 0x00000000.
5. The data flash is disabled (DFLCTL.DFLEN bit is 0).

Table 10.4 shows the relationship between the operating power control modes, the OPCM[1:0], and SOPCM bits settings.

SOPCMTSF flag (Operating Power Control Mode Transition Status Flag)

The SOPCMTSF flag indicates the switching control state when the operating power control mode is switched to or from Subosc-speed mode. This flag becomes 1 when the SOPCM bit is written, and 0 when mode transition completes. Read this flag and confirm that it is 0 before proceeding.

Table 10.4 shows each operating power control mode.

Table 10.4 Operating power control mode

Operating power control mode	OPCM[1:0] bits	SOPCM bit	Power consumption
High-speed mode	00b	0	High
Middle-speed mode	01b	0	↓
Low-speed mode	11b	0	↓
Subosc-speed mode	xxb	1	Low

10.2.8 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	—	—	REMC0REQEN	RXD2REQEN	RXD0REQEN	SNZDTCEN	—
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	SNZDTCEN	DTC Enable in Snooze mode 0: Disable DTC operation 1: Enable DTC operation	R/W
2	RXD0REQEN	RXD0 or SCK00 Snooze Request Enable 0: Ignore RXD0 or SCK00 edge in Software Standby mode 1: Detect RXD0 or SCK00 edge in Software Standby mode	R/W
3	RXD2REQEN	RXD2 or SCK20 Snooze Request Enable 0: Ignore RXD2 or SCK20 edge in Software Standby mode 1: Detect RXD2 or SCK20 edge in Software Standby mode	R/W
4	REMC0REQEN	RIN0 Snooze Request Enable 0: Ignore RIN0 edge in Software Standby mode 1: Detect RIN0 edge in Software Standby mode	R/W
6:5	—	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: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

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.

RXD0REQEN bit (RXD0 or SCK00 Snooze Request Enable)

The RXD0REQEN bit specifies whether to detect an edge of the RXD0 or SCK00 pin in Software Standby mode. To detect an edge of the RXD0 or SCK00 pin, set this bit before entering Software Standby mode. When this bit is set to 1, an edge of the RXD0 or SCK00 pin in Software Standby mode causes the MCU to enter Snooze mode.

RXD2REQEN bit (RXD2 or SCK20 Snooze Request Enable)

The RXD2REQEN bit specifies whether to detect an edge of the RXD2 or SCK20 pin in Software Standby mode. To detect an edge of the RXD2 or SCK20 pin, set this bit before entering Software Standby mode. When this bit is set to 1, an edge of the RXD2 or SCK20 pin in Software Standby mode causes the MCU to enter Snooze mode.

REMC0REQEN bit (RIN0 Snooze Request Enable)

The REMC0REQEN bit specifies whether to detect an edge of the RIN0 pin in software standby mode. To detect an edge of the RIN0 pin, set this bit before entering Software Standby mode. When this bit is set to 1, an edge of the RIN0 pin in Software Standby mode causes the MCU to enter Snooze mode.

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 10.6 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.

10.2.9 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:	—	—	—	—	ADNCRED	DTCNZRED	DTCZRED	—
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	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
3	ADNCRED	ADC Compare Mismatch Snooze End Enable 0: Disable the snooze end request 1: Enable the snooze end request	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W

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 10.7 as a condition to switch from Snooze mode to Software Standby mode, the corresponding bit in 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.

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 14, 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 14, Data Transfer Controller (DTC).

ADNCRED bit (ADC Compare Mismatch Snooze End Enable)

The ADNCRED bit specifies whether to enable transition from Snooze mode to Software Standby mode when ADC conversion is completed during Snooze mode and the A/D conversion end interrupt request signal (ADC_ENDI) is not generated. For details on the trigger conditions, see section 29, 12-bit A/D Converter (ADC12).

10.2.10 SNZEDCR1 : Snooze End Control Register 1

Base address: SYSC = 0x4001_E000

Offset address: 0x095

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	SAU1NCRED	SAU0NCRED	—	—	REMCNCRED	TMLOIED
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TMLOIED	TML32 Interrupt Snooze End Enable 0: Disable the snooze end request 1: Enable the snooze end request	R/W
1	REMCNCRED	REMC No-Interrupt Completion Snooze End Enable 0: Disable the snooze end request 1: Enable the snooze end request	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	SAU0NCRED	SAU0 No-Interrupt Error Completion Snooze End Enable 0: Disable the snooze end request 1: Enable the snooze end request	R/W
5	SAU1NCRED	SAU1 No-Interrupt Error Completion Snooze End Enable 0: Disable the snooze end request 1: Enable the snooze end request	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W

The SNZEDCR1 register controls the condition for switching from Snooze mode to Software Standby mode. In order to use a trigger shown in Table 10.7 as a condition to switch from Snooze mode to Software Standby mode, the corresponding bit in the SNZEDCR1 register must be set to 1.

The event that is used to return to Normal mode from Snooze mode as shown in Table 10.3 must not be enabled in the SNZEDCR1 register.

TMLOIED bit (TML32 Interrupt Snooze End Enable)

The TMLOIED bit specifies whether to enable a transition from Snooze mode to Software Standby mode on TML32_OUTI interrupt. For details on the trigger conditions, see section 19, 32-bit Interval Timer (TML32) .

REMCNCRED bit (REMC No-Interrupt Completion Snooze End Enable)

The REMCNCRED bit specifies whether to enable transition from Snooze mode to Software Standby mode when the data received is completed in Snooze mode and any interrupt for REMC has not occurred. For details on the trigger conditions, see section 26, Remote Control Signal Receiver (REMC).

SAU0NCRED bit (SAU0 No-Interrupt Error Completion Snooze End Enable)

The SAU0NCRED bit specifies whether to enable transition from Snooze mode to Software Standby mode when the data received with an error is completed in Snooze mode and any interrupt for SAU0 has not occurred. For details on the trigger conditions, see section 23, Serial Array Unit (SAU).

SAU1NCRED bit (SAU1 No-Interrupt Error Completion Snooze End Enable)

The SAU1NCRED bit specifies whether to enable transition from Snooze mode to Software Standby mode when the data received with an error is completed in Snooze mode and any interrupt for SAU1 has not occurred. For details on the trigger conditions, see section 23, Serial Array Unit (SAU).

10.2.11 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:	—	—	—	—	—	SNZREQEN26	—	SNZREQEN24	—	—	—	—	—	—	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:	—	—	—	—	—	—	—	—	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
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
16:8	—	These bits are read as 0. The write value should be 0.	R/W
17	SNZREQEN17	Enable KEY_INTKR snooze request 0: Disable the snooze request 1: Enable the snooze request	R/W
23:18	—	These bits are read as 0. The write value should be 0.	R/W
24	SNZREQEN24	Enable RTC alarm snooze request 0: Disable the snooze request 1: Enable the snooze request	R/W
25	—	This bit is read as 0. The write value should be 0.	R/W
26	SNZREQEN26	Enable TML32 snooze request 0: Disable the snooze request 1: Enable the snooze request	R/W
31:27	—	These bits are read as 0. The write value should be 0.	R/W

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 WUPEN 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 WUPEN 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).

10.2.12 PSMCR : Power Save Memory Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x09F

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

Bit	Symbol	Function	R/W
1:0	PSMC[1:0]	Power Save Memory Control 0 0: All SRAMs are on in Software Standby mode 0 1: 8 KB SRAM (0x2000_0000 to 0x2000_0FFF and 0x2000_4000 to 0x2000_4FFF) is on, in Software Standby mode 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

PSMC[1:0] bits (Power Save Memory Control)

The PSMC[1:0] bits select the SRAM retention area in Software Standby mode. Setting these bits to 01b (8 KB SRAM in Software Standby mode) reduces the supply current. Set the PSMCR register before executing a WFI instruction.

This register is protected by the PRCR.PRC1 bit.

10.2.13 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	—	—	—	—	—	—	—
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	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.

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 changed. For details, see section 2.4.5. Restrictions on Connecting an OCD Emulator.

10.2.14 LSMRWDIS : Low Speed Module R/W Disable Control Register

Base address: MSTP = 0x4004_7000

Offset address: 0x00C

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PRKEY[7:0]								WREN	—	—	—	—	IWDTDIS	WDTDIS	RTCRWDIS
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	RTCRWDIS	RTC Register R/W Enable Control Stop the RTC register R/W clock (only valid when LPOPT.LPOPTEN = 1) 0: RTC register R/W clock always on 1: RTC register R/W clock stops	R/W
1	WDTDIS	WDT Operate Clock Control Stop the WDT counter clock and register R/W clock (valid only when LPOPT.LPOPTEN = 1) 0: WDT operates as normal 1: Stop the WDT clock and register R/W clock	R/W
2	IWDTDIS	IWDT Register Clock Control Stop the IWDT register R/W clock (valid only when LPOPT.LPOPTEN = 1) 0: IWDT operates as normal 1: Stop the IWDT register R/W clock	R/W
6:3	—	These bits are read as 0. The write value should be 0.	R/W
7	WREN	Write Enable for bits [2:0] 0: Write protect for bits [2:0] 1: Write enable for bits [2:0]	R/W
15:8	PRKEY[7:0]	LSMRWDIS Key Code These bits control the write access to the LSMRWDIS register. To modify the LSMRWDIS register, write 0xA5 to the upper 8 bits and the target value to the lower 8 bits as a 16-bit unit.	W

RTCRWDIS bit (RTC Register R/W Enable Control)

[Setting conditions]

- This bit can only be modified when WREN is set to 1.
- When LPOPT.LPOPTEN = 1 and this bit is set to 1, this bit stops the RTC register R/W clock.

WDTDIS bit (WDT Operate Clock Control)

[Setting conditions]

- This bit can only be modified when WREN is set to 1.
- When LPOPT.LPOPTEN = 1 and this bit is set to 1, this bit stops the WDT operate clock.
- Do not set this bit to 1 when WDT is in auto start mode (OFS0.WDTSTRT = 0).
- Do not set this bit to 1 when WDT is operating.
- Set this bit to 1 to disable register start mode for WDT.

IWDTDIS bit (IWDT Register Clock Control)

[Setting conditions]

- This bit can only be modified when WREN is set to 1.
- When LPOPT.LPOPTEN = 1 and this bit is set to 1, this bit stops the IWDT register R/W clock.

- Do not set this bit to 1 when IWDT is in auto start mode (OFS0.IWDTSTRT = 0).
- Do not set this bit to 1 when IWDT is operating.

10.2.15 LPOPT : Lower Power Operation Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x04C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	LPOPTEN	—	—	—	BPFCLKDIS	—	—	—
Value after reset:	0	1	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
3	BPFCLKDIS	BPF Clock Disable Control Stop the Flash register R/W clock (valid only when LPOPT.LPOPTEN = 1) 0: Flash register R/W clock operates as normal 1: Flash register R/W clock stops	R/W
5:4	—	These bits are read as 0. The write value should be 0.	R/W
6	—	This bit is read as 1. The write value should be 1.	R/W
7	LPOPTEN	Lower Power Operation Enable 0: All lower power counter measure disable 1: All lower power counter measure enable	R/W

The LPOPT register is protected by the PRCR.PRC0 bit.

BPFCLKDIS bit (BPF Clock Disable Control)

[Setting condition]

- Do not set this bit to 1 when in OCD mode or UART (SAU) boot mode.
- Do not set this bit to 1 when operating code flash or data flash through the Flash register.
- Do not set this bit to 1 when operating data flash.
- Do not set this bit to 1 when system transfers power control mode (for example, transfer High-speed mode to Middle-speed mode or transfer High-speed mode to Low-speed mode).
- When LPOPT.LPOPTEN = 1 and this bit is set to 1, this bit stops the Flash register R/W clock.

LPOPTEN bit (Lower Power Operation Enable)

[Setting condition]

- Setting this bit to 1 decreases the MCU power consumption but creates limitations in the system.

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 8.2.1. 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 D, i = 31 to 0) in MSTPCRn (n = A to D) 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 DTC 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.

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					
	High-speed on-chip oscillator	Middle-speed on-chip oscillator	Low-speed on-chip oscillator	External clock input	Sub-clock oscillator	IWDT-dedicated on-chip oscillator
High-speed	Available	Available	Available	Available	Available	Available
Middle-speed	Available	Available	Available	Available	Available	Available
Low-speed	Available	Available	Available	Available	Available	Available
Subosc-speed	N/A	N/A	Available	N/A	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)

Example 2: From High-speed mode to Subosc-speed mode:

(Operation begins in High-speed mode)

1. Switch the clock source to sub-clock oscillator. Turn off HOCO, MOCO, LOCO and EXTAL.
2. Confirm that all clock sources other than the sub-clock oscillator are stopped.
3. Confirm that the SOPCCR.SOPCMTSF flag is 0 (indicates transition completed).
4. Set the SOPCCR.SOPCM bit to 1 (Subosc-speed mode).
5. Confirm that the SOPCCR.SOPCMTSF flag is 0 (indicates transition completed).

(Operation is now in Subosc-speed mode)

(2) Switching from a lower power mode to a higher power mode

Example 1: From Subosc-speed mode to High-speed mode:

(Operation begins in Subosc-speed mode)

1. Confirm that the SOPCCR.SOPCMTSF flag is 0 (indicates transition completed).
2. Set the SOPCCR.SOPCM bit to 0 (High-speed mode).
3. Confirm that the SOPCCR.SOPCMTSF flag is 0 (indicates transition completed).
4. Turn on the 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)

Example 2: 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)

10.5.2 Operating Range

Figure 10.2 to Figure 10.4 show the ICLK operating voltages and frequencies. However, peripheral module clocked by PCLKB is not equal to ICLK.

High-speed mode

The maximum operating frequency during a flash read is 48 MHz for ICLK. The operating voltage range during a flash read is 1.8 to 5.5 V.

During flash programming/erasure, the operating frequency range is 1 to 48 MHz and the operating voltage range is 1.8 to 5.5 V.

Figure 10.2 shows the operating voltages and frequencies in High-speed mode.

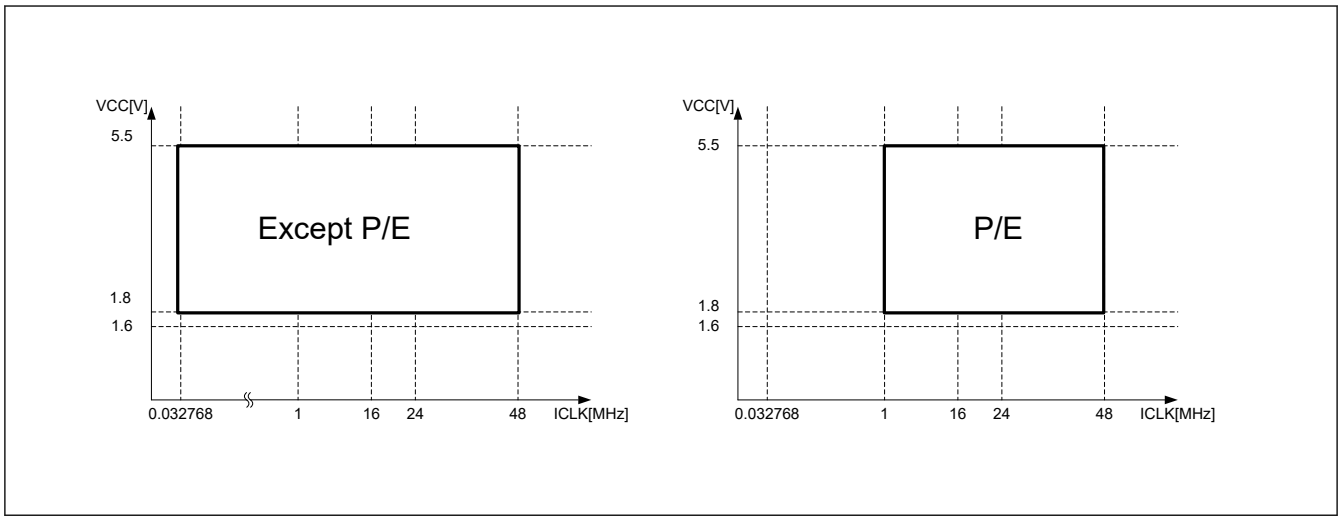


Figure 10.2 Operating voltages and frequencies in High-speed mode

Middle-speed mode

The power consumption of this mode is lower than that of High-speed mode under the same conditions.

The maximum operating frequency during a flash read is 24 MHz for ICLK. The operating voltage range during a flash read is 1.6 to 5.5 V. However, the maximum operating frequency during a flash read is 4 MHz when the operating voltage is 1.6 to 1.8 V.

During flash programming/erasure, the operating frequency range is 1 to 24 MHz and the operating voltage range is 1.6 to 5.5 V. However, the maximum operating frequency during flash programming/erasure is 4 MHz when the operating voltage is 1.6 to 1.8 V.

Figure 10.3 shows the operating voltages and frequencies in Middle-speed mode.

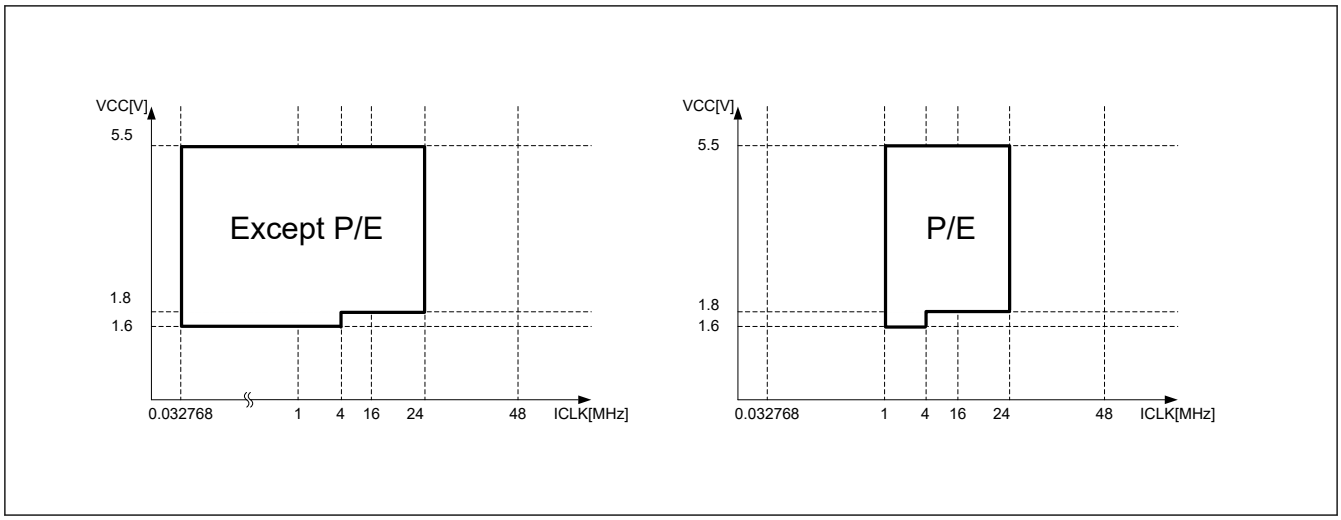


Figure 10.3 Operating voltages and frequencies in Middle-speed mode

Low-speed mode

The maximum operating frequency during a flash read is 1 MHz for ICLK. The operating voltage range during a flash read is 1.6 to 5.5 V.

During flash programming/erasure, the operating frequency is 1 MHz and the operating range is 1.6 to 5.5 V.

Figure 10.4 shows the operating voltages and frequencies in Low-speed mode.

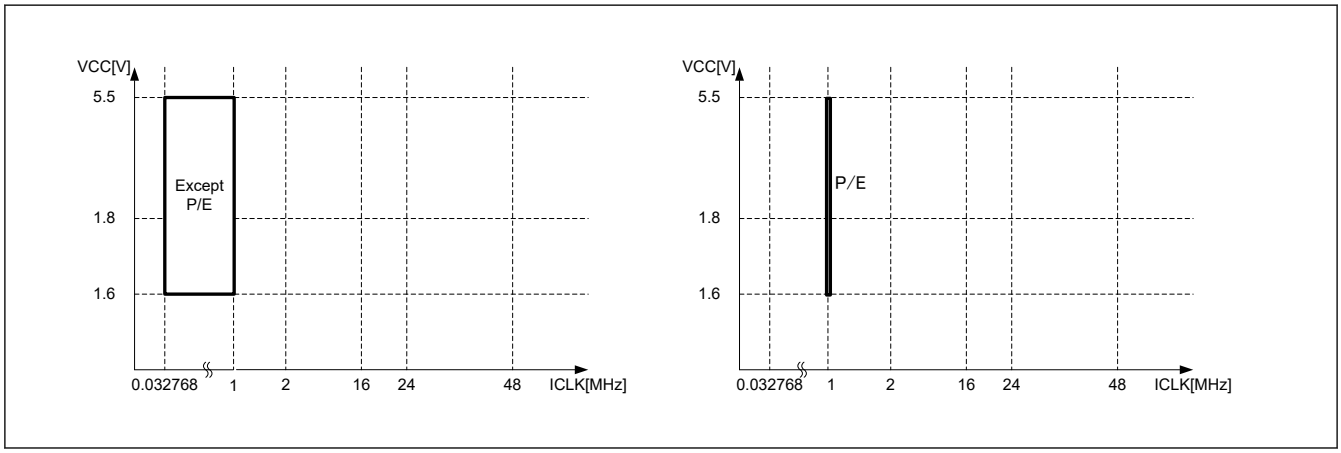


Figure 10.4 Operating voltages and frequencies in Low-speed mode

Subosc-speed mode

The maximum operating frequency during a flash read is 37.6832 kHz for ICLK. The operating voltage range during a flash read is 1.6 to 5.5 V. P/E operations for flash memory are prohibited.

Using the oscillators other than the sub-clock oscillator or low-speed on-chip oscillator is prohibited. Setting the SCKDIVCR register to a value other than 0x00000000 is also prohibited.

Figure 10.5 shows the operating voltages and frequencies in Subosc-speed mode.

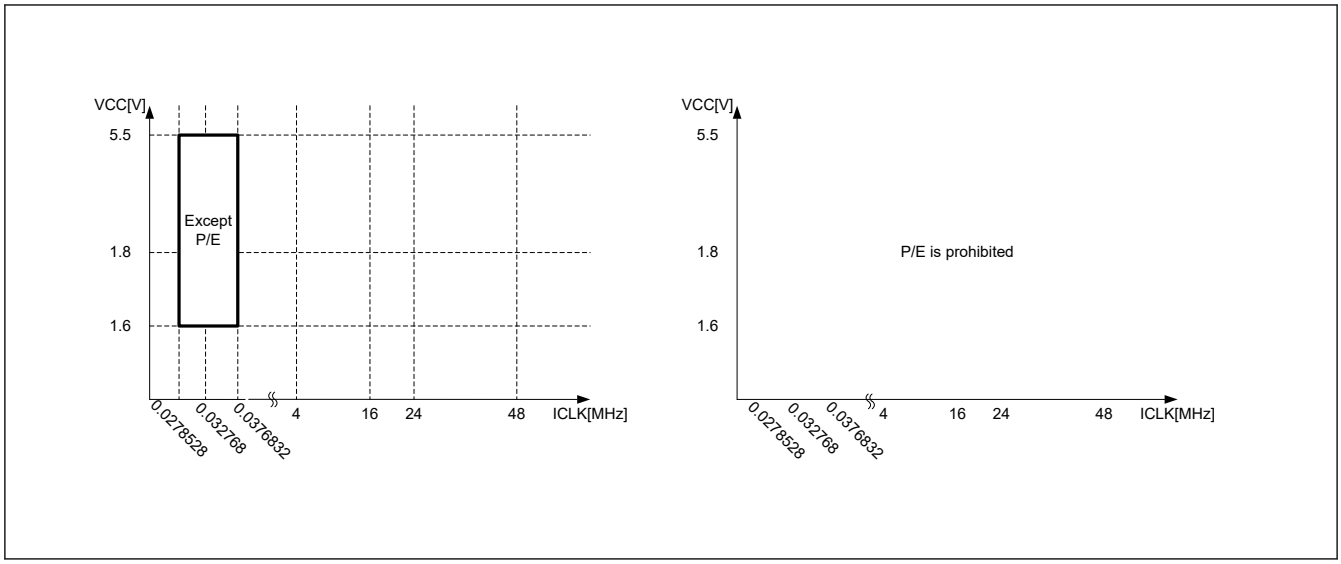


Figure 10.5 Operating voltages and frequencies in Subosc-speed mode

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 error reset
- A reset caused by an IWDT or a WDT underflow
- A debug reset

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 37, 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 condition:
   - 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 Transition to Software Standby Mode

When a WFI instruction is executed while SBYCR.SSBY bit is 1, the MCU enters Software Standby mode. In this mode, the CPU, most of the on-chip peripheral functions and 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 a significant reduction in power consumption because most of the oscillators stop in this mode. section 10.1. Overview shows the status of each on-chip peripheral functions and oscillators. Available resets or interrupts in Software Standby mode cause the MCU to cancel Software Standby mode. See section 10.1. Overview for available interrupt sources and section 12.2.7. WUPEN0 : Wake Up Interrupt Enable Register 0 for information on how to wake up the MCU from Software Standby mode. If using an interrupt to cancel Software Standby 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 Software Standby mode while the IWDT is in auto start mode and the OFS0.IWDTSTPCTL bit is 1 (IWDT stops in Sleep mode, Software Standby mode, and Snooze mode). Counting by 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 mode, Software Standby mode, and Snooze mode).

WDT stops counting when the MCU enters Software Standby mode.

Do not enter Software Standby mode while the flash memory performs a programming or erasing procedure. To enter Software Standby mode, execute a WFI instruction after the programming or erasing procedure completes.

10.7.2 Canceling Software Standby Mode

Software Standby mode is canceled by:

- An available interrupt shown in section 10.1. Overview
- 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 12.2.7. 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 section 10.1. Overview) 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.
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 37, 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 10.6 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), 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 10.6 is specified in section 37, Electrical Characteristics.

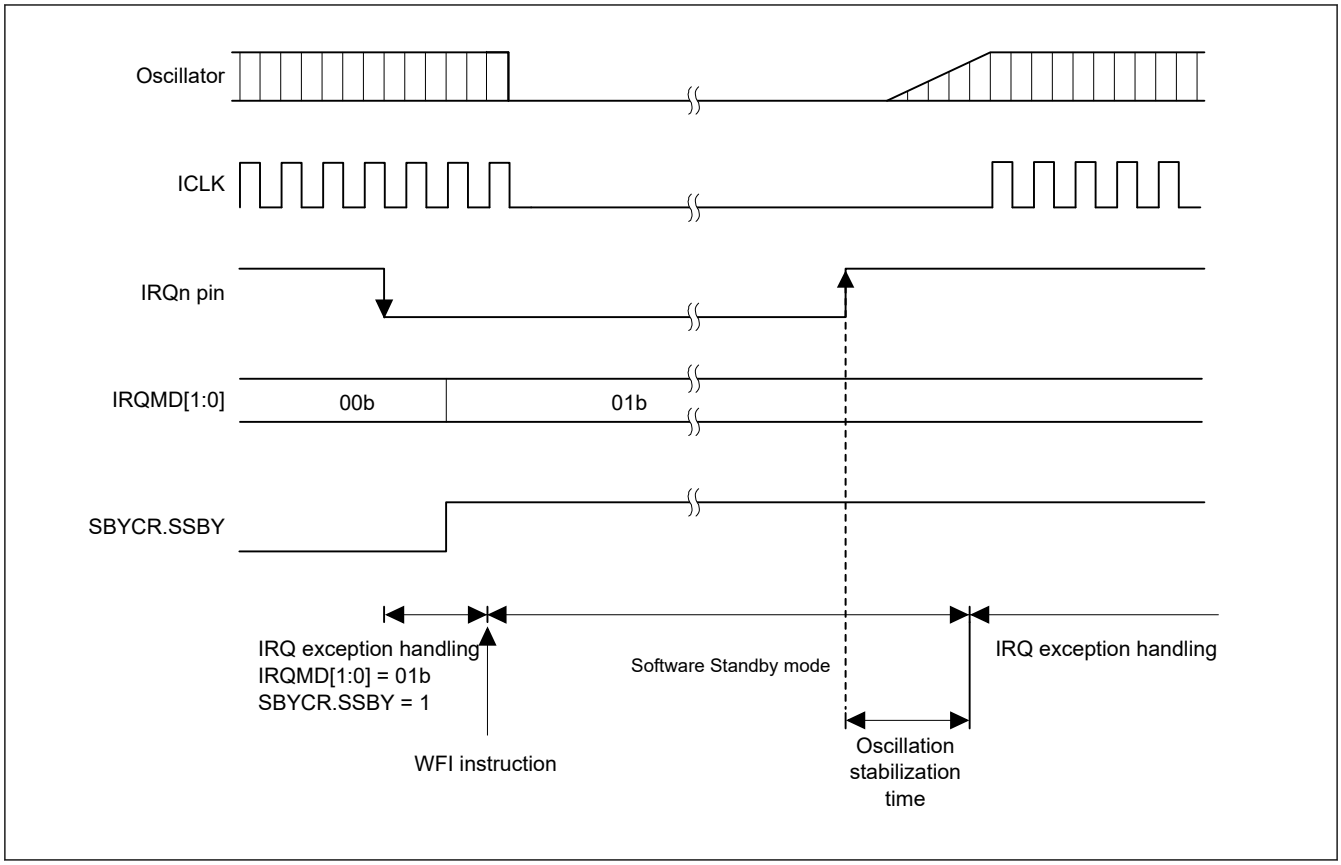


Figure 10.6 Example of Software Standby mode application

10.8 Snooze Mode

10.8.1 Transition to Snooze Mode

Figure 10.7 shows snooze mode entry configuration. When the snooze control circuit receives a snooze request in Software Standby mode, the MCU transfers to Snooze mode. In this mode, some peripheral modules operate without waking up the CPU. Table 10.2 shows the peripheral modules that can operate in Snooze mode. Also, DTC operation in Snooze mode can be selected by setting the SNZCR.SNZDTCEN bit.

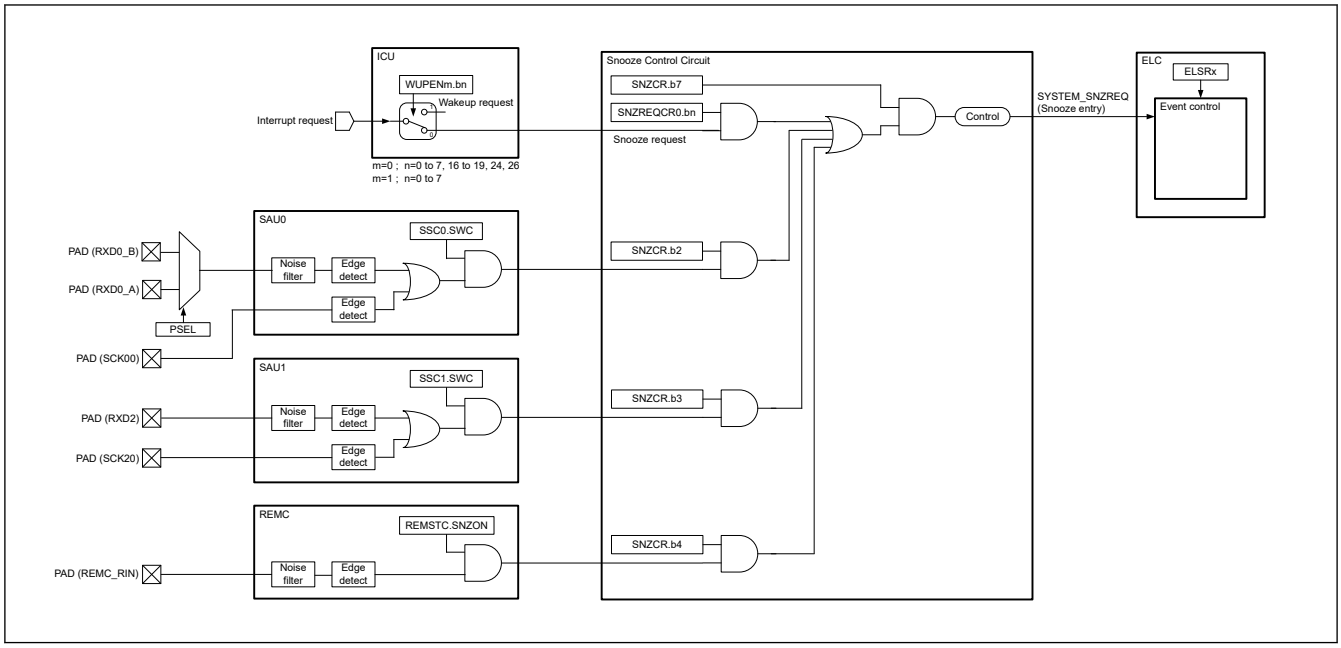


Figure 10.7 Snooze mode entry configuration

Table 10.6 shows the snooze requests to switch the MCU from Software Standby mode to Snooze mode. To use the listed snooze requests as a trigger to switch to Snooze mode, you must set the associated SNZREQENn bit of the SNZREQCR0 register or RXD0REQEN, RXD2REQEN, or REMC0REQEN bit of the SNZCR register before entering Software Standby mode.

Note: Do not enable multiple snooze requests at the same time.

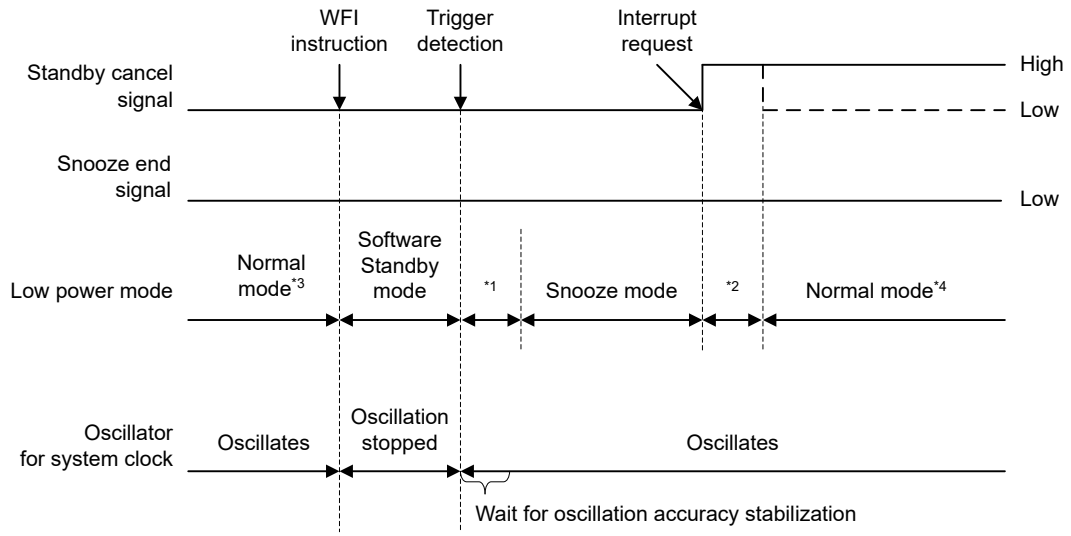
Table 10.6 Available snooze requests to switch to Snooze mode

Snooze request	Control Register	
	Register	Bit
PORT_IRQn (n = 0 to 7)	SNZREQCR0	SNZREQENn (n = 0 to 7)
KEY_INTKR	SNZREQCR0	SNZREQEN17
RTC_ALM_OR_PRD	SNZREQCR0	SNZREQEN24
TML32_OUTI	SNZREQCR0	SNZREQEN26
RXD0 falling/rising edge SCK00 falling/rising edge	SNZCR	RXD0REQEN
RXD2 falling/rising edge SCK20 falling/rising edge	SNZCR	RXD2REQEN
RIN0 falling/rising edge	SNZCR	REMC0REQEN

Clear the DTCST.DTCST bit to 0 before executing a WFI instruction except when using DTC in Snooze mode. If 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 CLIC 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 10.8 Canceling of Snooze mode when an interrupt request signal is generated

10.8.3 Returning from Snooze Mode to Software Standby Mode

Table 10.7 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 10.8 shows the snooze end conditions that consist of the snooze end requests and the conditions of the peripheral modules. The ADC12, REMC, SAU, and DTC modules can keep the MCU in Snooze mode until they complete the operation.

Figure 10.9 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 SNZEDCRn register. A snooze request is cleared automatically after returning to Software Standby mode.

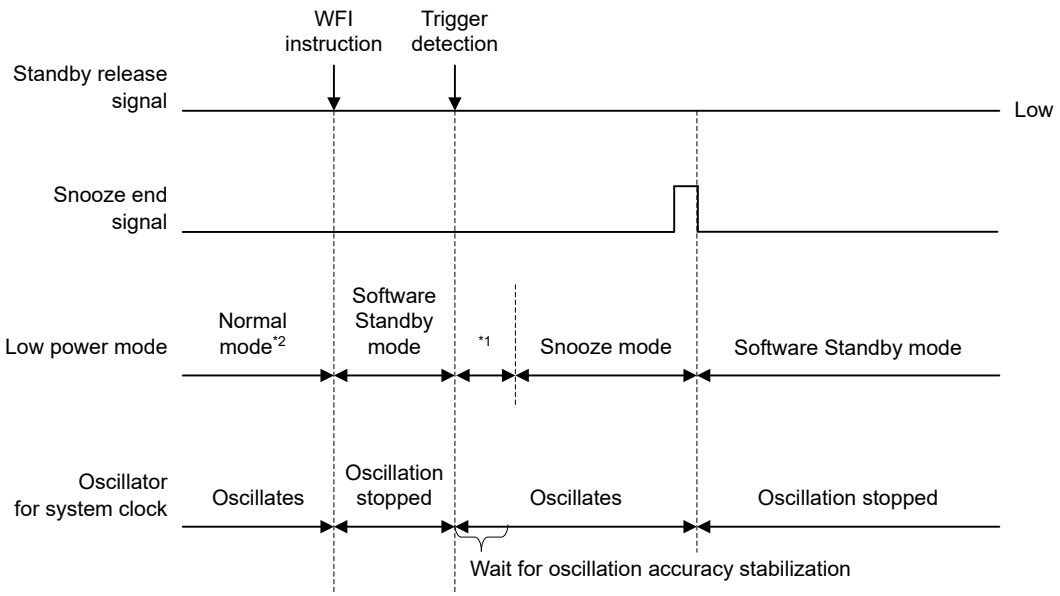
Table 10.7 Available snooze end requests (triggers to return to Software Standby mode)

Peripheral Module	Snooze end request	Enable/Disable Control	
		Register	Symbol
DTC	Not Last DTC Transmission Completion (DTC_TRANSFER)	SNZEDCR0	DTCNZRED
DTC	Last DTC transmission completion (DTC_COMPLETE)	SNZEDCR0	DTCZRED
ADC12	ADC Compare Mismatch Completion	SNZEDCR0	ADNCRED
TML32	TML32 Interrupt (TML32_OUTI)	SNZEDCR1	TMLOIED
REMC	REMC No-interrupt Completion	SNZEDCR1	REMCNCRED
SAU0	SAU0 No-interrupt Error Completion	SNZEDCR1	SAU0NCRED
SAU1	SAU1 No-interrupt Error Completion	SNZEDCR1	SAU1NCRED

Table 10.8 Snooze end conditions

Operating module when a snooze end request occurs	Snooze end request
DTC	The MCU transfers to the Software Standby mode after all of the modules listed in this table complete operation.
ADC12	
SAU (UART0, UART2, SPI00, SPI20)	
REMC	
Other than specified	The MCU transfers to the Software Standby mode immediately after a snooze end request is generated.

Note: If the DTC is used to activate the ADC12, REMC, or SAU, the MCU transitions to Software Standby mode after a snooze end request is generated.

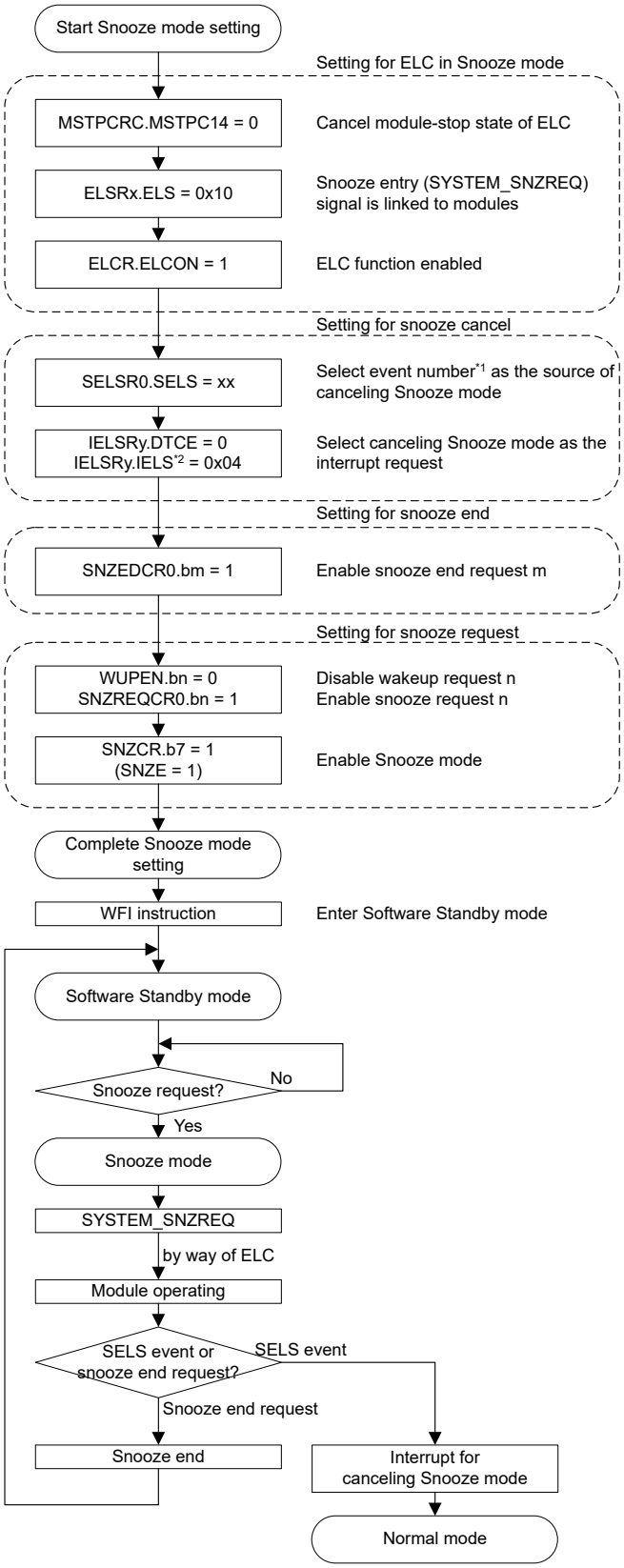


Note 1. Transition time from Software Standby mode to Snooze mode.
Note 2. Enable Snooze mode (SNZCR.SNZE = 1) immediately before switching to Software Standby mode.

Figure 10.9 Canceling of Snooze mode when an interrupt request signal is not generated

10.8.4 Snooze Operation Example

Figure 10.10 shows an example setting for using ELC in Snooze mode.



Note 1. See Table 12.4
Note 2. y = 7, 15, 23, 31

Figure 10.10 Setting example of using ELC in Snooze mode

(1) Asynchronous data transfer by SAU

The MCU can transmit and receive data in SAU asynchronous mode without CPU intervention. When using the SAU in Snooze mode, use one of the following operating modes:

- High-speed mode
- Middle-speed mode
- Low-speed mode.

Do not use Subosc-speed mode.

10.9 Usage Notes

10.9.1 Register Access

(1) Do not write to registers listed in this section in any of the following conditions:

[Registers]

- All registers with a peripheral name of SYSTEM.

[Conditions]

- OPCCR.OPCMTSF = 1 or SOPCCR.SOPCMTSF = 1 (during transition of the operating power control mode)
- During the time period from executing a WFI instruction to returning to Normal mode
- Flash P/E mode, data flash P/E mode

(2) Valid setting for the clock-related registers

Table 10.9 and Table 10.10 show the valid settings for the clock-related registers in each operating power control mode. Do not write any value other than the valid setting, otherwise it is ignored. Additionally, each register has some prohibited settings under certain conditions other than those related to the operating power control modes. See section 8, Clock Generation Circuit for these other conditions of each register.

Table 10.9 Valid settings for the clock-related registers (1)

Mode	Valid settings						
	SCKSCR.CKSEL[2:0] CKOCR.CKOSEL[2:0]	SCKDIVCR.ICK[2:0]	HOCOCR.HCSTP	MOCOCR.MCSTP	LOCOCR.LCSTP	MOSCCR.MOSTP	SOSCCR.SOSTP
High-speed Middle-speed Low-speed	000b (HOCO) 001b (MOCO) 010b (LOCO) 011b (EXTAL) 100b (SOSC)	000b (1/1) 001b (1/2) 010b (1/4) 011b (1/8) 100b (1/16) 101b (1/32) 110b (1/64)	0 (operating) 1 (stopped)	0 (operating) 1 (stopped)	0 (operating) 1 (stopped)	0 (operating) 1 (stopped)	0 (operating) 1 (stopped)
Subosc-speed	010b (LOCO) 100b (SOSC)	000b (1/1)	1 (stopped)	1 (stopped)	0 (operating) 1 (stopped)	1 (stopped)	0 (operating) 1 (stopped)

Table 10.10 Valid settings for the clock-related registers (2) (1 of 2)

Operating oscillator	Valid settings	
	SOPCCR.SOPCM	OPCCR.OPCM[1:0]
High-speed on-chip oscillator	0	00b, 01b, 11b
Middle-speed on-chip oscillator		
External clock input		

Table 10.10 Valid settings for the clock-related registers (2) (2 of 2)

Operating oscillator	Valid settings	
	SOPCCR.SOPCM	OPCCR.OPCM[1:0]
Low-speed on-chip oscillator	0, 1	00b, 01b, 11b
Sub-clock oscillator		
IWDT-dedicated on-chip oscillator		

(3) Do not write to registers listed in this section for the following condition:

[Registers]

- SCKSCR, OPCCR.

[Condition]

- SOPCCR.SOPCM = 1 (Subosc-speed mode).

(4) Do not write to registers listed in this section by DTC:

[Registers]

- MSTPCRA, MSTPCRB, MSTPCRC, MSTPCRD.

(5) Do not write to registers listed in this section in Snooze mode. They must be set before entering Software Standby mode:

[Registers]

- SNZCR, SNZEDCR0, SNZEDCR1, SNZREQCR0.

(6) Write access to registers listed in this section is invalid when PRCR.PRC1 bit is 0:

[Registers]

- SBYCR, SNZCR, SNZEDCR0, SNZEDCR1, SNZREQCR0, PSMCR, OPCCR, SOPCCR.

10.9.2 I/O Port Pin States

The I/O port pin states in Software Standby mode 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.

10.9.3 Module-Stop State of DTC

Before writing 1 to MSTPCRA.MSTPA22, clear the DTCST.DTCST bit of the DTC to 0. For details, see section 14, Data Transfer Controller (DTC).

10.9.4 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 startup source cannot be cleared. Always disable the associated interrupts before setting the module-stop bits.

10.9.5 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.

10.9.6 Writing to the WDT/IWDT Registers by DTC in Sleep Mode or Snooze Mode

Do not write to the WDT or IWDT registers by the DTC while WDT or IWDT is stopped after entering Sleep mode or Snooze mode.

10.9.7 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.

10.9.8 Snooze Mode Entry by Edge Detection of REMC (RIN0) and SAU (RXD0, RXD2, SCK00, SCK20) Signal

When using SAU (UART, SPI) and REMC in Snooze mode, set RXD0REQEN, RXD2REQEN, and REMC0REQEN bits of SNZCR register to 1 to detect the edge of the signal input from the related pins. Each module changes this MCU from Software Standby mode to Snooze mode. When the operation of the module has completed, it is possible to transition from Snooze mode to Software Standby mode or Normal mode.

10.9.9 Using REMC/SAU (UART, SPI) in Snooze Mode

When using SAU (UART, SPI) or REMC in Snooze mode, make sure that the Snooze request (edge detection of related terminal signal) does not conflict with the wakeup request set in WUPEN register. If there is a conflict, the operation of each module is not guaranteed.

10.9.10 Conditions of A/D Conversion Start in Snooze Mode

ADC12 can only be triggered by the ELC in Snooze mode. Do not use software trigger or hardware trigger (TAU0_ENDI1, RTC_ALM_OR_PRD, TML32_OUTI). When ADC12 is activated by ELC event signal in Software Standby mode, ADC12 should be in the Hardware trigger wait mode before entering the Software Standby mode. In Snooze mode, the association between the frequencies of the system clock (ICLK) and the ADC12 clock (PCLKB) should be ICLK:PCLKB = N:1 (N ≦ 8).

10.9.11 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)
- A/D conversion end (ADC_ENDI)
- Data operation circuit interrupt (DOC_DOPCI).

10.9.12 Module-Stop Function for ADC12

When entering Software Standby mode, it is recommended that you set the ADC12 module-stop state to reduce power consumption. In this case, the ADC12 can be available in Snooze mode by releasing the ADC12 module-stop using the DTC. Similarly, set the module-stop state using the DTC before returning to Software Standby mode from Snooze mode.

10.9.13 Module-Stop Function for an Unused Circuit

A circuit that is not used in user mode might not be reset, and might operate in an unstable state because the clocks are not supplied during an MCU reset. In this case, when the MCU transitions to Low-speed mode or Software Standby mode, the supply current can be increased to a value greater than that stated in this User's Manual. Therefore, initialize the unused circuit as shown in Figure 10.11.

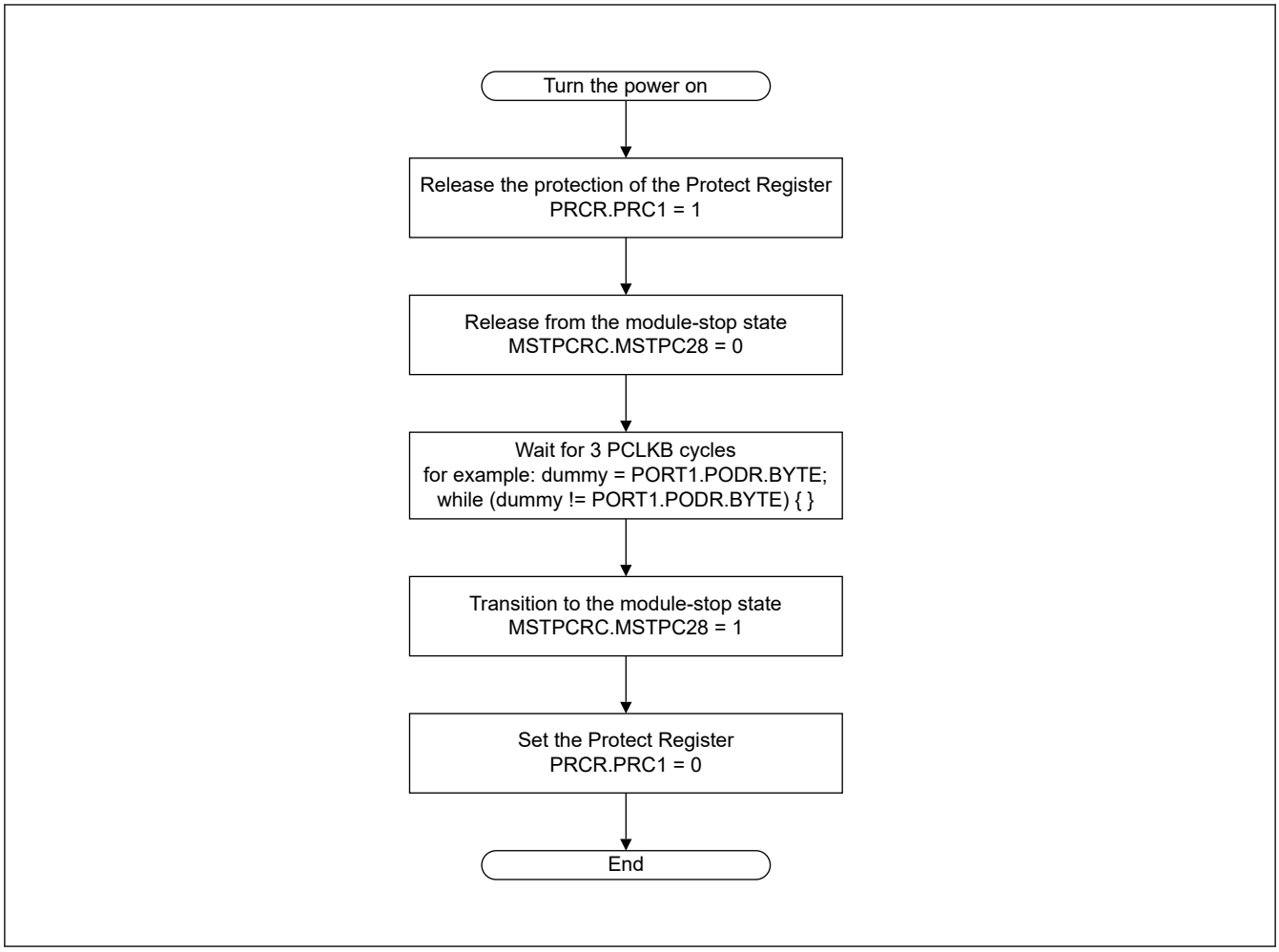


Figure 10.11 Example of initial setting flow for an unused circuit

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, HOCOCR, HOCOCR2, MOCOCR, CKOCR, HOCOUTCR, LOCOCR, LPOPT, OSMCR, MOSCCR, SOSCCR, SOMCR, SOMRG, MEMWAIT, LOCOUTCR, MOCOUTCR
PRC1	• Registers related to the low power modes: SBYCR, OPCCR, SNZCR, SNZEDCR0, SNZEDCR1, SNZREQCR0, SOPCCR, SYOCDCR, PSMCR
PRC3	• Registers related to the LVD: LVD1CR1, LVD1SR, LVD2CR1, LVD2SR, LVCMPCR, LVDLVLR, LVD1CR0, LVD2CR0

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]								—	—	—	—	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
7:4	—	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)

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.

12. Interrupt Controller Unit (ICU)

12.1 Overview

The Interrupt Controller Unit (ICU) controls which event signals are linked to the Core-Local Interrupt Controller (CLIC), 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: 52
	External pin interrupts	• Interrupt detection on low level*1, falling edge, rising edge, rising and falling edges. One of these detection methods can be set for each source. • Digital filter function supported • 8 sources, with interrupts from IRQi (i = 0 to 7) pins.
	Interrupt requests to CPU (CLIC)	• 32 interrupt requests are output to CLIC. • Maskable interrupt sources are classified into 8 groups, and one source can be selected individually from 31 sources that are classified into groups.
	DTC control	• The DTC can be activated using interrupt sources*2. • The method for selecting an interrupt source is the same as that of the interrupt request to CLIC.
Non-maskable interrupts*3	NMI pin interrupt	• Interrupt from the NMI pin • Interrupt detection on falling edge or rising edge • Digital filter function supported
	WDT underflow/refresh error*4	Interrupt on an underflow of the down-counter or occurrence of a refresh error
	IWDT underflow/refresh error*4	Interrupt on an underflow of the down-counter or occurrence of a refresh error
	Low voltage detection 1*4	Voltage monitor 1 interrupt of the voltage monitor 1 circuit (LVD_LVD1)
	Low voltage detection 2*4	Voltage monitor 2 interrupt of the voltage monitor 2 circuit (LVD_LVD2)
	RPEST	Interrupt on SRAM parity error
	RECCST	Interrupt on SRAM ECC error
	BUSST	Interrupt on BUS 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 interrupts which can be selected in the WUPEN0/1 register • Snooze mode: Return is initiated by interrupts which can be selected in the SELSR0 and WUPEN0/1 registers. See section 12.2.7. WUPEN0 : Wake Up Interrupt Enable Register 0, section 12.2.8. WUPEN1 : Wake Up Interrupt Enable Register 1, and section 12.2.10. SELSR0 : SYS Event Link Setting Register.

Note 1. Interrupt detection continues to flag an interrupt request if it is not cleared after a detection.
Note 2. For the DTC activation sources, see Table 12.4.
Note 3. Non-maskable interrupts can be enabled only once after a reset release.
Note 4. 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.

Figure 12.1 shows the ICU block diagram.

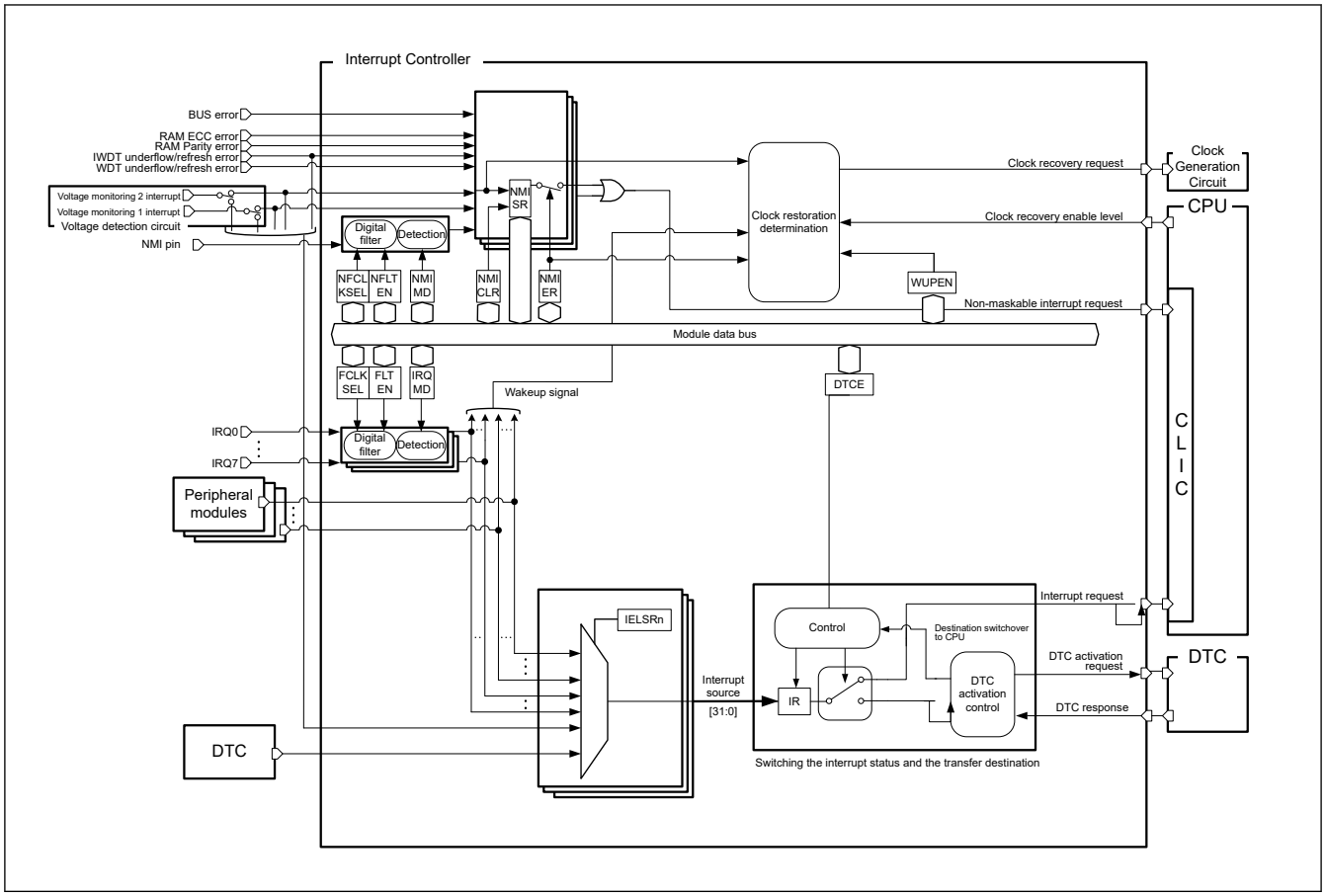


Figure 12.1 ICU block diagram

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 7)	Input	External interrupt request pins

12.2 Register Descriptions

This section does not describe the CLIC internal registers. For information about these registers, see reference 1 in section 2.8.6. References.

12.2.1 IRQCRi : IRQ Control Register i (i = 0 to 7)

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

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 31).
  The register value should be changed only when the value of the target IELSRn register is 0x0000.
- For a wakeup enable signal:
  Change the IRQCRi register setting before setting the target WUPEN0.IRQWUPEN[n] (n = 0 to 7). 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.

12.2.2 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:	—	—	—	BUSST	—	—	RECCST	RPEST	NMIST	—	—	—	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
6:4	—	These bits are read as 0.	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
11:10	—	These bits are read as 0.	R
12	BUSST	BUS Error Interrupt Status Flag 0: Interrupt not requested 1: Interrupt requested	R
15:13	—	These bits are read as 0.	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.

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.

BUSST flag (BUS Error Interrupt Status Flag)

The BUSST flag indicates a bus master error interrupt request.

[Setting condition]

When an interrupt is generated in response to a bus error.

[Clearing condition]

When 1 is written to the NMICLR.BUSCLR bit.

12.2.3 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:	—	—	—	BUSEN	—	—	RECCEN	RPEEN	NMIEN	—	—	—	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
6:4	—	These bits are read as 0. The write value should be 0.	R/W
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
11:10	—	These bits are read as 0. The write value should be 0.	R/W
12	BUSEN	Bus Error Interrupt Enable 0: Disabled 1: Enabled	R/W*1
15:13	—	These bits are read as 0. The write value should be 0.	R/W

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 a maskable interrupt.

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.

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.

BUSEN bit (Bus Error Interrupt Enable)

The BUSEN bit enables bus error interrupt as an NMI trigger.

12.2.4 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:	—	—	—	BUSCLR	—	—	RECCCLR	RPECLR	NMICLR	—	—	—	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
6:4	—	These bits are read as 0. The write value should be 0.	R/W
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
9	RECCCLR	SRAM ECC Error Interrupt Status Flag Clear 0: No effect 1: Clear the NMISR.RECCST flag	R/W*1
11:10	—	These bits are read as 0. The write value should be 0.	R/W
12	BUSCLR	Bus Error Interrupt Status Flag Clear 0: No effect 1: Clear the NMISR.BUSST flag	R/W*1
15:13	—	These bits are read as 0. The write value should be 0.	R/W

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.

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.

BUSCLR bit (Bus Error Interrupt Status Flag Clear)

Writing 1 to the BUSCLR bit clears the NMISR.BUSST flag. This bit is read as 0.

12.2.5 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

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.

12.2.6 IELSRn : ICU Event Link Setting Register n (n = 0 to 31)

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[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	IELS[4:0]	ICU Event Link Select 0x00: Disable interrupts to the associated CLIC or DTC module Others: Event signal number to be linked. For details, see section 12.3.3. ICU and DTC Event Number.	R/W
15:5	—	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 1. Writing 1 to the IR flag is prohibited.

The IELSRn register selects the IRQi source used by the CLIC. IELSRn corresponds to the CLIC IRQ input source number, where n = 0 to 31.

IELS[4:0] bits (ICU Event Link Select)

The IELS[4:0] bits link an event signal to the associated CLIC 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[4:0].

[Setting condition]

When an interrupt request is received from the associated peripheral module or IRQi pin.

[Clearing condition]

- When 0 is written to the IR flag. DTCE must be set to 0 before writing 0 to the IR flag.

To clear the IR flag:

1. Negate the interrupt input 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.

12.2.7 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:	—	—	—	—	—	TML32 WUPEN	—	RTCW UPEN	—	—	—	—	LVD2 WUPEN	LVD1 WUPEN	KEYW UPEN	IWDT WUPEN
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[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	IRQWUPEN[7:0]	IRQn (n = 0 to 7) Interrupt Software Standby/Snooze Mode Returns Enable, whereas IRQn corresponds to IRQWUPEN[n] 0: Software Standby/Snooze Mode returns by IRQn interrupt disabled 1: Software Standby/Snooze Mode returns by IRQn interrupt enabled	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W
16	IWDTWUPEN	IWDT Interrupt Software Standby/Snooze Mode Returns Enable 0: Software Standby/Snooze Mode returns by IWDT interrupt disabled 1: Software Standby/Snooze Mode returns by IWDT interrupt enabled	R/W
17	KEYWUPEN	Key Interrupt Software Standby/Snooze Mode Returns Enable 0: Software Standby/Snooze Mode returns by Key interrupt disabled 1: Software Standby/Snooze Mode returns by Key interrupt enabled	R/W
18	LVD1WUPEN	LVD1 Interrupt Software Standby/Snooze Mode Returns Enable 0: Software Standby/Snooze Mode returns by LVD1 interrupt disabled 1: Software Standby/Snooze Mode returns by LVD1 interrupt enabled	R/W
19	LVD2WUPEN	LVD2 Interrupt Software Standby/Snooze Mode Returns Enable 0: Software Standby/Snooze Mode returns by LVD2 interrupt disabled 1: Software Standby/Snooze Mode returns by LVD2 interrupt enabled	R/W
23:20	—	These bits are read as 0. The write value should be 0.	R/W
24	RTCWUPEN	RTC Interrupt Software Standby/Snooze Mode Returns Enable 0: Software Standby/Snooze Mode returns by RTC interrupt disabled 1: Software Standby/Snooze Mode returns by RTC interrupt enabled	R/W
25	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
26	TML32WUPEN	TML32 Interrupt Software Standby/Snooze Mode Returns Enable 0: Software Standby/Snooze Mode returns by TML32 interrupt disabled 1: Software Standby/Snooze Mode returns by TML32 interrupt enabled	R/W
31:27	—	These bits are read as 0. The write value should be 0.	R/W

The bits in this register control whether the associated interrupt can wake up the CPU from Software Standby/Snooze mode.

IRQWUPEN[7:0] bits (IRQn (n = 0 to 7) Interrupt Software Standby/Snooze Mode Returns Enable, whereas IRQn corresponds to IRQWUPEN[n])

The IRQWUPEN[7:0] bits enable the use of IRQn interrupts to cancel Software Standby/Snooze mode.

IWDTWUPEN bit (IWDT Interrupt Software Standby/Snooze Mode Returns Enable)

The IWDTWUPEN bit enables the use of IWDT interrupts to cancel Software Standby/Snooze mode.

KEYWUPEN bit (Key Interrupt Software Standby/Snooze Mode Returns Enable)

The KEYWUPEN bit enables the use of KEY interrupts to cancel Software Standby/Snooze mode.

LVD1WUPEN bit (LVD1 Interrupt Software Standby/Snooze Mode Returns Enable)

The LVD1WUPEN bit enables the use of LVD1 interrupts to cancel Software Standby/Snooze mode.

LVD2WUPEN bit (LVD2 Interrupt Software Standby/Snooze Mode Returns Enable)

The LVD2WUPEN bit enables the use of LVD2 interrupts to cancel Software Standby/Snooze mode.

RTCWUPEN bit (RTC Interrupt Software Standby/Snooze Mode Returns Enable)

The RTCWUPEN bit enables the use of RTC interrupts to cancel Software Standby/Snooze mode.

TML32WUPEN bit (TML32 Interrupt Software Standby/Snooze Mode Returns Enable)

The TML32WUPEN bit enables the use of TML32 interrupts to cancel Software Standby/Snooze mode.

12.2.8 WUPEN1 : Wake Up Interrupt Enable Register 1

Base address: ICU = 0x4000_6000

Offset address: 0x1A4

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:	—	—	—	—	—	—	—	—	COMPDET1WUPEN	COMPDET0WUPEN	UARTARXERI1WUPEN	UARTARXENDI1WUPEN	UARTARXERI0WUPEN	UARTARXENDI0WUPEN	IICA1WUPEN	IICA0WUPEN
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	IICA0WUPEN	IIC0_ENDI/IIC0_WUI Interrupt Software Standby/Snooze Mode Returns Enable 0: Software Standby/Snooze mode returns by IIC0_ENDI/IIC0_WUI interrupt disabled 1: Software Standby/Snooze mode returns by IIC0_ENDI/IIC0_WUI interrupt enabled	R/W

Bit	Symbol	Function	R/W
1	IICA1WUPEN	IIC1_ENDI/IIC1_WUI Interrupt Software Standby/Snooze Mode Returns Enable 0: Software Standby/Snooze mode returns by IIC1_ENDI/IIC1_WUI interrupt disabled 1: Software Standby/Snooze mode returns by IIC1_ENDI/IIC1_WUI interrupt enabled	R/W
2	UARTARXENDI0WUPEN	UARTA_RX_ENDI0 Interrupt Software Standby/Snooze Mode Returns Enable 0: Software standby/Snooze mode returns by UARTA_RX_ENDI0 interrupt disabled 1: Software standby/Snooze mode returns by UARTA_RX_ENDI0 interrupt enabled	R/W
3	UARTARXERI0WUPEN	UARTA_RX_ERI0 Interrupt Software Standby/Snooze Mode Returns Enable 0: Software standby/Snooze mode returns by UARTA_RX_ERI0 interrupt disabled 1: Software standby/Snooze mode returns by UARTA_RX_ERI0 interrupt enabled	R/W
4	UARTARXENDI1WUPEN	UARTA_RX_ENDI1 Interrupt Software Standby/Snooze Mode Returns Enable 0: Software standby/Snooze mode returns by UARTA_RX_ENDI1 interrupt disabled 1: Software standby/Snooze mode returns by UARTA_RX_ENDI1 interrupt enabled	R/W
5	UARTARXERI1WUPEN	UARTA_RX_ERI1 Interrupt Software Standby/Snooze Mode Returns Enable 0: Software standby/Snooze mode returns by UARTA_RX_ERI1 interrupt disabled 1: Software standby/Snooze mode returns by UARTA_RX_ERI1 interrupt enabled	R/W
6	COMPDET0WUPEN	COMP_DET0 Interrupt Software Standby/Snooze Mode Returns Enable 0: Software standby/Snooze mode returns by COMP_DET0 interrupt disabled 1: Software standby/Snooze mode returns by COMP_DET0 interrupt enabled	R/W
7	COMPDET1WUPEN	COMP_DET1 Interrupt Software Standby/Snooze Mode Returns Enable 0: Software standby/Snooze mode returns by COMP_DET1 interrupt disabled 1: Software standby/Snooze mode returns by COMP_DET1 interrupt enabled	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

The bits in this register control whether the associated interrupt can wakeup the CPU from Software Standby/Snooze mode.

IICA0WUPEN bit (IIC0_ENDI/IIC0_WUI Interrupt Software Standby/Snooze Mode Returns Enable)

The IICA0WUPEN bit enable the use of IIC0_ENDI/IIC0_WUI (IICA channel 0 communication complete/address match) interrupts to cancel Software Standby/Snooze mode.

IICA1WUPEN bit (IIC1_ENDI/IIC1_WUI Interrupt Software Standby/Snooze Mode Returns Enable)

The IICA1WUPEN bit enables the use of IIC1_ENDI/IIC1_WUI (IICA channel 1 communication complete/address match) interrupts to cancel Software Standby/Snooze mode.

UARTARXENDI0WUPEN bit (UARTA_RX_ENDI0 Interrupt Software Standby/Snooze Mode Returns Enable)

The UARTARXENDI0WUPEN bit enables the use of UARTA_RX_ENDI0 (UARTA0 receive transfer complete) interrupt to cancel Software Standby/Snooze mode.

UARTARXERI0WUPEN bit (UARTA_RX_ERI0 Interrupt Software Standby/Snooze Mode Returns Enable)

The UARTARXERI0WUPEN bit enables the use of UARTA_RX_ERI0 (UARTA0 receive transfer complete communication error generation) interrupt to cancel Software Standby/Snooze mode.

UARTARXENDI1WUPEN bit (UARTA_RX_ENDI1 Interrupt Software Standby/Snooze Mode Returns Enable)

The UARTARXENDI1WUPEN bit enables the use of UARTA_RX_ENDI1 (UARTA1 receive transfer complete) interrupt to cancel Software Standby/Snooze mode.

UARTARXERI1WUPEN bit (UARTA_RX_ERI1 Interrupt Software Standby/Snooze Mode Returns Enable)

The UARTARXERI1WUPEN bit enables the use of UARTA_RX_ERI1 (UARTA1 receive transfer complete communication error generation) interrupt to cancel Software Standby/Snooze mode.

COMPDET0WUPEN bit (COMP_DET0 Interrupt Software Standby/Snooze Mode Returns Enable)

The COMPDET0WUPEN bit enables the use of COMP_DET0 (Comparator Detection 0) interrupt to cancel Software Standby/Snooze mode.

COMPDET1WUPEN bit (COMP_DET1 Interrupt Software Standby/Snooze Mode Returns Enable)

The COMPDET1WUPEN bit enables the use of COMP_DET1 (Comparator Detection 1) interrupt to cancel Software Standby/Snooze mode.

12.2.9 IELEN : ICU Event Enable Register

Base address: ICU = 0x4000_6000

Offset address: 0x1C0

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

Bit	Symbol	Function	R/W
0	RTCEN	RTC Interrupt Enable (when LPOPT.LPOPTEN bit = 1) 0: Disabled 1: Enabled	R/W
1	IELEN	Parts Asynchronous Interrupts Enable (when LPOPTEN bit = 1) 0: Disabled 1: Enabled	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W

Bits in this register control whether the associated interrupt can be used.

RTCEN bit (RTC Interrupt Enable (when LPOPT.LPOPTEN bit = 1))

The RTCEN bit enables the use of RTC interrupt event.

IELEN bit (Parts Asynchronous Interrupts Enable (when LPOPTEN bit = 1))

The IELEN bit enables the use of parts asynchronous interrupts as follows:

- LVD_LVD1, LVD_LVD2, IWDT_NMIUNDF, PORT_IRQ0 to PORT_IRQ7, KEY_INTKR, TML32_OUTI, IIC0_ENDI0/IIC0_WUI, IIC1_ENDI/IIC1_WUI, UARTA_RX_ENDI0, UARTA_RX_ERI0,UARTA_RX_ENDI1, UARTA_RX_ERI1, COMP_DET0, COMP_DET1

12.2.10 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[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The SELSR0 register selects the events that wake up the CPU from Snooze mode. You can only use the events listed in Table 12.4 checked for "Canceling Snooze" column. Events specified in this register are defined as ICU_SNZCANCEL in Table 12.4.

The SELSR0 event interrupt will wake the CPU up from Snooze mode.

When ICU_SNZCANCEL is selected in IELSRn.IELS[7:0], the SELSR0 event interrupt to CPU is requested.

12.3 Vector Table

The ICU detects maskable and non-maskable interrupts. Interrupt priorities are set up in the CLIC. For information about these registers, see section 2, CPU.

12.3.1 Interrupt Vector Table

Table 12.3 describes the interrupt vector table. The interrupt vector addresses conform to the CLIC specifications (vectored mode).

Table 12.3 Interrupt vector table (1 of 2)

ID	IRQ number	Vector offset	Source	Description
0	—	0x000	—	Reserved
1	—	0x004	—	Reserved
2	—	0x008	—	Reserved
3	—	0x00C	Machine Timer	Machine software interrupt (msip)
4	—	0x010	—	Reserved
5	—	0x014	—	Reserved
6	—	0x018	—	Reserved
7	—	0x01C	Machine Timer	Machine timer interrupt (mtip)
8	—	0x020	—	Reserved
9	—	0x024	—	Reserved
10	—	0x028	—	Reserved
11	—	0x02C	—	Reserved
12	—	0x030	—	Reserved
13	—	0x034	—	Reserved
14	—	0x038	—	Reserved
15	—	0x03C	—	Reserved
16	—	0x040	—	Reserved
17	—	0x044	—	Reserved
18	—	0x048	—	Reserved
19	0	0x04C	ICU.IELSR0	Event selected in the ICU.IELSR0 register
20	1	0x050	ICU.IELSR1	Event selected in the ICU.IELSR1 register
21	2	0x054	ICU.IELSR2	Event selected in the ICU.IELSR2 register
22	3	0x058	ICU.IELSR3	Event selected in the ICU.IELSR3 register
23	4	0x05C	ICU.IELSR4	Event selected in the ICU.IELSR4 register
24	5	0x060	ICU.IELSR5	Event selected in the ICU.IELSR5 register
25	6	0x064	ICU.IELSR6	Event selected in the ICU.IELSR6 register
26	7	0x068	ICU.IELSR7	Event selected in the ICU.IELSR7 register
27	8	0x06C	ICU.IELSR8	Event selected in the ICU.IELSR8 register
28	9	0x070	ICU.IELSR9	Event selected in the ICU.IELSR9 register
29	10	0x074	ICU.IELSR10	Event selected in the ICU.IELSR10 register
30	11	0x078	ICU.IELSR11	Event selected in the ICU.IELSR11 register
31	12	0x07C	ICU.IELSR12	Event selected in the ICU.IELSR12 register
32	13	0x080	ICU.IELSR13	Event selected in the ICU.IELSR13 register
33	14	0x084	ICU.IELSR14	Event selected in the ICU.IELSR14 register
34	15	0x088	ICU.IELSR15	Event selected in the ICU.IELSR15 register
35	16	0x08C	ICU.IELSR16	Event selected in the ICU.IELSR16 register
36	17	0x090	ICU.IELSR17	Event selected in the ICU.IELSR17 register

Table 12.3 Interrupt vector table (2 of 2)

ID	IRQ number	Vector offset	Source	Description
37	18	0x094	ICU.IELSR18	Event selected in the ICU.IELSR18 register
38	19	0x098	ICU.IELSR19	Event selected in the ICU.IELSR19 register
39	20	0x09C	ICU.IELSR20	Event selected in the ICU.IELSR20 register
40	21	0x0A0	ICU.IELSR21	Event selected in the ICU.IELSR21 register
41	22	0x0A4	ICU.IELSR22	Event selected in the ICU.IELSR22 register
42	23	0x0A8	ICU.IELSR23	Event selected in the ICU.IELSR23 register
43	24	0x0AC	ICU.IELSR24	Event selected in the ICU.IELSR24 register
44	25	0x0B0	ICU.IELSR25	Event selected in the ICU.IELSR25 register
45	26	0x0B4	ICU.IELSR26	Event selected in the ICU.IELSR26 register
46	27	0x0B8	ICU.IELSR27	Event selected in the ICU.IELSR27 register
47	28	0x0BC	ICU.IELSR28	Event selected in the ICU.IELSR28 register
48	29	0x0C0	ICU.IELSR29	Event selected in the ICU.IELSR29 register
49	30	0x0C4	ICU.IELSR30	Event selected in the ICU.IELSR30 register
50	31	0x0C8	ICU.IELSR31	Event selected in the ICU.IELSR31 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 CLIC	" ✓" indicates the interrupt can be used as a CPU interrupt
Invoke DTC	" ✓" indicates the interrupt can be used to request DTC activation
Canceling Snooze	" ✓" indicates the interrupt can be used to request a return from Snooze mode
Canceling Software Standby mode	" ✓" indicates the interrupt can be used to request a return from Software Standby mode

Table 12.4 Event table (1 of 3)

Event number*3	Interrupt request source	Name	Connect to CLIC*6	Invoke DTC*7	Canceling Snooze	Canceling Software Standby
0x01	Port	PORT_IRQ0	✓	✓	✓	✓
0x02		PORT_IRQ1	✓	✓	✓	✓
0x03		PORT_IRQ2	✓	✓	✓	✓
0x04		PORT_IRQ3	✓	✓	✓	✓
0x05		PORT_IRQ4	✓	✓	✓	✓
0x06		PORT_IRQ5	✓	✓	✓	✓
0x07		PORT_IRQ6	✓	✓	✓	✓
0x08		PORT_IRQ7	✓	✓	✓	✓
0x09	DTC	DTC_COMPLETE	✓	—	✓*4	—
0x0B	ICU	ICU_SNZCANCEL	✓	—	✓	✓
0x0C	FLASH	FCU_FRDYI	✓	—	—	—
0x0D	LVD1	LVD_LVD1	✓	—	✓	✓
0x0E	LVD2	LVD_LVD2	✓	—	✓	✓

Table 12.4 Event table (2 of 3)

Event number[*3]	Interrupt request source	Name	Connect to CLIC[*6]	Invoke DTC[*7]	Canceling Snooze	Canceling Software Standby
0x10	LPW	SYSTEM_SNZREQ	—	✓	—	—
0x11	IWDT	IWDT_NMIUNDF	✓	—	✓	✓
0x12	CWDT	WDT_NMIUNDF	✓	—	—	—
0x13	RTC	RTC_ALM_OR_PRD	✓	—	✓	✓
0x2B	IICA	IIC0_ENDI/IIC0_WUI	✓	✓	✓	✓
0x2C		IIC1_ENDI/IIC1_WUI	✓	✓	✓	✓
0x30	KEYI	KEY_INTKR	✓	—	✓[*5]	✓[*5]
0x31	CAC	CAC_FEERI	✓	—	—	—
0x32		CAC_MENDI	✓	—	—	—
0x33		CAC_OVFI	✓	—	—	—
0x34	I/O Ports	IOPORT_GROUP1	✓	✓[*1]	—	—
0x35		IOPORT_GROUP2	✓	✓[*1]	—	—
0x38	TRNG	TRNG_RD_REQ	✓	—	—	—
0x39	ELC	ELC_SWEVT0	✓[*2]	✓	—	—
0x3A		ELC_SWEVT1	✓[*2]	✓	—	—
0x3B	SAU	SAU0_ENDI0	✓	✓	✓[*4]	—
0x3C		SAU0_ENDI1	✓	✓	✓[*4]	—
0x3D		SAU0_ENDI2	✓	✓	—	—
0x3E		SAU0_ENDI3	✓	✓	—	—
0x3F		SAU1_ENDI0	✓	✓	✓[*4]	—
0x40		SAU1_ENDI1	✓	✓	✓[*4]	—
0x43		SAU0_INTSRE0	✓	—	✓[*4]	—
0x44		SAU0_INTSRE1	✓	—	—	—
0x45		SAU1_INTSRE2	✓	—	✓[*4]	—
0x47	TAU	TAU0_ENDI0	✓	✓	—	—
0x48		TAU0_ENDI1	✓	✓	—	—
0x49		TAU0_ENDI2	✓	✓	—	—
0x4A		TAU0_ENDI3	✓	✓	—	—
0x4B		TAU0_ENDI4	✓	✓	—	—
0x4C		TAU0_ENDI5	✓	✓	—	—
0x4D		TAU0_ENDI6	✓	✓	—	—
0x4E		TAU0_ENDI7	✓	✓	—	—
0x4F		TAU0_MODE8_ENDI1	✓	✓	—	—
0x50		TAU0_MODE8_ENDI3	✓	✓	—	—
0x5B	TML32	TML32_OUTI	✓	✓	✓	✓
0x5C	REMC	REMC_OUTI	✓	✓	✓[*4]	—

Table 12.4 Event table (3 of 3)

Event number*3	Interrupt request source	Name	Connect to CLIC*6	Invoke DTC*7	Canceling Snooze	Canceling Software Standby
0x5D	UARTA	UARTA_TX_ENDI0	✓	✓	—	—
0x5E		UARTA_RX_ENDI0	✓	✓	✓	✓*8
0x5F		UARTA_RX_ERI0	✓	—	✓	✓*8
0x60		UARTA_TX_ENDI1	✓	✓	—	—
0x61		UARTA_RX_ENDI1	✓	✓	✓	✓*8
0x62		UARTA_RX_ERI1	✓	—	✓	✓*8
0x63	CMP	COMP_DET0	✓	—	✓	✓
0x64		COMP_DET1	✓	—	✓	✓
0x65	ADC12	ADC_ENDI	✓	✓	✓*4	—
0x69	DOC	DOC_DOPCI	✓	—	✓*4	—

Note 1. Only the first edge detection is valid.
Note 2. Only interrupts after DTC transfer are supported.
Note 3. For the setting of CPU and DTC interrupts, see Table 12.7.
Note 4. Using SELSR0.
Note 5. Only supported when KRCTL.KRMD = 1.
Note 6. Unintentional behavior may occur when events mark with "-" trigger DTC.
Note 7. Unintentional behavior may occur when events mark with "-" trigger interrupt.
Note 8. Only supported when operation clock of UARTA is UARTALCLK or UARTASCLK.

12.3.3 ICU and DTC Event Number

Table 12.5, Table 12.6 indicates the IELSRn.IELS[4:0] set values on the CPU interrupt or on DTC activation request. Table 12.7 indicates register set values of each event selection.

Table 12.5 ICU input link select (1) (1 of 2)

IELS[4:0]	Group 0 (interrupt ch IELSR0/8/16/24)	Group 1 (interrupt ch IELSR1/9/17/25)	Group 2 (interrupt ch IELSR2/10/18/26)	Group 3 (interrupt ch IELSR3/11/19/27)
0x00	Interrupt disabled	Interrupt disabled	Interrupt disabled	Interrupt disabled
0x01	PORT_IRQ0	PORT_IRQ1	PORT_IRQ2	PORT_IRQ3
0x02	DTC_COMPLETE	Setting prohibited	FCU_FRDYI	Setting prohibited
0x03	Setting prohibited	Setting prohibited	Setting prohibited	IWDT_NMIUNDF
0x04	LVD_LVD1	LVD_LVD2	Setting prohibited	Setting prohibited
0x05	Setting prohibited	RTC_ALM_OR_PRD	SYSTEM_SNZREQ	Setting prohibited
0x06	WDT_NMIUNDF	Setting prohibited	Setting prohibited	Setting prohibited
0x07	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0x08	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0x09	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0x0A	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0x0B	IIC0_ENDI/IIC0_WUI	IIC1_ENDI/IIC1_WUI	Setting prohibited	Setting prohibited
0x0C	Setting prohibited	Setting prohibited	Setting prohibited	KEY_INTKR
0x0D	Setting prohibited	ELC_SWEVT0	Setting prohibited	Setting prohibited
0x0E	TRNG_RD_REQ	Setting prohibited	Setting prohibited	Setting prohibited
0x0F	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0x10	Setting prohibited	Setting prohibited	CAC_FEERI	CAC_MENDI
0x11	Setting prohibited	Setting prohibited	IOPORT_GROUP1	Setting prohibited

Table 12.5 ICU input link select (1) (2 of 2)

IELS[4:0]	Group 0 (interrupt ch IELSR0/8/16/24)	Group 1 (interrupt ch IELSR1/9/17/25)	Group 2 (interrupt ch IELSR2/10/18/26)	Group 3 (interrupt ch IELSR3/11/19/27)
0x12	SAU0_ENDI0	SAU0_ENDI1	SAU0_ENDI2	SAU0_ENDI3
0x13	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0x14	Setting prohibited	SAU0_INTSRE0	Setting prohibited	SAU0_INTSRE1
0x15	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0x16	TAU0_ENDI0	TAU0_ENDI1	TAU0_ENDI2	TAU0_ENDI3
0x17	Setting prohibited	Setting prohibited	TAU0_MODE8_ENDI1	TAU0_MODE8_ENDI3
0x18	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0x19	TML32_OUTI	Setting prohibited	REMC_OUTI	Setting prohibited
0x1A	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0x1B	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0x1C	UARTA_TX_ENDI0	UARTA_RX_ENDI0	UARTA_RX_ERI0	Setting prohibited
0x1D	COMP_DET0	COMP_DET1	Setting prohibited	Setting prohibited
0x1E	Setting prohibited	Setting prohibited	Setting prohibited	ADC_ENDI
0x1F	Setting prohibited	Setting prohibited	Setting prohibited	DOC_DOPCI

Table 12.6 ICU input link select (2) (1 of 2)

IELS[4:0]	Group 4 (interupt ch IELSR4/12/20/28)	Group 5 (interupt ch IELSR5/13/21/29)	Group 6 (interupt ch IELSR6/14/12/30)	Group 7 (interupt ch IELSR7/15/23/31)
0x00	Interrupt disabled	Interrupt disabled	Interrupt disabled	Interrupt disabled
0x01	PORT_IRQ0	PORT_IRQ1	PORT_IRQ2	PORT_IRQ3
0x02	DTC_COMPLETE	Setting prohibited	FCU_FRDYI	Setting prohibited
0x03	Setting prohibited	Setting prohibited	Setting prohibited	IWDT_NMIUNDF
0x04	LVD_LVD1	LVD_LVD2	Setting prohibited	ICU_SNZCANCEL
0x05	Setting prohibited	RTC_ALM_OR_PRD	SYSTEM_SNZREQ	Setting prohibited
0x06	WDT_NMIUNDF	Setting prohibited	Setting prohibited	Setting prohibited
0x07	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0x08	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0x09	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0x0A	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0x0B	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0x0C	Setting prohibited	ELC_SWEVT1	Setting prohibited	KEY_INTKR
0x0D	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0x0E	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0x0F	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0x10	CAC_OVFI	Setting prohibited	Setting prohibited	Setting prohibited
0x11	Setting prohibited	Setting prohibited	IOPORT_GROUP2	PORT_IRQ7
0x12	SAU1_ENDI0	SAU1_ENDI1	Setting prohibited	Setting prohibited
0x13	Setting prohibited	PORT_IRQ5	PORT_IRQ6	Setting prohibited
0x14	Setting prohibited	SAU1_INTSRE2	Setting prohibited	Setting prohibited
0x15	TAU0_ENDI4	TAU0_ENDI5	TAU0_ENDI6	TAU0_ENDI7
0x16	PORT_IRQ4	Setting prohibited	Setting prohibited	Setting prohibited
0x17	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited

Table 12.6 ICU input link select (2) (2 of 2)

IELS[4:0]	Group 4 (interupt ch IELSR4/12/20/28)	Group 5 (interupt ch IELSR5/13/21/29)	Group 6 (interupt ch IELSR6/14/12/30)	Group 7 (interupt ch IELSR7/15/23/31)
0x18	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0x19	Setting prohibited	Setting prohibited	REMC_OUTI	Setting prohibited
0x1A	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0x1B	UARTA_TX_ENDI1	UARTA_RX_ENDI1	UARTA_RX_ERI1	Setting prohibited
0x1C	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0x1D	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0x1E	SAU0_ENDI0	SAU0_ENDI1	Setting prohibited	Setting prohibited
0x1F	TML32_OUTI	Setting prohibited	Setting prohibited	ADC_ENDI

Table 12.7 Register setting for event (1 of 2)

Name	IELSRn.IELS[4:0]							
	Group 0 (n = 0/8/16/24)	Group 1 (n = 1/9/17/25)	Group 2 (n = 2/10/18/26)	Group 3 (n = 3/11/19/27)	Group 4 (n = 4/12/20/28)	Group 5 (n = 5/13/21/29)	Group 6 (n = 6/14/22/30)	Group 7 (n = 7/15/23/31)
PORT_IRQ0	0x01	—	—	—	0x01	—	—	—
PORT_IRQ1	—	0x01	—	—	—	0x01	—	—
PORT_IRQ2	—	—	0x01	—	—	—	0x01	—
PORT_IRQ3	—	—	—	0x01	—	—	—	0x01
PORT_IRQ4	—	—	—	—	0x16	—	—	—
PORT_IRQ5	—	—	—	—	—	0x13	—	—
PORT_IRQ6	—	—	—	—	—	—	0x13	—
PORT_IRQ7	—	—	—	—	—	—	—	0x11
DTC_COMPLETE	0x02*1	—	—	—	0x02*1	—	—	—
ICU_SNZCANCEL	—	—	—	—	—	—	—	0x04*1
FCU_FRDYI	—	—	0x02*1	—	—	—	0x02*1	—
LVD_LVD1	0x04*1	—	—	—	0x04*1	—	—	—
LVD_LVD2	—	0x04*1	—	—	—	0x04*1	—	—
SYSTEM_SNZREQ	—	—	0x05*2	—	—	—	0x05*2	—
IWDT_NMIUNDF	—	—	—	0x03*1	—	—	—	0x03*1
WDT_NMIUNDF	0x06*1	—	—	—	0x06*1	—	—	—
RTC_ALM_OR_PRD	—	0x05*1	—	—	—	0x05*1	—	—
IIC0_ENDI/IIC0_WUI	0x0B	—	—	—	—	—	—	—
IIC1_ENDI/IIC1_WUI	—	0x0B	—	—	—	—	—	—
KEY_INTKR	—	—	—	0x0C*1	—	—	—	0x0C*1
CAC_FERRI	—	—	0x10*1	—	—	—	—	—
CAC_MENDI	—	—	—	0x10*1	—	—	—	—
CAC_OVFI	—	—	—	—	0x10*1	—	—	—
IOPORT_GROUP1	—	—	0x11	—	—	—	—	—
IOPORT_GROUP2	—	—	—	—	—	—	0x11	—
TRNG_RD_REQ	0x0E*1	—	—	—	—	—	—	—
ELC_SWEVT0	—	0x0D	—	—	—	—	—	—

Table 12.7 Register setting for event (2 of 2)

Name	IELSRn.IELS[4:0]							
	Group 0 (n = 0/8/16/24)	Group 1 (n = 1/9/17/25)	Group 2 (n = 2/10/18/26)	Group 3 (n = 3/11/19/27)	Group 4 (n = 4/12/20/28)	Group 5 (n = 5/13/21/29)	Group 6 (n = 6/14/22/30)	Group 7 (n = 7/15/23/31)
ELC_SWEVT1	—	—	—	—	—	0x0C	—	—
SAU0_ENDI0	0x12	—	—	—	0x1E	—	—	—
SAU0_ENDI1	—	0x12	—	—	—	0x1E	—	—
SAU0_ENDI2	—	—	0x12	—	—	—	—	—
SAU0_ENDI3	—	—	—	0x12	—	—	—	—
SAU1_ENDI0	—	—	—	—	0x12	—	—	—
SAU1_ENDI1	—	—	—	—	—	0x12	—	—
SAU0_INTSRE0	—	0x14*1	—	—	—	—	—	—
SAU0_INTSRE1	—	—	—	0x14*1	—	—	—	—
SAU1_INTSRE2	—	—	—	—	—	0x14*1	—	—
TAU0_ENDI0	0x16*1	—	—	—	—	—	—	—
TAU0_ENDI1	—	0x16*1	—	—	—	—	—	—
TAU0_ENDI2	—	—	0x16*1	—	—	—	—	—
TAU0_ENDI3	—	—	—	0x16*1	—	—	—	—
TAU0_ENDI4	—	—	—	—	0x15*1	—	—	—
TAU0_ENDI5	—	—	—	—	—	0x15*1	—	—
TAU0_ENDI6	—	—	—	—	—	—	0x15*1	—
TAU0_ENDI7	—	—	—	—	—	—	—	0x15*1
TAU0_MODE8_ENDI1	—	—	0x17*1	—	—	—	—	—
TAU0_MODE8_ENDI3	—	—	—	0x17*1	—	—	—	—
TML32_OUTI	0x19	—	—	—	0x1F	—	—	—
REMC_OUTI	—	—	0x19	—	—	—	0x19	—
UARTA_TX_ENDI0	0x1C	—	—	—	—	—	—	—
UARTA_RX_ENDI0	—	0x1C	—	—	—	—	—	—
UARTA_RX_ERI0	—	—	0x1C*1	—	—	—	—	—
UARTA_TX_ENDI1	—	—	—	—	0x1B	—	—	—
UARTA_RX_ENDI1	—	—	—	—	—	0x1B	—	—
UARTA_RX_ERI1	—	—	—	—	—	—	0x1B*1	—
COMP_DET0	0x1D*1	—	—	—	—	—	—	—
COMP_DET1	—	0x1D*1	—	—	—	—	—	—
ADC_ENDI	—	—	—	0x1E	—	—	—	0x1F
DOC_DOPCI	—	—	—	0x1F*1	—	—	—	—

Note 1. Use for CPU interrupt only.
Note 2. Use for DTC interrupt only.

12.4 Interrupt Operation

The ICU performs the following functions:

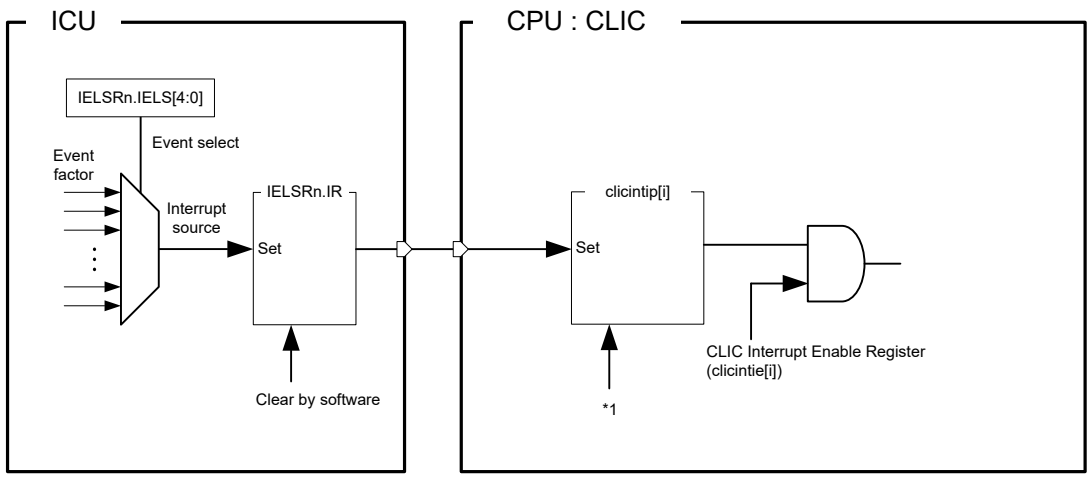
- Detecting interrupts

- Enabling and disabling interrupts
- Selecting interrupt request destinations such as CPU interrupt, DTC activation.

12.4.1 Interrupt Detection Selection

The ICU selects an event source input from a peripheral function interrupt or an external terminal interrupt with IELSRn.IELS[4:0].

The accepted interrupt source sets the IELSRn.IR to 1 and sends an interrupt request to the CLIC. Each interrupt input has a dedicated interrupt pending bit in the clicintip[i] register.



Note 1. The clearing mechanism is different between Selective Hardware Vectoring (SHV) mode and non-SHV (common code) mode. When a Selective Hardware Vectoring (SHV) interrupt is selected and occurred, the HW automatically clears the corresponding pending bit. In this case, SW does not need to clear the pending bit at all in the service routine. When non-SHV (common code) interrupt is selected, HW clears the corresponding pending bit only when SW uses a 'csrrsi/csrrci mnxti' instruction to select this interrupt and return its entry address.

Figure 12.2 Interrupt path of the ICU and CPU (CLIC)

12.4.2 Detecting Interrupts

External pin interrupt requests are detected by either the edge or the level (falling edge, rising edge, rising and falling edges, or low level) of the interrupt signal. Set the IRQMD[1:0] bits in the IRQCRi register to select the detection mode for the IRQi pins. For interrupt sources associated with peripherals module, see Table 12.3 and Table 12.4.

12.5 Interrupt setting procedure

12.5.1 Enabling Interrupt Requests

The following steps show the procedure for enabling an interrupt request:

1. Set the CLIC Interrupt Enable register (clicintie[i]).
2. Set the IELSRn.IELS[4:0] bits as the interrupt source.
3. Specify the operation settings for the event source, such as 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 software standby mode cancellation (WUPEN register setting).
2. Clear the interrupt source setting (IELSRn.IELS[4:0] = 0x00).
3. Clear the interrupt status flag (IELSRn.IR = 0).
4. Clear the CLIC Interrupt Enable register (clicintie[i]) and the CLIC Interrupt Pending register (clicintip[i]) for edge triggered interrupts.

12.5.3 Polling for interrupts

The following steps show the procedure for polling for interrupt requests:

1. Set the CLIC Interrupt Enable register (clicintie[i]).
2. Set the IELSRn.IELS[4:0] bits as the interrupt source.
3. Specify the operation settings for the event source, such as software standby mode cancellation (WUPEN register setting).
4. Poll the CLIC Interrupt Pending register (clicintip[i]).

12.5.4 Selecting Interrupt Request Destinations

The interrupt output destination, CPU or DTC, can be independently selected for each interrupt source.

The available destinations are fixed for each interrupt, as described in Table 12.3, Table 12.4, Table 12.5 and Table 12.6.

Use an interrupt request destination setting that is indicated by a "✓" in the event list.

If the DTC is selected as the destination for requests from an IRQi pin, the IRQMD[1:0] bits in IRQCRi must be set 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 CLIC. Set the IELSRn.IELS[4:0] bits 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[4:0] 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 12.8 shows operation when the DTC is the interrupt request destination.

Table 12.8 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 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 controls the interrupt generation 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 14.2 in section 14, Data Transfer Controller (DTC).

12.5.5 Digital Filter

A digital filter function is provided for the external interrupt request pins IRQi, (i = 0 to 7) and the NMI pin interrupt. It samples input signals on the filter with PCLKB sampling clock and removes any signal with a pulse width less than 3 sampling cycles.

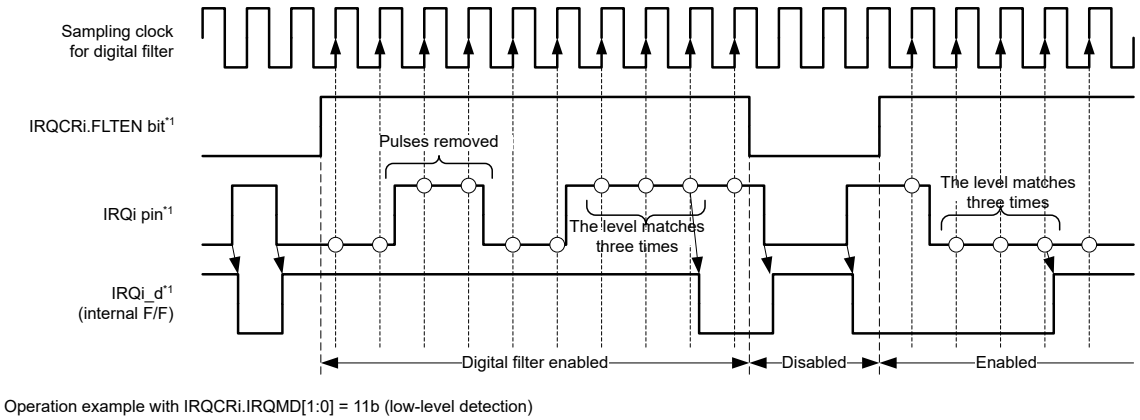
To use the digital filter for an IRQi pin:

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 7).
2. Set the IRQCRi.FLTEN bit (i = 0 to 7) 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 12.3 shows an example of digital filter operation.



Note 1. i = 0 to 7

Figure 12.3 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 7) to 0 (digital filter disabled).
3. Set the IRQMD[1:0] bits of the given IRQCRi register (i = 0 to 7) 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[4:0] bits and the IELSRn.DTCE bit to 0.

- If the IRQ pin is to be used for DTC activation, set the IELSRn.IELS[4:0] bits and the IELSRn.DTCE bit to 1.

12.6 Non-Maskable Interrupt Operation

The following sources can trigger a non-maskable interrupt:

- NMI pin 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 error interrupt

Non-maskable interrupts can only be used with the CPU, not to activate the DTC. 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, use the following procedure:

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.

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 CLIC.

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:

Non-maskable interrupt:

- Use the NMIER register to enable the target interrupt request.

Maskable interrupt:

- Select the interrupt source that enables return from the Software Standby.

- Use the WUPEN register to enable the target interrupt request.
- Select the CPU as the interrupt request destination.
- Enable the interrupt in the CLIC.

Interrupt requests through IRQn pins that do not satisfy these conditions are not detected while the clock is stopped in Software Standby mode

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 event that you want to trigger a return to Normal mode from Snooze mode in SELSR0.SELS[7:0].
2. Set the value 0x04 (ICU_SNZCANCEL) in IELSRn.IELS[4:0] (n =7/15/23/31).
3. Select the CPU as the interrupt request destination.
4. Enable the interrupt in the CLIC.

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.

13. Buses

13.1 Overview

The bus subsystem handles routing of transactions between bus masters and bus slaves in the device. In addition, the bus module provides bus error monitoring, error status logging, and flash read protection.

Table 13.1 lists the bus specifications, Figure 13.1 shows the bus configuration, and Table 13.2 lists the addresses assigned for each bus.

For details on bus error monitoring and error status logging, see section 13.3. Bus Error Monitoring. For details on the flash read protection function, see section 13.4. Flash Read Protection.

Table 13.1 Bus specifications

Bus type		Description
Main bus	Instruction bus (CPU)	• Connected to CPU instruction bus • Connected to on-chip memory and internal peripheral bus
	Data bus (CPU)	• Connected to CPU data bus • Connected to on-chip memory and internal peripheral bus
	DMA bus	• Connected to DTC • Connected to on-chip memory and internal peripheral bus
Slave interface	Memory bus 1	Connected to code flash memory
	Memory bus 4	Connected to SRAM0
	Internal peripheral bus 1	Connected to system control related to peripheral modules (except Debug Module)
	Internal peripheral bus 3	Connected to peripheral modules (CAC, I/O Ports, WDT, IWDT, ADC12, CMP, DOC, ICU, ELC, KINT, TAU, SAU, RTC, IICA, UARTA, DAC8, TSN, TRNG and CRC)
	Internal peripheral bus 9	Connected to flash memory (in P/E (Programming/Erasure))
	Internal peripheral bus 10	Connected to Debug Module (DM)
	Internal peripheral bus 11	Connected to data flash memory

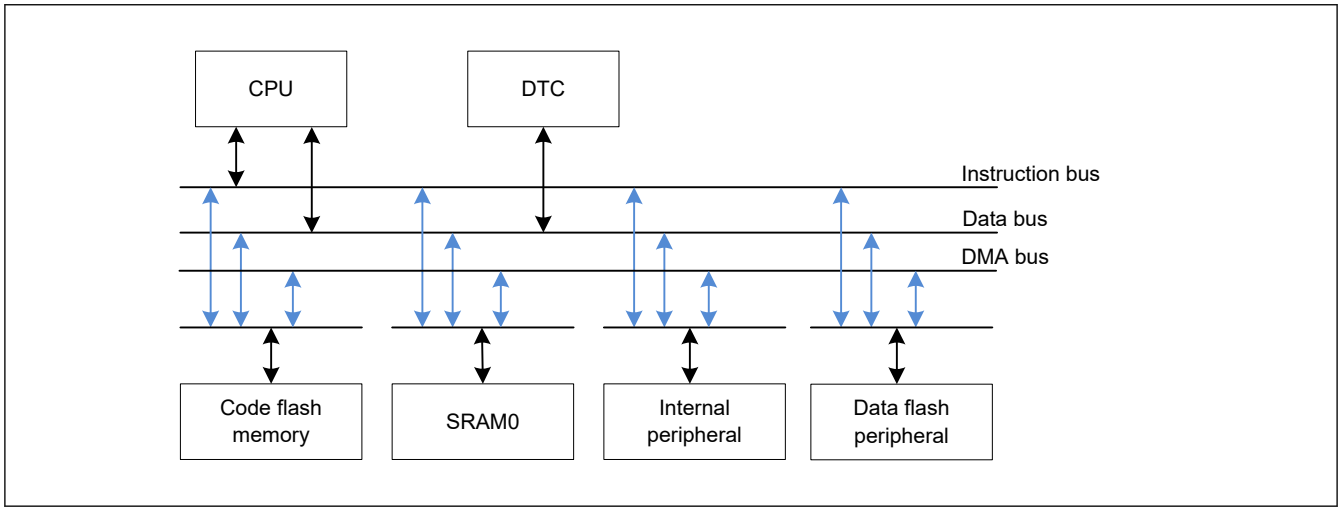


Figure 13.1 Bus configuration

Table 13.2 Addresses assigned for each bus (1 of 2)

Address	Bus	Area
0x0000_0000 to 0x01FF_FFFF	Memory bus 1	Code flash memory
0x2000_0000 to 0x2000_7FFF	Memory bus 4	SRAM0
0x4000_0000 to 0x4001_8FFF	Internal peripheral bus 1	Peripheral registers
0x4001_A000 to 0x4001_BFFF	Internal peripheral bus 1	Peripheral registers

Table 13.2 Addresses assigned for each bus (2 of 2)

Address	Bus	Area
0x4001_C000 to 0x4001_CFFF	Internal peripheral bus 9	Peripheral registers
0x4001_D000 to 0x4001_FFFF	Internal peripheral bus 1	Peripheral registers
0x4004_0000 to 0x400B_FFFF	Internal peripheral bus 3	Peripheral registers
0x4010_0000 to 0x407D_FFFF	Internal peripheral bus 11	Data flash memory
0x407E_0000 to 0x407F_FFFF	Internal peripheral bus 9	Flash memory (in P/E)
0xE200_0000 to 0xE67F_FFFF	Internal peripheral bus 1	Peripheral registers
0xE680_0000 to 0xE680_0FFF	Internal peripheral bus 10	Debug Module (DM) registers

Note: All internal peripherals belong to the same slave bus.

13.2 Description of Buses

13.2.1 Main Buses

The main buses consist of the CPU instruction bus, CPU data bus, and DMA bus and are connected to the following slave buses:

- Code flash memory
- SRAM0
- Internal peripherals

The CPU instruction bus is used for instruction fetches by the CPU. The CPU data bus is used for data accesses by the CPU.

The DMA bus is used for accesses by the DTC module.

13.2.2 Slave Interface

For connections from the main buses to the slave interfaces, see the slave interfaces in section 13.1. Overview.

Bus accesses from the CPU instruction bus, CPU data bus, and DMA bus are arbitrated and have the following fixed priority order:

DMA bus > CPU data bus > CPU instruction bus

Requests from different masters to different slave buses can proceed simultaneously.

13.2.3 Parallel Operations

Parallel operations are possible when different bus masters request access to different slave buses (code flash memory bus, SRAM0 bus, and internal peripheral bus). Figure 13.2 shows an example of parallel operations. In this example, the CPU uses the instruction and data buses for simultaneous access to the flash and SRAM, respectively. Additionally, the DTC simultaneously uses the DMA bus for access to a peripheral bus.

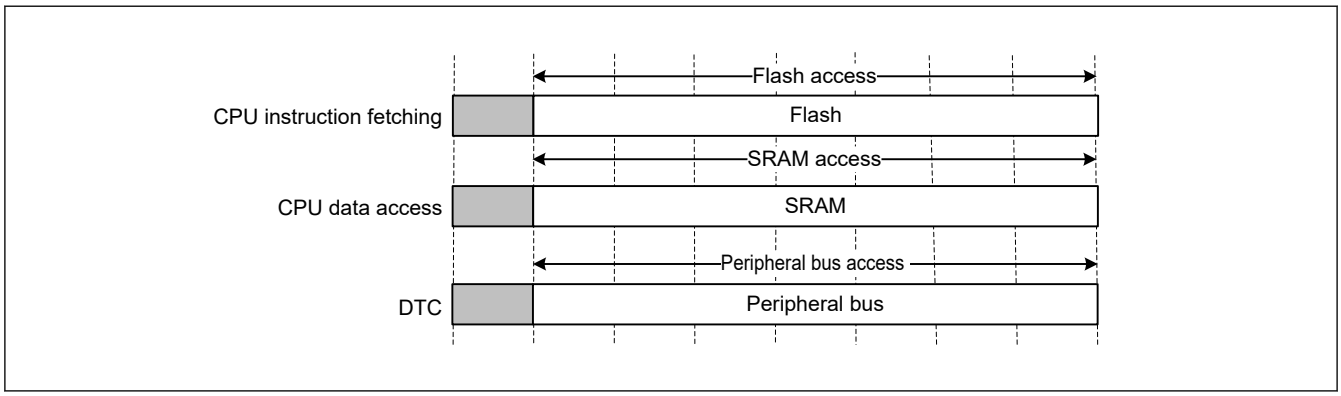


Figure 13.2 Example of parallel operations

13.2.4 Restriction on Endianness

Memory space must be little-endian to execute code on the CPU core.

13.3 Bus Error Monitoring

The monitoring system monitors accesses to each individual area. When a violation is detected, an error is generated and recorded.

13.3.1 Error Types that Occur by Bus

Three types of errors can occur on each master bus:

- Illegal address access error
- Slave bus error
- Illicit memory access error

Illegal address access errors occur when a bus master attempts to access a reserved region instead of accessing a bus slave in the system.

Slave bus errors occur when the bus slave reports an error back to the requesting master using the AHB-Lite error response protocol. Illicit memory access errors occur when the CPU operates incorrectly and attempts an inappropriate access such as an instruction fetch to peripheral modules or a write access to code flash.

13.3.2 Operation when a Bus Error Occurs

When a bus error occurs (due to illegal address access, illicit memory access, or slave bus error), operation is not guaranteed and the error is returned to the requesting master IP. If the error is due to CPU access and error reporting is enabled using BUSMCNTx.IERES, a Bus Error Non-Maskable Interrupt (NMI) or a Bus Error Reset is requested, depending on the value of the BUSCNTOAD.OAD bit. The NMI request must be enabled using the ICU.NMIER.BUSEN bit for the CPU to receive the error. If the NMI request is not enabled, the CPU is not notified of the error. Software must ensure no errors have occurred by checking the error status register (BUSnERRSTAT) before enabling the NMI using ICU.NMIER.BUSEN. The non-maskable interrupt status is indicated in ICU.NMISR.BUSST and can be cleared using ICU.NMICLR.BUSCLR. The reset status is indicated in SYSTEM.RSTSR1.BUSERRF.

In addition, the bus error information is stored in the corresponding BUSnERRSTAT and BUSnERRADD registers. These registers are cleared by a system reset other than the reset from a Bus Error Reset. To continue the system operation after a bus error occurs and after the cause of the bus error is confirmed by the error status registers, a system reset is required. Otherwise, the next bus error cannot be recorded.

If the error is from DTC access, the DTC does not receive the bus error and the transfer continues. The CPU must poll the bus error status and address registers to confirm if an error occurred during the DTC access.

13.3.3 Conditions for Issuing Illegal Address Access Errors

Table 13.3 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.

Table 13.3 Conditions leading to illegal address access errors (1 of 2)

Address	Slave bus name	Main buses		
		Instruction bus (CPU)	Data bus (CPU)	DMA bus
0x0000_0000 to 0x01FF_FFFF	Memory bus 1	—	—	—
0x0200_0000 to 0x1FFF_FFFF	Reserved	E	E	E
0x2000_0000 to 0x2000_7FFF	Memory bus 4	—	—	—
0x2000_8000 to 0x3FFF_FFFF	Reserved	E	E	E
0x4000_0000 to 0x4001_8FFF	Internal peripheral bus 1	—	—	—
0x4001_9000 to 0x4001_9FFF	Reserved	E	E	E
0x4001_A000 to 0x4001_FFFF	Internal peripheral bus 1, 9	—	—	—

Table 13.3 Conditions leading to illegal address access errors (2 of 2)

Address	Slave bus name	Main buses		
		Instruction bus (CPU)	Data bus (CPU)	DMA bus
0x4002_0000 to 0x4003_FFFF	Reserved	E	E	E
0x4004_0000 to 0x400B_FFFF	Internal peripheral bus 3	—	—	—
0x400C_0000 to 0x400F_FFFF	Reserved	E	E	E
0x4010_0000 to 0x407F_FFFF	Internal peripheral bus 9, 11	—	—	—
0x4080_0000 to 0xE1FF_FFFF	Reserved	E	E	E
0xE200_0000 to 0xE680_0FFF	Internal peripheral bus 1, 10	—	—	—
0xE680_1000 to 0xFFFF_FFFF	Reserved	E	E	E

Note: E indicates the path where an illegal address access error occurs.
— indicates the path where an illegal address access error does not occur.
Note: The bus module detects an access error resulting from access to reserved area, for example if no area is assigned for the slave.
0x0200_0000 to 0x1FFF_FFFF: Access error detection.
0x0000_0000 to 0x01FF_FFFF: No access error detection (Memory bus 1).

13.3.4 Conditions for Causing Slave Bus Error When Accessing the Debug Module (DM)

Slave bus errors occur when the bus slave reports an error back to the requesting master using the AHB-Lite error response protocol.

Accesses to the debug module (DM) are allowed only by the CPU when in debug mode. A slave bus error occurs when the CPU attempts to access the debug module (DM) in non-debug mode. In addition, any access to the DM by the DTC triggers a slave bus error response.

13.3.5 Conditions for Issuing Illicit Memory Access Errors

The IEC60730 standard mandates checking that the CPU is operating correctly.

This MCU provides a function to trigger an NMI or a reset request when an illicit memory area is accessed by the CPU. Access to the area indicated as not allowed is shown in Figure 13.3, is detected as illicit and triggers the illicit memory access error. Access to a reserved area triggers the illegal address access error by the bus decoder.

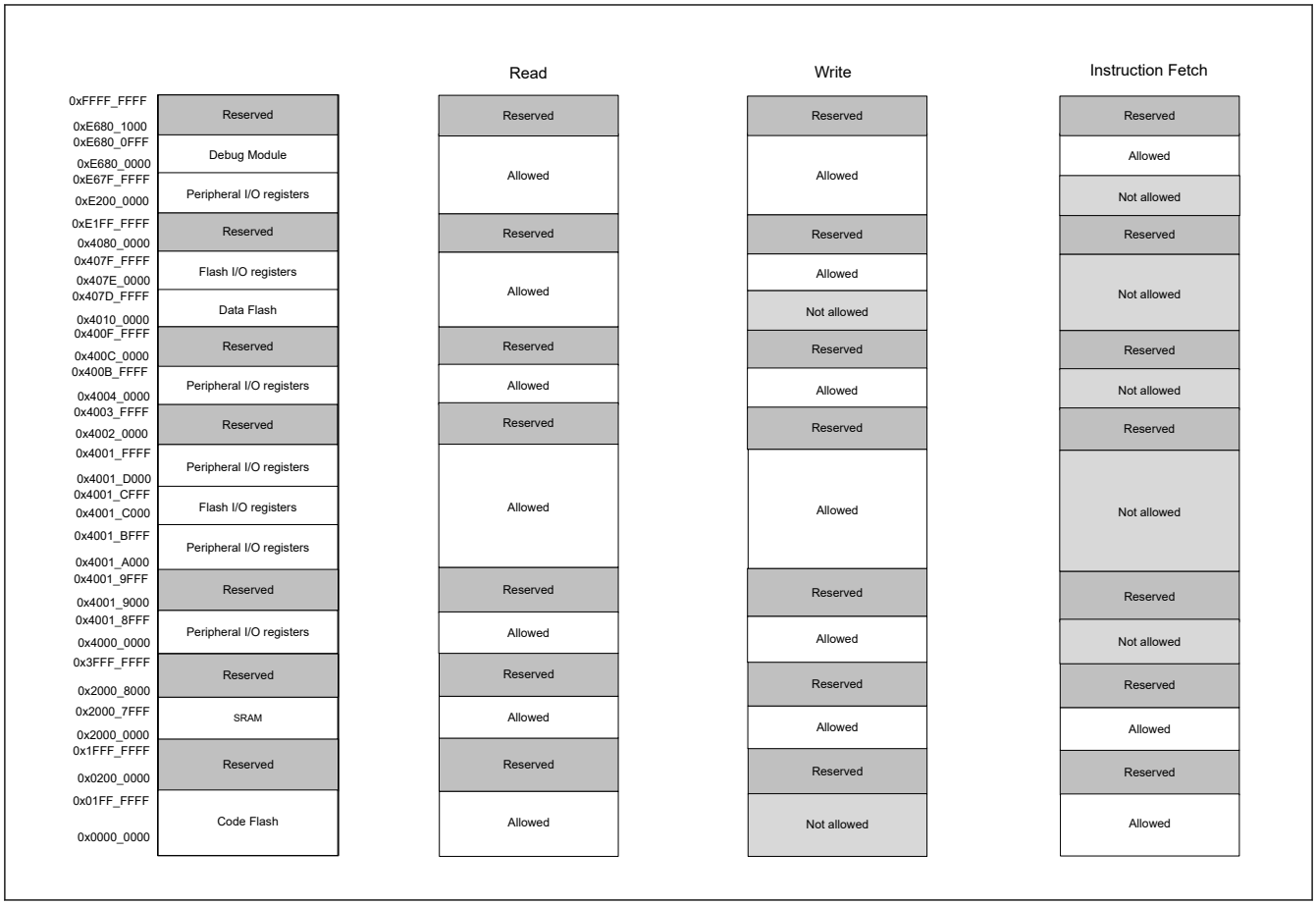


Figure 13.3 Illicit access areas (indicated as not allowed)

13.4 Flash Read Protection

13.4.1 Function of Flash Read Protection

The flash read protection function can be used to protect a specified range of the code flash memory region (0x00000000 to 0x0001FFFF) against data read access by the CPU and DTC.

Data access requests to the protected region are discarded without triggering any error, NMI, or reset.

Fetching of instructions in the specified range by the CPU is still possible. Note that even if program code is to be executed from the protected region, reading data that is placed in the protected region is not possible.

Figure 13.4 shows how the flash read protection function prevents data access to the protected region (between the Start Address and End Address) while allowing normal instruction fetches. The start and end addresses can be set using the Flash Read Protection Start Address Register (FLRPROTS) and the Flash Read Protection End Address Register (FLRPROTE), respectively in the Option-Setting Memory.

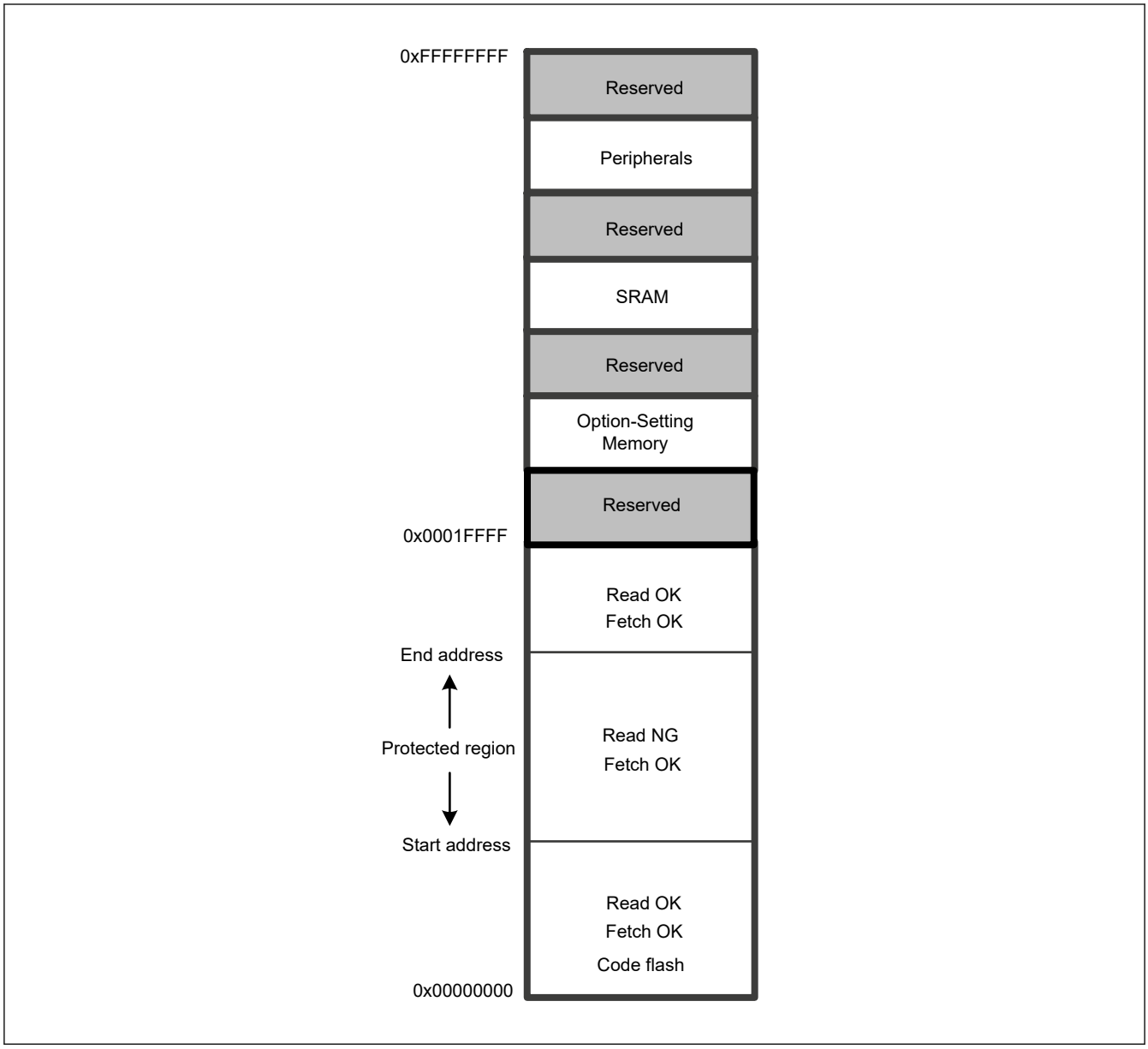


Figure 13.4 Flash read protection (preventing data access to protected region)

13.4.2 Setting of Flash Read Protection

The flash read protection in the option-setting memory is set by a flash memory programmer, serial programming using an on-chip debugger, or self-programming. After the settings are made, read access to addresses in the code flash memory area between the start and end of the flash read protection becomes impossible.

The program to be executed should not be placed on the last three words from the bottom of the code flash or SRAM region. The CPU is able to prefetch instructions up to three words ahead from the current Program Counter (PC).

13.4.3 Data Placement in Code Region

Data associated with code to be executed from the protected region must be placed in an area outside the protected region. In addition, because vector fetches are regarded as data accesses, the interrupt vector table must also be placed outside the protected region.

13.5 Register Descriptions

13.5.1 BUSMCNTx : Control Register x (x = INST, DAT, DMA)

Base address: BUS = 0x4000_3000

Offset address: 0x1000 (BUSMCNTINST)
0x1004 (BUSMCNTDAT)
0x1008 (BUSMCNTDMA)

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	IERES	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	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/W
15	IERES	Ignore Error Responses 0: Bus errors are reported 1: Bus errors are not reported	R/W

Note: Changing reserved bits from the initial value of 0 is prohibited. Operation during the change is not guaranteed.

Bus errors are reported to the CPU through a non-maskable interrupt request. In addition to setting the IERES bit to 0, the NMI request must be enabled using the ICU.NMIER.BUSEN bit for the CPU to receive the error. If the NMI request is enabled after a bus error occurs, the status and address of any new errors are not recorded in BUSnERRSTAT and BUSnERRADD. Software must ensure no errors have occurred by checking the error status register (BUSnERRSTAT) before enabling the NMI using ICU.NMIER.BUSEN.

IERES bit (Ignore Error Responses)

The IERES bit specifies the enable or disable of an error response of the AHB-Lite protocol.

Table 13.4 lists the registers associated with each bus type.

Table 13.4 Associations between bus types and registers

Bus type	Control Register	Bus Error Address Register	Bus Error Status Register
Instruction bus (CPU)	BUSMCNTINST	BUS1ERRADD	BUS1ERRSTAT
Data bus (CPU)	BUSMCNTDAT	BUS2ERRADD	BUS2ERRSTAT
DMA bus	BUSMCNTDMA	BUS3ERRADD	BUS3ERRSTAT

13.5.2 BUSCNTOAD : Bus Control Error Operation After Detection Register

Base address: BUS = 0x4000_3000

Offset address: 0x1400

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 These bits enable or disable writes to the OAD bit.	R/W*1

Note: Changing reserved bits from the initial value of 0 is prohibited. Operation during the change is not guaranteed.

Note 1. Write data is not retained.

OAD bit (Operation After Detection)

The OAD bit generates either a reset or non-maskable interrupt when any type of bus error occurs (illegal address access error, slave bus error, or illicit memory access error). See section 13.3.1. Error Types that Occur by Bus for more information about the types of errors.

When the OAD bit is set, simultaneously write 0xA5 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 OAD bit. When writing to the OAD bit, simultaneously write 0xA5 to the KEY[7:0] bits. When other values are written, the OAD bit is not updated. The KEY[7:0] bits are always read as 0x00.

13.5.3 BUSnERRADD : Bus Error Address Register n (n = 1, 2, 3)

Base address: BUS = 0x4000_3000

Offset address: 0x1800 (n = 1)
0x1810 (n = 2)
0x1820 (n = 3)

Bit position:	31																												0
Bit field:	BERAD[31:0]																												
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	x

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

BUSnERRADD is cleared by a system reset other than the reset from a Bus Error Reset. For more information, see section 5, Resets.

The error address of only the first error received by a bus master is recorded in BUSnERRADD. Subsequent errors received by the same bus master are not recorded until a system reset (other than a Bus Error Reset) has occurred.

Table 13.4 lists the registers associated with each bus type.

BERAD[31:0] bits (Bus Error Address)

The BERAD[31:0] bits store the accessed address when a bus error occurred.

The value is valid only when one of the flags (ILLERRSTAT, SLERRSTAT, ILTERRSTAT) in BUSnERRSTAT (n = 1, 2, 3) is set to 1.

For more information, see the description of ILLERRSTAT, SLERRSTAT, and ILTERRSTAT flags in section 13.5.4. BUSnERRSTAT : BUS Error Status Register n (n = 1, 2, 3).

13.5.4 BUSnERRSTAT : BUS Error Status Register n (n = 1, 2, 3)

Base address: BUS = 0x4000_3000

Offset address: 0x1804 (n = 1)
0x1814 (n = 2)
0x1824 (n = 3)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	ILLERRSTAT	SLERRSTAT	ILTERRSTAT	—	—	—	—	ACCSTAT
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ACCSTAT	Error Access Status flag Access status when the error occurred: 0: Read access 1: Write access	R
4:1	—	These bits are read as 0.	R
5	ILTERRSTAT	Illicit Memory Access Error Status flag 0: No bus error occurred 1: Bus error occurred	R
6	SLERRSTAT	Slave Bus Error Status flag 0: No bus error occurred 1: Bus error occurred	R
7	ILLERRSTAT	Illegal Address Access Error Status flag 0: No bus error occurred 1: Bus error occurred	R

BUSnERRSTAT is cleared by a system reset other than the reset from Bus Error Reset. For more information, see section 5, Resets.

The error status of only the first error received by a bus master is recorded in BUSnERRSTAT. Subsequent errors received by the same bus master are not recorded until a system reset (other than a Bus Error Reset) has occurred.

Table 13.4 lists the registers associated with each bus type.

ACCSTAT flag (Error Access Status flag)

The ACCSTAT flag indicates the access status, write or read access, when a bus error occurs. The value is valid only when one of the flags (ILLERRSTAT, SLERRSTAT, ILTERRSTAT) in BUSnERRSTAT (n = 1, 2, 3) is set to 1.

ILTERRSTAT flag (Illicit Memory Access Error Status flag)

The ILTERRSTAT flag indicates whether an illicit memory access has occurred. When an illicit memory access error occurs, the access address and status of write or read access are stored. In addition, the ILLERRSTAT flag is set to 1.

SLERRSTAT flag (Slave Bus Error Status flag)

The SLERRSTAT flag indicates whether a bus slave has returned an error on the bus. When a slave bus error occurs, the access address and status of write or read access are stored. In addition, the SLERRSTAT flag is set to 1.

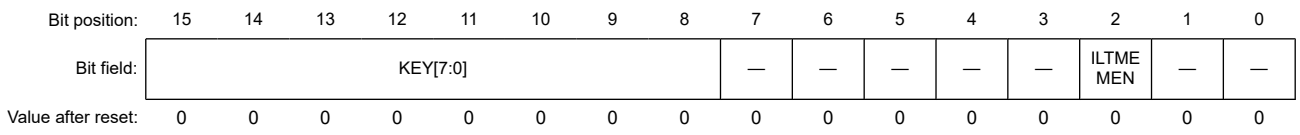
ILLERRSTAT flag (Illegal Address Access Error Status flag)

The ILLERRSTAT flag indicates whether an illegal address access error occurred on the bus. When a bus error occurs, the access address and status of write or read access are stored. In addition, the ILTERRSTAT flag is set to 1.

13.5.5 ILTMEMCTL : Illicit Memory Access Detection Control Register

Base address: BUS = 0x4000_3000

Offset address: 0xC00



Bit	Symbol	Function	R/W
1:0	—	These bits are read as 0. The write value should be 0.	R/W
2	ILTMEMEN	Illicit Memory Access Detection Enable 0: Disabled 1: Enabled	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
15:8	KEY[7:0]	Key Code These bits enable or disable writes to the ILTMEMEN bit.	R/W*1

Note: Changing reserved bits from the initial value of 0 is prohibited. Operation during the change is not guaranteed.
Note 1. Write data is not retained.

ILTMEMEN bit (Illicit Memory Access Detection Enable)

The ILTMEMEN bit is used to enable or disable the illicit memory access detection function. When enabled, a Bus Error NMI/Reset request is generated when an illicit memory access occurs.

When the ILTMEMEN bit is set, simultaneously write 0xA5 to the KEY[7:0] bits using halfword access.

KEY[7:0] bit (Key Code)

The KEY[7:0] bits enable or disable writes to the ILTMEMEN bit. When writing to the ILTMEMEN bit, simultaneously write 0xA5 to the KEY[7:0] bits. When other values are written, the ILTMEMEN bit is not updated. The KEY[7:0] bits are always read as 0x00.

13.6 Register Descriptions (Option-Setting Memory)

All flash read protection registers are option-setting memory. Option-setting memory refers to a set of registers that are available for selecting the state of the microcontroller after a reset. The option-setting memory is allocated in the code flash. For more details, see section 6, Option-Setting Memory.

13.6.1 FLRPROTS : Flash Read Protection Start Address Register

Address: 0x0000_0418/0x0000_2418*1

Bit position:	31							23	0
Bit field:	—	—	—	—	—	—	—	—	FLRPROTS[23:0]
Value after reset:	0	0	0	0	0	0	0	0	The value set by user

Bit	Symbol	Function	R/W
23:0	FLRPROTS[23:0]	Region Start Address Address where the region starts, for use in region determination. The value range is from 0x0000_0000 to 0x0001_FFFC, excluding reserved areas. The lower 2 bits are read as 0. When programming to the code flash, the lower 2 bits write value should be 0.	R/W
31:24	—	These bits are read as 0. When programming to the code flash, the write value should be 0.	R/W

Note 1. The address of these registers changes when the boot swap is set.

13.6.2 FLRPROTE : Flash Read Protection End Address Register

Address: 0x0000_041C/0x0000_241C*1

Bit position:	31							23	0
Bit field:	—	—	—	—	—	—	—	—	FLRPROTE[23:0]
Value after reset:	0	0	0	0	0	0	0	0	The value set by user

Bit	Symbol	Function	R/W
23:0	FLRPROTE[23:0]	Region End Address Address where the region ends, for use in region determination. The value range is from 0x0000_0003 to 0x0001_FFFF, excluding reserved areas. The lower 2 bits are read as 1. When programming to the code flash, the lower 2 bits write value should be 1.	R/W

Bit	Symbol	Function	R/W
31:24	—	These bits are read as 0. When programming to the code flash, the write value should be 0.	R/W

Note 1. The address of these registers changes when the boot swap is set.

13.6.3 FLRPROTAC : Flash Read Protection Access Control Register

Address: 0x0000_0438/0x0000_2438*1

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:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DIS
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	*2

Bit	Symbol	Function	R/W
0	DIS	Flash Read Protection Disable 0: Enabled 1: Disabled	R/W
31:1	—	These bits are read as 1. When programming to the code flash, the write value should be 1.	R/W

Note 1. The address of these registers changes when the boot swap is set.
Note 2. The value set by user

DIS bit (Flash Read Protection Disable)

The DIS bit enables or disables the flash read protection function. If enabled, the code flash memory region within the limits set up by FLRPROTS and FLRPROTE is protected from data access by the CPU and DTC.

13.7 Usage Notes

13.7.1 Notes on the Use of a Debugger

The memory cannot be debugged if the flash read protection function is enabled. Disable the flash read protection when debugging a program. OCD debug is only valid when FLRPROTAC register is 0xFFFF_FFFF.

13.8 References

1. *RISC-V External Debug Support* (Version 0.13.2).

14. Data Transfer Controller (DTC)

14.1 Overview

A Data Transfer Controller (DTC) module is provided for transferring data when activated by an interrupt request.

Table 14.1 lists the DTC specifications and Figure 14.1 shows DTC block diagram.

Table 14.1 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.
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
Module-stop function	Module-stop state can be set to reduce power consumption

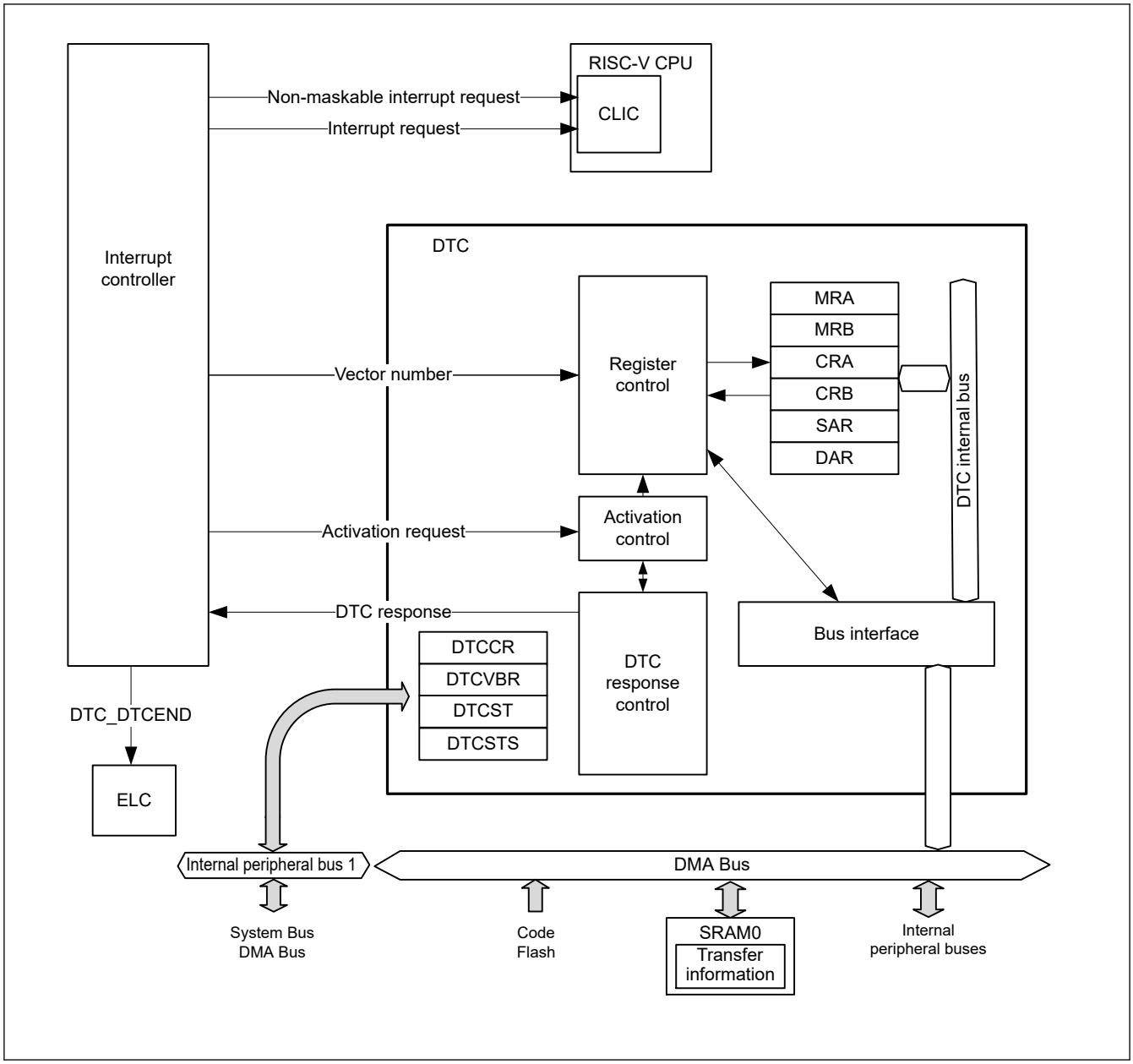


Figure 14.1 DTC block diagram

See section 12.1. Overview in section 12, Interrupt Controller Unit (ICU) for the connections between the DTC and CLIC in the CPU.

14.2 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.

14.2.1 MRA : DTC Mode Register A

Base address: DTCVBR

Offset address: 0x03 + 0x4 × Vector number
(Inaccessible directly from the CPU. See section 14.3.1. 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	—
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 14.3.1. Allocating Transfer Information and DTC Vector Table.

14.2.2 MRB : DTC Mode Register B

Base address: DTCVBR

Offset address: 0x02 + 0x4 × Vector number
(Inaccessible directly from the CPU. See section 14.3.1. 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.	—

Bit	Symbol	Function	R/W
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 MRA.SZ[1:0] = 01b +4 when MRA.SZ[1:0] = 10b 1 1: DAR value is decremented after data transfer: -1 when MRA.SZ[1:0] = 00b -2 when MRA.SZ[1:0] = 01b -4 when MRA.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 14.3.1. 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 14.3.

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 14.4.6. Chain Transfer.

14.2.3 SAR : DTC Transfer Source Register

Base address: DTCVBR

Offset address: 0x04 + 0x4 × Vector number
(Inaccessible directly from the CPU. See section 14.3.1. 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	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 14.3.1. 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.

14.2.4 DAR : DTC Transfer Destination Register

Base address: DTCVBR

Offset address: 0x08 + 0x4 × Vector number
(Inaccessible directly from the CPU. See section 14.3.1. 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	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 14.3.1. 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.

14.2.5 CRA : DTC Transfer Count Register A

Base address: DTCVBR

Offset address: 0x0E + 0x4 × Vector number
(Inaccessible directly from the CPU. See section 14.3.1. 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 14.3.1. 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.

14.2.6 CRB : DTC Transfer Count Register B

Base address: DTCVBR

Offset address: 0x0C + 0x4 × Vector number
(Inaccessible directly from the CPU. See section 14.3.1. 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 14.3.1. Allocating Transfer Information and DTC Vector Table.

14.2.7 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.

14.2.8 DTCVBR : DTC Vector Base Register

Base address: DTC = 0x4000_5400

Offset address: 0x04

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	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.

14.2.9 DTCST : DTC Module Start Register

Base address: DTC = 0x4000_5400

Offset address: 0x0C

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

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

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

For details on these transitions, see section 14.9. Low Power Consumption Function and section 10, Low Power Modes.

14.2.10 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.

14.3 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 31. For an enabled interrupt, the specific DTC interrupt source associated with each interrupt vector number *n* is selected in ICU.IELSRn.IELS[4:0] where n = 0 to 31, as listed in section 12.3.2. Event Number in section 12, Interrupt Controller Unit (ICU). For activation by software, see section 15.2.2. 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.

14.3.1 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.

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 14.2 shows the relationship between the DTC vector table and transfer information. Figure 14.3 shows the allocation of transfer information in the SRAM area.

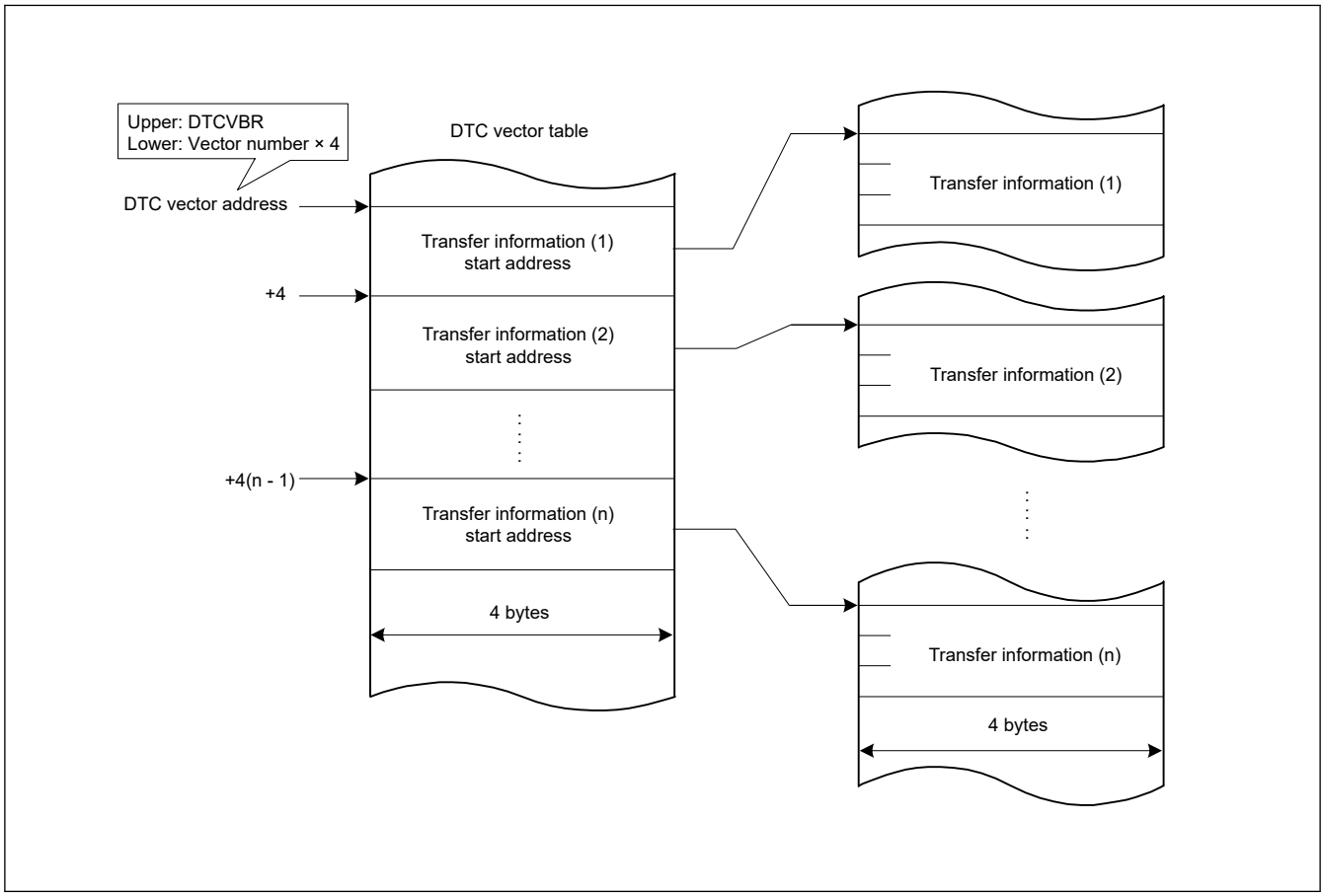


Figure 14.2 DTC vector table and transfer information

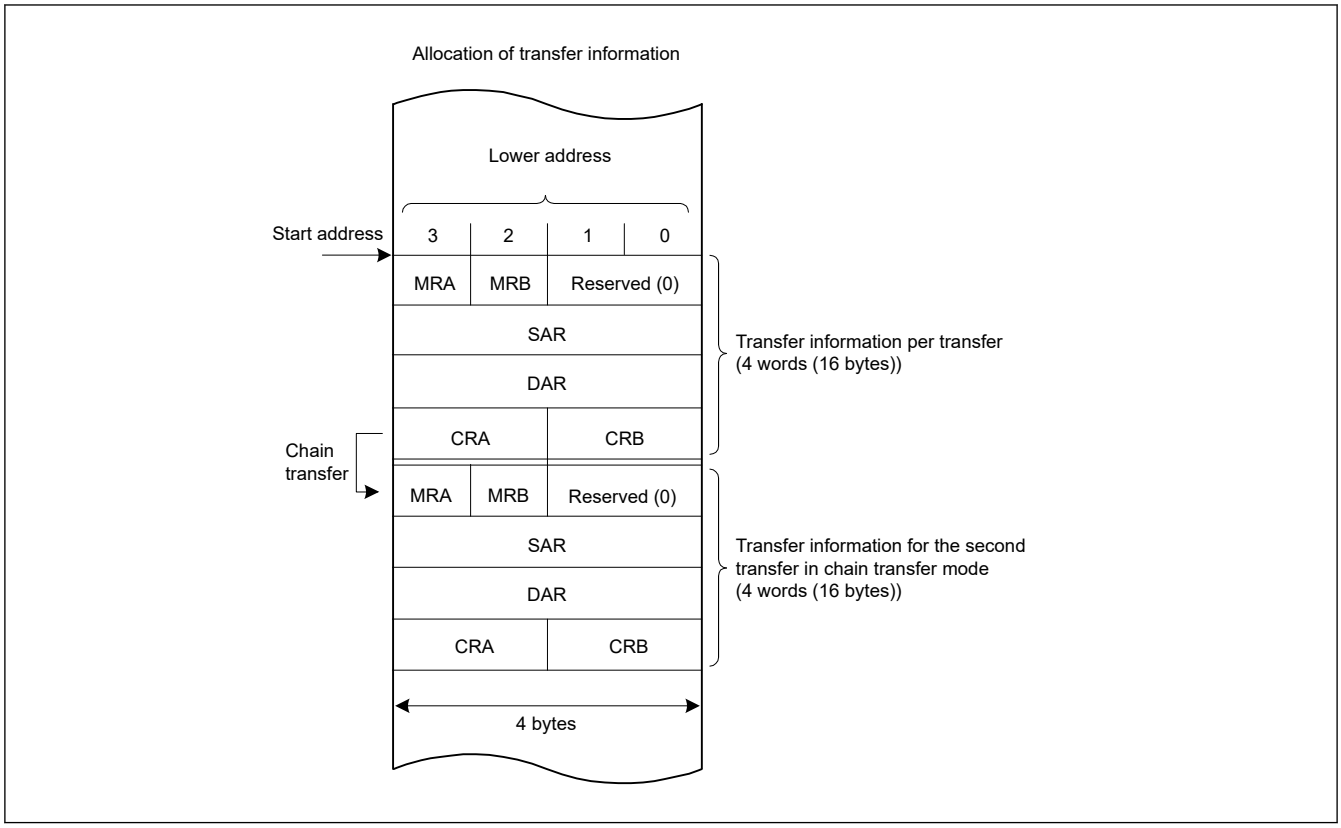


Figure 14.3 Allocation of transfer information in the SRAM area

14.4 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 14.2 describes the DTC transfer modes.

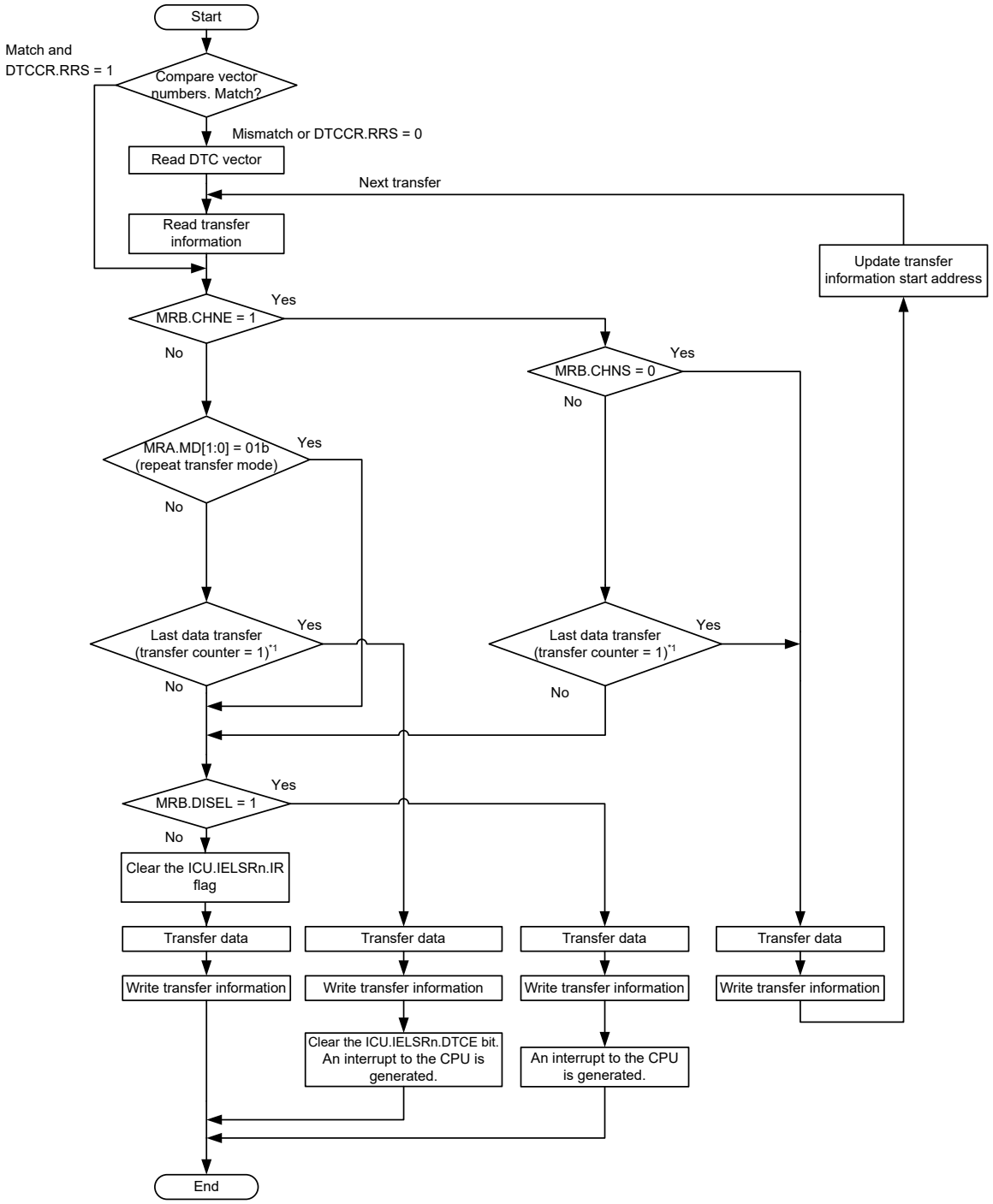
Table 14.2 DTC transfer modes

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 14.4 shows the operation flow of the DTC. Table 14.3 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 14.4 DTC operation flow

Table 14.3 Chain transfer conditions

First transfer				Second transfer*3				DTC transfer
CHNE bit	CHNS bit	DISEL bit	Transfer counter*1 *2	CHNE bit	CHNS bit	DISEL bit	Transfer counter*1 *2	
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.

14.4.1 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 14.12 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.

14.4.2 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 14.4 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 14.4 Transfer information write-back skip conditions and applicable registers

MRA.SM[1:0] bits		MRB.DM[1:0] bits		SAR register	DAR register
b3	b2	b3	b2		
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		

14.4.3 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 14.5 lists register functions in normal transfer mode, and Figure 14.5 shows the memory map of normal transfer mode.

Table 14.5 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.

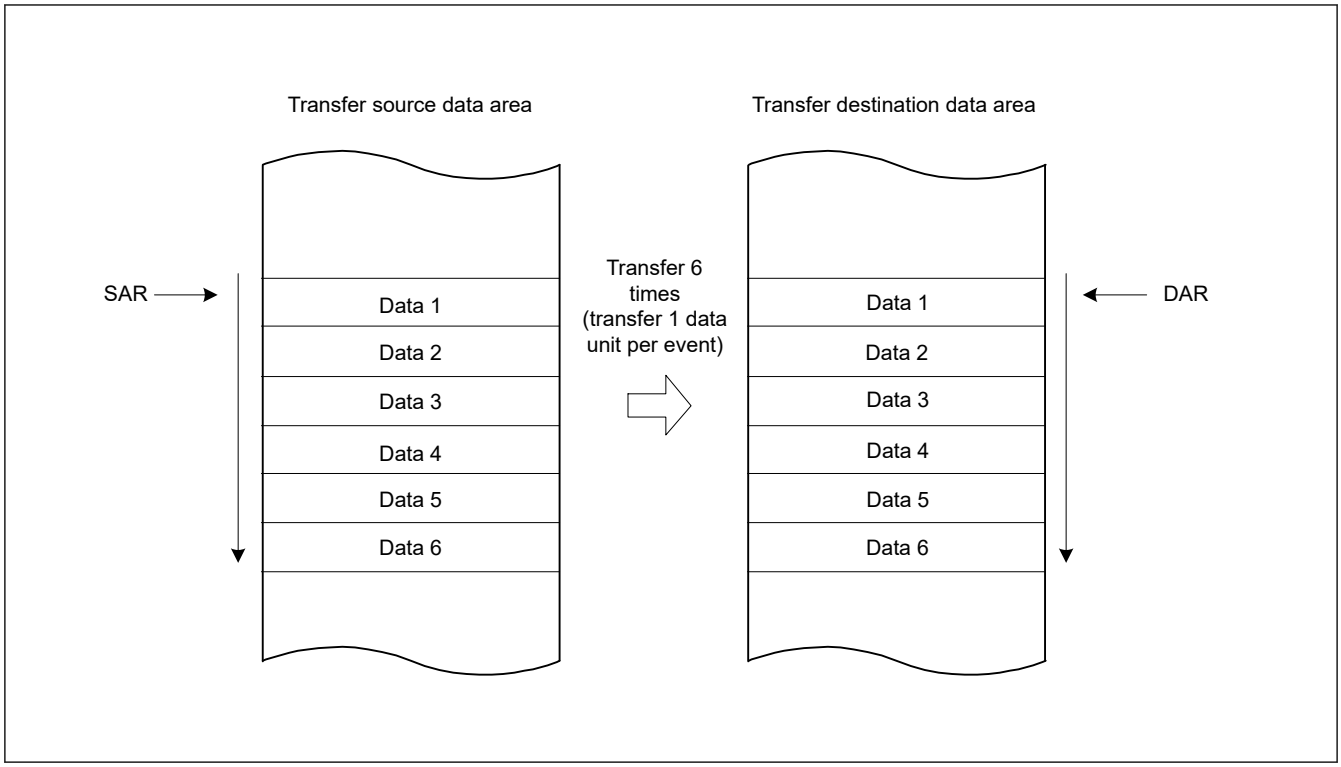


Figure 14.5 Memory map of normal transfer mode (MRA.SM[1:0] = 10b, MRB.DM[1:0] = 10b, CRA = 0x0006)

14.4.4 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 14.6 lists the register functions in repeat transfer mode, and Figure 14.6 shows the memory map of repeat transfer mode.

Table 14.6 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.

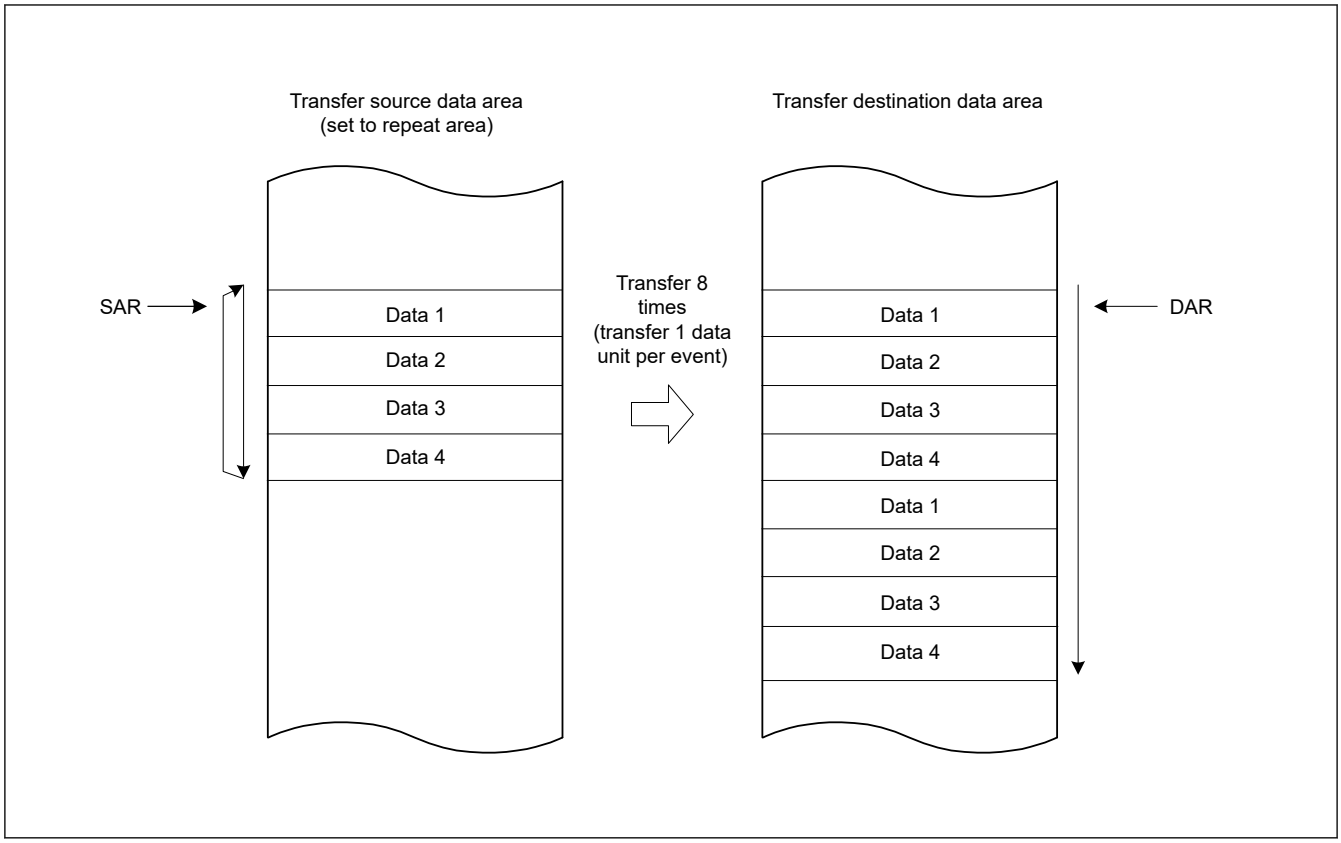


Figure 14.6 Memory map of repeat transfer mode when transfer source is a repeat area (MRA.SM[1:0] = 10b, MRB.DM[1:0] = 10b, CRAH = 0x04)

14.4.5 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 14.7 lists the register functions in block transfer mode, and Figure 14.7 shows the memory map for block transfer mode.

Table 14.7 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.

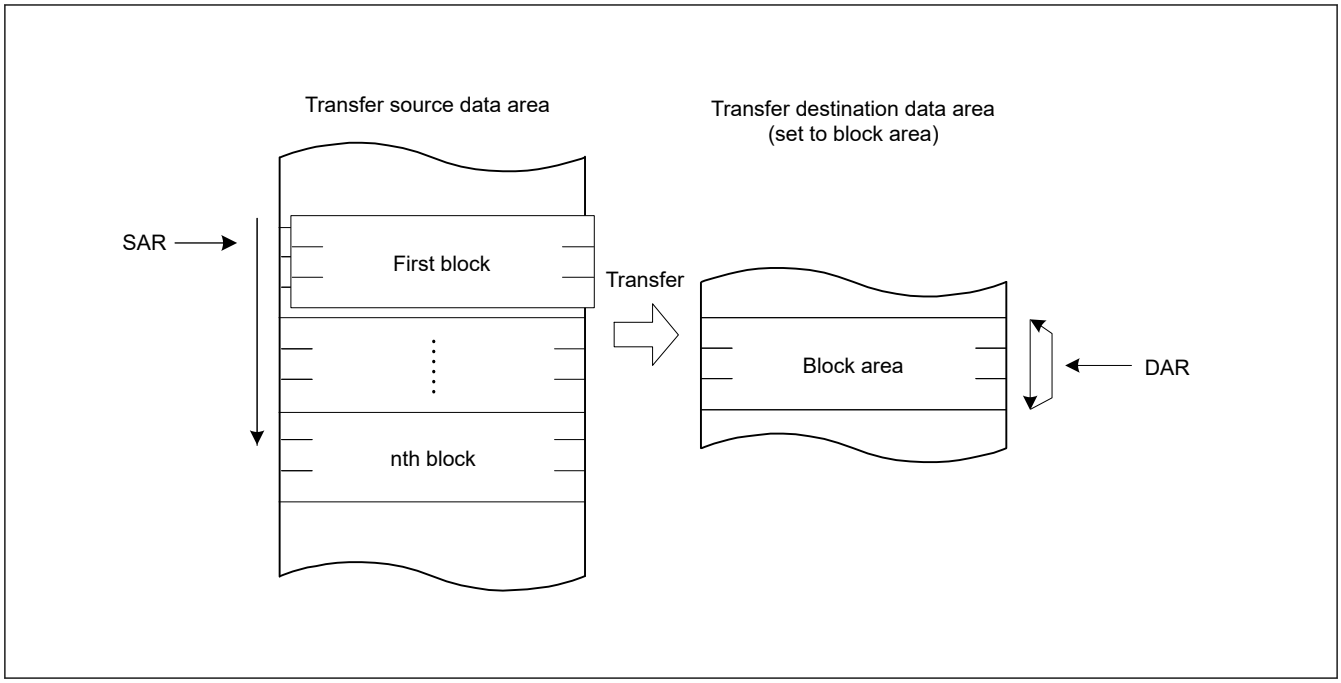


Figure 14.7 Memory map of block transfer mode

14.4.6 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 14.8 shows a chain transfer operation.

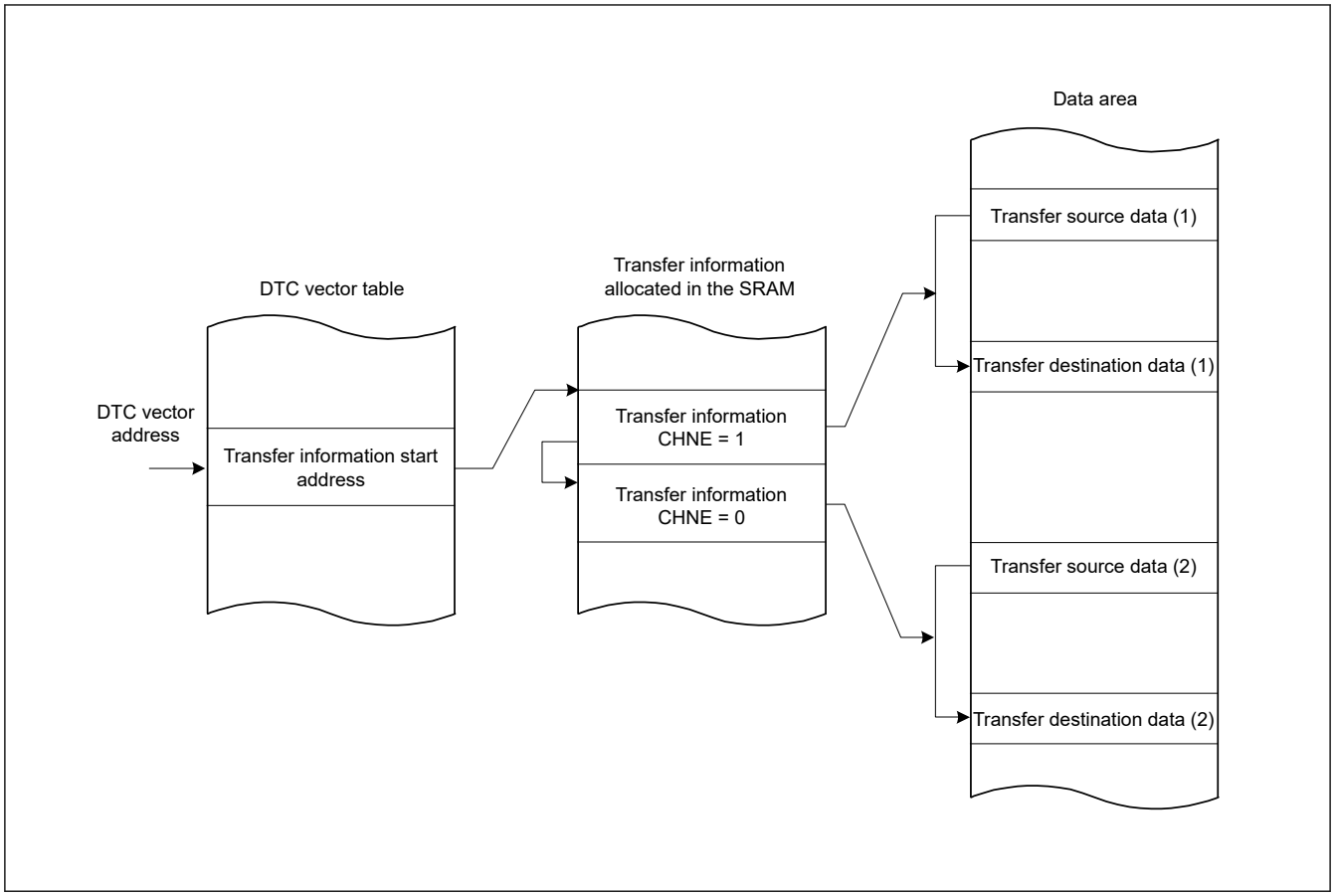


Figure 14.8 Chain transfer operation

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 14.3.

14.4.7 Operation Timing

Figure 14.9 to Figure 14.12 are timing diagrams that show the minimum number of execution cycles.

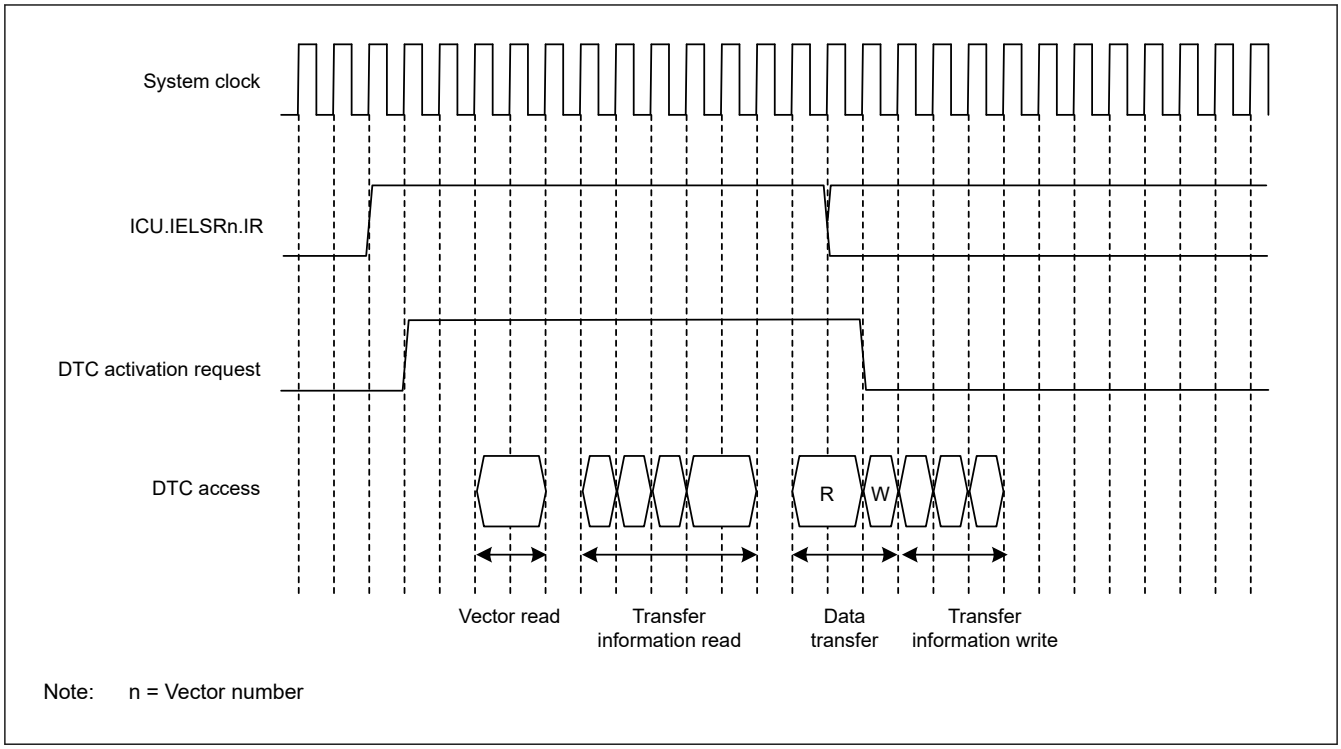


Figure 14.9 Example 1 of DTC operation timing in normal transfer and repeat transfer modes

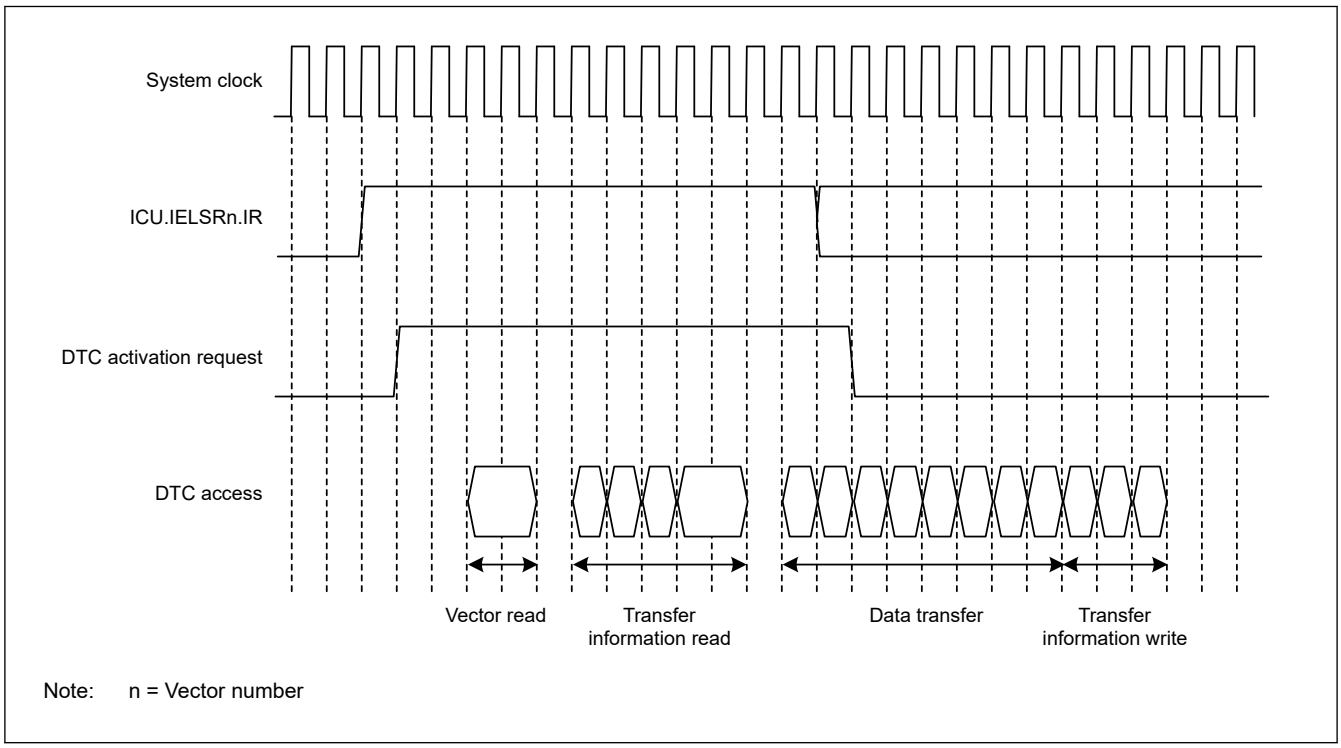
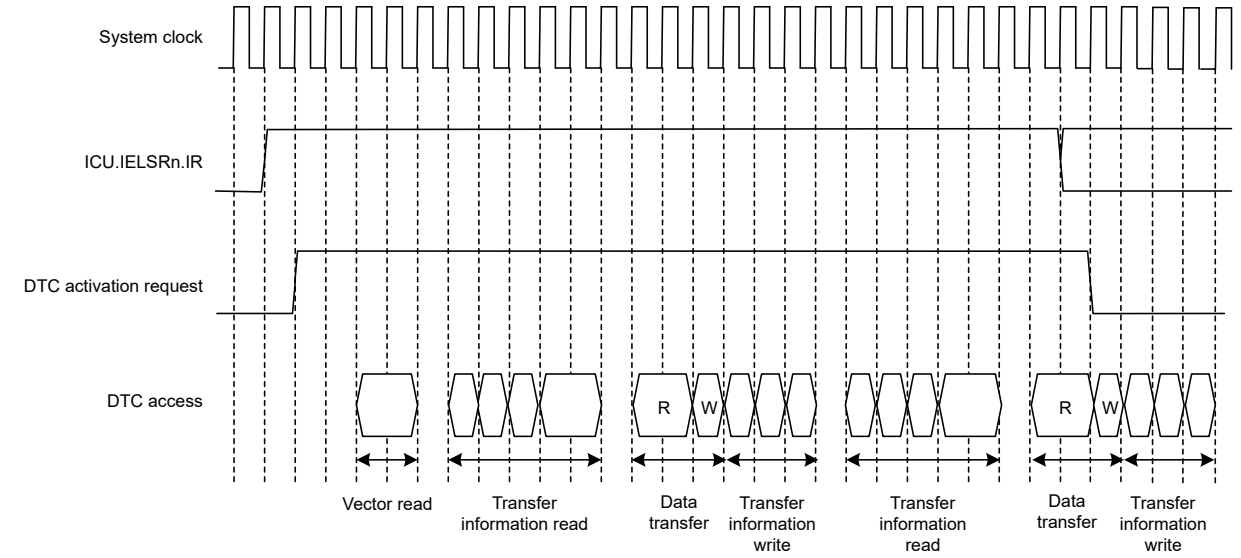
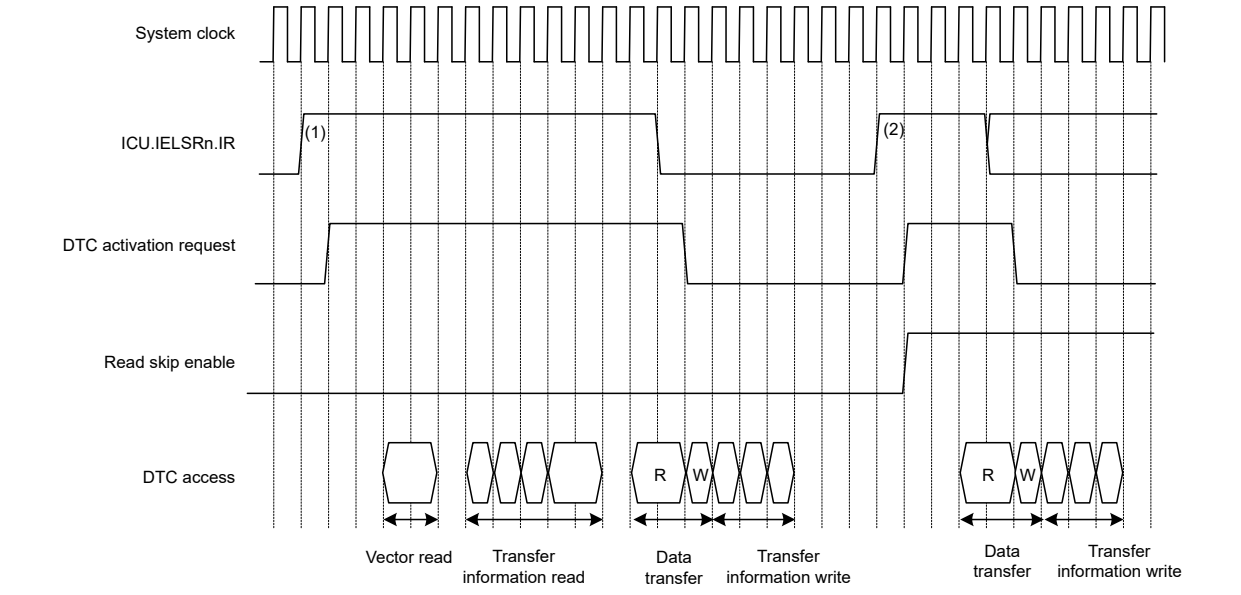


Figure 14.10 Example 2 of DTC operation timing in block transfer mode when the block size = 4



Note: n = Vector number

Figure 14.11 Example 3 of DTC operation timing for chain transfer



Note: n = Vector number
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 14.12 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

14.4.8 Execution Cycles of DTC

Table 14.8 lists the execution cycles of single data transfer of the DTC. For the order of the execution states, see section 14.4.7. Operation Timing.

Table 14.8 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 34, SRAM, section 35, 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 14.8 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.

14.4.9 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.

14.5 DTC Setting Procedure

Before using the DTC, set the DTC Vector Base Register (DTCVBR). Set the ICU.IELSRn.IELS[4:0] bits to 0 to disable the interrupt in the CLIC and follow the procedure in Table 14.9 to set the DTC.

Table 14.9 DTC setting procedure

No.	Step Name	Description
1	Set the DTCCR.RRS bit to 0	Set the DTCCR.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 14.2. Register Descriptions. To allocate transfer information, see section 14.3.1. 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 14.3.1. Allocating Transfer Information and DTC Vector Table.
4	Set the DTCCR.RRS bit to 1	Set the DTCCR.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[4:0] as interrupt source. The interrupt should be enabled in the CLIC.	Set the ICU.IELSRn.DTCE bit to 1. Set ICU.IELSRn.IELS[4:0] as interrupt sources that trigger DTC. The interrupt must be enabled in the CLIC. 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.

14.6 Examples of DTC Usage

14.6.1 Normal Transfer

This section provides an example of DTC usage and its application when receiving 128 bytes of data from an SAU.

(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 and CHNS bits can be set to any value. Set the SDRmn register address of the SAU 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 SAU 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[4:0] as the SAU interrupt. The interrupt must be enabled in the CLIC. Set the DTCST.DTCST bit to 1.

(4) SAU settings

Enable the SAUm_ENDIn interrupt by setting the SMRmn.MD0 bit in the SAU to 0. If a reception error occurs during the SAU 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 SAU is complete, an SAUm_ENDIn interrupt is generated to activate the DTC. The DTC transfers the received byte from the SDRmn register of the SAU 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 SAUm_ENDIn interrupt request is generated for the CPU. Complete the process in the handling routine for this interrupt.

Note: m: Unit number (m = 0, 1), n: Channel number (n = 1, 3), mn = 01, 03, 11

14.6.2 Chain transfer

This section provides an example of chain transfer by the DTC and describes its use in the output of pulses by the Timer Array Unit (TAU). You can use chain transfer to transfer compare data and change the period of the multiple PWM output for the TAU.

For the first of the chain transfers, normal transfer mode is specified for transfer to the TDR0n register. For the second transfer, normal transfer mode is specified for transfer to the TDR0p register. For the third transfer of the chained transfer, normal transfer mode for transfer to the TDR0q register 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 timer interrupt with the TDR0n register as an activating source for the DTC.

(1) First transfer information setting

Set up transfer to the TDR0n 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 halfword-sized transfer (MRA.SZ[1:0] = 01b).
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 TDR0n 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 TDR0p 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 halfword-sized transfer (MRA.SZ[1:0] = 01b).
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 TDR0p 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 TDR0q 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 halfword-sized transfer (MRA.SZ[1:0] = 01b).
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 TDR0q 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 TDR0q register immediately after the transfer control information for use in the TDR0n and TDR0p registers.

(5) DTC vector table

In the DTC vector table, set the address where the transfer control information for use in transfer to the TDR0n and TDR0p registers starts.

(6) ICU setting and DTC module activation

1. Set the ICU.IELSRx.DTCE bit associated with the timer interrupt (TAU0_ENDIn).
2. Set the ICU.IELSRx.IELS[4:0] bits and specify the timer interrupt (TAU0_ENDIn).
3. Set the DTCST.DTCST bit to 1.

(7) TAU settings

For details, see section 18.8.3. Operation for the Multiple PWM Output Function.

1. Set the TOE0 and TOM0 registers so that the TDR0p and TDR0q registers operate as output compare registers.
2. Set the default compare values in the TDR0p and TDR0q registers.
3. Set the default period values in the TDR0n register.
4. Set 1 to the output bit in PghPFS.PDR, and set 11001b to the Peripheral Select bits in PghPFS.PSEL[4:0].

(8) TAU activation

Set the TS0.TS[n], TS[p], and TS[q] bits to 1 to start the TCR0n, TCR0p, and TCR0q counter.

(9) DTC transfer

Each time a timer interrupt (TAU0_ENDIn) is generated with the TDR0n register, the next compare values are transferred to the TDR0p and TDR0q registers.

(10) Interrupt handling

After the specified rounds of data transfer are complete, for example when the value in the CRA register for TAU transfer becomes 0, a TAU timer interrupt request is issued for the CPU. Complete the process for this interrupt in the handling routine.

14.6.3 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 14.13 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.

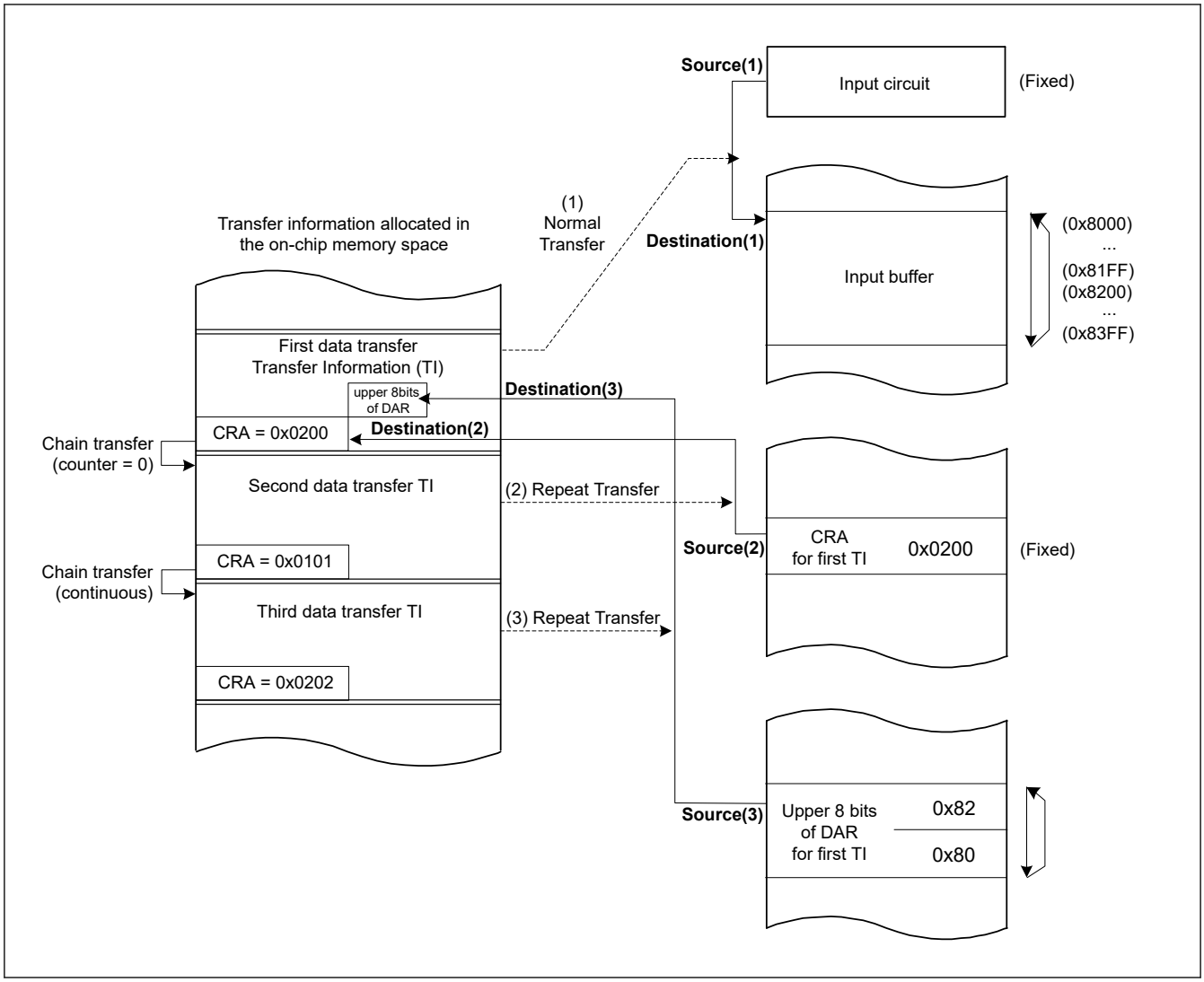


Figure 14.13 Chain transfer when counter = 0

14.7 Interrupt

14.7.1 Interrupt Sources

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 CLIC and the ICU.IELSRn.IELS[4:0] 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 CLIC priority.

14.8 Event Link

The DTC can produce an event link request on completion of one transfer request.

14.9 Low Power Consumption Function

Before transitioning to the module-stop state, or Software Standby mode, set the DTCST.DTCST bit to 0, and then perform the operations described in the following sections. 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

Use the settings described in section 10.7.1. Transition to Software Standby Mode.

If DTC transfer operations are in progress when the WFI instruction is executed, the transition to Software Standby mode is executed after the completion of the DTC transfer.

(3) 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, 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 12.4.2. Detecting Interrupts, then execute the WFI instruction.

14.10 Usage Notes

14.10.1 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.

15. Event Link Controller (ELC)

15.1 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 15.1 lists the ELC specifications, and Figure 15.1 shows a block diagram.

Table 15.1 ELC specifications

Item	Description
Event link function	52 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.

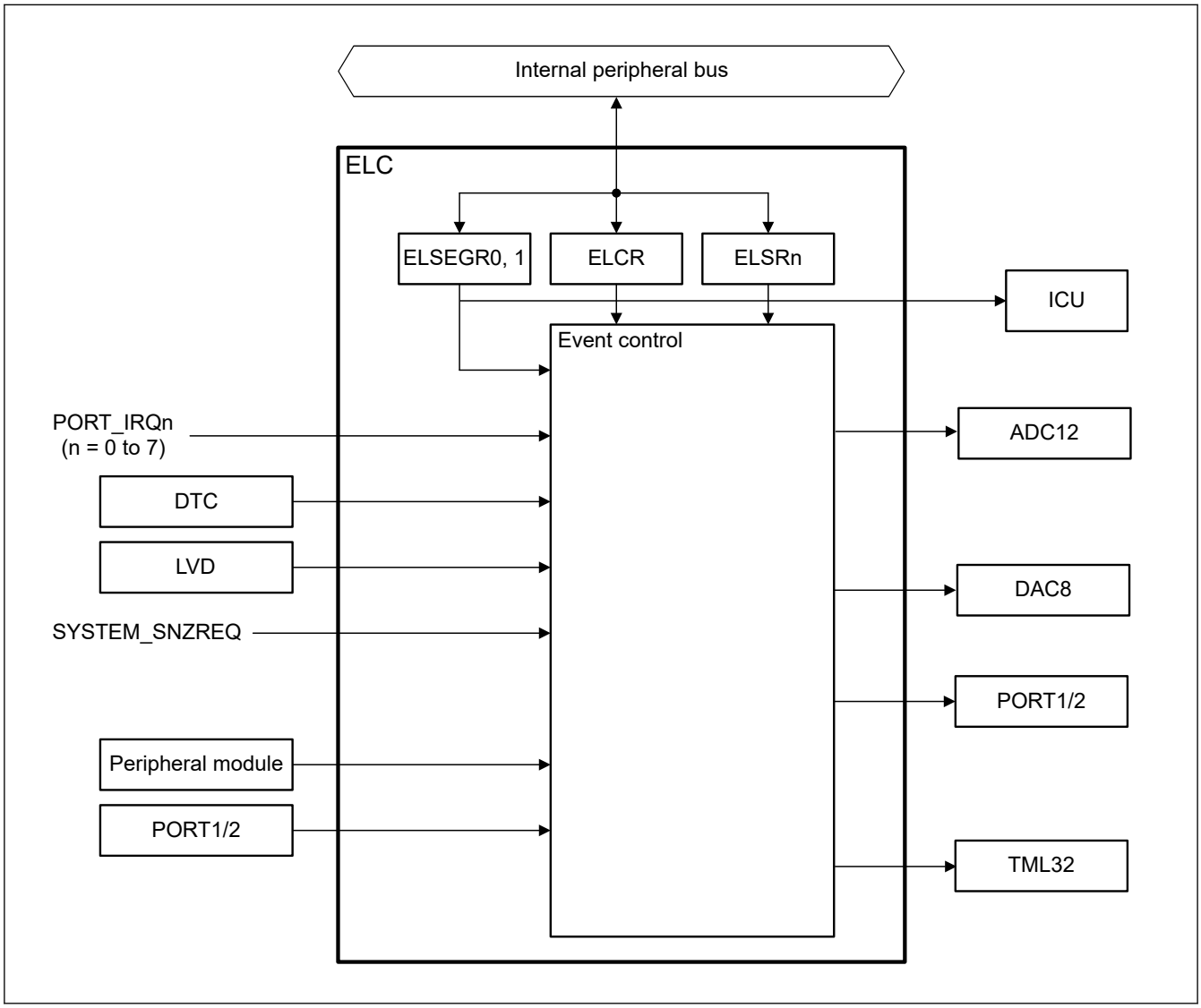


Figure 15.1 ELC block diagram

15.2 Register Descriptions

15.2.1 ELCR : Event Link Controller Register

Base address: ELC = 0x4004_1000

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

The ELCR register controls the ELC operation.

15.2.2 ELSEGRn : Event Link Software Event Generation Register n (n = 0, 1)

Base address: ELC = 0x4004_1000

Offset address: 0x02 + 0x02 × 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

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.

15.2.3 ELSRn : Event Link Setting Register n (n = 8, 14, 15, 19, 20, 23)

Base address: ELC = 0x4004_1000

Offset address: 0x10 + 0x04 × n

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	ELS[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	ELS[7:0]	Event Link Select 0x00: Event output disabled for the associated peripheral module 0x01: Number setting for the event signal to be linked ⋮ ⋮ 0xAA: Number setting for the event signal to be linked Others: Settings prohibited	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W

The ELSRn register specifies an event signal to be linked to each peripheral module. Table 15.2 shows the association between the ELSRn register and the peripheral modules. Table 15.3 shows the association between the event signal names set in the ELSRn register and the signal numbers.

Table 15.2 Association between the ELSRn registers and peripheral functions

Register name	Peripheral function (module)	Event name
ELSR8	ADC12	ELC_AD00
ELSR14	PORT1	ELC_PORT1
ELSR15	PORT2	ELC_PORT2
ELSR19	DAC8ch1	ELC_DAC8ch1
ELSR20	DAC8ch0	ELC_DAC8ch0
ELSR23	TML32	ELC_TML32

Table 15.3 Association between event signal names set in ELSRn.ELS[7:0] bits and signal numbers (1 of 3)

Event number	Interrupt request source	Name	Description
0x01	Port	PORT_IRQ0*1	External pin interrupt 0
0x02		PORT_IRQ1*1	External pin interrupt 1
0x03		PORT_IRQ2*1	External pin interrupt 2
0x04		PORT_IRQ3*1	External pin interrupt 3
0x05		PORT_IRQ4*1	External pin interrupt 4
0x06		PORT_IRQ5*1	External pin interrupt 5
0x07		PORT_IRQ6*1	External pin interrupt 6
0x08		PORT_IRQ7*1	External pin interrupt 7
0x0A	DTC	DTC0_DTCEND*3	DTC transfer end
0x0D	LVD	LVD_LVD1	Voltage monitor 1 interrupt
0x0E		LVD_LVD2	Voltage monitor 2 interrupt
0x10	Low Power Mode	SYSTEM_SNZREQ*2 *3	Snooze entry

Table 15.3 Association between event signal names set in ELSRn.ELS[7:0] bits and signal numbers (2 of 3)

Event number	Interrupt request source	Name	Description
0x11	IWDT	IWDT_NMIUNDF	IWDT underflow
0x12	WDT	WDT_NMIUNDF	WDT underflow
0x13	RTC	RTC_ALM_OR_PRD	Alarm, fixed-period interrupt signal
0x2B	IICA	IIC0_ENDI/IIC0_WUI	IICA0 communication complete/address match wakeup
0x2C		IIC1_ENDI/IIC1_WUI	IICA1 communication complete/address match wakeup
0x34	I/O Port	IOPORT_GROUP1	Port 1 event
0x35		IOPORT_GROUP2	Port 2 event
0x39	ELC	ELC_SWEVT0	Software event 0
0x3A		ELC_SWEVT1	Software event 1
0x3B	SAU	SAU0_ENDI0	UART0 transmission transfer end or buffer empty interrupt/SPI00 transfer end or buffer empty interrupt/IIC00 transfer end
0x3C		SAU0_ENDI1	UART0 reception transfer end/SPI01 transfer end or buffer empty interrupt/IIC01 transfer end
0x3D		SAU0_ENDI2	UART1 transmission transfer end or buffer empty interrupt/SPI10 transfer end or buffer empty interrupt/IIC10 transfer end
0x3E		SAU0_ENDI3	UART1 reception transfer end/SPI11 transfer end or buffer empty interrupt/IIC11 transfer end
0x3F		SAU1_ENDI0	UART2 transmission transfer end or buffer empty interrupt/SPI20 transfer end or buffer empty interrupt/IIC20 transfer end
0x40		SAU1_ENDI1	UART2 reception transfer end/SPI21 transfer end or buffer empty interrupt/IIC21 transfer end
0x43		SAU0_INTSRE0	UART0 reception communication error occurrence
0x44		SAU0_INTSRE1	UART1 reception communication error occurrence
0x45		SAU1_INTSRE2	UART2 reception communication error occurrence

Table 15.3 Association between event signal names set in ELSRn.ELS[7:0] bits and signal numbers (3 of 3)

Event number	Interrupt request source	Name	Description
0x47	TAU	TAU0_ENDI0	Interrupt when counting/capturing is completed
0x48		TAU0_ENDI1	Interrupt when counting/capturing is completed
0x49		TAU0_ENDI2	Interrupt when counting/capturing is completed
0x4A		TAU0_ENDI3	Interrupt when counting/capturing is completed
0x4B		TAU0_ENDI4	Interrupt when counting/capturing is completed
0x4C		TAU0_ENDI5	Interrupt when counting/capturing is completed
0x4D		TAU0_ENDI6	Interrupt when counting/capturing is completed
0x4E		TAU0_ENDI7	Interrupt when counting/capturing is completed
0x4F		TAU0_MODE8_ENDI1	Interrupt when counting/capturing is completed in channel 1 when 8 bit mode is selected
0x50		TAU0_MODE8_ENDI3	Interrupt when counting/capturing is completed in channel 3 when 8 bit mode is selected
0x5B	TML32	TML32_OUTI	Interrupt output
0x5D	UARTA	UARTA_TX_ENDI0	Transmit transfer complete, buffer empty interrupt
0x5E		UARTA_RX_ENDI0	Receive transfer complete interrupt
0x5F		UARTA_RX_ERI0	Receive transfer complete communication error generation interrupt
0x60		UARTA_TX_ENDI1	Transmit transfer complete, buffer empty interrupt
0x61		UARTA_RX_ENDI1	Receive transfer complete interrupt
0x62		UARTA_RX_ERI1	Receive transfer complete communication error generation interrupt
0x63	CMP	COMP_DET0	Comparator detection 0
0x64		COMP_DET1	Comparator detection 1
0x65	ADC12	ADC_ENDI	A/D conversion end
0x69	DOC	DOC_DOPCI*3	Data operation circuit interrupt

Note 1. Only pulse (edge detection) is supported.
Note 2. ELSR8, 14, 15, 23 can select this event.
Note 3. This event occurs in Snooze mode.

15.3 Operation

15.3.1 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.

15.3.2 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 15.4 lists the operations of modules when an event occurs.

Table 15.4 Module operations when event occurs

Module	Operations when event is input
ADC12	Start A/D 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 1 Port 2
TML32	Start timer operation
DAC8	Start D/A conversion

15.3.3 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 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 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.

15.4 Usage Notes

15.4.1 Linking DTC Transfer End Signals as Events

When linking the DTC transfer end signals as events, do not set the same peripheral module as the DTC transfer destination and event link destination. If set, the peripheral module might be started before DTC transfer to the peripheral module is complete.

15.4.2 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).

Some modules can perform in Snooze mode. For more information, see Table 15.3 and section 10, Low Power Modes.

15.4.3 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 15.3 and section 10, Low Power Modes.

15.4.4 ELC Delay Time

In Figure 15.2, module A accesses module B through the ELC. There is a delay time in the ELC between module A and module B. Table 15.5 shows the ELC delay time.

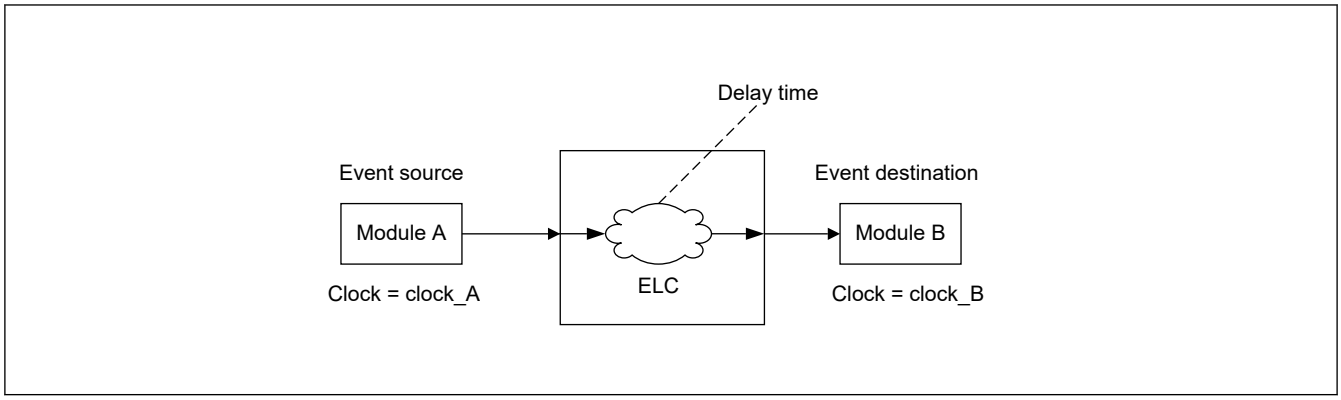


Figure 15.2 ELC delay time

Table 15.5 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

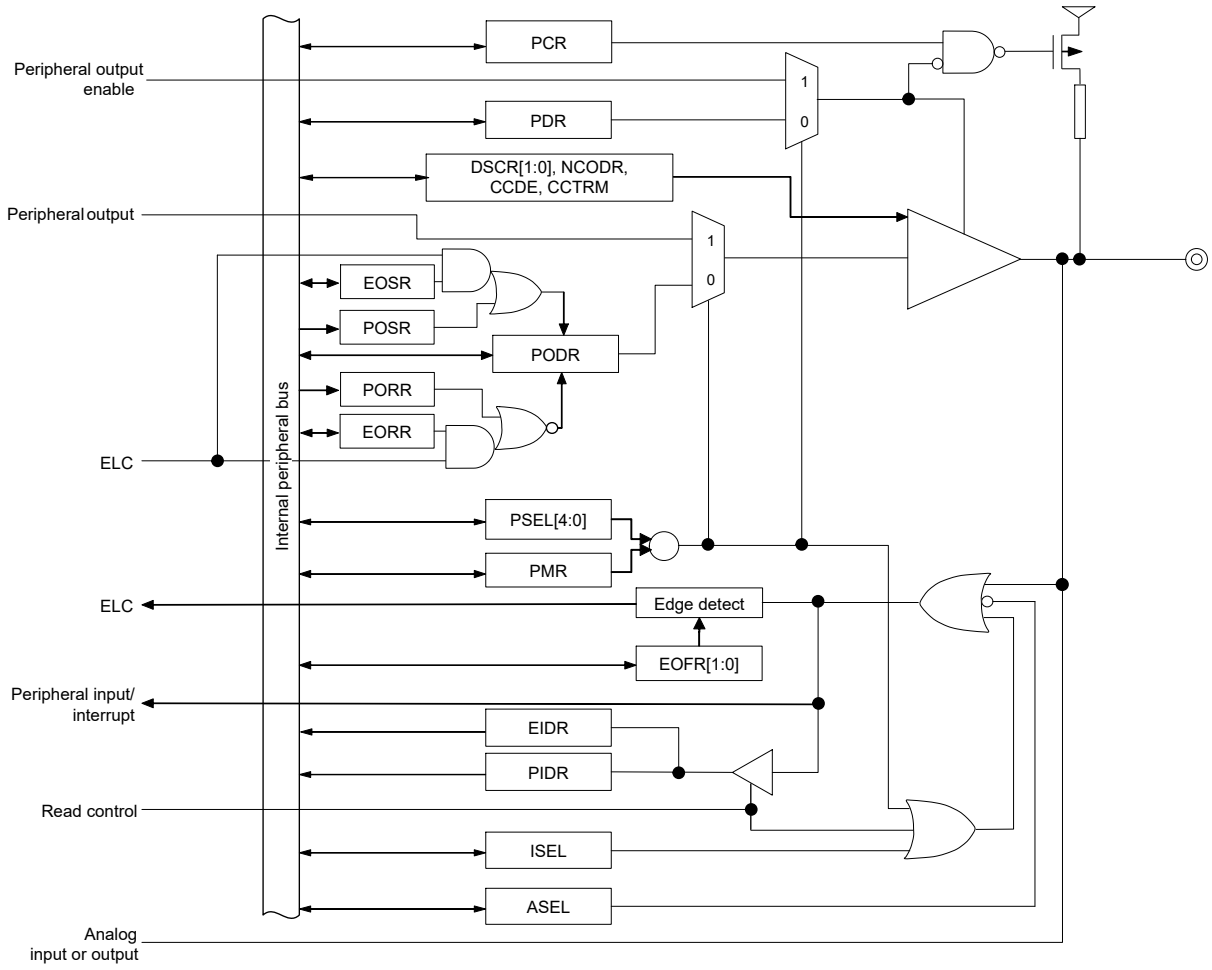
16. I/O Ports

16.1 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, port group function for the ELC.

All pins 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 16.1 shows a connection diagram for the I/O port registers. The configuration of the I/O ports differs depending on the package. Table 16.1 lists the I/O port specifications by package, and Table 16.2 lists the port functions.



Note: This figure shows a basic port configuration. The configuration differs depending on the ports.

Figure 16.1 Connection diagram for I/O port registers

Table 16.1 shows the I/O port specifications and Table 16.2 shows the port functions.

Table 16.1 I/O port specifications (1 of 2)

Port	Package		Package		Package		Package	
	48 QFN	Number of pins	32 QFN	Number of pins	24 QFN	Number of pins	16 WLCSP	Number of pins
Port 0	P000 to P003, P006 to P011	10	P000 to P003, P006 to P007	6	P000 to P003	4	P000 to P003	4
Port 1	P100 to P111	12	P100 to P111	12	P100 to P107	8	P100, P107	2

Table 16.1 I/O port specifications (2 of 2)

Port	Package 48 QFN	Number of pins	Package 32 QFN	Number of pins	Package 24 QFN	Number of pins	Package 16 WLCSP	Number of pins
Port 2	P200 to P207	8	P200 to P203	4	P200, P203	2	P200, P203	2
Port 3	P300 to P307	8	P300 to P303	4	P300 to P303	4	P300 to P303	4
Port 4	P400 to P403	4	—	—	—	—	—	—

Table 16.2 I/O port functions

Port	Port name	Current control	Input pull-up	Open drain output	5V-tolerant	I/O
Port 0	P000 to P001	—	✓	✓	—	Input/Output
	P002 to P003, P006 to P007	—	✓	—	—	Input/Output
	P008 to P009	—	✓	✓	—	Input/Output
	P010 to P011	—	✓	✓	✓	Input/Output
Port 1	P100	✓	✓	✓	—	Input/Output
	P101 to P103	—	✓	✓	✓	Input/Output
	P104 to P111	—	✓	✓	—	Input/Output
Port 2	P200	—	—	—	—	Input
	P201 to P207	—	✓	✓	—	Input/Output
Port 3	P300	—	✓	✓	—	Input/Output
	P301	—	✓	✓	—	Input/Output
	P302 to P303	✓	✓	✓	—	Input/Output
	P304 to P308	—	✓	✓	—	Input/Output
Port 4	P400 to P401	—	✓	—	—	Input/Output
	P402 to P403	—	✓	✓	—	Input/Output

Note: ✓: Available
—: Setting prohibited

16.2 Register Descriptions

16.2.1 PCNTR1/PODR/PDR : Port Control Register 1

Base address: PORTi = 0x4004_0000 + 0x0020 × i (i = 0, 3, 4)
PORTj = 0x4004_0000 + 0x0020 × j (j = 1, 2)

Offset address: 0x000 (PCNTR1/PODR)
0x002 (PDR)

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	Pin and Pjn Direction 0: Input (functions as an input pin) 1: Output (functions as an output pin)	R/W
31:16	PODR15 to PODR00	Pin and Pjn Output Data 0: Low output 1: High output	R/W

Note: i = 0, 3 to 4, j = 1 to 2, n = 00 to 15

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 (Pin and Pjn 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 PORTi.PCNTR1.PDRn and PORTj.PCNTR1.PDRn bits. The I/O direction can be specified in 1-bit units. Bits associated with non-existent pins are reserved. Reserved bits are read as 0. The write value should be 0. P200 is input-only ports, PDRn bits are reserved. The PDRn bit in the PORTi.PCNTR1 and PORTj.PCNTR1 register serves the same function as the PDR bit in the PFS.PmnPFS register.

PODRn bits (Pin and Pjn 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. P200 is input-only ports, PODRn bits are reserved. The PODRn bit in the PORTi.PCNTR1 and PORTj.PCNTR1 registers serves the same function as the PODR bit in the PFS.PmnPFS register.

16.2.2 PCNTR2/EIDR/PIDR : Port Control Register 2

Base address: PORTi = 0x4004_0000 + 0x0020 × i (i = 0, 3, 4)
PORTj = 0x4004_0000 + 0x0020 × j (j = 1, 2)

Offset address: 0x004 (PCNTR2/EIDR)
0x006 (PIDR)

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	Pin and Pjn State 0: Low level 1: High level	R
31:16	EIDR15 to EIDR00	Port Event Input Data*1 When an ELC_PORTj signal occurs: 0: Low input 1: High input	R

Note: i = 0, 3 to 4, j = 1 to 2, n = 00 to 15
Note 1. • j = 1 to 2 for EIDR only
• Supported for ports 1 and 2 (j = 1 to 2)

The Port Control Register 2 (PCNTR2/EIDR/PIDR) allows read access to the Pmn state using 32-bit or 16-bit access.

The PCNTR2 specifies the Pin and Pjn 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 (Pin and Pjn State)

The PIDRn bits reflect the individual pin states of the port, regardless of the values set in PmnPFS.PMR, PORTi.PCNTR1.PDRn and PORTj.PCNTR1.PDRn. The PIDRn bit in the PORTi.PCNTR2 and PORTj.PCNTR2 registers serves the same function as the PIDR bit in the PFS.PmnPFS register.

A pin state cannot be reflected in PIDRn when one of the following functions is enabled:

- Sub-clock oscillator (SOSC)
- Analog function (ASEL = 1)

EIDRn bits (Port Event Input Data)

The EIDRn bits latch a pin state when an ELC_PORTj signal occurs. Pin states can only be input to EIDRn when PmnPFS.PMR and PORTj.PCNTR1.PDRn are 0.

When the PmnPFS.ASEL bit is set to 1, the associated pin state is not reflected in EIDRn.

16.2.3 PCNTR3/PORR/POSR : Port Control Register 3

Base address: PORTi = 0x4004_0000 + 0x0020 × i (i = 0, 3, 4)
PORTj = 0x4004_0000 + 0x0020 × j (j = 1, 2)

Offset address: 0x008 (PCNTR3/PORR)
0x00A (POSR)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	PORR15	PORR14	PORR13	PORR12	PORR11	PORR10	PORR09	PORR08	PORR07	PORR06	PORR05	PORR04	PORR03	PORR02	PORR01	PORR00
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:	POSR15	POSR14	POSR13	POSR12	POSR11	POSR10	POSR09	POSR08	POSR07	POSR06	POSR05	POSR04	POSR03	POSR02	POSR01	POSR00
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	Pin and Pjn Output Set 0: No effect on output 1: High output	W
31:16	PORR15 to PORR00	Pin and Pjn Output Reset 0: No effect on output 1: Low output	W

Note: i = 0, 3 to 4, j = 1 to 2, 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 (Pin and Pjn Output Set)

POSR changes PODR when set by a software write. For example, for P100, when PORT1.PCNTR3.POSR00 = 1, PORT1.PCNTR1.PODR00 outputs 1. Bits associated with non-existent pins are reserved. The write value should always be 0. P200 is input-only port, PORRn bits are reserved.

PORRn bits (Pin and Pjn Output Reset)

PORR changes PODR when reset by a software write. For example, for P100, when PORT1.PCNTR3.POSR00 = 1, PORT1.PCNTR1.PODR00 outputs 0. Bits associated with non-existent pins are reserved. The write value should always be 0. P200 is input-only port, PORRn bits are reserved.

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.

16.2.4 PCNTR4/EORR/EOSR : Port Control Register 4

Base address: PORTj = 0x4004_0000 + 0x0020 × j (j = 1, 2)

Offset address: 0x00C (PCNTR4/EORR)
0x00E (EOSR)

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	Pin and Pjn Event Output Set When an ELC_PORTj signal occurs 0: No effect on output 1: High output	R/W
31:16	EORR15 to EORR0	Pin and Pjn Event Output Reset When an ELC_PORTj signal occurs 0: No effect on output 1: Low output	R/W

Note: j = 1, 2, n = 00 to 15

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 (Pin and Pjn Event Output Set)

EOSR changes PODR when set because an ELC_PORTj signal occurs. For example, for P100 if PORT1.PCNTR4.EOSR00 is set to 1 when the ELC_PORTj occurs, PORT1.PCNTR1.PODR00 outputs 1. Bits associated with non-existent pins are reserved. The write value should always be 0. P200 is input only, so PORT2.PCNTR4.EOSR00 bit is reserved.

EORRn bits (Pin and Pjn Event Output Reset)

EORR changes PODR when reset because an ELC_PORTj signal occurs. For example, for P100 if PORT1.PCNTR4.EORR00 = 1 when the ELC_PORTj occurs, PORT1.PCNTR1.PODR00 outputs 0. Bits associated with non-existent pins are reserved. The write value should always be 0. P200 is input only, so PORT2.PCNTR4.EORR00 bit is reserved.

Note: When EORRn or EOSRn is set, writing to PODRn, PORRn, and POSRn is prohibited.

Note: EORRn and EOSRn should not be set at the same time.

16.2.5 PmnPFS/PmnPFS_HA/PmnPFS_BY : Port mn Pin Function Select Register (m = 0 to 4, n = 00 to 15)

Base address: PFS = 0x4004_0800

Offset address: 0x000 + 0x040 × m + 0x004 × n (PmnPFS)
0x002 + 0x040 × m + 0x004 × n (PmnPFS_HA)
0x003 + 0x040 × m + 0x004 × n (PmnPFS_BY)

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	0	0	0	0	0	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
1	PIDR	Pmn State 0: Low level 1: High level	R
2	PDR	Port Direction 0: Input (functions as an input pin) 1: Output (functions as an output pin)	R/W
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
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
9:7	—	These bits are read as 0. The write value should be 0.	R/W
11:10	DSCR[1:0]	Port Output Current Select*3 0 0: 2 mA 0 1: 5 mA 1 x: 10 mA	R/W
13:12	EOFR[1:0]	Event on Falling (EOF)/Event on Rising (EOR)*2 0 0: Don't care 0 1: Detect rising edge 1 0: Detect falling edge 1 1: Detect both edges	R/W
14	ISEL	IRQ Input Enable 0: Not used as an IRQn input pin 1: Used as an IRQn input pin	R/W
15	ASEL	Analog Input Enable 0: Not used as an analog pin 1: Used as an analog pin	R/W
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
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
31:29	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. The initial value of P203, P300 to P303 is not 0x00000000. The initial value of P203 is 0x00000010, P300, P301 and P303 are 0x00010010, and P302 is 0x00010000.
Note 2. Supported for port 1 and port 2.
Note 3. Support for P100, P302, and P303 only when the port selected bits in CCDE is enabled. See section 16.2.8. CCDE : Output Current Control Enable Register.

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.

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 (except for P203, P300, P301, and P303). 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 Output Current Select)

The DSCR[1:0] bits are used to set the output currents of the port pins selected as output current control pin in the Output Current Control Enable register (CCDE) or to place them in the high-impedance state. For current control, the pins are at the low level and produce output currents controlled at 2 mA, 5 mA, or 10 mA.

EOFR[1:0] bits (Event on Falling (EOF)/Event on Rising (EOR))

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 CPU. 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. This setting can be used in combination with the peripheral functions, although an IRQn (external pin interrupt) of the same number must only be enabled for one pin.

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 ISEL bit for an unspecified IRQn is reserved. 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. For details on the peripheral settings for each product, see section 16.6. Peripheral Select Settings for Each Product.

16.2.6 PWPR : Write-Protect Register

Base address: PFS = 0x4004_0800

Offset address: 0x503

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.

16.2.7 PRWCNTR : Port Read Wait Control Register

Base address: PFS = 0x4004_0800

Offset address: 0x50F

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

Bit	Symbol	Function	R/W
1:0	WAIT[1:0]	Wait Cycle Control 0 0: Setting prohibited 0 1: Insert a 1-cycle wait 1 0: Insert a 2-cycle wait 1 1: Insert a 3-cycle wait	W
7:2	—	The write value should be 0.	W

WAIT[1:0] bits (Wait Cycle Control)

The WAIT[1:0] bits specify the number of wait cycles for accessing the PCNTR2 and PFS registers.

16.2.8 CCDE : Output Current Control Enable Register

Base address: PORGA = 0x4009_1000

Offset address: 0x48

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	CCDE02	CCDE01	CCDE00
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CCDE00	CCDE00 (P100) output control function 0: Disable (digital I/O) 1: Enable (current control function)	R/W
1	CCDE01	CCDE01 (P302) output control function 0: Disable (digital I/O) 1: Enable (current control function)	R/W
2	CCDE02	CCDE02 (P303) output control function 0: Disable (digital I/O) 1: Enable (current control function)	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

The CCDE register is an 8-bit read/write register that controls setting or resetting of the port output current control.

CCDE0n bit (Port output control function)

The CCDE0n bit is used to specify the use of P100, P302, and P303 as output current control port pins in 1-bit units. The setting of the corresponding port Output Current Control register (PFS.DSCR[1:0]) controls a pin for which this function is selected to be at the low level and produces the selected output current (low-level output current) or to be in the high impedance state.

Note:
- When a port is used as output current control pin, set the output current control function and then set the corresponding bit in the PFS.PDR register to output mode.
- The state of a pin takes 10 µs to become stable after 1 is written to the corresponding bit of the CCDE register.

16.2.9 CCTRM : Output Current Control Trimming Register

Base address: PORGA = 0x4009_1000

Offset address: 0x81

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

Bit	Symbol	Function	R/W
3:0	IADJ[3:0]	Output current control trimming These bits adjust the output current of current control function port.	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W

The CCTRM register is an 8-bit read/write register that controls setting or resetting of the port output current.

IADJ[3:0] bits (Output current control trimming)

The IADJ[3:0] bits adjust or fine tune the current of all current control support pins (P100, P302, and P303) at the same time when used as output current control port to match the selected specification. Each port output current depends on the setting of the corresponding port Output Current Control register (PFS.DSCR[1:0]) when CCDE.CCDE0n bit is 1.

Table 16.3 shows the approximate output current at each IADJ[3:0] setting when VCC = 4 V, PFS.DSCR[1:0] = 00 (= 2 mA). This shows the approximate value of output current and it is not guaranteed. So use this register to adjust the output current to match the selected specification.

Table 16.3 Output current adjustment at VCC = 4 V, PFS.DSCR[1:0] = 00

IADJ[3:0]	Output current [mA]
0x0	2.42
0x1	2.25
0x2	2.08
0x3	1.91
0x4	1.81
0x5	1.70
0x6	1.60
0x7	1.49
0x8	1.42
0x9	1.35
0xA	1.29
0xB	1.22
0xC	1.17
0xD	1.13
0xE	1.08
0xF	1.03

Note: The output current listed in this table is an approximate value and not a guarantee.

16.3 Operation

16.3.1 General I/O Ports

All pins except P300, P301, P302 and P303 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 16.2. Register Descriptions.

Each port has the following bits:

- 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_PORT1 or 2 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_PORT1 or 2 signal occurs
- Event Output Reset bit (EORRn), which indicates the output value when an ELC_PORT1 or 2 signal occurs

16.3.2 Port Function Select

The following port functions are available for configuring each pin:

- I/O configuration: CMOS output or open-drain output, and pull-up control
- 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
- 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 16.2.5. PmnPFS/PmnPFS_HA/PmnPFS_BY : Port mn Pin Function Select Register (m = 0 to 4, n = 00 to 15).

16.3.3 Output Current Control Function

The P100, P302, and P303 ports support output current control function with the following associated registers:

- CCDE — Output Current Control Enable register, which selects pins to enable the output current control function that produces low-level output current.

  Note: After PFS.PDR register for output mode and 1 are written to the corresponding bit of the CCDE register, the state of a pin takes 10 µs to become stable.

- PmnPFS.DSCR[1:0] (Pmn = P100, P302, and P303) — Port output current control bit, which selects the constant low-level output current to 2 mA, 5 mA or 10 mA when output current control function enabled by CCDE.CCDE0n bit is set to 1.
- CCTRM — Output Current Control Trimming register, which is used to adjust or fine tune the low-level output current of all output current control function enable port to match the selected specification.

16.3.4 Port Group Interrupt Function for the ELC

In the MCU, Ports 1 to 2 are assigned for the ELC port group function.

16.3.4.1 Behavior when ELC_PORT1 or 2 is Input from the ELC

The MCU supports the two functions described in this section when an ELC_PORT1 or 2 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_PORT1 or 2 signal comes from the ELC, the input enable of the I/O cell is asserted, and then data from the external pins is read into the EIDR bit.

For the GPO function (PDR = 1) or the peripheral mode (PMR = 1), 0 is input to the EIDR bit from the external pins.

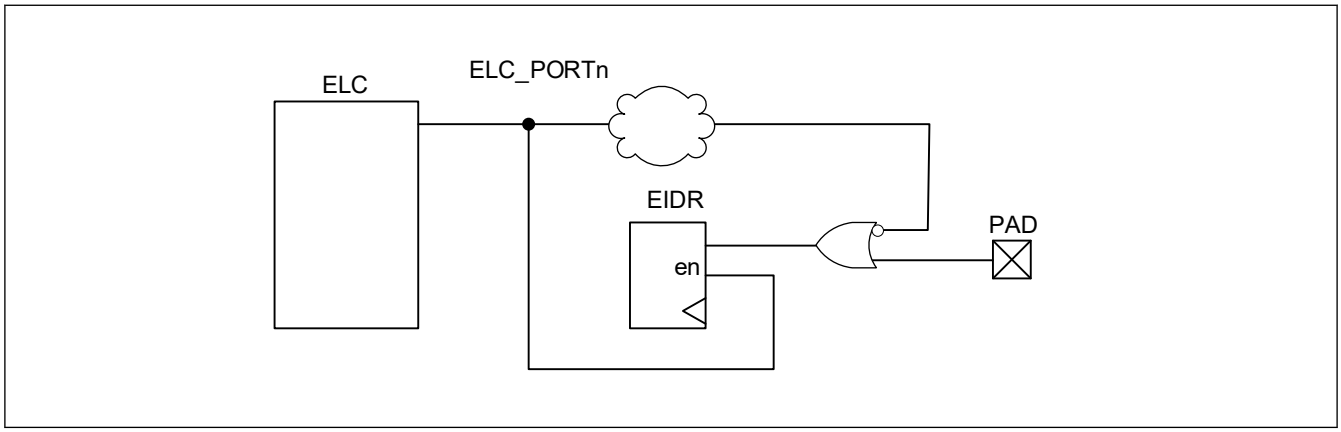


Figure 16.2 Event ports input data

(2) Output from PODR by EOSR and EORR

When an ELC_PORT1 or 2 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 ELC_PORT1 or 2 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_PORT1 or 2 occurs, the PODR register outputs 0 to the external pin. Otherwise, when EORR = 0, the PODR value is retained.

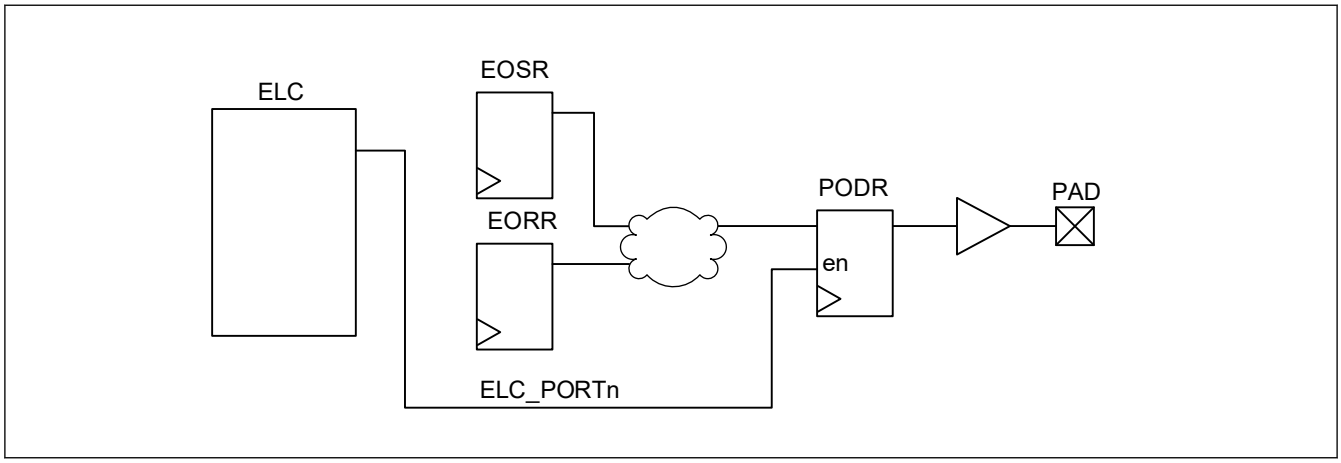


Figure 16.3 Event ports output data

16.3.4.2 Behavior when an Event Pulse is Output to the 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 16.2.5. PmnPFS/PmnPFS_HA/PmnPFS_BY : Port mn Pin Function Select Register (m = 0 to 4, 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 1, when the data is input from P100 to P103, the data of those 4 pins is organized by OR logic. This data is formed into a one-shot pulse that goes to the ELC. The operation of Port 2 is the same.

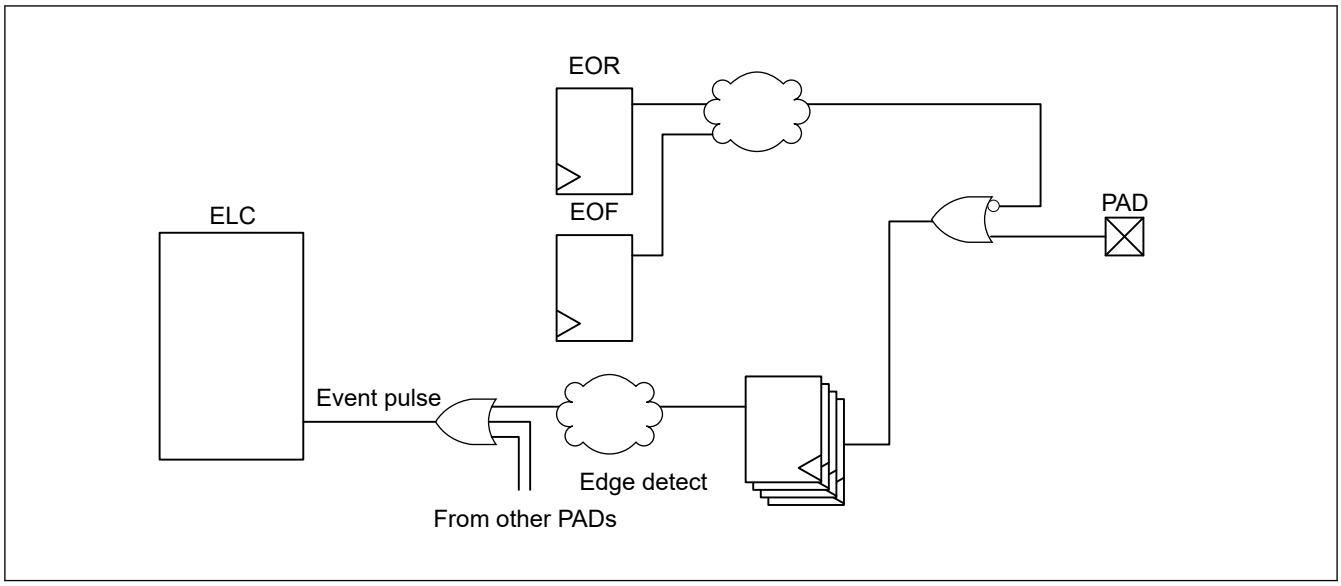


Figure 16.4 Generation of event pulse

16.3.5 Wait Function for Port Read

Wait cycles for reading the port input data can be set in the PRWCNTR.WAIT[1:0] bits as follows:

- Read port input data (PIDR) by reading the PCNTR2 or PFS register
- Latch port pin state to Event on Falling/Event on Rising (EOFR) when a PORT1 or 2 signal occurs.

The number of access cycles is the value in PRWCNTR.WAIT[1:0] plus 1. For example, if PRWCNTR.WAIT[1:0] is set as 2'b10, then the wait is 2 cycles, and access is 3 clock cycles.

Table 16.4 shows the relationship of voltage, frequency, and wait cycles.

Table 16.4 Relationship between voltage, frequency, and wait cycles

VCC	Access cycles*1	Wait cycles*2
More than 2.7 V	2 to 4	1 to 3
2.4 to 2.7 V	3 to 4	2 to 3
1.8 to 2.4 V	4	3
1.6 to 1.8 V	2 to 4	1 to 3

Note 1. Bus latency is not included.
Note 2. Number of wait cycles to set in Port Read Wait Control Register (PRWCNTR).

16.4 Handling of Unused Pins

Table 16.5 shows how to handle unused pins.

Table 16.5 Handling of unused pins

Pin name	Description
P203/MD	Use as a mode pin
RES	Connect to VCC through a resistor (pulling up)
P200/NMI	Connect to VCC through a resistor (pulling up)
P300/EXTAL	When the external clock input is not used, set the MOSCCR.MOSTP bit to 1 (general port P300). When this pin is not used as port P300, configure it in the same way as ports P0x to P4x.
XT1	When the sub-clock oscillator is not used, set the SOSCCR.SOSTP bit to 1 and connect the associated pin to VSS (pulled down) through a resistor.
XT2	When the sub-clock oscillator is not used, set the SOSCCR.SOSTP bit to 1 and leave the associated pin open.
Other ports P0x to P4x	• If the direction setting is for 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 setting is for output (PCNTR1.PDRn = 1), keep pin open*1
P006 to P007, P400 to P401	• If the direction setting is for input (PCNTR1.PDRn = 0), connect the associated pin to AVCC (pulled up) through a resistor or to AVSS (pulled down) through a resistor*1 • If the direction setting is for output (PCNTR1.PDRn = 1), keep pin open *1
P003/AVREFM	• If the direction setting is for input (PCNTR1.PDRn = 0), connect the associated pin to AVSS (pulled down) through a resistor*1 • If the direction setting is for output (PCNTR1.PDRn = 1), keep pin open*1
P002/AVREFP	• If the direction setting is for input (PCNTR1.PDRn = 0), connect the associated pin to AVCC (pulled up) through a resistor*1 • If the direction setting is for output (PCNTR1.PDRn = 1), keep pin open*1

Note 1. Clear the PmnPFS.PMR, PmnPFS.ISEL, PmnPFS.PCR, and PmnPFS.ASEL bits to 0.
Note 2. P203, P300, P301, and P303 should be enabled for input pull-up from the initial value (PmnPFS.PCR = 1).

16.5 Usage Notes

16.5.1 Note on Using SAU Function

To use the SAU function, when SAU is selected by Peripheral Select bit (PSEL) = 10001b, the output buffer for all SAU ports except SCK/SCL ports is turned on by default. At the same time, the output signal from SAU is maintained at logic 1 during input mode. To switch SAU ports to input mode, configure the Open-Drain Control bit (NCODR) to 1. This action places the output into a Hi-Z state, allowing the signal from the terminal to enter the input buffer. See Figure 16.5.

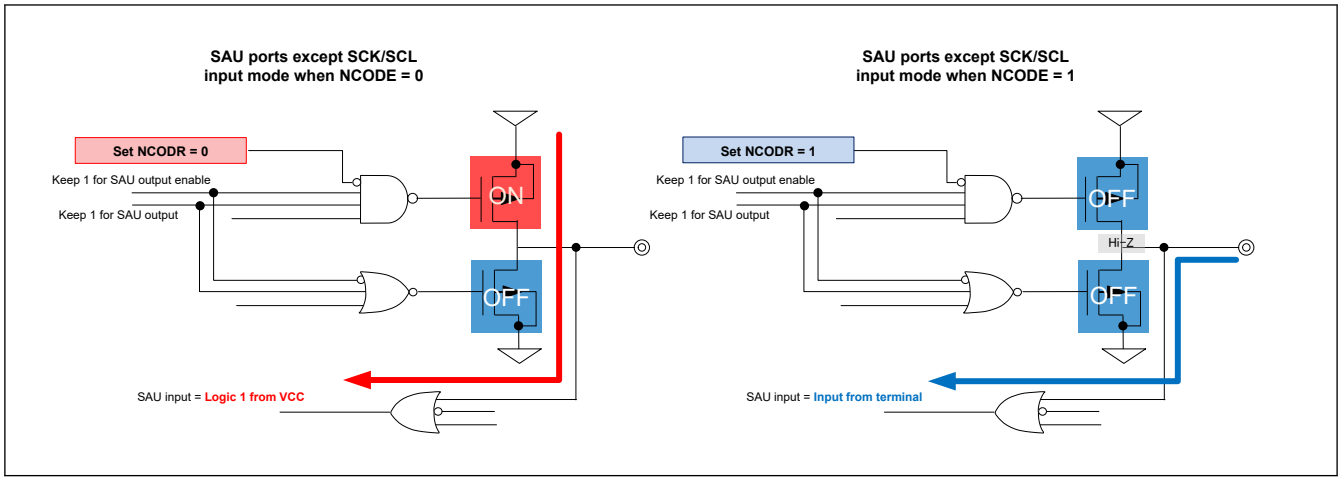


Figure 16.5 SAU ports except SCK/SCL input mode

For SCK/SCL ports, set the port direction according to the SAU function by setting the Port Direction bit (PDR) in the Port mn Pin Function Select Register (PmnPFS.PDR).

16.5.2 Note on Using TAU Function

To use the TAU function, when TAU is selected by the Peripheral Select bit (PSEL) = 11001b, set the port direction depending on the TAU function by the Port Direction bit (PDR) in the Port mn Pin Function Select Register (PmnPFS.PDR). When port is used as input, for example TIxx, set PDR bits = 0 (port direction = input). When port is used as output, for example TOxx, set PDR bits = 1 (port direction = output).

Note: xx indicates the TAU port name 00 to 07 with _A/_B pin group.

16.5.3 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.
2. Set 1 to the PFSWE bit in the PWPR register. This enables writing to the PmnPFS register.
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.
7. Set 1 to the B0WI bit in the PWPR register. This disables writing to the PFSWE bit in the PWPR register.

16.5.4 Procedure for Using Port Group Input

To use the port group input (port 1 and port 2):

1. Set the IELSRn.IELS[4:0] bits to all 0 to ignore unexpected pulses. For more information, see section 15, 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.

16.5.5 Port Output Data Register (PODR) Summary

This register outputs data as follows:

1. Outputs 0 if PCNTR4.EORR is set to 1 when ELC_PORT1 or 2 signal occurs.
2. Outputs 1 if PCNTR4.EOSR is set to 1 when ELC_PORT1 or 2 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 section 16.5.5. Port Output Data Register (PODR) Summary and 3. from the list occur at the same time, the higher priority event section 16.5.5. Port Output Data Register (PODR) Summary is executed.

16.5.6 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.

16.5.7 Notes for Handling Non-existing Pins in Small Pin Package Product

In small pin package (32- /24- /16-pin) product, there are some pins that have been removed. Do not set I/O related register that makes the setting of non-existing pin different from the initial value after reset. Keep the setting that relates to those pins at initial value, otherwise GPIO input will float and create unnecessary current.

16.6 Peripheral Select Settings for Each Product

This section describes the pin function select configuration by the PmnPFS register. Some pin names have a _A, _B, _C, _D, _E, or _F suffix. The suffix can be ignored when assigning functionality, but assigning the same function to two or more pins simultaneously is prohibited. Only the allowed values (functions) should be specified in the PSEL bits of PmnPFS. If a value that is not allowed for the register is specified, the correct operation is not guaranteed.

Table 16.6 Register settings for I/O pin functions on Port 0

PSEL[4:0] settings	Function	Pin									
		P000	P001	P002	P003	P006	P007	P008	P009	P010	P011
00000b (value after reset)	Hi-Z/cJTAG	Hi-Z									
00110b	UARTA	—	—	—	—	—	—	RxDA1	TxDA0	RxDA0	CLKA0
00111b	IICA	—	—	—	—	—	—	—	—	SDAA1	SCLA1
10001b	SAU	RxD1	TxD1	—	—	—	—	SI10/SDA10	SCK10/ SCL10	—	—
10010b	KINT	—	—	—	—	—	—	—	—	—	—
11001b	TAU	TO00	TI00	—	—	—	—	—	—	—	TI07_B/ TO07_B
ASEL bit	ADC12	ANI16	ANI17	ANI0	ANI1	ANI4	ANI5	—	—	—	—
	DAC8	DACOUT0	DACOUT1	—	—	—	—	—	—	—	—
	CMP	IVREF0*1	—	—	—	—	—	—	—	—	—
ISEL bit		IRQ6_A	IRQ5_A	—	IRQ7_A	IRQ4_A	IRQ3_A	—	—	—	—
DSCR[1:0] bits	Current control	—	—	—	—	—	—	—	—	—	—
NCODR bit	N-ch open-drain	✓	✓	—	—	—	—	✓	✓	✓	✓
PCR bit	Pull-up	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Number of pins	48-pin product	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	32-pin product	✓	✓	✓	✓	✓	✓	—	—	—	—
	24-pin product	✓	✓	✓	✓	—	—	—	—	—	—
	16-pin product	✓	✓	✓	✓	—	—	—	—	—	—

✓: Available
—: Setting prohibited
Recommend using pins that have a letter appended to their names, for instance "_A" or "_B", to indicate group membership. For the interface, the AC portion of the electrical characteristics is measured for each group.

Note 1. IVREF0 function is not supported in 16-pin product.

Table 16.7 Register settings for I/O pin functions on Port 1

PSEL[4:0] settings	Function	Pin											
		P100	P101	P102	P103	P104	P105	P106	P107	P108	P109	P110	P111
00000b (value after reset)	Hi-Z/cJtag	Hi-Z											
00111b	IICA	—	—	SCLA0_B	SDAA0_B	—	—	—	—	—	—	—	—
01001b	CMP (Digital)	—	—	—	—	—	VCOUT0	—	—	—	—	—	—
01010b	CAC	—	—	—	—	—	—	—	CACREF	—	—	—	—
10001b	SAU	SO00/ TxD0_A	SCK20/ SCL20	SI20/ SDA20/ RxD2	SO20/ TxD2	SCK11/ SCL11	SI11/ SDA11	SO11	—	—	—	—	—
10010b	KINT	—	—	—	—	—	—	—	—	KR04	KR05	—	—
11001b	TAU	TI05/TO05	TI02_B/ TO02_B	TI01/TO01	TI02_A/ TO02_A	—	—	—	TI03_A/ TO03_A	—	—	—	—
11011b	RTC	—	—	—	—	—	RTC1HZ	—	—	—	—	—	—
ASEL bit	ADC12	—	—	—	—	—	ANI18	ANI19	—	—	—	—	—
	CMP	—	—	—	—	IVREF1	—	IVCMP0	IVCMP1	—	—	—	—
ISEL bit		IRQ6_C	IRQ7_C	IRQ2_A	—	—	—	—	—	IRQ4_B	IRQ5_B	IRQ7_B	IRQ6_B
DSCR[1:0] bits	Current control	2/5/10 mA	—	—	—	—	—	—	—	—	—	—	—
NCODR bit	N-ch open-drain	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
PCR bit	Pull-up	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Number of pins	48-pin product	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	32-pin product	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	24-pin product	✓	✓	✓	✓	✓	✓	✓	✓	—	—	—	—
	16-pin product	✓	—	—	—	—	—	—	✓	—	—	—	—

✓: Available
—: Setting prohibited
Recommend using pins that have a letter appended to their names, for instance "_A" or "_B", to indicate group membership. For the interface, the AC portion of the electrical characteristics is measured for each group.

Table 16.8 Register settings for I/O pin functions Port 2

PSEL[4:0] settings	Function	Pin							
		P200	P201	P202	P203	P204	P205	P206	P207
00000b (value after reset)	Hi-Z/cJtag	Hi-Z			Input pull-up	Hi-Z			
01001b	CLKOUT	—	—	CLKOUT_B	—	—	—	—	—
01110b	REMC	—	—	RIN0	—	—	—	—	—
10001b	SAU	—	—	—	—	SCK21/SCL21	SI21/SDA21	SO21	SCK01/SCL01
10010b	KINT	—	—	—	—	—	—	—	KR02
ASEL bit		—	—	—	—	—	—	—	—
ISEL bit		NMI	IRQ3_C	IRQ2_C	—	—	—	—	—
DSCR[1:0] bits	Current control	—	—	—	—	—	—	—	—
NCODR bit	N-ch open-drain	—	✓	✓	✓	✓	✓	✓	✓
PCR bit	Pull-up	—	✓	✓	✓	✓	✓	✓	✓
Number of pins	48-pin product	✓	✓	✓	✓	✓	✓	✓	✓
	32-pin product	✓	✓	✓	✓	—	—	—	—
	24-pin product	✓	—	—	✓	—	—	—	—
	16-pin product	✓	—	—	✓	—	—	—	—

✓: Available
—: Setting prohibited
Recommend using pins that have a letter appended to their names, for instance "_A" or "_B", to indicate group membership. For the interface, the AC portion of the electrical characteristics is measured for each group.

Table 16.9 Register settings for input/output pin functions (PORT3)

PSEL[4:0] settings	Function	Pin							
		P300	P301	P302	P303	P304	P305	P306	P307
00000b (value after reset)	Hi-Z/cJTAG	TCKC	TMSC	Hi-Z	Input pull-up	Hi-Z			
00111b	IICA	—	—	SCLA0_A	SDAA0_A	—	—	—	—
01001b	CLKOUT/CMP (Digital)	—	—	VCOUT1	CLKOUT_A	—	—	—	—
10001b	SAU	SCK00/SCL00	SI00/SDA00/ RxD0_A	TxD0_B	RxD0_B	SO01	SI01/SDA01	—	—
10010b	KINT	—	—	—	—	KR00	KR01	KR03	—
11001b	TAU	TI07_A/TO07_A	TI06/TO06	TI03_B/TO03_B	TI04/TO04	—	—	—	—
ASEL bit		—	—	—	—	—	—	—	—
ISEL bit		IRQ0_A	IRQ1_A	IRQ3_B	IRQ2_B	—	—	—	—
DSCR[1:0] bits	Current control	—	—	2/5/10 mA	2/5/10 mA	—	—	—	—
NCODR bit	N-ch open-drain	✓	✓	✓	✓	✓	✓	✓	✓
PCR bit	Pull-up	✓	✓	✓	✓	✓	✓	✓	✓
Number of pins	48-pin product	✓	✓	✓	✓	✓	✓	✓	✓
	32-pin product	✓	✓	✓	✓	—	—	—	—
	24-pin product	✓	✓	✓	✓	—	—	—	—
	16-pin product	✓	✓	✓	✓	—	—	—	—

✓: Available
—: Setting prohibited
Recommend using pins that have a letter appended to their names, for instance "_A" or "_B", to indicate group membership. For the interface, the AC portion of the electrical characteristics is measured for each group.

Table 16.10 Register settings for I/O pin functions on Port 4

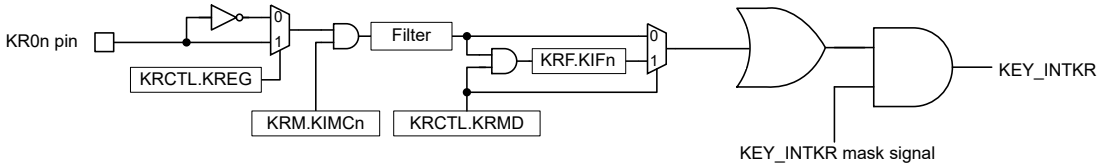
PSEL[4:0] settings	Function	Pin			
		P400	P401	P402	P403
00000b (value after reset)	Hi-Z/cJtag	Hi-Z			
00110b	UARTA	—	—	CLKA1	TxDA1
10001b	SAU	—	—	—	SO10
ASEL bit	ADC12	ANI2	ANI3	—	—
ISEL bit		—	—	—	—
DSCR[1:0] bits	Current control	—	—	—	—
NCODR bit	N-ch open-drain	—	—	✓	✓
PCR bit	Pull-up	✓	✓	✓	✓
Number of pins	48-pin product	✓	✓	✓	✓
	32-pin product	—	—	—	—
	24-pin product	—	—	—	—
	16-pin product	—	—	—	—

✓: Available
—: Setting prohibited
Recommend using pins that have a letter appended to their names, for instance "_A" or "_B", to indicate group membership. For the interface, the AC portion of the electrical characteristics is measured for each group.

17. Key Interrupt Function (KINT)

17.1 Overview

The key interrupt function (KINT) generates the key interrupt by detecting rising or falling edge on the key interrupt input pins. Figure 17.1 shows a block diagram and Table 17.1 lists the input pins.



Note: n = 0 to 5

Figure 17.1 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 17.1 KINT I/O pins

Pin name	I/O	Function
KR00 to KR05	Input	Key interrupt input pins

The number of key interrupt input channels differs, depending on the product. This is shown in Table 17.2.

Table 17.2 Number of key interrupts

24-pin products	32-pin products	48-pin products
—	2 ch (KR04 to KR05)	6 ch (KR00 to KR05)

17.2 Register Descriptions

17.2.1 KRCTL : Key Return Control Register

Base address: KINT = 0x4008_0000

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 KR05 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.KIF5) 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 5), and sets the detection edge.

17.2.2 KRF : Key Return Flag Register

Base address: KINT = 0x4008_0000

Offset address: 0x04

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	KIF5	KIF4	KIF3	KIF2	KIF1	KIF0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	KIF0 to KIF5	Key Interrupt Flag n 0: No interrupt detected 1: Interrupt detected	R/W
7:6	—	These bits are read as 0. The write value should be 0.	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.

17.2.3 KRM : Key Return Mode Register

Base address: KINT = 0x4008_0000

Offset address: 0x08

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	KIMC5	KIMC4	KIMC3	KIMC2	KIMC1	KIMC0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	KIMC0 to KIMC5	Key Interrupt Mode Control n 0: Do not detect key interrupt signals 1: Detect key interrupt signals	R/W
7:6	—	These bits are read as 0. The write value should be 0.	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 16, I/O Ports.

17.3 Operation

17.3.1 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.

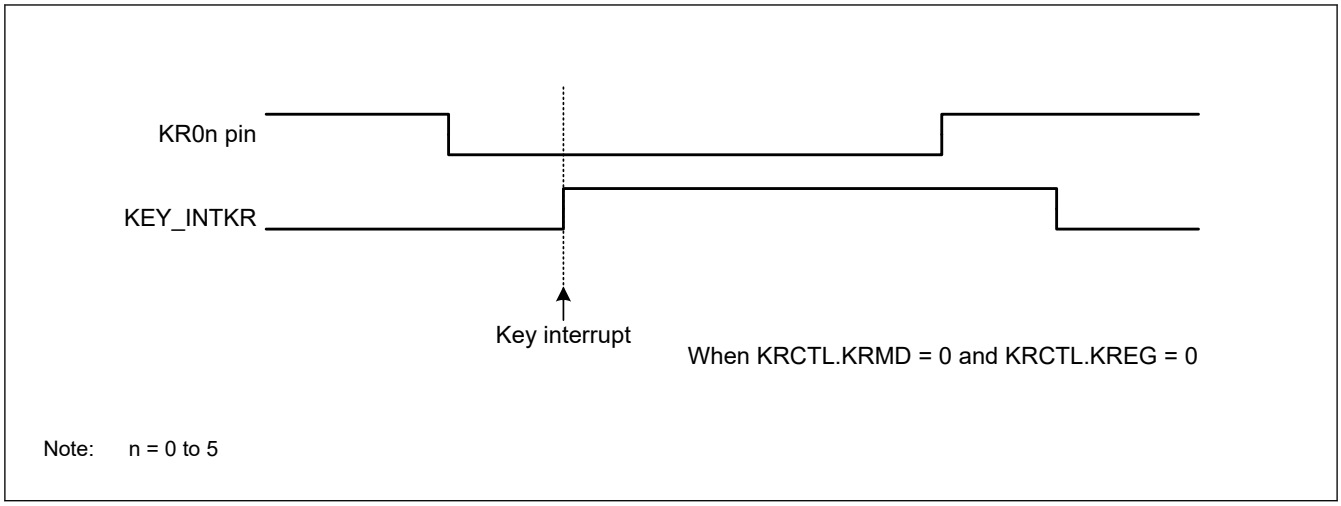


Figure 17.2 Operation of KEY_INTKR signal when a key interrupt is input to a single channel

Figure 17.3 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 17.3.

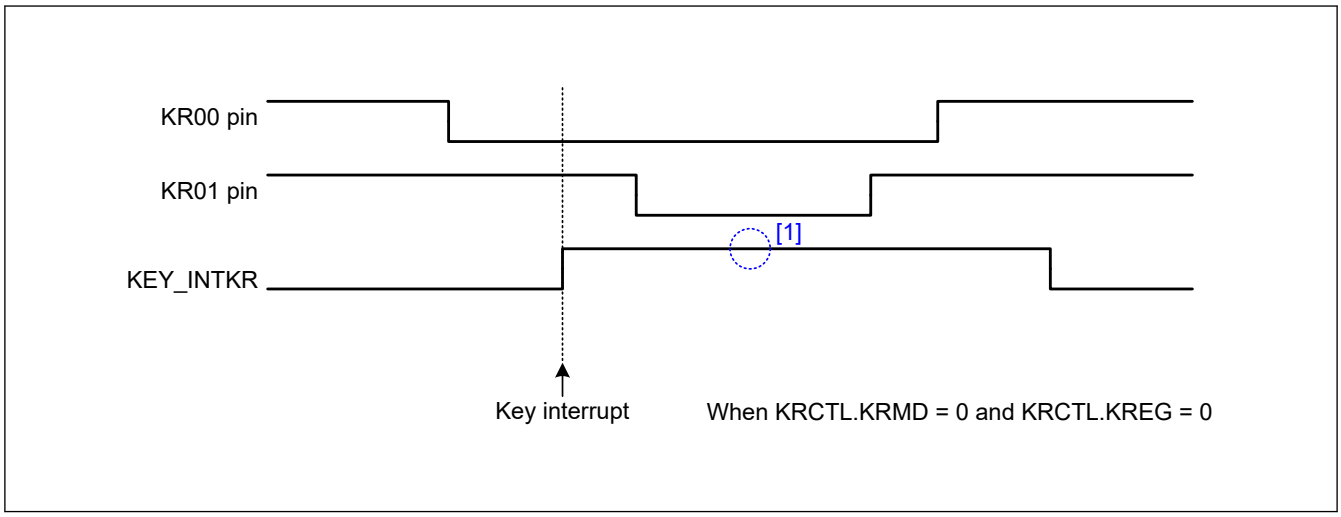


Figure 17.3 Operation of KEY_INTKR signal when key interrupts are input to multiple channels

17.3.2 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 17.4 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.

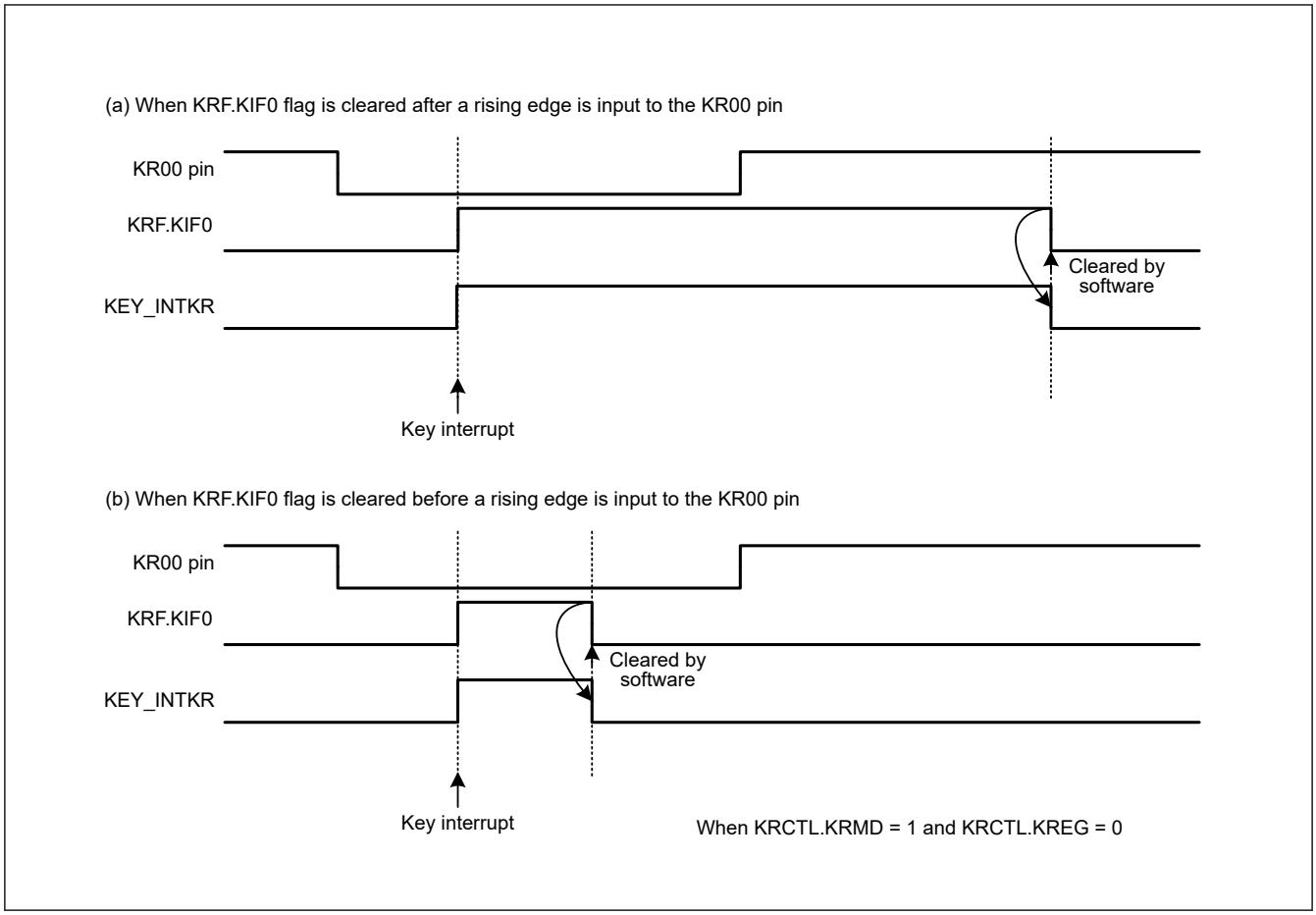


Figure 17.4 Basic operation of KEY_INTKR signal when key interrupt flag is used

Figure 17.5 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 17.5.

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.

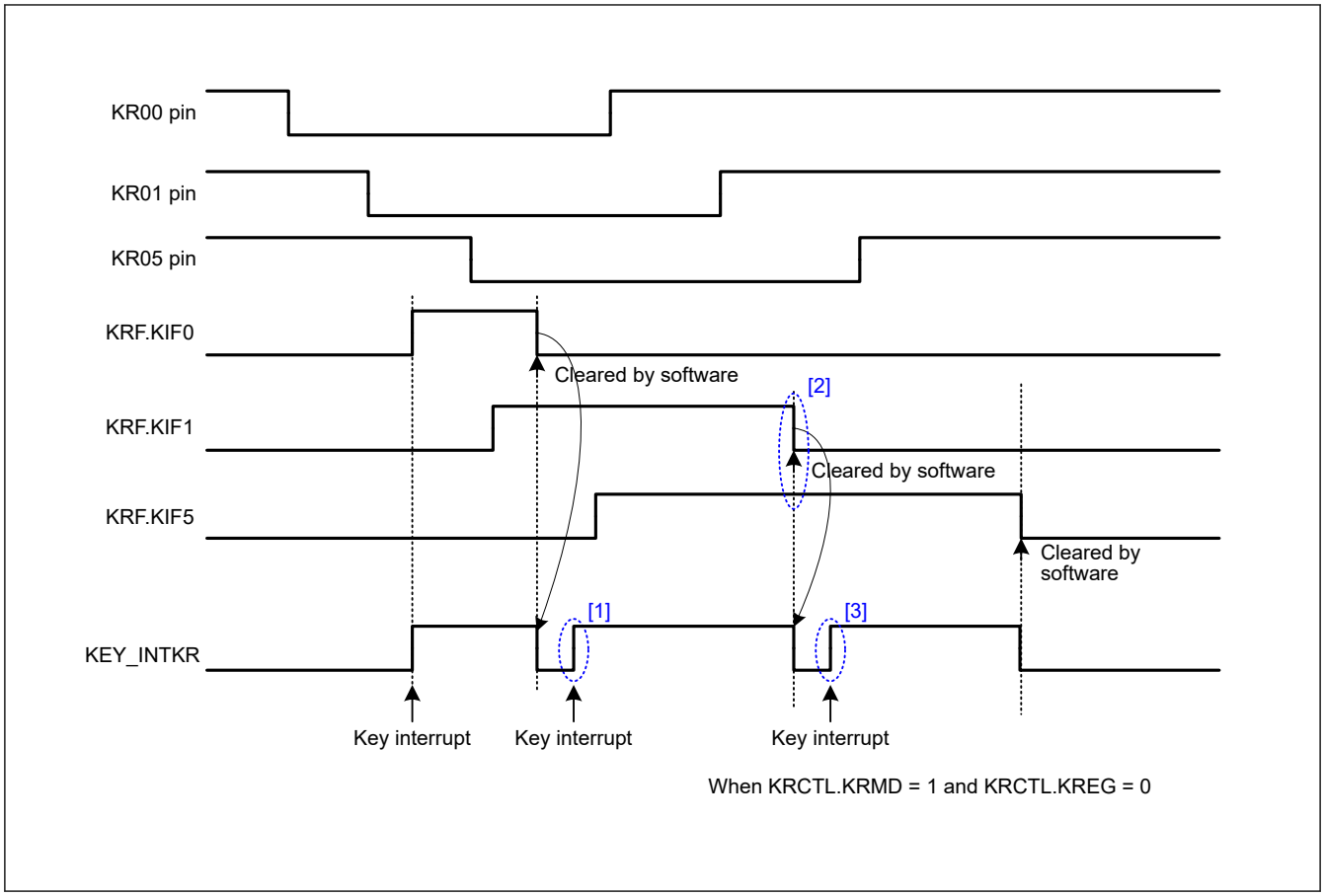


Figure 17.5 Operation of KEY_INTKR signal when key interrupts are input to multiple channels

17.4 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.
- When using KINT to trigger ADC in Snooze mode through ELC, it is necessary to configure DTC to repeat transfer mode to clear KEY_INTKR level interrupt.

Figure 17.6 shows an example setting for using KINT to trigger ADC in Snooze mode with KEY_INTKR interrupt cleared by DTC.

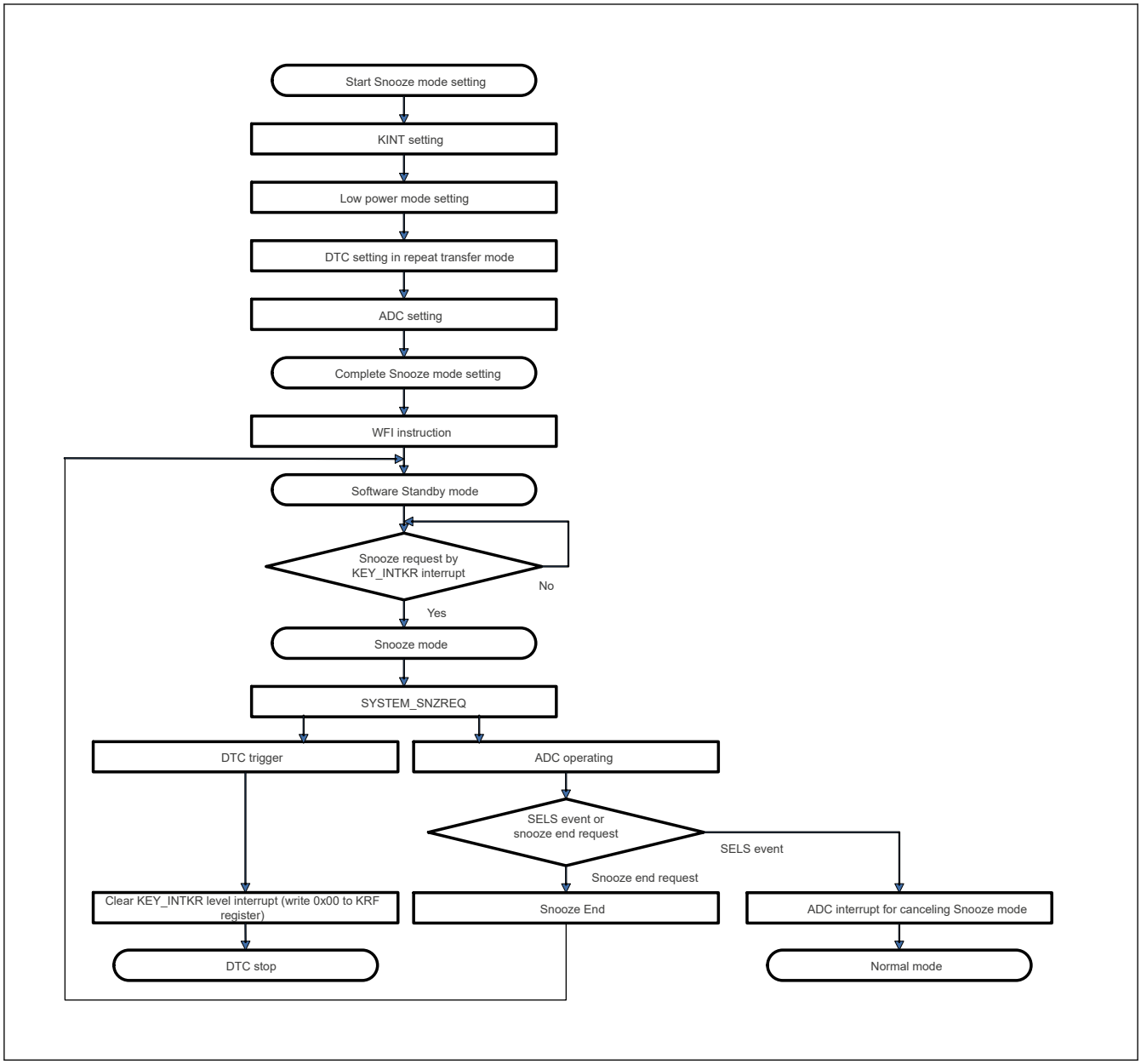


Figure 17.6 Example setting for using KINT to trigger ADC in Snooze mode with KEY_INTKR interrupt cleared by DTC

Figure 17.7 shows an example setting for DTC in repeat transfer mode.

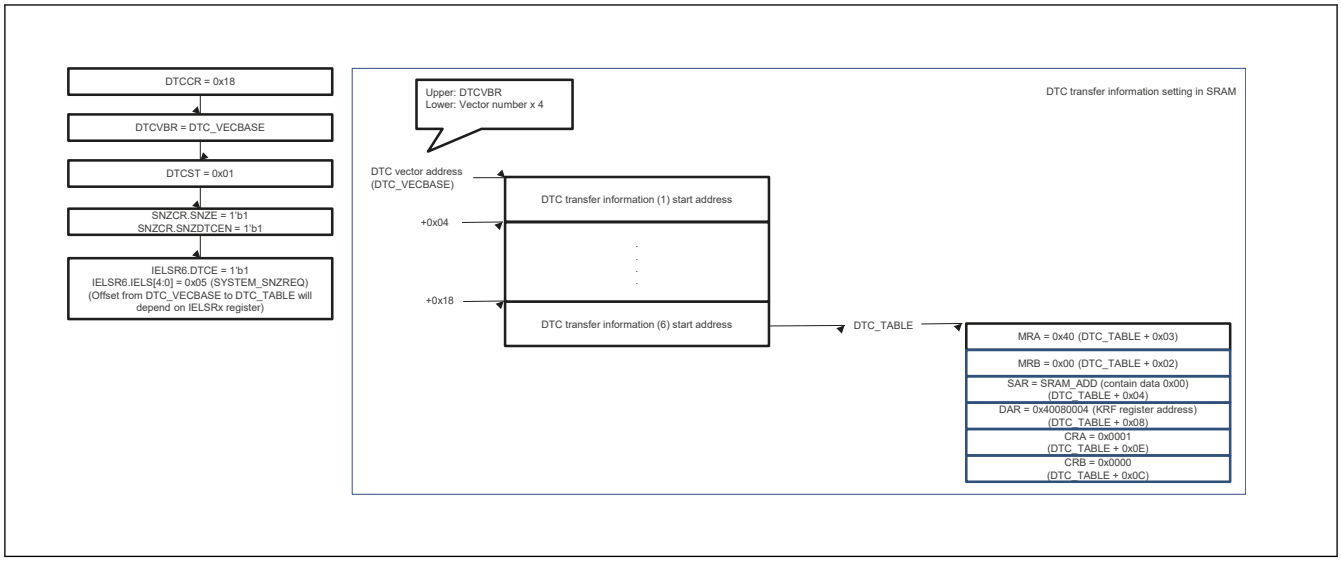


Figure 17.7 Example setting for DTC in repeat transfer mode

18. Timer Array Unit (TAU)

18.1 Overview

The timer array unit has eight 16-bit timers.

Each 16-bit timer is called a channel and can be used as an independent timer. In addition, two or more channels can be used to create a high-accuracy timer.

Figure 18.1 shows the channel configuration in timer array unit.

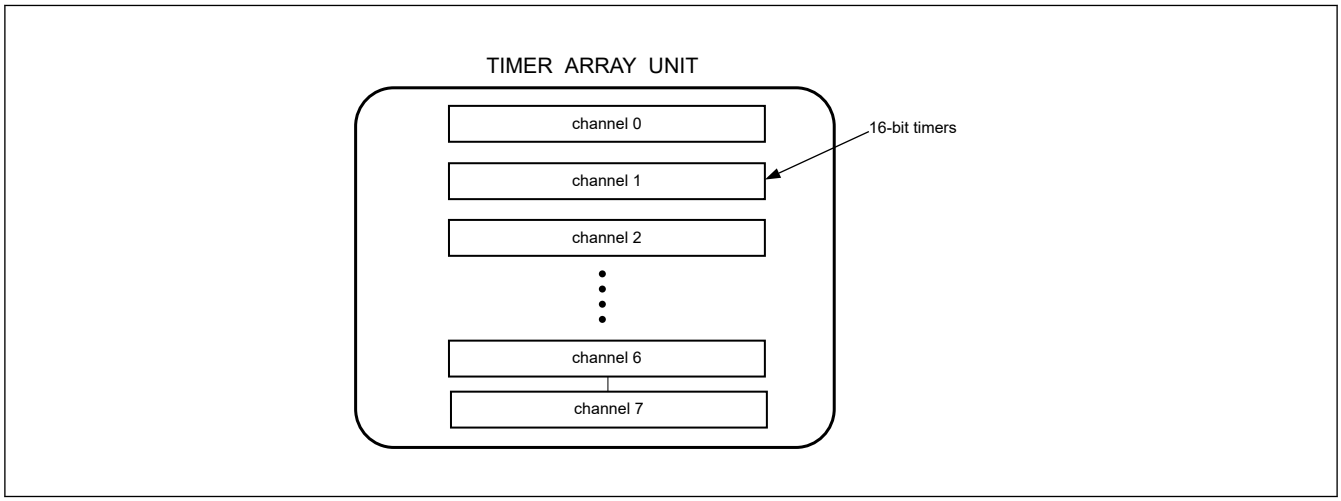


Figure 18.1 Channel configuration

It is possible to use the 16-bit timer of channels 1 and 3 as two 8-bit timers (higher and lower). The functions that can use channels 1 and 3 as 8-bit timers are as follows:

- Interval timer (upper or lower 8-bit timer) and square wave output (lower 8-bit timer only)
- External event counter (lower 8-bit timer only)
- Delay counter (lower 8-bit timer only)

Channel 7 can be used to realize LIN-bus communication operating in combination with UART2 of the serial array unit.

Table 18.1 lists the TAU functions and Figure 18.2 to Figure 18.11 show each functional image.

Table 18.1 TAU functions

Parameter		Description
Independent channel operation function *1	Interval timer	Each timer can be used as a reference timer that generates an interrupt (TAU0_ENDIn) at fixed intervals.
	Square wave output	A toggle operation is performed each time TAU0_ENDIn interrupt is generated and a square wave with a duty factor of 50% is output from a timer output pin (TO0n).
	External event counter	Each timer can be used as an event counter that generates an interrupt when the number of the valid edges of a signal input to the timer input pin (TI0n) has reached a specific value.
	Divider function (only channel 0)	A clock input from a timer input pin (TI00) is divided and output from an output pin (TO00).
	Input pulse interval measurement	Counting is started by the valid edge of a pulse signal input to a timer input pin (TI0n). The count value of the timer is captured at the valid edge of the next pulse. In this way, the interval of the input pulse can be measured.
	Measurement of high- or low-level width of input signal	Counting is started by a single edge of the signal input to the timer input pin (TI0n), and the count value is captured at the other edge. In this way, the high-level or low-level width of the input signal can be measured.
	Delay counter	Counting is started at the valid edge of the signal input to the timer input pin (TI0n), and an interrupt is generated after any delay period.
Simultaneous channel operation function *2	One-shot pulse output	Two channels are used as a set to generate a one-shot pulse with a specified output timing and a specified pulse width.
	PWM (Pulse Width Modulation) output	Two channels are used as a set to generate a pulse with a specified period and a specified duty factor.
	Multiple PWM (Pulse Width Modulation) output	By extending the PWM function and using one master channel and two or more slave channels, up to seven types of PWM signals that have a specific period and a specified duty factor can be generated.
8-bit timer operation function (channels 1 and 3 only) *3		The 8-bit timer operation function makes it possible to use a 16-bit timer channel in a configuration consisting of two 8-bit timer channels.
LIN-bus supporting function (channel 7 only) *4	Detection of wakeup signal	The timer starts counting at the falling edge of a signal input to the serial data input pin (RxD2) of UART2 and the count value of the timer is captured at the rising edge. In this way, a low-level width can be measured. If the low-level width is greater than a specific value, it is recognized as a wakeup signal.
	Detection of break field	The timer starts counting at the falling edge of a signal input to the serial data input pin (RxD2) of UART2 after a wakeup signal is detected, and the count value of the timer is captured at the rising edge. In this way, a low-level width is measured. If the low-level width is greater than a specific value, it is recognized as a break field.
	Measurement of pulse width of sync field	After a break field is detected, the low-level width and high-level width of the signal input to the serial data input pin (RxD2) of UART2 are measured. From the bit interval of the sync field measured in this way, a baud rate is calculated.

Note 1. This function can be used without being affected by the operation mode of other channels. For details, see section 18.7. Independent Channel Operation Function of Timer Array Unit.

Note 2. This function can be used to combine a master channel (a reference timer mainly controlling the cycle) and slave channels (timers operating according to the master channel). For details, see section 18.8. Simultaneous Channel Operation Function of Timer Array Unit.

Note 3. There are several rules for using 8-bit timer operation function. For details, see section 18.3.2. Basic Rules of 8-bit Timer Operation Function (Channels 1 and 3 only).

Note 4. Timer array unit is used to check whether signals received in LIN-bus communication match the LIN-bus communication format. For details about setting up the operations used to implement the LIN-bus, see section 18.2.15. ISC : Input Switch Control register and section 18.7.5. Operation for Input Signal High- or Low-Level Width Measurement.

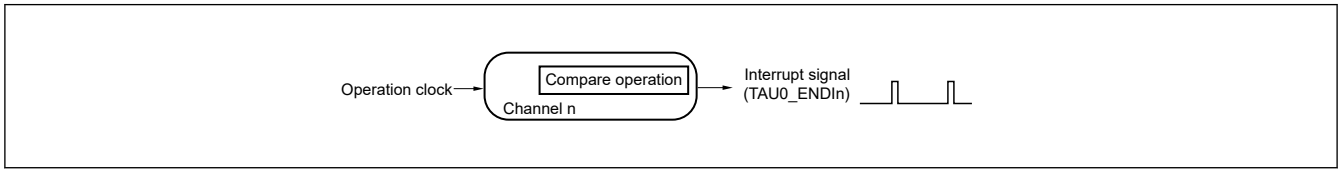


Figure 18.2 Functional image of interval timer

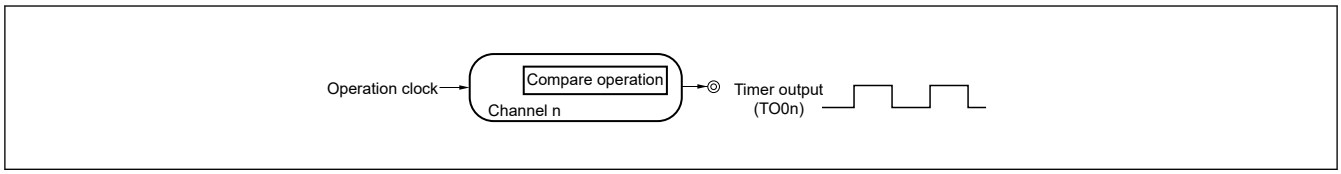


Figure 18.3 Functional image of square wave output

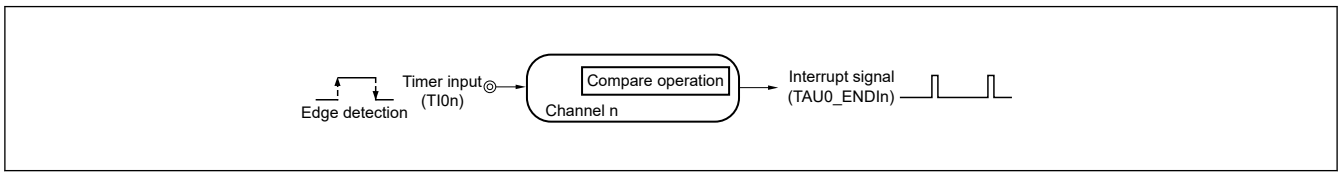


Figure 18.4 Functional image of external event counter

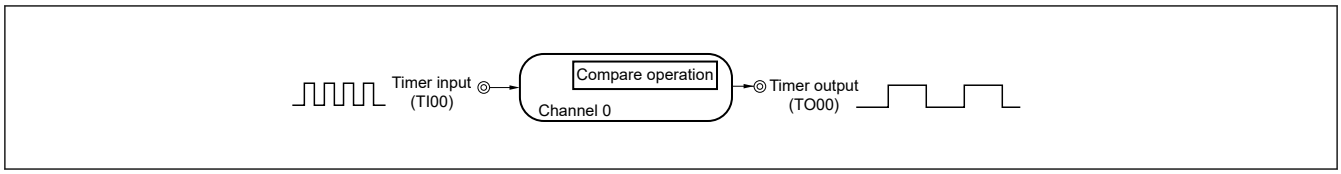


Figure 18.5 Functional image of divider function

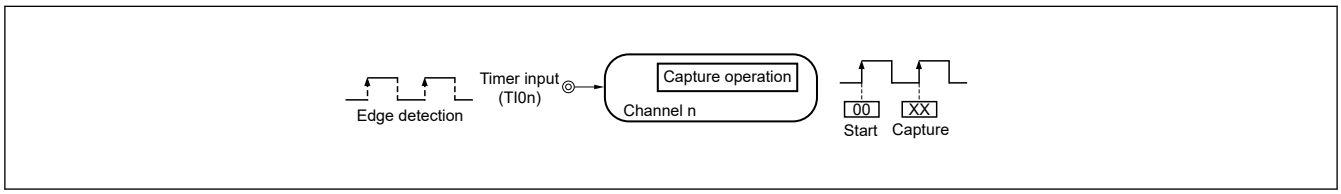


Figure 18.6 Functional image of input pulse interval measurement

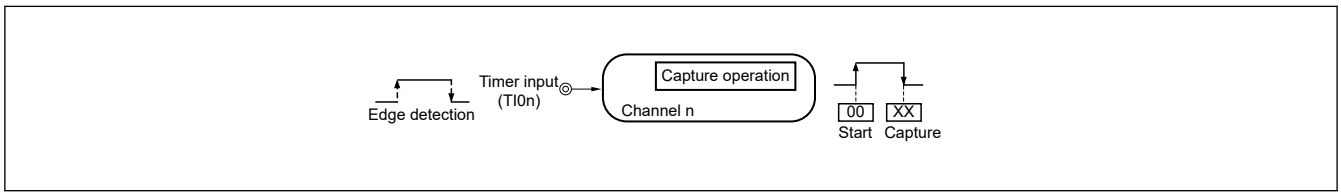


Figure 18.7 Functional image of measurement of high- or low-level width of input signal

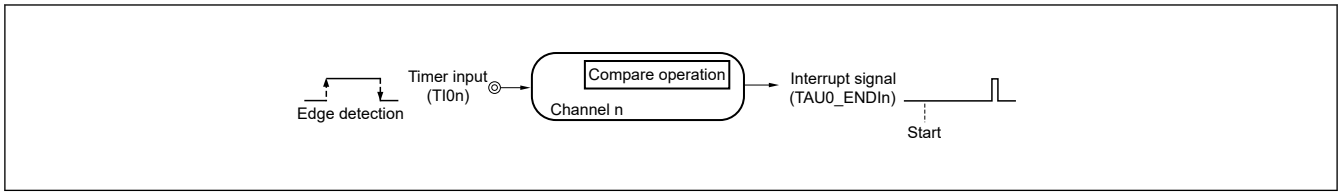


Figure 18.8 Functional image of delay counter

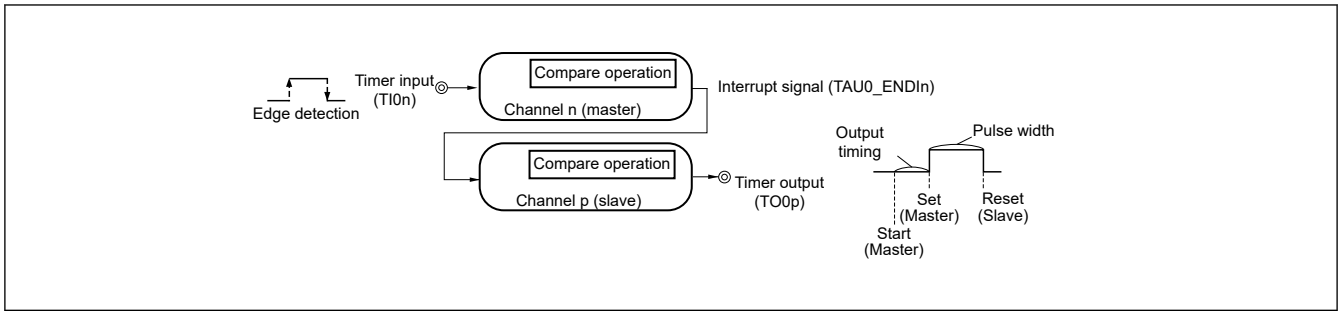


Figure 18.9 Functional image of one-shot pulse output

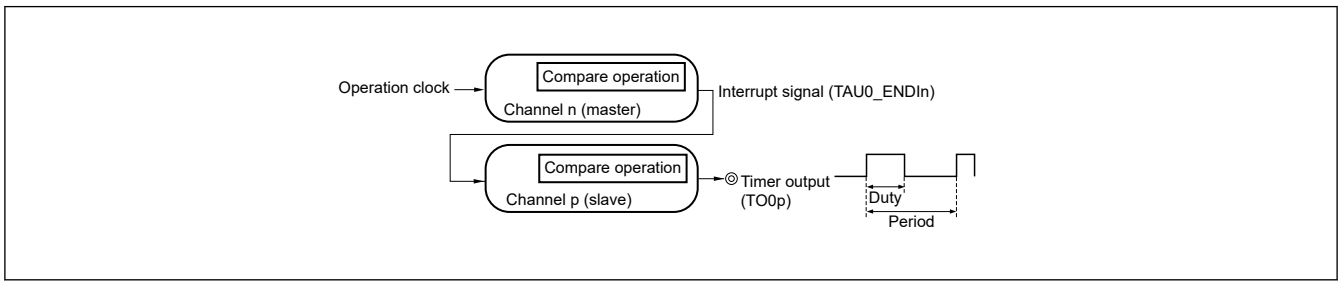


Figure 18.10 Functional image of PWM output

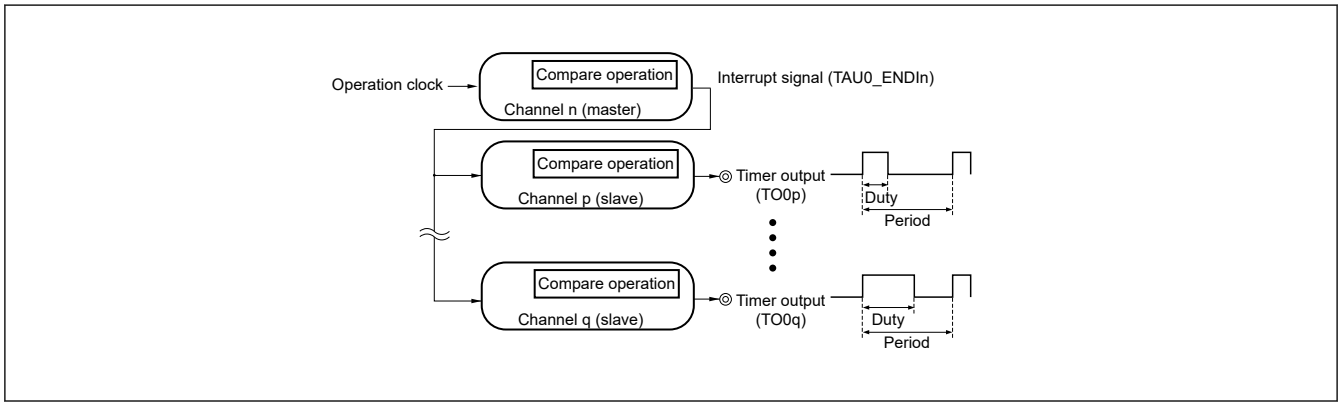


Figure 18.11 Functional image of multiple PWM output

Note: n: Channel number (n = 0 to 7), p, q: Slave channel number (n < p < q ≤ 7)

Timer array unit includes the hardware shown in Table 18.2.

Table 18.2 Configuration of timer array unit

Item	Configuration
Timer/counter	Timer counter register 0n (TCR0n)
Register	Timer data register 0n (TDR0n)
Timer input	TI00 to TI07, RxD2 pin (for LIN-bus)
Timer output	TO00 to TO07 pins, output controller

Figure 18.12 shows the block diagram of the timer array unit. Figure 18.13 to Figure 18.18 show a block diagram for each channel.

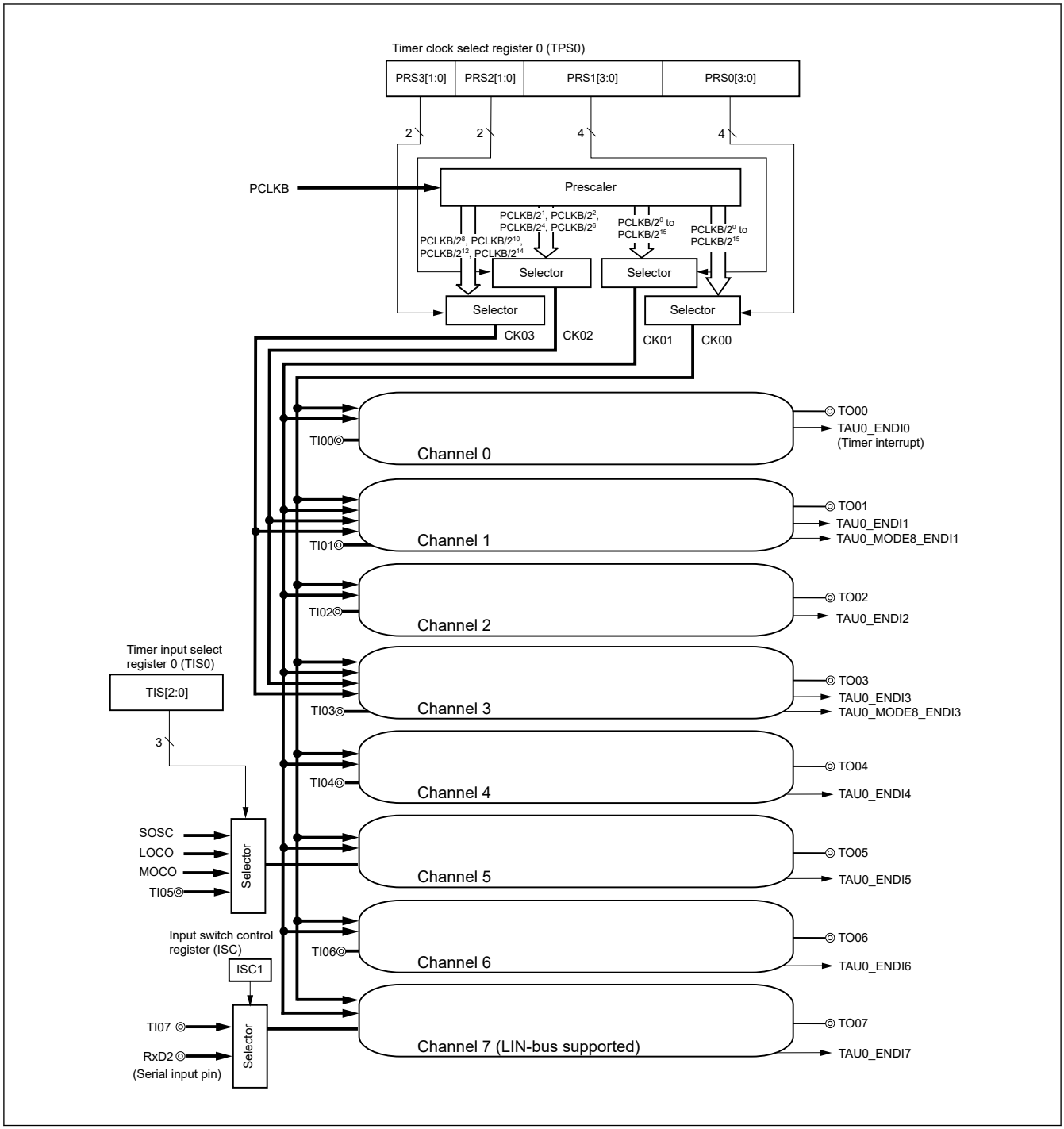


Figure 18.12 Entire configuration of timer array unit

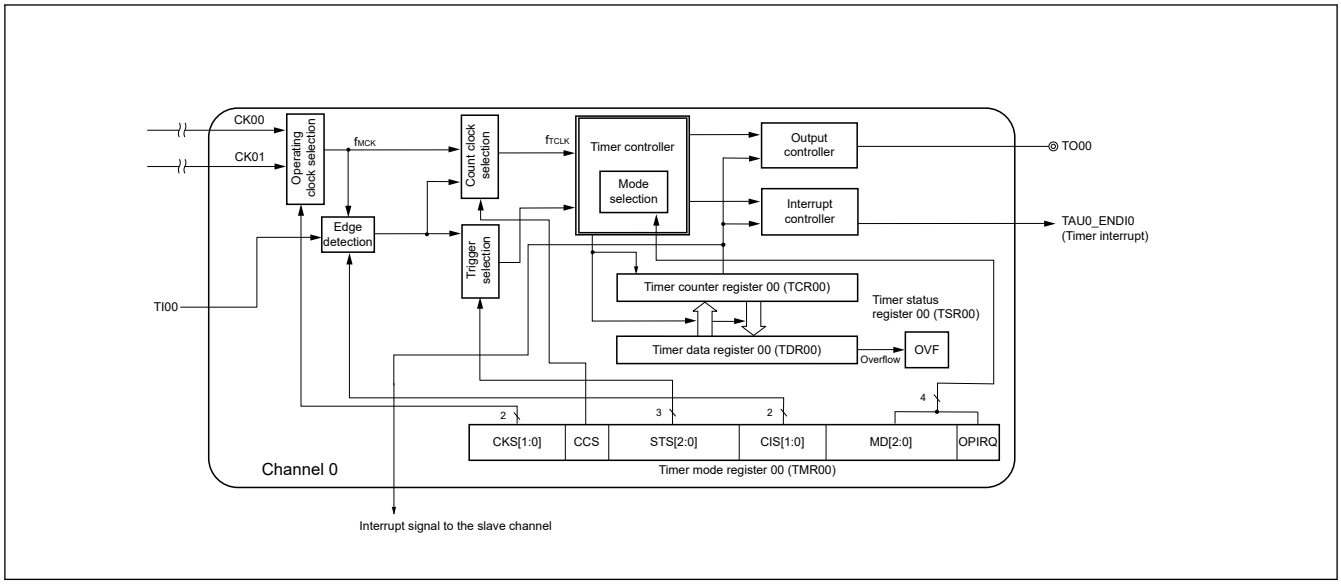


Figure 18.13 Internal block diagram of channel 0 of timer array unit

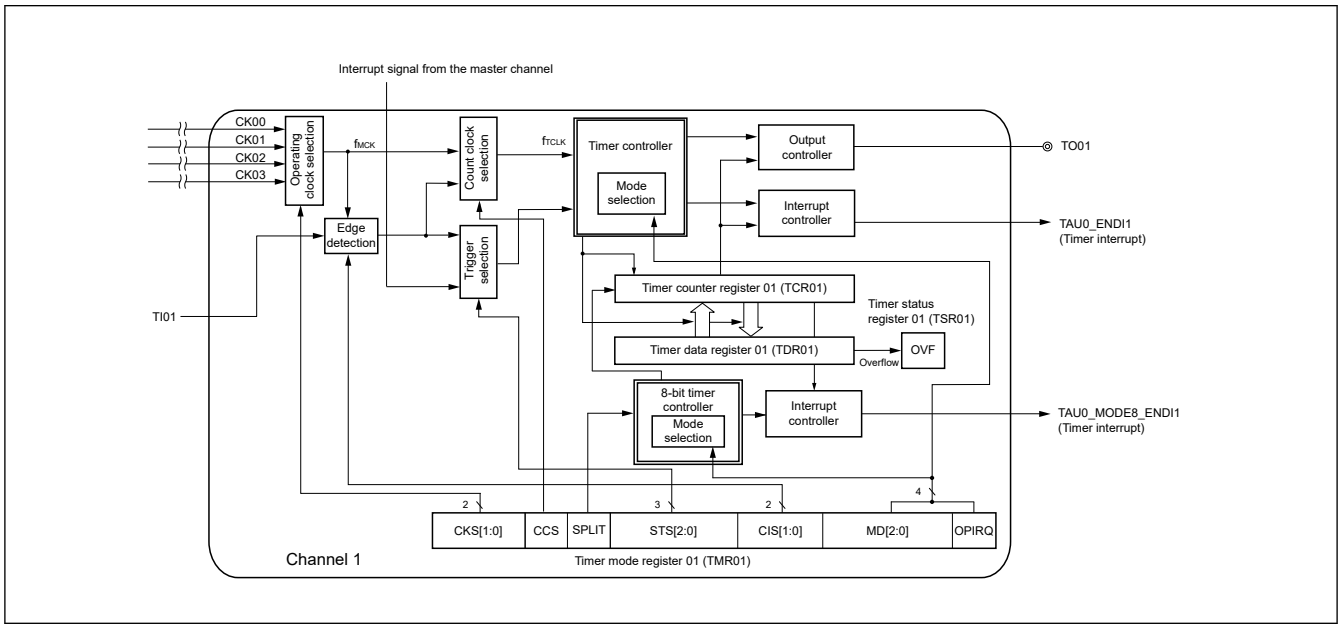
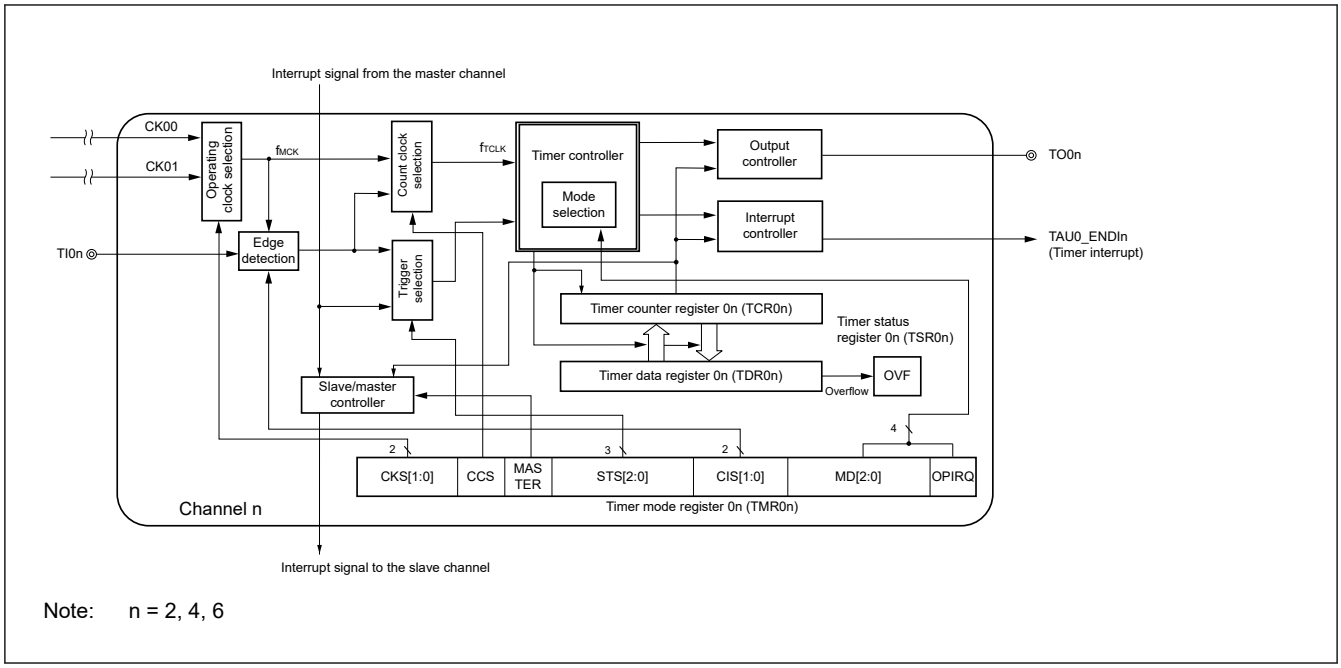


Figure 18.14 Internal block diagram of channel 1 of timer array unit



Note: n = 2, 4, 6

Figure 18.15 Internal block diagram of channels 2, 4, and 6 of timer array unit

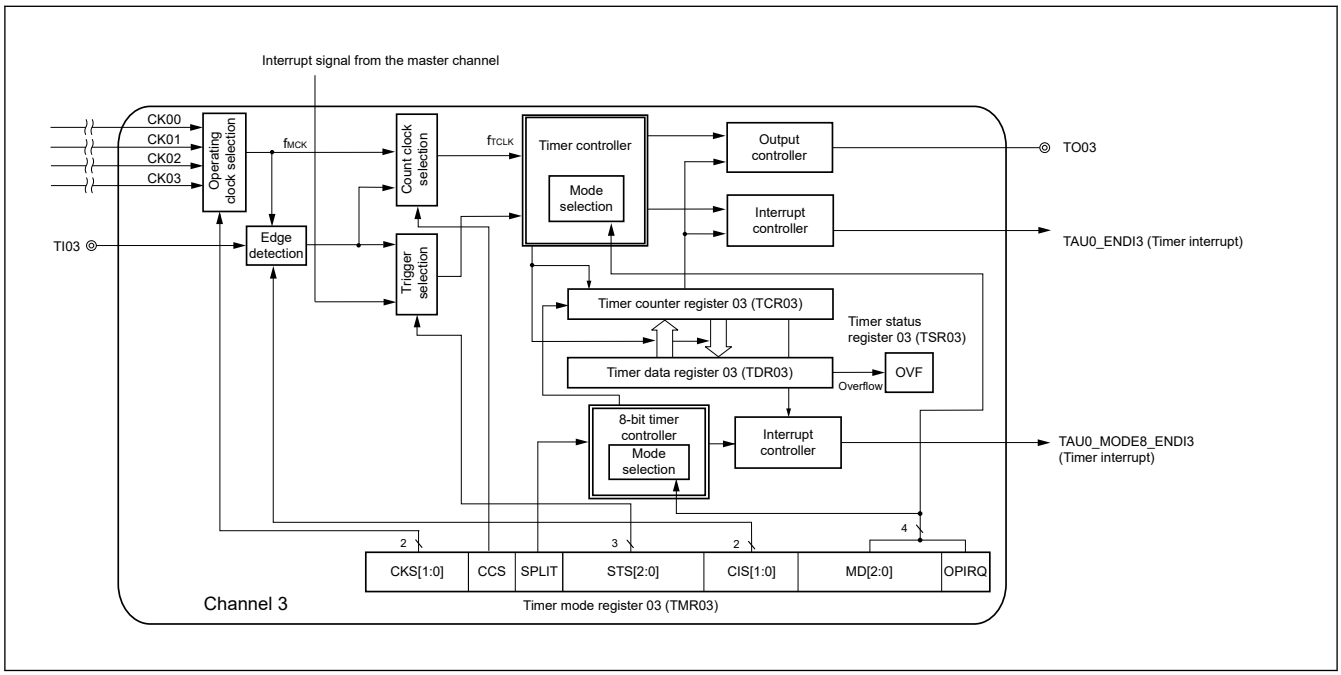


Figure 18.16 Internal block diagram of channel 3 of timer array unit

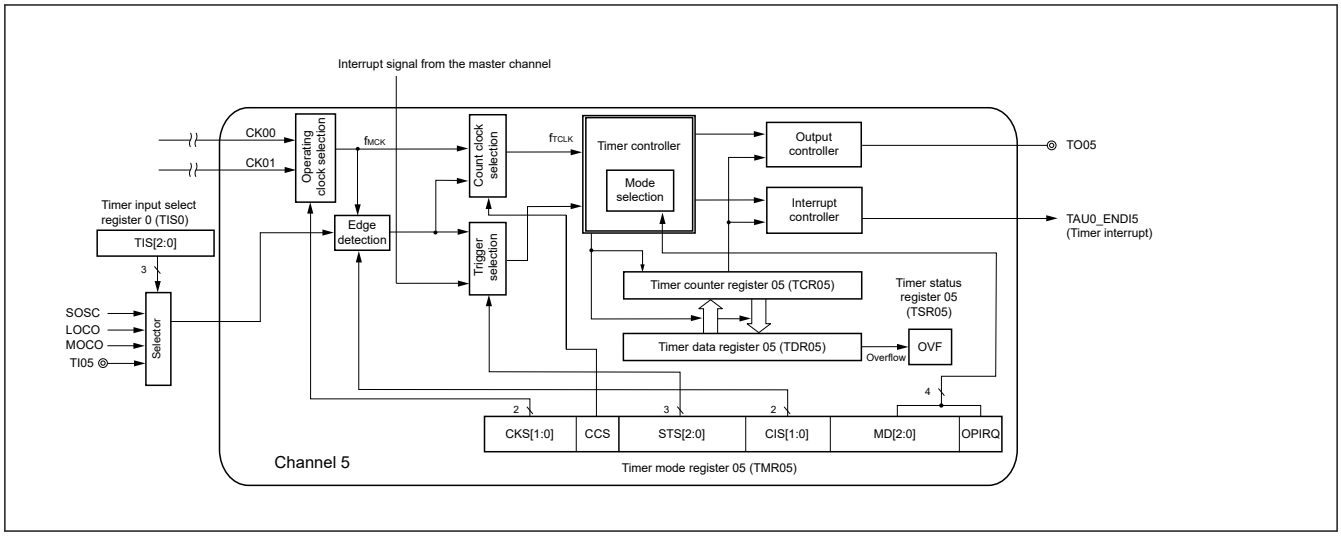


Figure 18.17 Internal block diagram of channel 5 of timer array unit

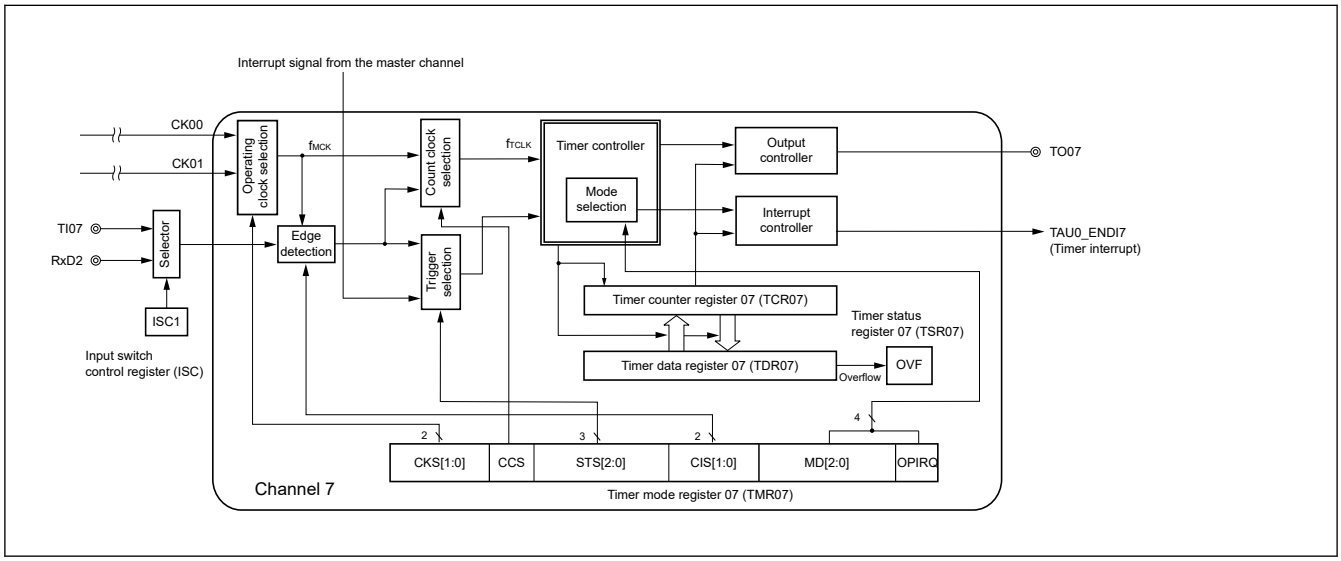


Figure 18.18 Internal block diagram of channel 7 of timer array unit

18.2 Register Descriptions

18.2.1 TCR0n : Timer Counter Register 0n (n = 0 to 7)

Base address: TAU = 0x4009_5000

Offset address: 0x0080 + 0x2 × n

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	n/a	16-bit clock count result for channel n	R

The TCR0n register is a 16-bit read-only register used to count clocks.

The value of this counter is incremented or decremented in synchronization with the rising edge of a count clock. Whether the counter is incremented or decremented depends on the operation mode that is selected by the MD[2:0] bits and OPIRQ

bit of timer mode register 0n (TMR0n) (see section 18.2.4. TMR0n : Timer Mode Register 0n (n = 0, 2, 4, 5, 6, 7) and section 18.2.5. TMR0n : Timer Mode Register 0n (n = 1, 3)).

The count value can be read by reading timer counter register 0n (TCR0n). The count value is set to 0xFFFF in the following cases.

- When the reset signal is generated
- When counting of the slave channel has been completed in the PWM output mode
- When counting of the slave channel has been completed in the delay count mode
- When counting of the master/slave channel has been completed in the one-shot pulse output mode
- When counting of the slave channel has been completed in the multiple PWM output mode

The count value is cleared to 0x0000 in the following cases.

- When the start trigger is input in the capture mode
- When capturing has been completed in the capture mode

Note: The count value is not captured to timer data register 0n (TDR0n) even when the TCR0n register is read.

The value read from the TCR0n register varies depending on the change to the operation mode and operating state as shown in the Table 18.3.

Table 18.3 Timer counter register 0n (TCR0n) read value in various operation modes

Operation mode	Count mode	Value read from the timer counter register 0n (TCR0n)*1			
		Value when the operation mode is changed after releasing reset	Value when count operation is temporarily stopped (TT0.TT[n] = 1)	Value when the operation mode is changed after count operation was temporarily stopped (TT0.TT[n] = 1)	Value when waiting fora start trigger after one count
Interval timer mode	Countdown	0xFFFF	Value when counting is stopped	Undefined	—
Capture mode	Count-up	0x0000	Value when counting is stopped	Undefined	—
Event counter mode	Countdown	0xFFFF	Value when counting is stopped	Undefined	—
One-count mode	Countdown	0xFFFF	Value when counting is stopped	Undefined	0xFFFF
Capture & one-count mode	Count-up	0x0000	Value when counting is stopped	Undefined	Captured value of TDR0n register + 1

Note 1. This indicates the value read from the TCR0n register when channel n has stopped operating as a timer (TE0.TE[n] = 0) and has been enabled to operate as a counter (TS0.TS[n] = 1). The read value is held in the TCR0n register until the count operation starts.

18.2.2 TDR0n/TDR01x/TDR03x : Timer Data Register 0n (n = 0 to 7) (x = L, H)

Base address: TAU = 0x4009_5000

Offset address: 0x0000 (TDR00)
0x0002 (TDR01/TDR01L)
0x0003 (TDR01H)
0x0004 (TDR02)
0x0006 (TDR03/TDR03L)
0x0007 (TDR03H)
0x0008 (TDR04)
0x000A (TDR05)
0x000C (TDR06)
0x000E (TDR07)

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	n/a	16-bit Timer capture result or setting compare data for channel n	R/W

This is a 16-bit register from which a capture function and a compare function can be selected.

The capture or compare function can be switched by selecting an operation mode by using the TMR0n.MD[2:0] bits and TMR0n.OPIRQ bit of timer mode register 0n (TMR0n).

The value of the TDR0n register can be changed at any time. This register can be read or written in 16-bit units.

In addition, for the TDR01 and TDR03 registers, while in the 8-bit timer mode (when the TMR01.SPLIT and TMR03.SPLIT bits are 1), it is possible to read and write the data in 8-bit units, with TDR01H and TDR03H used as the higher 8 bits, and TDR01L and TDR03L used as the lower 8 bits.

(i) When timer data register 0n (TDR0n) is used as compare register

Counting down is started from the value set to the TDR0n register. When the count value reaches 0x0000, an interrupt signal (TAU0_ENDIn) is generated. The TDR0n register holds its value until it is rewritten.

Note: The TDR0n register does not perform a capture operation even if a capture trigger is input, when it is set to the compare function.

(ii) When timer data register 0n (TDR0n) is used as capture register

The count value of timer counter register 0n (TCR0n) is captured to the TDR0n register when the capture trigger is input.

A valid edge of the TI0n pin can be selected as the capture trigger. This selection is made by timer mode register 0n (TMR0n).

18.2.3 TPS0 : Timer Clock Select Register 0

Base address: TAU = 0x4009_5000

Offset address: 0x00B6

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	PRS3[1:0]		—	—	PRS2[1:0]		PRS1[3:0]				PRS0[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	PRS0[3:0]	Selection of operation clock (CK00)*1 *2 *3 0x0: PCLKB 0x1: PCLKB/2 0x2: PCLKB/2^2 0x3: PCLKB/2^3 0x4: PCLKB/2^4 0x5: PCLKB/2^5 0x6: PCLKB/2^6 0x7: PCLKB/2^7 0x8: PCLKB/2^8 0x9: PCLKB/2^9 0xA: PCLKB/2^{10} 0xB: PCLKB/2^{11} 0xC: PCLKB/2^{12} 0xD: PCLKB/2^{13} 0xE: PCLKB/2^{14} 0xF: PCLKB/2^{15}	R/W
7:4	PRS1[3:0]	Selection of operation clock (CK01)*1 *2 *3 0x0: PCLKB 0x1: PCLKB/2 0x2: PCLKB/2^2 0x3: PCLKB/2^3 0x4: PCLKB/2^4 0x5: PCLKB/2^5 0x6: PCLKB/2^6 0x7: PCLKB/2^7 0x8: PCLKB/2^8 0x9: PCLKB/2^9 0xA: PCLKB/2^{10} 0xB: PCLKB/2^{11} 0xC: PCLKB/2^{12} 0xD: PCLKB/2^{13} 0xE: PCLKB/2^{14} 0xF: PCLKB/2^{15}	R/W
9:8	PRS2[1:0]	Selection of operation clock (CK02)*1 *4 0x0: PCLKB/2 0x1: PCLKB/2^2 0x2: PCLKB/2^4 0x3: PCLKB/2^6	R/W
11:10	—	These bits are read as 0. The write value should be 0.	R/W
13:12	PRS3[1:0]	Selection of operation clock (CK03)*1 *4 0x0: PCLKB/2^8 0x1: PCLKB/2^{10} 0x2: PCLKB/2^{12} 0x3: PCLKB/2^{14}	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. When changing the clock selected for PCLKB, stop the timer array unit (TT0 = 0x00FF).
Note 2. If PCLKB (undivided) is selected as the operation clock (CK0k) and TDR0n is set to 0x0000 (n = 0 to 7), interrupt requests output from the timer array units cannot be used.
Note 3. Waveform of the clock to be selected in the TPS0 register which becomes high level for one period of PCLKB from its rising edge. For details, see section 18.4.1. Count Clock (f_{TCLK}).
Note 4. The timer array unit must also be stopped if the operating clock (f_{MCK}) or the valid edge of the signal input from the TI0n pin is selected.

The TPS0 register is a 16-bit register used to select two types or four types of operation clocks (CK00, CK01, CK02, CK03) that are commonly supplied to each channel. CK00 is selected by using the PRS0[3:0] bits, and CK01 is selected by using the PRS1[3:0] bits. In addition, only for channels 1 and 3, CK02 and CK03 can be also selected. CK02 is selected by using the PRS2[1:0] bits, and CK03 is selected by using the PRS3[1:0] bits.

Rewriting of the TPS0 register during timer operation is possible only in the following cases.

- If the PRS0[3:0] bits can be rewritten (n = 0 to 7):
  All channels for which CK00 is selected as the operation clock (TMR0n.CKS[1:0] = 00b) are stopped (TE0.TE[n] = 0).
- If the PRS1[3:0] bits can be rewritten (n = 0 to 7):
  All channels for which CK01 is selected as the operation clock (TMR0n.CKS[1:0] = 01b) are stopped (TE0.TE[n] = 0).
- If the PRS2[1:0] bits can be rewritten (n = 1, 3):
  All channels for which CK02 is selected as the operation clock (TMR0n.CKS[1:0] = 10b) are stopped (TE0.TE[n] = 0).
- If the PRS3[1:0] bits can be rewritten (n = 1, 3):
  All channels for which CK03 is selected as the operation clock (TMR0n.CKS[1:0] = 11b) are stopped (TE0.TE[n] = 0).

PRS0[3:0] bits (Selection of operation clock (CK00))

The input sources that can be selected with the PRS0[3:0] bits are shown in Table 18.4.

PRS1[3:0] bits (Selection of operation clock (CK01))

The input sources that can be selected with the PRS1[1:0] bits are shown in Table 18.4.

Table 18.4 Selection of operation clock (PRSk (k = 0, 1))

PRSk[3:0]	Selection of operation clock (CK0k)*1 (k = 0, 1)						
		PCLKB = 2 MHz	PCLKB = 5 MHz	PCLKB = 10 MHz	PCLKB = 20 MHz	PCLKB = 32 MHz	PCLKB = 48 MHz
0x0	PCLKB	2 MHz	5 MHz	10 MHz	20 MHz	32 MHz	48 MHz
0x1	PCLKB/2	1 MHz	2.5 MHz	5 MHz	10 MHz	16 MHz	24 MHz
0x2	PCLKB/2^2	500 kHz	1.25 MHz	2.5 MHz	5 MHz	8 MHz	12 MHz
0x3	PCLKB/2^3	250 kHz	625 kHz	1.25 MHz	2.5 MHz	4 MHz	6 MHz
0x4	PCLKB/2^4	125 kHz	313 kHz	625 kHz	1.25 MHz	2 MHz	3 MHz
0x5	PCLKB/2^5	62.5 kHz	156 kHz	313 kHz	625 kHz	1 MHz	1.5 MHz
0x6	PCLKB/2^6	31.3 kHz	78.1 kHz	156 kHz	313 kHz	500 kHz	750 kHz
0x7	PCLKB/2^7	15.6 kHz	39.1 kHz	78.1 kHz	156 kHz	250 kHz	375 kHz
0x8	PCLKB/2^8	7.81 kHz	19.5 kHz	39.1 kHz	78.1 kHz	125 kHz	187.5 kHz
0x9	PCLKB/2^9	3.91 kHz	9.77 kHz	19.5 kHz	39.1 kHz	62.5 kHz	93.8 kHz
0xA	PCLKB/2^{10}	1.95 kHz	4.88 kHz	9.77 kHz	19.5 kHz	31.3 kHz	46.9 kHz
0xB	PCLKB/2^{11}	977 Hz	2.44 kHz	4.88 kHz	9.77 kHz	15.6 kHz	23.4 kHz
0xC	PCLKB/2^{12}	488 Hz	1.22 kHz	2.44 kHz	4.88 kHz	7.81 kHz	11.7 kHz
0xD	PCLKB/2^{13}	244 Hz	610 Hz	1.22 kHz	2.44 kHz	3.91 kHz	5.86 kHz
0xE	PCLKB/2^{14}	122 Hz	305 Hz	610 Hz	1.22 kHz	1.95 kHz	2.93 kHz
0xF	PCLKB/2^{15}	61.0 Hz	153 Hz	305 Hz	610 Hz	977 Hz	1.46 kHz

Note: If PCLKB (undivided) is selected as the operation clock (CK0k) and TDR0n is set to 0x0000 (n = 0 to 7), interrupt requests output from timer array units cannot be used.

Note: Waveform of the clock to be selected in the TPS0 register which becomes high level for one period of PCLKB from its rising edge. For details, see section 18.4.1. Count Clock (f_{TCLK}).

Note 1. When changing the clock selected for PCLKB, stop timer array unit (TT0 = 0x00FF).

PRS2[1:0] bits (Selection of operation clock (CK02))

The input sources that can be selected with the PRS2[1:0] bits are shown in Table 18.5.

Table 18.5 Selection of operation clock (PRS2[1:0])

PRS2[1:0]	Selection of operation clock (CK02)*1						
		PCLKB = 2 MHz	PCLKB = 5 MHz	PCLKB = 10 MHz	PCLKB = 20 MHz	PCLKB = 32 MHz	PCLKB = 48 MHz
00b	PCLKB/2	1 MHz	2.5 MHz	5 MHz	10 MHz	16 MHz	24 MHz
01b	PCLKB/2^2	500 kHz	1.25 MHz	2.5 MHz	5 MHz	8 MHz	12 MHz
10b	PCLKB/2^4	125 kHz	313 kHz	625 kHz	1.25 MHz	2 MHz	3 MHz
11b	PCLKB/2^6	31.3 kHz	78.1 kHz	156 kHz	313 kHz	500 kHz	750 kHz

Note 1. When changing the clock selected for PCLKB, stop timer array unit (TT0 = 0x00FF).
The timer array unit must also be stopped if the operating clock (f_{MCK}) or the valid edge of the signal input from the TI0n pin is selected.

PRS3[1:0] bits (Selection of operation clock (CK03))

The input sources that can be selected with the PRS3[1:0] bits are shown in Table 18.6.

Table 18.6 Selection of operation clock (PRS3[1:0])

PRS3[1:0]	Selection of operation clock (CK03)*1						
		PCLKB = 2 MHz	PCLKB = 5 MHz	PCLKB = 10 MHz	PCLKB = 20 MHz	PCLKB = 32 MHz	PCLKB = 48 MHz
00b	PCLKB/2^8	7.81 kHz	19.5 kHz	39.1 kHz	78.1 kHz	125 kHz	188 kHz
01b	PCLKB/2^{10}	1.95 kHz	4.88 kHz	9.77 kHz	19.5 kHz	31.3 kHz	46.9 kHz
10b	PCLKB/2^{12}	488 Hz	1.22 kHz	2.44 kHz	4.88 kHz	7.81 kHz	11.7 kHz
11b	PCLKB/2^{14}	122 Hz	305 Hz	610 Hz	1.22 kHz	1.95 kHz	2.93 kHz

Note 1. When changing the clock selected for PCLKB, stop timer array unit (TT0 = 0x00FF).
The timer array unit must also be stopped if the operating clock (f_{MCK}) or the valid edge of the signal input from the TI0n pin is selected.

By using channels 1 and 3 in the 8-bit timer mode and specifying CK02 or CK03 as the operation clock, the interval times shown in Table 18.7 can be achieved by using the interval timer function.

Table 18.7 Interval times available for operation clock CK02 or CK03

Clock		Interval time*1 (PCLKB = 32 MHz)			
		10 µs	100 µs	1 ms	10 ms
CK02	PCLKB/2	✓	—	—	—
	PCLKB/2^2	✓	—	—	—
	PCLKB/2^4	✓	✓	—	—
	PCLKB/2^6	✓	✓	—	—
CK03	PCLKB/2^8	—	✓	✓	—
	PCLKB/2^{10}	—	✓	✓	—
	PCLKB/2^{12}	—	—	✓	✓
	PCLKB/2^{14}	—	—	✓	✓

Note: For details of a signal of PCLKB/2^j selected with the TPS0 register, see section 18.4.1. Count Clock (f_{TCLK}).

Note 1. The margin is within 5%.

18.2.4 TMR0n : Timer Mode Register 0n (n = 0, 2, 4, 5, 6, 7)

Base address: TAU = 0x4009_5000

Offset address: 0x0090 (TMR00)
0x0094 (TMR02)
0x0098 (TMR04)
0x009A (TMR05)
0x009C (TMR06)
0x009E (TMR07)

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CKS[1:0]		—	CCS	MASTER	STS[2:0]			CIS[1:0]		—	—	MD[2:0]			OPIRQ
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	OPIRQ	Setting of starting count and interrupt	R/W
3:1	MD[2:0]	Selection of operation mode at channel n 0 0 0: Interval timer mode 0 1 0: Capture mode 0 1 1: Event counter mode 1 0 0: One-count mode 1 1 0: Capture and one-count mode Others: Setting prohibited	R/W
5:4	—	These bits are read as 0. The write value should be 0.	R/W
7:6	CIS[1:0]	Selection of TI0n pin input valid edge 0 0: Falling edge 0 1: Rising edge 1 0: Both edges (when low-level width is measured) Start trigger: Falling edge, Capture trigger: Rising edge 1 1: Both edges (when high-level width is measured) Start trigger: Rising edge, Capture trigger: Falling edge	R/W
10:8	STS[2:0]	Setting of start trigger or capture trigger of channel n 0 0 0: Only software trigger start is valid (other trigger sources are unselected). 0 0 1: Valid edge of the TI0n pin input is used as both the start trigger and capture trigger. 0 1 0: Both the edges of the TI0n pin input are used as a start trigger and a capture trigger. 1 0 0: Interrupt signal of the master channel is used (when the channel is used as a slave channel with the simultaneous channel operation function). Others: Setting prohibited	R/W
11[*1]	MASTER	Selection between using channel n independently or simultaneously with another channel (as a slave or master) 0: Operates in independent channel operation function or as slave channel in simultaneous channel operation function. 1: Operates as master channel in simultaneous channel operation function.	R/W
12	CCS	Selection of counter clock (f_{TCLK}) of channel n 0: Operating clock (f_{MCK}) specified by the CKS[1:0] bits 1: Valid edge of input signal input from the TI0n pin. • In channel 5, valid edge of input signal selected by the TIS0 register • In channel 7, valid edge of input signal selected by the ISC register	R/W
13	—	This bit is read as 0. The write value should be 0.	R/W
15:14	CKS[1:0]	Selection of operation clock (f_{MCK}) of channel n 0 0: Operation clock CK00 set by timer clock select register 0 (TPS0) 0 1: Operation clock CK02 set by timer clock select register 0 (TPS0) 1 0: Operation clock CK01 set by timer clock select register 0 (TPS0) 1 1: Operation clock CK03 set by timer clock select register 0 (TPS0)	R/W

Note: The timer array unit must be stopped (TT0 = 0x00FF) if the clock selected for PCLKB is changed even if the operating clock specified by using the CKS[1:0] bits (f_{MCK}) or the valid edge of the signal input from the TI0n pin is selected as the count clock (f_{TCLK}).

Note 1. Not supported when n = 0, 5, 7. The TMR00.MASTER, TMR05.MASTER, and TMR07.MASTER bits are fixed to 0. Writing to these bits are ignored.

The TMR0n register sets an operation mode of channel n. This register is used to select the operation clock (f_{MCK}), select the count clock, select the master or slave, select the 16-bit timer, specify the start trigger and capture trigger, select the valid edge of the timer input, and specify the operation mode (interval, capture, event counter, one-count, or capture and one-count).

Rewriting the TMR0n register is prohibited when the register is in operation (when TE0.TE[n] = 1). However, the CIS[1:0] bits can be rewritten even while the register is operating with some functions (when TE0.TE[n] = 1). For details, see section 18.7. Independent Channel Operation Function of Timer Array Unit and section 18.8. Simultaneous Channel Operation Function of Timer Array Unit.

OPIRQ bit (Setting of starting count and interrupt)

Table 18.8 lists operation mode that can be selected with MD[2:0] bits and OPIRQ bit.

Table 18.8 Operation mode selected with OPIRQ bit

Operation mode (MD[2:0])	OPIRQ	Setting of starting count and interrupt
• Interval timer mode (000b) • Capture mode (010b)	0	Timer interrupt is not generated when counting is started (timer output does not change, either).
	1	Timer interrupt is generated when counting is started (timer output also changes).
Event counter mode (011b)	0	Timer interrupt is not generated when counting is started (timer output does not change, either).
One-count mode (100b)*1	0	Start trigger is invalid during counting operation. At that time, interrupt is not generated.
	1	Start trigger is valid during counting operation *2. At that time, interrupt is generated.
Capture & one-count mode (110b)	0	Timer interrupt is not generated when counting is started (timer output does not change, either). Start trigger is invalid during counting operation. At that time interrupt is not generated.
Other than above		Setting prohibited

Note 1. In one-count mode, interrupt output (TAU0_ENDIn) when starting a count operation and TO0n output are not controlled.
Note 2. If the start trigger (TS0.TS[n] = 1) is issued during operation, the counter is initialized, and recounting is started (interrupt request does not occur).

MD[2:0] bits (Selection of operation mode at channel n)

The operation in each mode varies depending on the TMR0n.OPIRQ bit (see Table 18.8).

Table 18.9 lists operation mode that can be selected with MD[2:0] bits.

Table 18.9 Operation mode selected with MD[2:0] bits (1 of 2)

MD[2:0]	Operation mode of channel n	Corresponding function	Count operation of TCR
000b	Interval timer mode	Interval timer or Square wave output or Divider function or PWM output (master)	Counting down
010b	Capture mode	Input pulse interval measurement	Counting up
011b	Event counter mode	External event counter	Counting down
100b	One-count mode	Delay counter or One-shot pulse output or PWM output (slave)	Counting down

Table 18.9 Operation mode selected with MD[2:0] bits (2 of 2)

MD[2:0]	Operation mode of channel n	Corresponding function	Count operation of TCR
110b	Capture & one-count mode	Measurement of high- or low- level width of input signal	Counting up
Other than above	Setting prohibited		

CIS[1:0] bits (Selection of TI0n pin input valid edge)

If both the edges are specified when the value of the STS[2:0] bits is other than 010b, set the CIS[1:0] bits to 10b.

STS[2:0] bits (Setting of start trigger or capture trigger of channel n)

These bits are used for setting the start trigger or capture trigger of channel n.

MASTER bit (Selection between using channel n independently or simultaneously with another channel (as a slave or master))

Only channel 2, 4, or 6 can be set as a master channel (MASTER = 1). Be sure to use channel 0, 5, or 7 which is fixed to 0. Regardless of the bit setting, channel 0 operates as master, because it is the highest channel. Clear the MASTER bit to 0 for a channel that is used with the independent channel operation function.

CCS bit (Selection of counter clock (f_{TCLK}) of channel n)

Counter clock (f_{TCLK}) is used for the counter, output controller, and interrupt controller.

CKS[1:0] bits (Selection of operation clock (f_{MCK}) of channel n)

Operation clock (f_{MCK}) is used by the edge detector. A count clock (f_{TCLK}) and a sampling clock are generated depending on the setting of the CCS bit.

The operation clocks CK02 and CK03 can only be selected for channels 1 and 3.

18.2.5 TMR0n : Timer Mode Register 0n (n = 1, 3)

Base address: TAU = 0x4009_5000

Offset address: 0x0092 (TMR01)
0x0096 (TMR03)

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CKS[1:0]		—	CCS	SPLIT	STS[2:0]			CIS[1:0]		—	—	MD[2:0]			OPIRQ
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	OPIRQ	Setting of starting count and interrupt	R/W
3:1	MD[2:0]	Selection of operation mode at channel n 0 0 0: Interval timer mode 0 1 0: Capture mode 0 1 1: Event counter mode 1 0 0: One-count mode 1 1 0: Capture & one-count mode Others: Setting prohibited	R/W
5:4	—	These bits are read as 0. The write value should be 0.	R/W
7:6	CIS[1:0]	Selection of TI0n pin input valid edge 0 0: Falling edge 0 1: Rising edge 1 0: Both edges (when low-level width is measured) Start trigger: Falling edge, Capture trigger: Rising edge 1 1: Both edges (when high-level width is measured) Start trigger: Rising edge, Capture trigger: Falling edge	R/W

Bit	Symbol	Function	R/W
10:8	STS[2:0]	Setting of start trigger or capture trigger of channel n 0 0 0: Only software trigger start is valid (other trigger sources are unselected). 0 0 1: Valid edge of the TI0n pin input is used as both the start trigger and capture trigger. 0 1 0: Both the edges of the TI0n pin input are used as a start trigger and a capture trigger. 1 0 0: Interrupt signal of the master channel is used (when the channel is used as a slave channel with the simultaneous channel operation function). Others: Setting prohibited	R/W
11	SPLIT	Selection of 8 or 16-bit timer operation for channels 1 and 3 0: Operates as 16-bit timer (Operates in independent channel operation function or as slave channel in simultaneous channel operation function.) 1: Operates as 8-bit timer	R/W
12	CCS	Selection of counter clock (f_{TCLK}) of channel n 0: Operating clock (f_{MCK}) specified by the CKS[1:0] bits 1: Valid edge of input signal input from the TI0n pin	R/W
13	—	This bit is read as 0. The write value should be 0.	R/W
15:14	CKS[1:0]	Selection of operation clock (f_{MCK}) of channel n 0 0: Operation clock CK00 set by timer clock select register 0 (TPS0) 0 1: Operation clock CK02 set by timer clock select register 0 (TPS0) 1 0: Operation clock CK01 set by timer clock select register 0 (TPS0) 1 1: Operation clock CK03 set by timer clock select register 0 (TPS0)	R/W

Note: The timer array unit must be stopped (TT0 = 0x00FF) if the clock selected for PCLKB is changed, even if the operating clock specified by using the CKS[1:0] bits (f_{MCK}) or the valid edge of the signal input from the TI0n pin is selected as the count clock(f_{TCLK}).

The TMR0n register sets an operation mode of channel n. This register is used to select the operation clock (f_{MCK}), select the count clock, select the master or slave, select the 16 or 8-bit timer (only for channels 1 and 3), specify the start trigger and capture trigger, select the valid edge of the timer input, and specify the operation mode (interval, capture, event counter, one-count, or capture & one-count).

Rewriting the TMR0n register is prohibited when the register is in operation (when TE0.TE[n] = 1). However, the CIS[1:0] bits can be rewritten even while the register is operating with some functions (when TE0.TE[n] = 1). For details, see section 18.7. Independent Channel Operation Function of Timer Array Unit and section 18.8. Simultaneous Channel Operation Function of Timer Array Unit.

OPIRQ bit (Setting of starting count and interrupt)

Table 18.8 lists operation mode that can be selected with MD[2:0] bits and OPIRQ bit.

MD[2:0] bits (Selection of operation mode at channel n)

The operation in each mode varies depending on the OPIRQ bit (see Table 18.8).

Table 18.9 lists operation mode that can be selected with MD[2:0] bits.

CIS[1:0] bits (Selection of TI0n pin input valid edge)

If both the edges are specified when the value of the STS[2:0] bits is other than 010b, set the CIS[1:0] bits to 10b.

STS[2:0] bits (Setting of start trigger or capture trigger of channel n)

These bits are used for setting the start trigger or capture trigger of channel n.

SPLIT bit (Selection of 8 or 16-bit timer operation for channels 1 and 3)

This bit is used to select 8 or 16-bit timer operation for channels 1 and 3.

CCS bit (Selection of counter clock (f_{TCLK}) of channel n)

Counter clock (f_{TCLK}) is used for the counter, output controller, and interrupt controller.

CKS[1:0] bits (Selection of operation clock (f_{MCK}) of channel n)

Operation clock (f_{MCK}) is used by the edge detector. A count clock (f_{TCLK}) and a sampling clock are generated depending on the setting of the CCS bit.

18.2.6 TSR0n : Timer Status Register 0n (n = 0 to 7)

Base address: TAU = 0x4009_5000

Offset address: 0x00A0 + 0x2 × n

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OVF
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	OVF	Counter overflow state of channel n 0: Overflow does not occur 1: Overflow occurs	R
15:1	—	These bits are read as 0. The write value should be 0.	R

The TSR0n register indicates the overflow state of the counter of channel n.

The TSR0n register is valid only in the capture mode (TMR0n.MD[2:0] = 010b) and capture & one-count mode (TMR0n.MD[2:0] = 110b). See Table 18.10 for the operation of the OVF bit in each operation mode and set or clear conditions.

The 8 lower-order bits of a TSR0n register can be handled as TSR0nL, which can be read by an 8-bit access.

OVF bit (Counter overflow state of channel n)

When OVF = 1, this flag is cleared (OVF = 0) when the next value is captured without overflow.

Table 18.10 shows the OVF bit operation, set, and clear conditions in each operation mode.

Table 18.10 OVF bit operation, set and clear conditions in each operation mode

Timer operation mode	OVF bit	Set and clear conditions
• Capture mode • Capture & one-count mode	clear	When no overflow has occurred upon capturing
	set	When an overflow has occurred upon capturing
• Interval timer mode • Event counter mode • One-count mode	clear	— (Use prohibited)
	set	

Note: The OVF bit does not change immediately after the counter has overflowed, but changes upon the subsequent capture.

18.2.7 TE0 : Timer Channel Enable Status Register 0

Base address: TAU = 0x4009_5000

Offset address: 0x00B0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	TEH3	—	TEH1	—	TE[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	TE[7:0]	Indication of operation enabled or disabled state of channel n These bits display whether operation of the lower 8-bit timer for the TE[1] and TE[3] bits is enabled or disabled when channel 1 or 3 is in the 8-bit timer mode. 0: Operation is disabled 1: Operation is enabled	R
8	—	This bit is read as 0.	R
9	TEH1	Indication of whether operation of the higher 8-bit timer is enabled or disabled when channel 1 is in the 8-bit timer mode 0: Operation is disabled 1: Operation is enabled	R
10	—	This bit is read as 0.	R
11	TEH3	Indication of whether operation of the higher 8-bit timer is enabled or disabled when channel 3 is in the 8-bit timer mode 0: Operation is disabled 1: Operation is enabled	R
15:12	—	These bits are read as 0.	R

The TE0 register is used to enable or disable the timer operation of each channel.

Each bit of the TE0 register corresponds to each bit of the timer channel start register 0 (TS0) and the timer channel stop register 0 (TT0). When a bit of the TS0 register is set to 1, the corresponding bit of this register is set to 1. When a bit of the TT0 register is set to 1, the corresponding bit of this register is cleared to 0.

The lower 8 bits of the TE0 register can be read by an 8-bit access.

18.2.8 TS0 : Timer Channel Start Register 0

Base address: TAU = 0x4009_5000

Offset address: 0x00B2

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	TSH3	—	TSH1	—	TS[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	TS[7:0]	Operation enable (start) trigger of channel n These bits are the trigger to enable operation (start operation) of the lower 8-bit timer for the TS[1] and TS[3] bits when channel 1 or 3 is in the 8-bit timer mode. 0: No trigger operation 1: The TE0.TE[n] bit is set to 1 and the count operation becomes enabled	R/W
8	—	This bit is read as 0. The write value should be 0.*1	R/W
9	TSH1	Trigger to enable operation (start operation) of the higher 8-bit timer when channel 1 is in the 8-bit timer mode 0: No trigger operation 1: The TE0.TEH1 bit is set to 1 and the count operation becomes enabled	R/W
10	—	This bit is read as 0. The write value should be 0.*1	R/W
11	TSH3	Trigger to enable operation (start operation) of the higher 8-bit timer when channel 3 is in the 8-bit timer mode 0: No trigger operation 1: The TE0.TEH3 bit is set to 1 and the count operation becomes enabled	R/W
15:12	—	These bits are read as 0. The write value should be 0.*1	R/W

Note: When switching from a function that does not use TI0n pin input to one that does, the following wait period is required from when timer mode register 0n (TMR0n) is set until the TS[n] (TSH1, TSH3) bit is set to 1.
When the TI0n pin noise filter is enabled (TNFEN.TNFEN0n = 1): 4 cycles of the operation clock (f_{MCK})
When the TI0n pin noise filter is disabled (TNFEN.TNFEN0n = 0): 2 cycles of the operation clock (f_{MCK})

Note: When the TS0 register is read, 0 is always read.

Note 1. Be sure to clear bits [15:12], bit [10], and bit [8] to 0.

The TS0 register is a trigger register that is used to initialize timer counter register 0n (TCR0n) and start the counting operation of each channel.

When a bit of this register is set to 1, the corresponding bit of timer channel enable status register 0 (TE0) is set to 1. The TS[n], TSH1, TSH3 bits are immediately cleared when operation is enabled (TE0.TE[n], TEH1, TEH3 = 1), because they are trigger bits.

18.2.9 TT0 : Timer Channel Stop Register 0

Base address: TAU = 0x4009_5000

Offset address: 0x00B4

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	TTH3	—	TTH1	—	TT[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	TT[7:0]	Operation stop trigger of channel n These bits are the trigger to stop operation of the lower 8-bit timer for the TT[1] and TT[3] bits when channel 1 or 3 is in the 8-bit timer mode. 0: No trigger operation 1: The TE0.TE[n] bit is cleared to 0 and the count operation is stopped	R/W
8	—	This bit is read as 0. The write value should be 0.*1	R/W
9	TTH1	Trigger to stop operation of the higher 8-bit timer when channel 1 is in the 8-bit timer mode 0: No trigger operation 1: The TE0.TEH1 bit is cleared to 0 and the count operation is stopped	R/W
10	—	This bit is read as 0. The write value should be 0.*1	R/W
11	TTH3	Trigger to stop operation of the higher 8-bit timer when channel 3 is in the 8-bit timer mode 0: No trigger operation 1: The TE0.TEH3 bit is cleared to 0 and the count operation is stopped	R/W
15:12	—	These bits are read as 0. The write value should be 0.*1	R/W

Note: When the TT0 register is read, 0 is always read.
Note 1. Be sure to clear bits [15:12], bit [10], and bit [8] to 0.

The TT0 register is a trigger register that is used to stop the counting operation of each channel.

When a bit of this register is set to 1, the corresponding bit of timer channel enable status register 0 (TE0) is cleared to 0. The TT0.TT[n], TTH1, TTH3 bits are immediately cleared when operation is stopped (TE0.TE[n], TEH1, TEH3 = 1), because they are trigger bits.

18.2.10 TIS0 : Timer Input Select Register 0

Base address: PORGA = 0x4009_1000

Offset address: 0x0004

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

Bit	Symbol	Function	R/W
2:0	TIS[2:0]	Selection of timer input used with channel 5 0 0 0: Input signal of timer input pin (TI05) 0 1 1: Middle-speed on-chip oscillator (MOCO) 1 0 0: Low-speed on-chip oscillator (LOCO) 1 0 1: Sub-clock oscillator (SOSC) Others: Setting prohibited	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: Make sure that both the high-level and low-level widths of timer input to be selected are no less than $1/f_{MCK}$ + 10 ns. Therefore, when selecting SOSC as PCLKB, the TIS[2] bit cannot be set to 1.

The TIS0 register is used to select channel 5 timer input.

18.2.11 TOE0 : Timer Output Enable Register 0

Base address: TAU = 0x4009_5000

Offset address: 0x00BA

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	TOE[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	TOE[7:0]	Enabling or disabling timer output for channel n 0: Disables timer output 1: Enables timer output	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W

The TOE0 register is used to enable or disable timer output of each channel.

Channel n for which timer output has been enabled becomes unable to rewrite the value of the TO[n] bit of timer output register 0 (TO0) described later by software, and the value reflecting the setting of the timer output function through the count operation is output from the timer output pin (TO0n).

The lower 8 bits of the TOE0 register can be set with an 8-bit access.

TOE[7:0] bits (Enabling or disabling timer output for channel n)

When set TOE[n] = 0

The corresponding TO0.TO[n] bit does not reflect timer operation with this setting, so the output level of the TO0.TO[n] bit is fixed to the level written to the TO0 register.

Writing to the TO0.TO[n] bit is enabled and the level set in the TO0.TO[n] bit is output from the TO0n pin.

When set TOE[n] = 1

The corresponding TO0.TO[n] bit reflects timer operation with this setting, so the output waveform is generated.

Writing to the TO0.TO[n] bit is ignored.

Note: n: Channel number (n = 0 to 7).

18.2.12 TO0 : Timer Output Register 0

Base address: TAU = 0x4009_5000

Offset address: 0x00B8

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	TO[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	TO[7:0]	Timer output of channel n 0: Timer output value is 0 1: Timer output value is 1	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W

The TO0 register is a buffer register of timer output of each channel.

The value of each bit in this register is output from the timer output pin (TO0n) of each channel.

The TO0n bit on this register can be rewritten by software only when timer output is disabled (TOE0.TOE[n] = 0). When timer output is enabled (TOE0.TOE[n] = 1), rewriting this register by software is ignored, and the value is changed only by the timer operation.

To use the port functions multiplexed with the inputs and outputs of timer array units, select the function by setting the PSEL[4:0] bits in Port gh Pin Function Select Register (PghPFS). For details, see section 16, I/O Ports.

The lower 8 bits of the TO0 register can be set with an 8-bit access.

Note: n: Channel number (n = 0 to 7)
gh: Port number (g = 0 to 4, h = 00 to 15)

18.2.13 TOL0 : Timer Output Level Register 0

Base address: TAU = 0x4009_5000

Offset address: 0x00BC

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

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
7:1	TOL[6:0]	Control of timer output of channel n 0: Positive logic output (active-high) 1: Negative logic output (active-low)	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W

Note: If the value of this register is rewritten during timer operation, the timer output logic is inverted when the timer output signal changes next, instead of immediately after the register value is rewritten.

The TOL0 register controls the timer output level of each channel.

The setting of the inverted output of channel n by this register is reflected at the timing of set or reset of the timer output signal while the timer output is enabled (TOE0.TOE[n] = 1) in the Slave channel output mode (TOM0.TOM[n − 1] = 1). In the master channel output mode (TOM0.TOM[n − 1] = 0), this register setting is invalid.

The lower 8 bits of the TOL0 register can be set with an 8-bit access.

Note: n: Channel number (n = 0 to 7)

18.2.14 TOM0 :Timer Output Mode Register 0

Base address: TAU = 0x4009_5000

Offset address: 0x00BE

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

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
7:1	TOM[6:0]	Control of timer output mode of channel n 0: Master channel output mode (to produce toggled output by timer interrupt request signal (TAU0_ENDIn)) 1: Slave channel output mode (output is set by the timer interrupt request signal (TAU0_ENDIn) of the master channel, and reset by the timer interrupt request signal (TAU0_ENDIp) of the slave channel)	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W

The TOM0 register is used to control the timer output mode of each channel.

When a channel is used for the independent channel operation function, set the corresponding bit of the channel to be used to 0.

When a channel is used for the simultaneous channel operation function (PWM output, one-shot pulse output, or multiple PWM output), set the corresponding bit of the master channel to 0 and the corresponding bit of the slave channel to 1.

The setting of each channel n by this register is reflected at the timing when the timer output signal is set or reset while the timer output is enabled (TOE0.TOE[n] = 1).

The lower 8 bits of the TOM0 register can be set with an 8-bit access.

Note: n: Channel number (n = 0 to 7) (n = 0, 2, 4, 6 for master channel), p: Slave channel number (n < p ≤ 7)

For details of the relation between the master channel and slave channel, see section 18.3.1. Basic Rules of Simultaneous Channel Operation Function.

18.2.15 ISC : Input Switch Control register

Base address: PORGA = 0x4009_1000

Offset address: 0x0003

Bit position:	7	6	5	4	3	2	1	0
Bit field:	ISC76[1:0]		—	ISC43[1:0]		—	ISC1	—
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	ISC1[*1]	Switching channel 7 input of timer array unit 0: Uses the input signal of the TI07 pin as a timer input (normal operation) 1: Input signal of the RxD2 pin is used as timer input (detects the wakeup signal and measures the low width of the break field and the pulse width of the sync field).	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W
4:3	ISC43[1:0][*2]	Switch of the serial clock input source of SPI00 0 0: Input signal of the SCK00 pin (normal operation) 1 0: TO01 output signal Others: Setting prohibited	R/W
5	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
7:6	ISC76[1:0]*3	Switch of the serial clock input source of SPI01 0 0: Input signal of the SCK01 pin (normal operation) 1 0: TO01 output signal Others: Setting prohibited	R/W

Note 1. When the LIN-bus communication function is used, select the input signal of the RxD2 pin by setting the ISC1 bit to 1.
Note 2. When UART mode or simplified I^2C mode is to be selected for channel 0, set the ISC43[1:0] bits to 0.
Note 3. When UART mode or simplified I^2C mode is to be selected for channel 1, set the ISC76[1:0] bits to 0.

ISC1 bit (Switching channel 7 input of timer array unit)

The ISC1 bit is used to realize a LIN-bus communication operation by UART2 in coordination with the timer array unit. When this bit is set to 1, the input signal of the serial data input (RxD2) pin is selected as a timer input, so that wake up signal can be detected, the low width of the break field, and the pulse width of the sync field can be measured by the timer.

ISC43[1:0] bits (Switch of the serial clock input source of SPI00)

The ISC43[1:0] bits are used to select the serial data input source and serial clock input source of SPI00. These bits can be used to select the SCK00 pin input or TO01 output signal as the serial clock input source of SPI00.

ISC76[1:0] bits (Switch of the serial clock input source of SPI01)

The ISC76[1:0] bits are used to select the serial data input source and serial clock input source of SPI01. These bits can be used to select the SCK01 pin input or TO01 output signal as the serial clock input source of SPI01.

18.2.16 TNFEN : TAU Noise Filter Enable Register

Base address: PORGA = 0x4009_1000

Offset address: 0x0001

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TNFEN07	TNFEN06	TNFEN05	TNFEN04	TNFEN03	TNFEN02	TNFEN01	TNFEN00
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TNFEN00	Enabling or disabling use of the noise filter for the TI00 pin 0: Turns the noise filter off 1: Turns the noise filter on	R/W
1	TNFEN01	Enabling or disabling use of the noise filter for the TI01 pin 0: Turns the noise filter off 1: Turns the noise filter on	R/W
2	TNFEN02	Enabling or disabling use of the noise filter for the TI02 pin 0: Turns the noise filter off 1: Turns the noise filter on	R/W
3	TNFEN03	Enabling or disabling use of the noise filter for the TI03 pin 0: Turns the noise filter off 1: Turns the noise filter on	R/W
4	TNFEN04	Enabling or disabling use of the noise filter for the TI04 pin 0: Turns the noise filter off 1: Turns the noise filter on	R/W
5	TNFEN05	Enabling or disabling use of the noise filter for the TI05 pin 0: Turns the noise filter off 1: Turns the noise filter on	R/W
6	TNFEN06	Enabling or disabling use of the noise filter for the TI06 pin 0: Turns the noise filter off 1: Turns the noise filter on	R/W

Bit	Symbol	Function	R/W
7	TNFEN07	Enabling or disabling use of the noise filter for the TI07 pin*1 0: Turns the noise filter off 1: Turns the noise filter on	R/W

Note 1. For the TI07 pin, it can be switched by setting the ISC1 bit of the ISC register.
ISC.ISC1 = 0: Whether or not to use the noise filter of the TI07 pin can be selected.
ISC.ISC1 = 1: Whether or not to use the noise filter of the RxD2 pin can be selected.

The TNFEN registers are used to set whether the noise filter can be used for the timer input signal to each channel.

Enable the noise filter by setting the corresponding bits to 1 on the pins in need of noise removal.

When the noise filter is enabled, after synchronization with the operating clock (f_{MCK}) for the target channel, whether the signal keeps the same value for two clock cycles is detected.

When the noise filter is disabled, the input signal is only synchronized with the operating clock (f_{MCK}) for the target channel*1.

Note 1. For details, see (2) When valid edge of input signal via the TI0n pin is selected (TMR0n.CCS = 1), section 18.4.2. Timing of the Start of Counting, and section 18.6. Timer Input (TI0n) Control.

18.3 Basic Rules of Timer Array Unit

18.3.1 Basic Rules of Simultaneous Channel Operation Function

When simultaneously using multiple channels, namely, a combination of a master channel (a reference timer mainly counting the cycle) and slave channels (timers operating according to the master channel), the following rules apply.

1. Only an even channel (channels 0, 2, 4, etc.) can be set as a master channel.
2. Any channel, except channel 0, can be set as a slave channel.
3. The slave channel must be lower than the master channel.
   Example: If channel 2 is set as a master channel, channel 3 or those that follow (channels 3, 4, 5, etc.) can be set as a slave channel.
4. Two or more slave channels can be set for one master channel.
5. When two or more master channels are to be used, slave channels with a master channel between them may not be set.
   Example: If channels 0 and 4 are set as master channels, channels 1 to 3 can be set as the slave channels of master channel 0. Channels 5 to 7 cannot be set as the slave channels of master channel 0.
6. The operating clock for a slave channel in combination with a master channel must be the same as that of the master channel. The TMR0n.CKS[1:0] bits of the slave channel that operates in combination with the master channel must be the same value as that of the master channel.
7. A master channel can transmit TAU0_ENDIn (interrupt), start software trigger, and count clock to the lower channels.
8. A slave channel can use TAU0_ENDIn (interrupt), a start software trigger, or the count clock of the master channel as a source clock, but cannot transmit its own TAU0_ENDIn (interrupt), start software trigger, or count clock to channels with lower channel numbers.
9. A master channel cannot use TAU0_ENDIn (interrupt), a start software trigger, or the count clock from the other higher master channel as a source clock.
10. To simultaneously start channels that operate in combination, the channel start trigger bit (TS0.TS[n]) of the channels in combination must be set at the same time.
11. During the counting operation, a TS0.TS[n] bit of a master channel or TS0.TS[n] bits of all channels which are operating simultaneously can be set. It cannot be applied to TS0.TS[n] bits of slave channels alone.
12. To stop the channels in combination simultaneously, the channel stop trigger bit (TT0.TT[n]) of the channels in combination must be set at the same time.
13. CK02 and CK03 cannot be selected while channels are operating simultaneously, because the operating clocks of master channels and slave channels have to be synchronized.
14. Timer mode register 00 (TMR00) has no master bit (it is fixed to 0). However, as channel 0 is the highest channel, it can be used as a master channel during simultaneous operation.

The rules of the simultaneous channel operation function are applied in a channel group (a master channel and slave channels forming one simultaneous channel operation function).

If two or more channel groups that do not operate in combination are specified, the basic rules of the simultaneous channel operation function in this section do not apply to the channel groups.

Note: n: Channel number (n = 0 to 7)

Figure 18.19 shows an example of how TAU can be used.

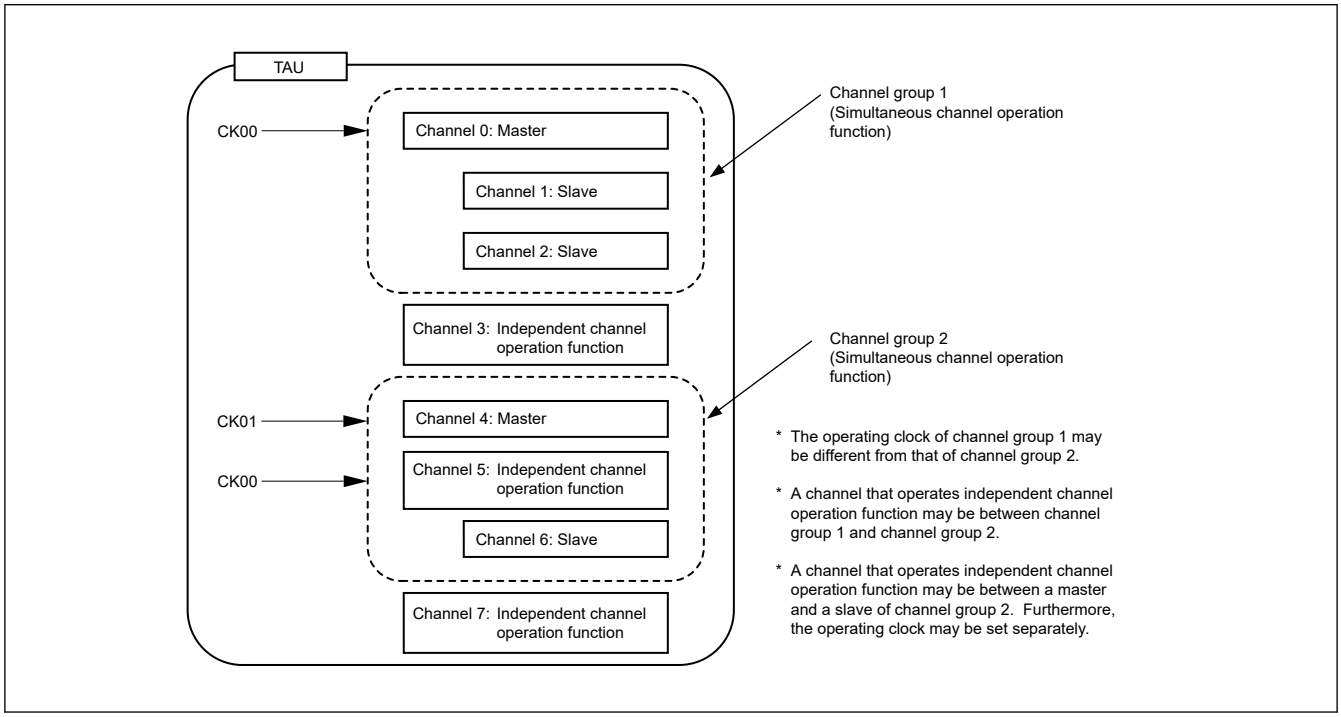


Figure 18.19 TAU utilization example

18.3.2 Basic Rules of 8-bit Timer Operation Function (Channels 1 and 3 only)

The 8-bit timer operation function makes it possible to use a 16-bit timer channel in a configuration consisting of two 8-bit timer channels.

This function can only be used for channels 1 and 3, and there are several rules for using it.

The basic rules for this function are as follows:

1. The 8-bit timer operation function applies only to channels 1 and 3.
2. When using 8-bit timers, set the SPLIT bit of timer mode register 0n (TMR0n) to 1.
3. The higher 8 bits can be operated as the interval timer function.
4. At the start of operation, the higher 8 bits output TAU0_MODE8_ENDI1 and TAU0_MODE8_ENDI3 (an interrupt) (which is the same operation performed when the TMR0n.OPIRQ bit is set to 1).
5. The operation clock of the higher 8 bits is selected according to the CKS[1:0] bits of the lower-bit TMR0n register.
6. For the higher 8 bits, the TS0.TSH1 and TSH3 bits are manipulated to start channel operation and the TT0.TTH1 and TTH3 bits are manipulated to stop channel operation. The channel state can be checked using the TE0.TEH1 and TEH3 bits.
7. The lower 8 bits operate according to the TMR0n register settings. The following three functions support operation of the lower 8 bits:
   - Interval timer function and square wave output function
   - External event counter function
   - Delay count function

8. For the lower 8 bits, the TS0.TS[1] and TS[3] bits are manipulated to start channel operation and the TT0.TT[1], TT[3] bits are manipulated to stop channel operation. The channel state can be checked using the TE0.TE[1] and TE[3] bits.
9. During 16-bit operation, manipulating the TS0.TSH1, TSH3, TT0.TTH1, and TTH3 bits are invalid. The TS0.TS[1], TS[3], TT0.TT[1], and TT[3] bits are manipulated to operate channels 1 and 3. The TE0.TEH1, and TEH3 bits are not changed.
10. For the 8-bit timer function, the simultaneous operation functions (one-shot pulse, PWM, and multiple PWM) cannot be used.

Note: n: Channel number (n = 1, 3)

18.4 Operations of Counters

18.4.1 Count Clock (f_{TCLK})

The count clock (f_{TCLK}) of the timer array unit can be selected between following by CCS bit of timer mode register 0n (TMR0n).

- Operation clock (f_{MCK}) specified by the TMR0n.CKS[1:0] bits
- Valid edge of input signal input from the TI0n pin

Because the timer array unit is designed to operate in synchronization with PCLKB, the timings of the count clock (f_{TCLK}) are shown below.

(1) When operation clock (f_{MCK}) specified by the TMR0n.CKS[1:0] bits is selected (TMR0n.CCS = 0)

The count clock (f_{TCLK}) is between PCLKB to PCLKB/2^{15} by setting of timer clock select register 0 (TPS0). When a divided PCLKB is selected in the TPS0 register, a signal becomes high level for one period from PCLKB rising edge. When a PCLKB is selected, high level is held for a signal.

Counting of timer counter register 0n (TCR0n) delayed by one period of PCLKB from rising edge of the count clock, because of synchronization with PCLKB. But this is described as "counting at rising edge of the count clock", as a matter of convenience.

Figure 18.20 shows the count clock (f_{TCLK}) timing from PCLKB when TMR0n.CCS = 0.

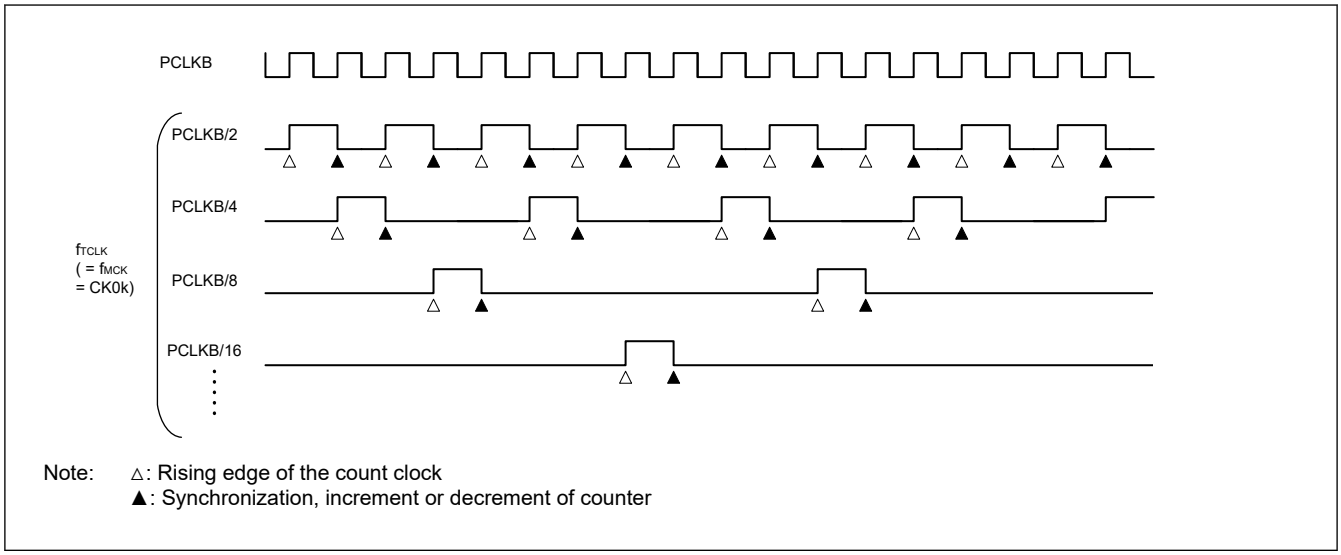


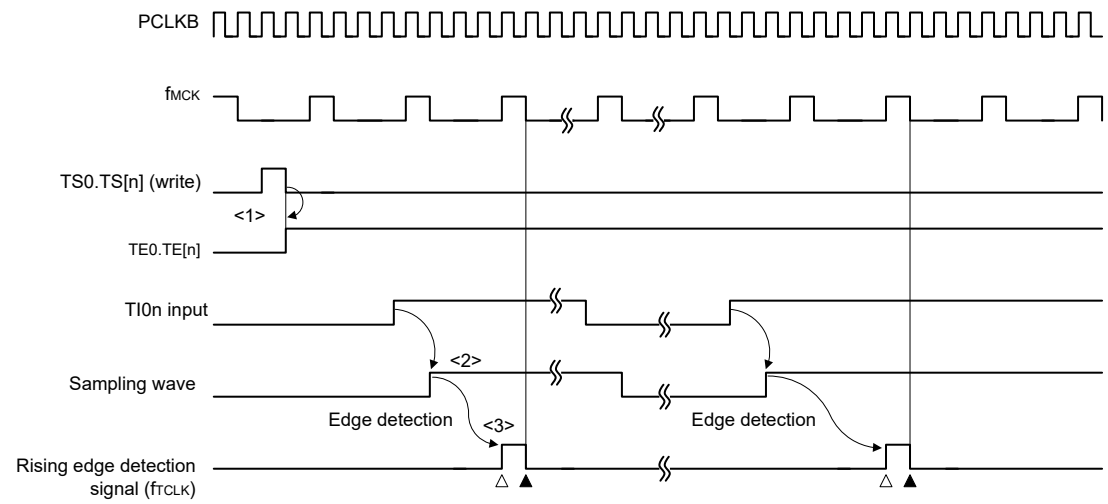
Figure 18.20 Timing of PCLKB and count clock (f_{TCLK}) (when TMR0n.CCS = 0)

(2) When valid edge of input signal via the TI0n pin is selected (TMR0n.CCS = 1)

The count clock (f_{TCLK}) becomes the signal that detects valid edge of input signal via the TI0n pin and synchronizes next rising f_{MCK}. The count clock (f_{TCLK}) is delayed for 1 to 2 period of f_{MCK} from the input signal via the TI0n pin (when a noise filter is used, the delay becomes 3 to 4 clock).

Counting of timer counter register 0n (TCR0n) is delayed by one period of PCLKB from the rising edge of the count clock, because of synchronization with PCLKB. This is described as "counting at valid edge of input signal via the TI0n pin", as a matter of convenience.

Figure 18.21 shows the count clock (f_{TCLK}) timing from PCLKB when TMR0n.CCS = 1 and noise filter unused.



Note: △: Rising edge of the count clock
▲: Synchronization, increment or decrement of counter
Note: f_{MCK}: Operation clock of channel n, n: Channel number (n = 0 to 7)
Note: The waveform of the input signal via TI0n pin of the input pulse interval measurement, the measurement of high or low width of input signal, and the delay counter, the one-shot pulse output are the same.

Figure 18.21 Timing of PCLKB and count clock (f_{TCLK}) (when TMR0n.CCS = 1, noise filter unused)

<1> Setting TS0.TS[n] bit to 1 enables the timer to be started and to become wait state for valid edge of input signal via the TI0n pin.

<2> The rise of input signal through the TI0n pin is sampled by f_{MCK}.

<3> The edge is detected by the rising of the sampled signal and the detection signal (count clock) is output.

18.4.2 Timing of the Start of Counting

Timer counter register 0n (TCR0n) becomes enabled to operation by setting of TS[n] bit of timer channel start register 0 (TS0).

Operations from count operation enabled state to timer counter register 0n (TCR0n) count start is shown in Table 18.11.

Table 18.11 Operations from the count operation enabled state to the start of counting by a Timer Counter Register 0n (TCR0n) (1 of 2)

Timer operation mode	Operation when TS0.TS[n] = 1 is set
• Interval timer mode	No operation is carried out from start trigger detection (TS0.TS[n] = 1) until count clock generation. The first count clock loads the value of the TDR0n register to the TCR0n register and the subsequent count clock performs count down operation (see (1) Operation in interval timer mode).
• Event counter mode	Writing 1 to the TS0.TS[n] bit loads the value of the TDR0n register to the TCR0n register. If detect edge of TI0n input. The subsequent count clock performs count down operation (see (2) Operation in event counter mode).
• Capture mode	No operation is carried out from start trigger detection (TS0.TS[n] = 1) until count clock generation. The first count clock loads 0x0000 to the TCR0n register and the subsequent count clock performs count up operation (see (3) Operation in capture mode (input pulse interval measurement)).

Table 18.11 Operations from the count operation enabled state to the start of counting by a Timer Counter Register 0n (TCR0n) (2 of 2)

Timer operation mode	Operation when TS0.TS[n] = 1 is set
• One-count mode	The waiting-for-start-trigger state is entered by writing 1 to the TS0.TS[n] bit while the timer is stopped (TE0.TE[n] = 0). No operation is carried out from start trigger detection until count clock generation. The first count clock loads the value of the TDR0n register to the TCR0n register and the subsequent count clock performs count down operation (see (4) Operation in one-count mode).
• Capture & one-count mode	The waiting-for-start-trigger state is entered by writing 1 to the TS0.TS[n] bit while the timer is stopped (TE0.TE[n] = 0). No operation is carried out from start trigger detection until count clock generation. The first count clock loads 0x0000 to the TCR0n register and the subsequent count clock performs count up operation (see (5) Operation in capture & one-count mode (high-level width measurement)).

Note: n: Channel number (n = 0 to 7)

18.4.3 Operations of Counters

The counter operation in each mode is explained in the sections that follow.

(1) Operation in interval timer mode

<1> Operation is enabled (TE0.TE[n] = 1) by writing 1 to the TS0.TS[n] bit. Timer counter register 0n (TCR0n) holds the initial value until count clock generation.

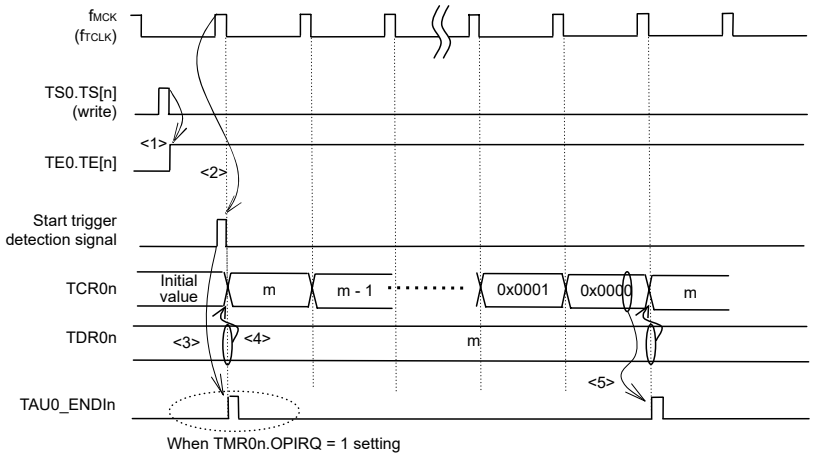
<2> A start trigger is generated at the first count clock after operation is enabled.

<3> When the TMR0n.OPIRQ bit is set to 1, TAU0_ENDIn is generated by the start trigger.

<4> By the first count clock after the operation enable, the value of timer data register 0n (TDR0n) is loaded to the TCR0n register and counting starts in the interval timer mode.

<5> When the TCR0n register counts down and its count value is 0x0000, TAU0_ENDIn is generated and the value of timer data register 0n (TDR0n) is loaded to the TCR0n register and counting keeps on.

Figure 18.22 shows the timing during operation in interval timer mode.



Note: n: Channel number (n = 0 to 7)
Note: In the first cycle operation of count clock after writing the TS0.TS[n] bit, an error at a maximum of one clock is generated since count start delays until count clock has been generated. When the information on count start timing is necessary, an interrupt can be generated at count start by setting TMR0n.OPIRQ = 1.
Note: f_{MCK}, the start trigger detection signal, and TAU0_ENDIn become active between one clock in synchronization with PCLKB.

Figure 18.22 Timing during operation in interval timer mode

(2) Operation in event counter mode

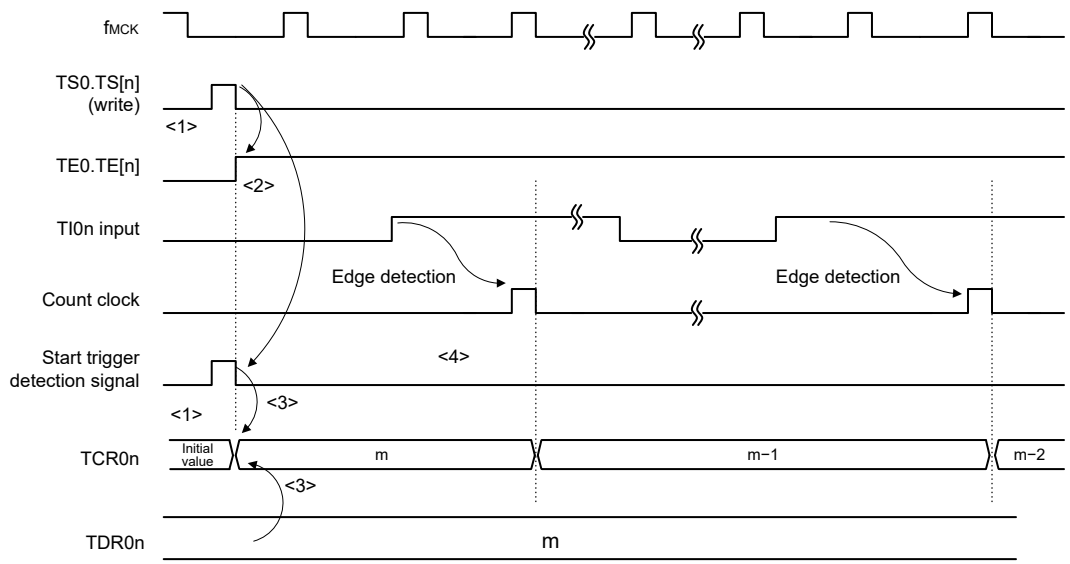
<1> Timer counter register 0n (TCR0n) holds its initial value while operation is stopped (TE0.TE[n] = 0).

<2> Operation is enabled (TE0.TE[n] = 1) by writing 1 to the TS0.TS[n] bit.

<3> As soon as 1 has been written to the TS0.TS[n] bit and 1 has been set to the TE0.TE[n] bit, the value of timer data register 0n (TDR0n) is loaded to the TCR0n register to start counting.

<4> After that, the TCR0n register value is counted down according to the count clock of the valid edge of the TI0n input.

Figure 18.23 shows the timing during operation in event counter mode.



Note: n: Channel number (n = 0 to 7)

Note: Figure 18.23 shows the timing when the noise filter is not used. By making the noise filter on-state, the edge detection becomes 2 f_{MCK} cycles (it sums up to 3 to 4 cycles) later than the normal cycle of TI0n input. The error per one period occurs be the asynchronous between the period of the TI0n input and that of the count clock (f_{MCK}).

Figure 18.23 Timing during operation in event counter mode

(3) Operation in capture mode (input pulse interval measurement)

<1> Operation is enabled (TE0.TE[n] = 1) by writing 1 to the TS0.TS[n] bit.

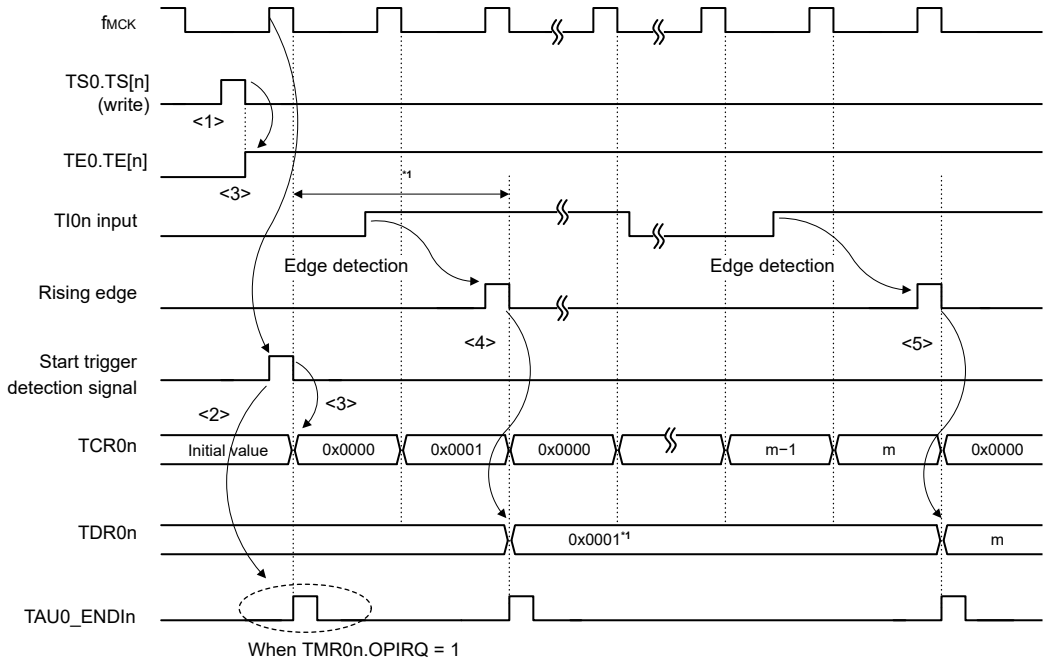
<2> Timer counter register 0n (TCR0n) holds the initial value until count clock generation.

<3> A start trigger is generated at the first count clock after operation is enabled. And the value of 0x0000 is loaded to the TCR0n register and counting starts in the capture mode. (when the TMR0n.OPIRQ bit is set to 1, TAU0_ENDIn is generated by the start trigger.)

<4> On detection of the valid edge of the TI0n input, the value of the TCR0n register is captured to timer data register 0n (TDR0n) and TAU0_ENDIn is generated. However, this captured value is meaningless. The TCR0n register keeps on counting from 0x0000.

<5> On next detection of the valid edge of the TI0n input, the value of the TCR0n register is captured to timer data register 0n (TDR0n) and TAU0_ENDIn is generated.

Figure 18.24 shows the timing during operation in capture mode (Input Pulse Interval Measurement).



Note: n: Channel number (n = 0 to 7)

Note: In the first cycle operation of count clock after writing the TS0.TS[n] bit, an error at a maximum of one clock is generated since count start delays until count clock has been generated. When the information on count start timing is necessary, an interrupt can be generated at count start by setting TMR0n.OPIRQ = 1.

Note: Figure 18.24 shows the timing when the noise filter is not used. By making the noise filter on-state, the edge detection becomes 2 f_{MCK} cycles (it sums up to 3 to 4 cycles) later than the normal cycle of TI0n input. The error per one period occurs be the asynchronous between the period of the TI0n input and that of the count clock (f_{MCK}).

Note 1. If a clock has been input to TI0n (the trigger exists) when capturing starts, counting starts when a trigger is detected, even if no edge is detected. Therefore, the first captured value (<4>) does not determine a pulse interval (in the above figure, 0x0001 just indicates two clock cycles but does not determine the pulse interval) and so the user can ignore it.

Figure 18.24 Timing during operation in capture mode (input pulse interval measurement)

(4) Operation in one-count mode

<1> Operation is enabled (TE0.TE[n] = 1) by writing 1 to the TS0.TS[n] bit.

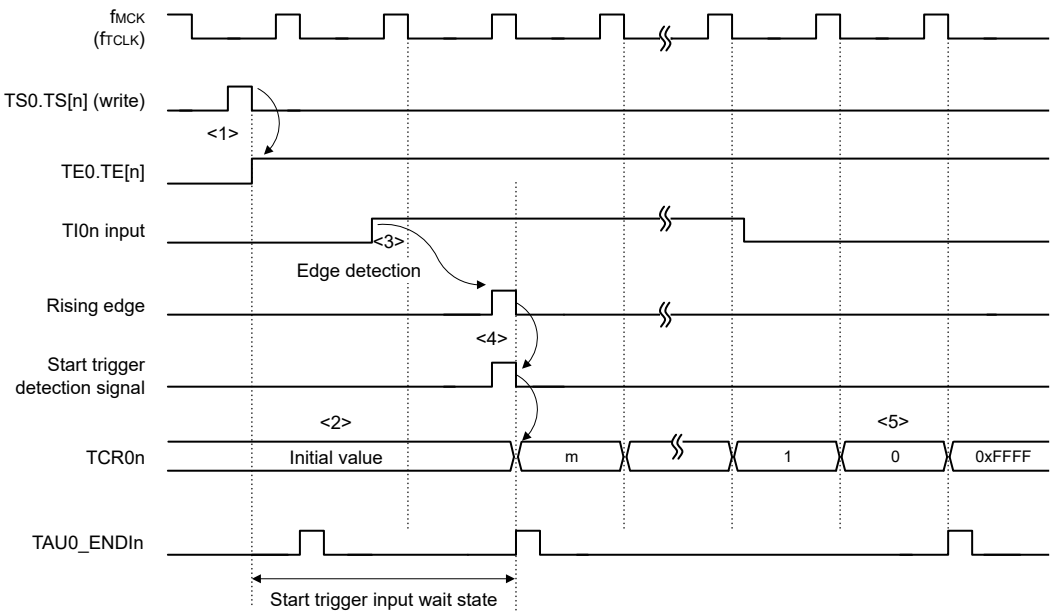
<2> Timer counter register 0n (TCR0n) holds the initial value until start trigger generation.

<3> Rising edge of the TI0n input is detected.

<4> On start trigger detection, the value of timer data register 0n (TDR0n) is loaded to the TCR0n register and count starts.

<5> When the TCR0n register counts down and its count value is 0x0000, TAU0_ENDIn is generated and the value of the TCR0n register becomes 0xFFFF and counting stops.

Figure 18.25 shows the timing during operation in one-count mode.



Note: n: Channel number (n = 0 to 7)
Note: Figure 18.25 shows the timing when the noise filter is not used. By making the noise filter on-state, the edge detection becomes 2 f_{MCK} cycles (it sums up to 3 to 4 cycles) later than the normal cycle of TI0n input. The error per one period occurs be the asynchronous between the period of the TI0n input and that of the count clock (f_{MCK}).

Figure 18.25 Timing during operation in one-count mode

(5) Operation in capture & one-count mode (high-level width measurement)

<1> Operation is enabled (TE0.TE[n] = 1) by writing 1 to the TS[n] bit of timer channel start register 0 (TS0).

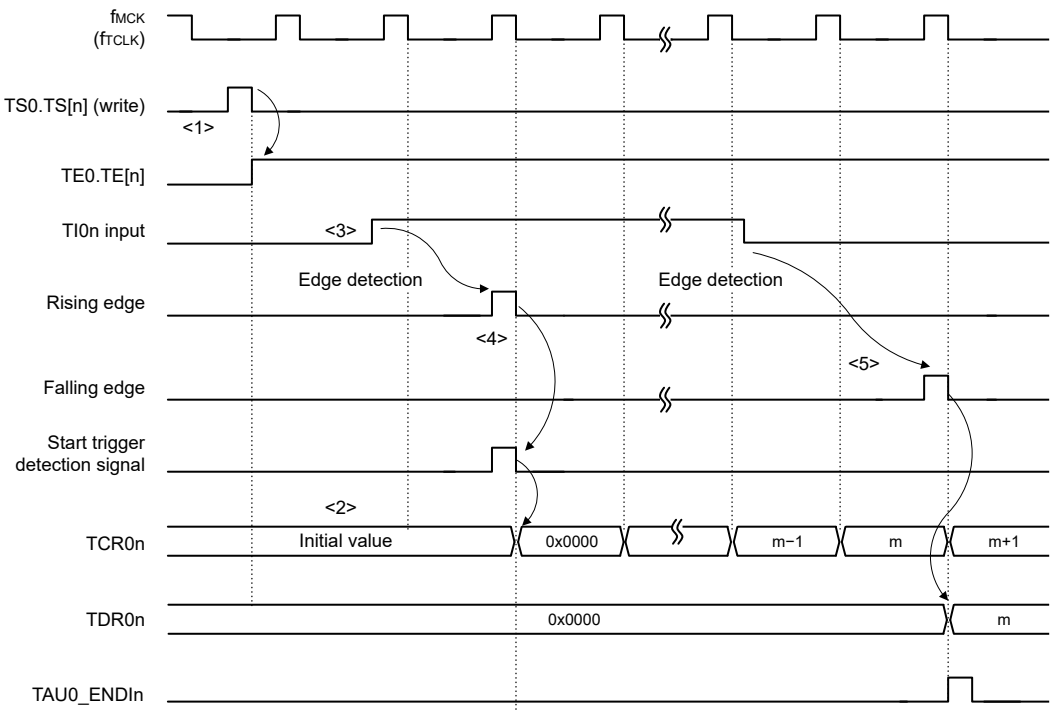
<2> Timer counter register 0n (TCR0n) holds the initial value until start trigger generation.

<3> Rising edge of the TI0n input is detected.

<4> On start trigger detection, the value of 0x0000 is loaded to the TCR0n register and count starts.

<5> On detection of the falling edge of the TI0n input, the value of the TCR0n register is captured to timer data register 0n (TDR0n) and TAU0_ENDIn is generated.

Figure 18.26 shows the timing during operation in capture & one-count mode (High-Level Width Measurement).



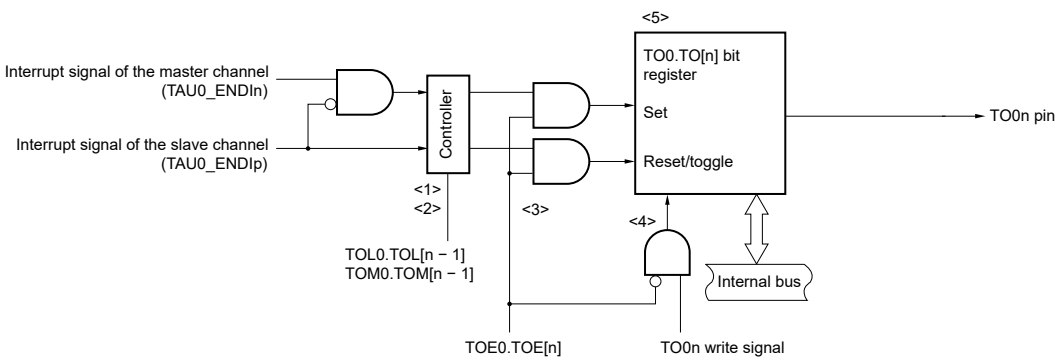
Note: n: Channel number (n = 0 to 7)
Note: Figure 18.26 shows the timing when the noise filter is not used. By making the noise filter on-state, the edge detection becomes 2 f_{MCK} cycles (it sums up to 3 to 4 cycles) later than the normal cycle of TI0n input. The error per one period occurs be the asynchronous between the period of the TI0n input and that of the count clock (f_{MCK}).

Figure 18.26 Timing during operation in capture & one-count mode (high-level width measurement)

18.5 Channel Output (TO0n Pin) Control

18.5.1 TO0n Pin Output Circuit Configuration

Figure 18.27 shows the TO0n pin output circuit.



Note: n: Channel number (n = 0 to 7)

Figure 18.27 Output circuit configuration

The following describes the TO0n pin output circuit.

<1> When TOM0.TOM[n − 1] = 0 (master channel output mode), the set value of timer output level register 0 (TOL0) is ignored and only TAU0_ENDIp (slave channel timer interrupt) is transmitted to timer output register 0 (TO0).

<2> When TOM0.TOM[n − 1] = 1 (slave channel output mode), both TAU0_ENDIn (master channel timer interrupt) and TAU0_ENDIp (slave channel timer interrupt) are transmitted to the TO0 register.

At this time, the TOL0 register becomes valid and the signals are controlled as follows:

When TOL0.TOL[n − 1] = 0: Positive logic output (TAU0_ENDIn → set, TAU0_ENDIp → reset)

When TOL0.TOL[n − 1] = 1: Negative logic output (TAU0_ENDIn → reset, TAU0_ENDIp → set)

When TAU0_ENDIn and TAU0_ENDIp are simultaneously generated, (0% output of PWM), TAU0_ENDIp (reset signal) takes priority, and TAU0_ENDIn (set signal) is masked.

<3> While timer output is enabled (TOE0.TOE[n] = 1), TAU0_ENDIn (master channel timer interrupt) and TAU0_ENDIp (slave channel timer interrupt) are transmitted to the TO0 register. Writing to the TO0 register (TO0n write signal) becomes invalid.

When TOE0.TOE[n] = 1, the TO0n pin output never changes with signals other than interrupt signals.

To initialize the TO0n pin output level, it is necessary to set timer operation is stopped (TOE0.TOE[n] = 0) and to write a value to the TO0 register.

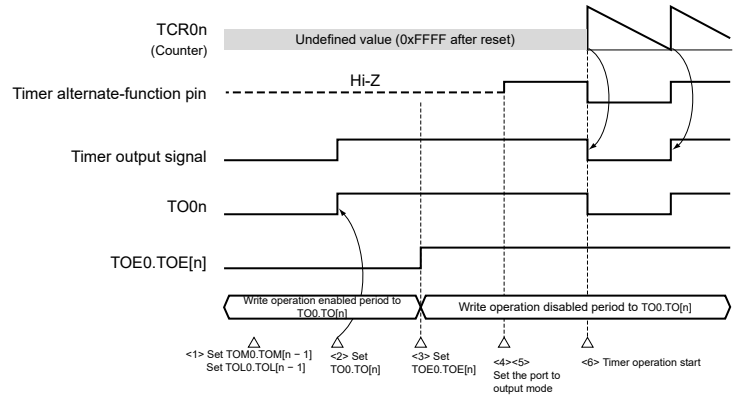
<4> While timer output is disabled (TOE0.TOE[n] = 0), writing to the TO0.TO[n] bit to the target channel (TO0n write signal) becomes valid. When timer output is disabled (TOE0.TOE[n] = 0), neither TAU0_ENDIn (master channel timer interrupt) nor TAU0_ENDIp (slave channel timer interrupt) is transmitted to the TO0 register.

<5> The TO0 register can always be read, and the TO0n pin output level can be checked.

Note: n: Channel number
n = 0 to 7 (n = 0, 2, 4, 6 for master channel)
p: Slave channel number
n < p ≤ 7

18.5.2 TO0n Pin Output Setting

Figure 18.28 shows the procedure and state transitions from the initial settings of a TO0n output pin to the start of timer operation.



Note: n: Channel number (n = 0 to 7)

Figure 18.28 State transitions from the settings for timer output to the start of timer operation

<1> The operation mode of timer output is set.

- TOM0.TOM[n − 1] bit (0: Master channel output mode, 1: Slave channel output mode)
- TOL0.TOL[n − 1] bit (0: Positive logic output, 1: Negative logic output)

<2> The timer output signal is set to the initial state by setting timer output register 0 (TO0).

<3> The timer output operation is enabled by writing 1 to the TOE0.TOE[n] bit (writing to the TO0 register is disabled).

<4> The port is set to peripheral output by the PSEL[4:0] bit of Port gh Pin Function Select Register (PghPFS) (see section 16.2.5. PmnPFS/PmnPFS_HA/PmnPFS_BY : Port mn Pin Function Select Register (m = 0 to 4, n = 00 to 15)).

<5> The port I/O setting is set to output by the PDR bit of PghPFS register.

<6> The timer operation is enabled (TS0.TS[n] = 1).

Note: n: Channel number (n = 0 to 7)
gh: Port number (g = 0 to 4, h = 00 to 15)

18.5.3 Cautions on Channel Output Operation

(1) Changing values set in the registers TO0, TOE0, and TOL0 during timer operation

Since the timer operations (operations of timer counter register 0n (TCR0n) and timer data register 0n (TDR0n)) are independent of the TO0n output circuit and changing the values set in timer output register 0 (TO0), timer output enable register 0 (TOE0), and timer output level register 0 (TOL0) does not affect the timer operation, the values can be changed during timer operation. To output an expected waveform from the TO0n pin by timer operation, however, set the TO0, TOE0, TOL0, and TOM0 registers to the values stated in the register setting example of each operation shown in section 18.6. Timer Input (TI0n) Control and section 18.7. Independent Channel Operation Function of Timer Array Unit.

When the values set to the TOE0, and TOM0 registers (but not the TO0 register) are changed close to the occurrence of the timer interrupt (TAU0_ENDIn) of each channel, the waveform output to the TO0n pin might differ, depending on whether the values are changed immediately before or immediately after the timer interrupt (TAU0_ENDIn) occurs.

Note: n: Channel number (n = 0 to 7)

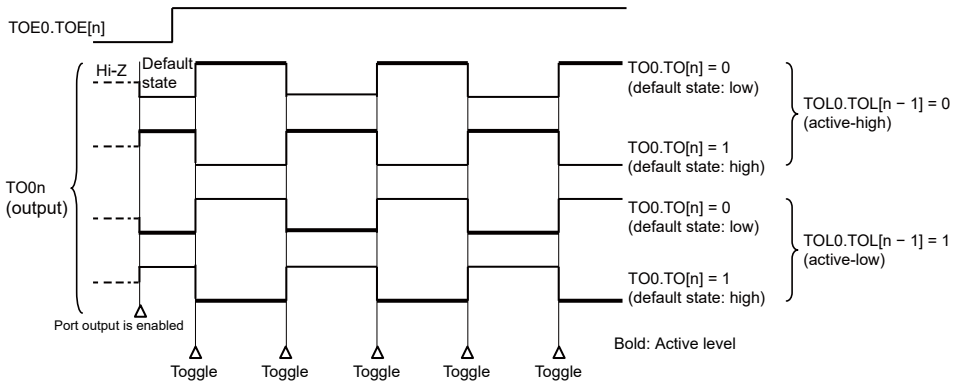
(2) Default level of TO0n pin and output level after timer operation start

The change in the output level of the TO0n pin when timer output register 0 (TO0) is written while timer output is disabled (TOE0.TOE[n] = 0), the initial level is changed, and then timer output is enabled (TOE0.TOE[n] = 1) before port output is enabled, is shown below.

When operation starts with master channel output mode (TOM0.TOM[n − 1] = 0) setting

The setting of timer output level register 0 (TOL0) is invalid when master channel output mode (TOM0.TOM[n − 1] = 0). When the timer operation starts after setting the default level, the toggle signal is generated and the output level of the TO0n pin is inverted.

Figure 18.29 shows the output state of the TO0n pin with the output toggled (TOM0.TOM[n − 1] = 0).



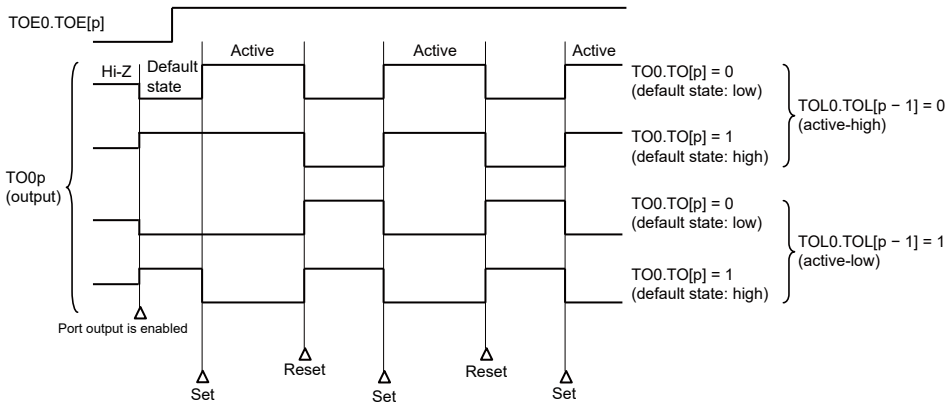
Note: Toggle: Toggle signal to invert the output on the TO0n pin
Note: n: Channel number (n = 0 to 7)

Figure 18.29 TO0n pin output states with toggled output (TOM0.TOM[n − 1] = 0)

When operation starts with slave channel output mode (TOM0.TOM[p − 1] = 1) setting (PWM output)

When slave channel output mode (TOM0.TOM[p − 1] = 1), the active level is determined by timer output level register 0 (TOL0) setting.

Figure 18.30 shows the output state of the TO0p pin with PWM output (TOM0.TOM[p − 1] = 1).



Note: Set: The output signal of the TO0p pin changes from inactive level to active level.
Reset: The output signal of the TO0p pin changes from active level to inactive level.
Note: p: Channel number (p = 1 to 7)

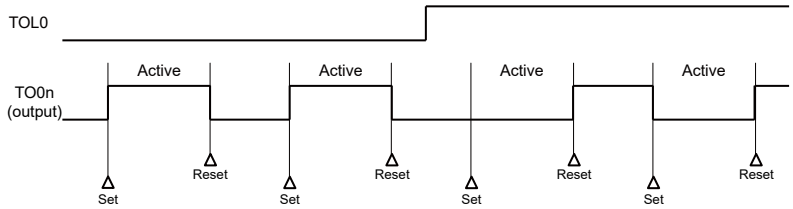
Figure 18.30 TO0p pin output states with PWM output (TOM0.TOM[p − 1] = 1)

(3) Operation of TO0n pin in slave channel output mode (TOM0.TOM[n − 1] = 1)

When the relevant bit of timer output level register 0 (TOL0) is changed during timer operation

When the TOL0 register setting has been changed during timer operation, the setting becomes valid at the generation timing of the TO0n pin change condition. Rewriting the TOL0 register does not change the output level of the TO0n pin.

The operation when TOM0.TOM[n − 1] is set to 1 and the value of the TOL0 register is changed while the timer is operating (TE0.TE[n] = 1) is shown Figure 18.31.



Note: Set: The output signal of the TO0n pin changes from inactive level to active level.
Reset: The output signal of the TO0n pin changes from active level to inactive level.
Note: n: Channel number (n = 0 to 7)

Figure 18.31 Operation when the relevant bit of the TOL0 register is changed during timer operation

Set and reset timing

To realize 0% and 100% output at PWM output, the TO0n pin and TO0.TO[n] bit set timing at master channel timer interrupt (TAU0_ENDIn) generation is delayed by 1 count clock by the slave channel.

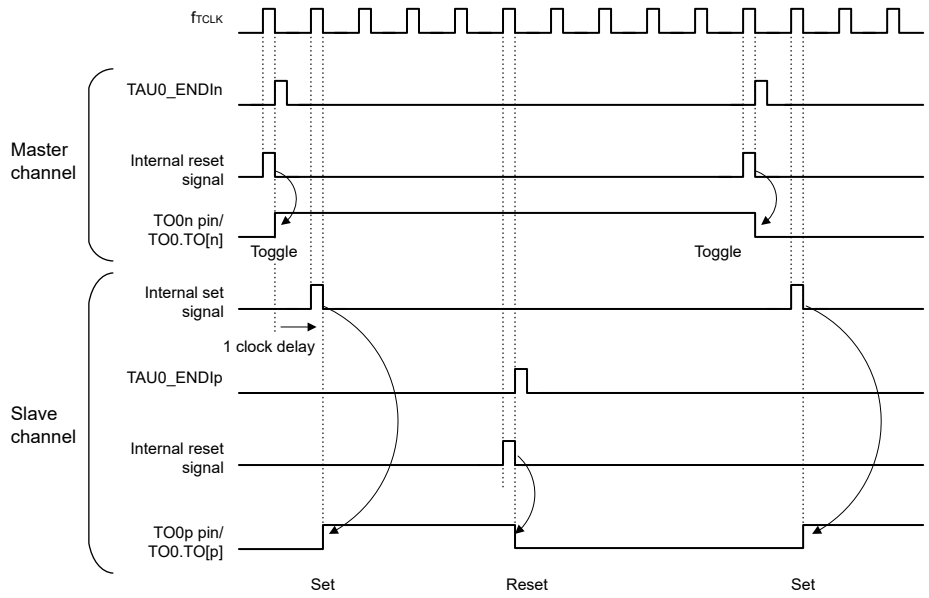
If the set condition and reset condition are generated at the same time, a higher priority is given to the latter.

Figure 18.32 shows the states of operation following set and reset signals when the master and slave channels are set as follows.

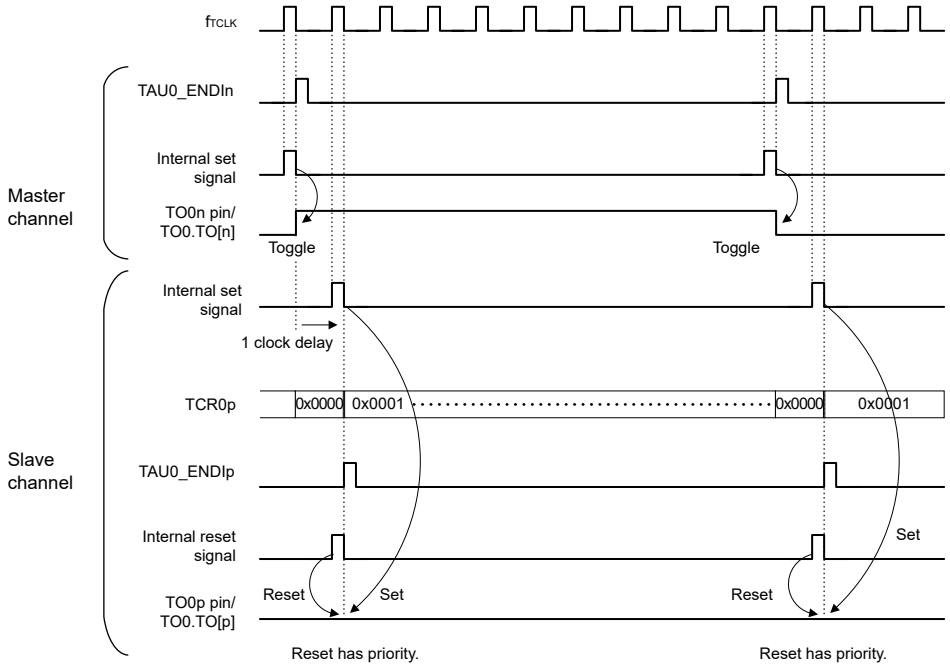
Master channel: TOE0.TOE[n] = 1, TOM0.TOM[n − 1] = 0, TOL0.TOL[n − 1] = 0

Slave channel: TOE0.TOE[p] = 1, TOM0.TOM[p − 1] = 1, TOL0.TOL[p − 1] = 0

(1) Basic timing during operation



(2) Timing during operation with the duty cycle set to 0%



Note: Internal reset signal: TO0n pin reset/toggle signal
Internal set signal: TO0n pin set signal

Note: n: Channel number
n = 0 to 7 (n = 0, 2, 4, 6 for master channel)
p: Slave channel number
n < p ≤ 7

Figure 18.32 States of operation following set and reset signals

18.5.4 Collective Manipulation of TO0.TO[n] Bit

In timer output register 0 (TO0), the setting bits for all the channels are located in one register in the same way as timer channel start register 0 (TS0). Therefore, the TO0.TO[n] bit of all the channels can be manipulated collectively.

Only the desired bits can also be manipulated by enabling writing only to the TO0.TO[n] bits (TOE0.TOE[n] = 0) that correspond to the relevant bits of the channel used to perform output (TO0n).

Table 18.12 shows an example of the TO0.TO[n] bit collective manipulation.

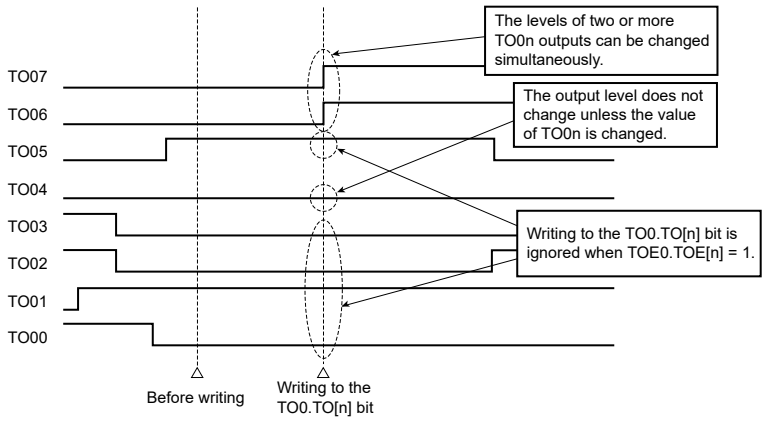
Table 18.12 Example of TO0.TO[n] bit collective manipulation

Bit position:		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Before writing	TO0	— 0	— 0	— 0	— 0	— 0	— 0	— 0	— 0	TO[7] 0	TO[6] 0	TO[5] 1	TO[4] 0	TO[3] 0	TO[2] 0	TO[1] 1	TO[0] 0
	TOE0	— 0	— 0	— 0	— 0	— 0	— 0	— 0	— 0	TOE[7] 0	TOE[6] 0	TOE[5] 1	TOE[4] 0	TOE[3] 1	TOE[2] 1	TOE[1] 1	TOE[0] 1
Data to be written	TO0	0	0	0	0	0	0	0	0	1 ↓ ●	1 ↓ ●	0 ↓ ×	0 ↓ ●	0 ↓ ×	0 ↓ ×	1 ↓ ×	1 ↓ ×
After writing	TO0	— 0	— 0	— 0	— 0	— 0	— 0	— 0	— 0	TO[7] 1	TO[6] 1	TO[5] 1	TO[4] 0	TO[3] 0	TO[2] 0	TO[1] 1	TO[0] 0

Writing is done only to the TO0.TO[n] bit with TOE0.TOE[n] = 0, and writing to the TO0.TO[n] bit with TOE0.TOE[n] = 1 is ignored.

TO0n (channel output) to which TOE0.TOE[n] = 1 is set is not affected by the write operation. Even if the write operation is done to the TO0.TO[n] bit, it is ignored and the output change by timer operation is normally done.

Figure 18.33 shows an example of a TO0.TO[n] bit collective manipulation.



Note: n: Channel number (n = 0 to 7)

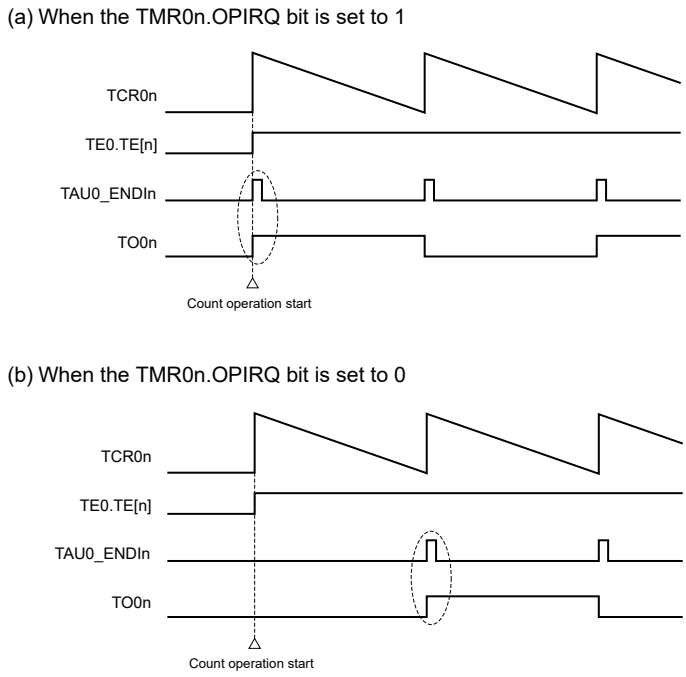
Figure 18.33 TO0n pin states by collective manipulation of TO0.TO[n] bits

18.5.5 Timer Interrupts and TO0n Outputs When Counting is Started

In the interval timer mode or capture mode, the TMR0n.OPIRQ bit in timer mode register 0n (TMR0n) sets whether or not to generate a timer interrupt at count start.

When the TMR0n.OPIRQ bit is set to 1, the count operation start timing can be known by the timer interrupt (TAU0_ENDIn) generation. In the other modes, neither timer interrupt at count operation start nor TO0n output is controlled.

Figure 18.34 shows operation examples when the interval timer mode (TOE0.TOE[n] = 1, TOM0.TOM[n − 1] = 0) is set.



Note: n: Channel number (n = 0 to 7)

Figure 18.34 Examples of the operation of timer interrupts and TO0n outputs when counting is started

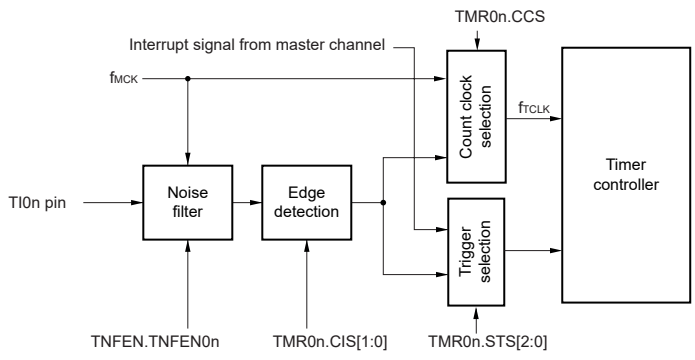
When the TMR0n.OPIRQ bit is set to 1, a timer interrupt (TAU0_ENDIn) is output at count operation start, and TO0n performs a toggle operation.

When the TMR0n.OPIRQ bit is set to 0, a timer interrupt (TAU0_ENDIn) is not output at count operation start, and TO0n does not change either. After counting one cycle, TAU0_ENDIn is output and TO0n performs a toggle operation.

18.6 Timer Input (TI0n) Control

18.6.1 TI0n Input Circuit Configuration

A signal is input from a timer input pin, goes through a noise filter and an edge detector, and is sent to a timer controller. Enable the noise filter for the pin in need of noise removal. Figure 18.35 shows the configuration of the input circuit.

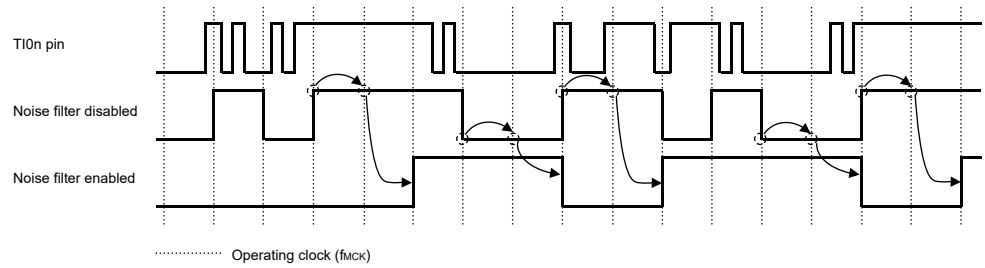


Note: n: Channel number (n = 0 to 7)

Figure 18.35 Input circuit configuration

18.6.2 Noise Filter

When the noise filter is disabled, the input signal is only synchronized with the operating clock (f_{MCK}) for channel n. When the noise filter is enabled, after synchronization with the operating clock (f_{MCK}) for channel n, whether the signal keeps the same value for two clock cycles is detected. Figure 18.36 shows differences in waveforms output from the noise filter between when the noise filter is enabled and disabled.



Note: n: Channel number (n = 0 to 7)

Figure 18.36 Sampling waveforms through TI0n input pin with noise filter enabled and disabled

18.6.3 Cautions on Channel Input Operation

When a timer input pin is set as unused, the operating clock is not supplied to the noise filter. Therefore, after settings are made to use the timer input pin, the following wait time is necessary before a trigger is specified to enable operation of the channel corresponding to the timer input pin.

1. Noise filter is disabled
   When the CCS and STS[1:0] bits in the timer mode register 0n (TMR0n) are all 0s and then one of them is set to 1, wait for at least two cycles of the operating clock (f_{MCK}), and then set the operation enable trigger bit in the timer channel start register 0 (TS0).
2. Noise filter is enabled
   When the CCS and STS[1:0] bits in the timer mode register 0n (TMR0n) are all 0s and then one of them is set to 1, wait for at least four cycles of the operating clock (f_{MCK}), and then set the operation enable trigger bit in the timer channel start register 0 (TS0).

18.7 Independent Channel Operation Function of Timer Array Unit

18.7.1 Operation as an Interval Timer or for Square Wave Output

(1) Interval timer

The timer array unit can be used as a reference timer that generates TAU0_ENDIn (timer interrupt) at fixed intervals. The interrupt generation period can be calculated by the following expression.

Generation period of TAU0_ENDIn (timer interrupt) = Period of count clock × (Set value of TDR0n + 1)

(2) Operation for square wave output

TO0n performs a toggle operation as soon as TAU0_ENDIn has been generated, and outputs a square wave with a duty factor of 50%.

The period and frequency for outputting a square wave from TO0n can be calculated by the following expressions.

- Period of square wave output from TO0n = Period of count clock × (Set value of TO0n + 1) × 2
- Frequency of square wave output from TO0n = Frequency of count clock / {(Set value of TDR0n + 1) × 2}

Timer counter register 0n (TCR0n) operates as a down counter in the interval timer mode.

The TCR0n register loads the value of timer data register 0n (TDR0n) at the first count clock after the channel start trigger bit (TS[n], TSH1, TSH3) of timer channel start register 0 (TS0) is set to 1. If the OPIRQ bit of timer mode register 0n

(TMR0n) is 0 at this time, TAU0_ENDIn is not output and the output on TO0n is not toggled. If the OPIRQ bit of the TMR0n register is 1, TAU0_ENDIn is output and the output on TO0n is toggled.

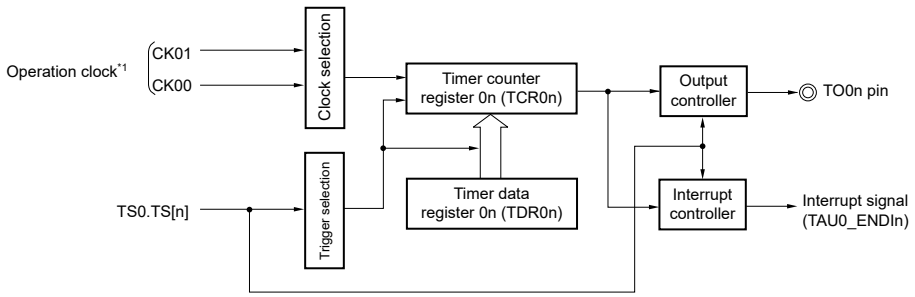
After that, the TCR0n register count down in synchronization with the count clock.

When TCR0n = 0x0000, TAU0_ENDIn is output and the output on TO0n is toggled at the next count clock. At the same time, the TCR0n register loads the value of the TDR0n register again. After that, the same operation is repeated.

The TDR0n register can be rewritten at any time. The new value of the TDR0n register becomes valid from the next period.

Note: n: Channel number (n = 0 to 7)

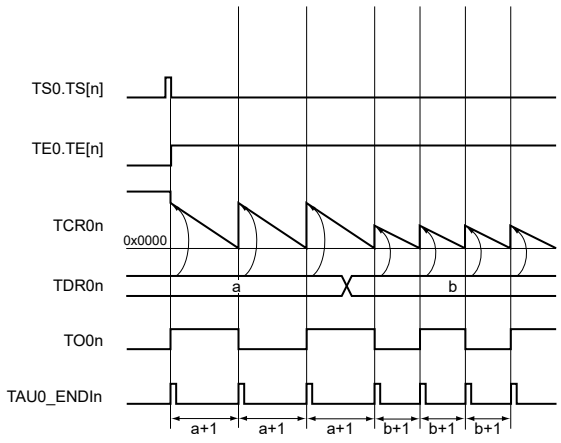
Figure 18.37 shows a block diagram of the operation as an interval timer or a square wave output.



Note: n: Channel number (n = 0 to 7)
Note 1. For channels 1 and 3, the clock can be selected as CK00, CK01, CK02, or CK03.

Figure 18.37 Block diagram for operation as an interval timer or for square wave output

Figure 18.38 shows an example of basic timing during operation as an interval timer or for square wave output (TMR0n.OPIRQ = 1).



Note: n: Channel number (n = 0 to 7)

Figure 18.38 Example of basic timing during operation as an interval timer or for square wave output (TMR0n.OPIRQ = 1)

Table 18.13 to Table 18.18 show register settings and procedure for operation when the interval timer or square wave output.

Table 18.13 Example of TMR0n settings for operation as an interval timer or for square wave output

Bit	Symbol	Set value	Function
0	OPIRQ	1/0	Setting of operation when counting is started 0: Neither generates TAU0_ENDIn nor inverts timer output when counting is started 1: Generates TAU0_ENDIn and inverts timer output when counting is started
3:1	MD[2:0]	000b	Operation mode of channel n 0 0 0: Interval timer mode
5:4	—	00b	Fixed to 0
7:6	CIS[1:0]	00b	Selection of TI0n pin input edge 0 0: Set to 00b because the TI0n input pin is not to be used
10:8	STS[2:0]	000b	Start trigger selection 0 0 0: Selects only software start
11	— (n = 0, 5, 7)	0	Fixed to 0 (channels 0, 5, 7)
	SPLIT (n = 1, 3)	1/0	Setting of SPLIT bit (channels 1, 3) 0: 16-bit timer mode 1: 8-bit timer mode
	MASTER (n = 2, 4, 6)	0	Setting of MASTER bit (channels 2, 4, 6) 0: Independent channel operation function
12	CCS	0	Count clock selection 0: Selects operation clock (f_{MCK})
13	—	0	Fixed to 0
15:14	CKS[1:0]	00b to 11b	Selection of the operating clock (f_{MCK}) 0 0: Selects CK00 as the operating clock for channel n 0 1: Selects CK02 as the operating clock (this can only be selected for channels 1 and 3) 1 0: Selects CK01 as the operating clock for channel n 1 1: Selects CK03 as the operating clock (this can only be selected for channels 1 and 3)

Table 18.14 Example of TO0 settings for operation as an interval timer or for square wave output

Bit	Symbol	Set value	Function
n	TO[n]	1/0	Timer output of channel n 0: Outputs 0 from TO0n 1: Outputs 1 from TO0n

Table 18.15 Example of TOE0 settings for operation as an interval timer or for square wave output

Bit	Symbol	Set value	Function
n	TOE[n]	1/0	Enabling or disabling timer output for channel n 0: Stops the TO0n output operation by counting operation 1: Enables the TO0n output operation by counting operation

Table 18.16 Example of TOL0 settings for operation as an interval timer or for square wave output

Bit	Symbol	Set Value	Function
n	— (n = 0)	0	Fixed to 0 (channels 0)
	TOL[n − 1] (n = 1 to 7)		Control of timer output of channel n (channels 1 to 7) 0: Sets this bit to 0 when TOM0.TOM[n − 1] = 0 (master channel output mode)

Table 18.17 Example of TOM0 settings for operation as an interval timer or for square wave output

Bit	Symbol	Set value	Function
n	— (n = 0)	0	Fixed to 0 (channels 0)
	TOM[n − 1] (n = 1 to 7)		Control of timer output mode of channel n (channels 1 to 7) 0: Sets master channel output mode

Note: n: Channel number (n = 0 to 7)

Table 18.18 Procedure for operations when the interval timer or square wave output function is to be used (1 of 2)

	Step	Software operation		Hardware state
TAU default setting	<1>	Sets timer clock select register 0 (TPS0). Determines clock frequencies of CK00 to CK03.		—
Channel default setting	<2>	Sets timer mode register 0n (TMR0n) (determines operation mode of channel). Sets interval (period) value to timer data register 0n (TDR0n).		Channel stops operating. (Clock is supplied and some power is consumed.)
	<3>	To use the TO0n output Clears the TOM0.TOM[n − 1] bit to 0 (master channel output mode). Clears the TOL0.TOL[n − 1] bit to 0.		The TO0n pin goes into Hi-Z output state.
		Sets the TO0.TO[n] bit and determines default level of the TO0n output.	→	The TO0n default setting level is output when the corresponding bit of the port direction register (PDR) is in the output mode.
		Sets the TOE0.TOE[n] bit to 1 and enables operation of TO0n.	→	TO0n does not change because channel stops operating.
		Sets the corresponding bit of the port direction register (PDR) to 1.	→	The TO0n pin outputs the TO0n set level.
Operation start	<4>	(Sets the TOE0.TOE[n] bit to 1 only if using TO0n output and resuming operation.). Sets the TS0.TS[n] (TSH1, TSH3) bit to 1. The TS0.TS[n] (TSH1, TSH3) bit automatically returns to 0 because it is a trigger bit.	→	TE0.TE[n] (TEH1, TEH3) = 1, and count operation starts. Value of the TDR0n register is loaded to timer counter register 0n (TCR0n). TAU0_ENDIn is generated and TO0n performs toggle operation if the OPIRQ bit of the TMR0n register is 1.
During operation	<5>	Set values of the TMR0n register, TOM0.TOM[n − 1], and TOL0.TOL[n − 1] bits cannot be changed. Set value of the TDR0n register can be changed. The TCR0n register can always be read. The TSR0n register is not used. Set values of the TO0 and TOE0 registers can be changed.		Counter (TCR0n) counts down. When count value reaches 0x0000, the value of the TDR0n register is loaded to the TCR0n register again and the count operation is continued. By detecting TCR0n = 0x0000, TAU0_ENDIn is generated and TO0n performs toggle operation. After that, the above operation is repeated.
Operation stop	<6>	The TT0.TT[n] (TTH1, TTH3) bit is set to 1. The TT0.TT[n] (TTH1, TTH3) bit automatically returns to 0 because it is a trigger bit.	→	TE0.TE[n] (TEH1, TEH3) = 0, and count operation stops. The TCR0n register holds count value and stops. The TO0n output is not initialized and retains its current state.
	<7>	The TOE0.TOE[n] bit is cleared to 0 and value is set to the TO0.TO[n] bit. To resume operation, go to step <4>. To terminate the operation, go to step <8>	→	The TO0n pin outputs the TO0.TO[n] bit set level.

Table 18.18 Procedure for operations when the interval timer or square wave output function is to be used (2 of 2)

	Step	Software operation		Hardware state
TAU stop	<8>	To hold the TO0n pin output level Sets the PSEL[4:0] bits to 00000b after the value to be held is set to the corresponding bit of the port output data register (PODR). When holding the TO0n pin output level is not necessary. Setting not required.	→	The TO0n pin output level is held by port function.

Note: n: Channel number (n = 0 to 7)

18.7.2 Operation as an External Event Counter

The timer array unit can be used as an external event counter that counts the number of times the valid input edge (external event) is detected in the TI0n pin. When a specified count value is reached, the event counter generates an interrupt. The specified number of counts can be calculated by the following expression.

Specified number of counts = Set value of TDR0n + 1

Timer counter register 0n (TCR0n) operates as a down counter in the event counter mode.

The TCR0n register loads the value of timer data register 0n (TDR0n) by setting any TS0.TS[n] bits to 1.

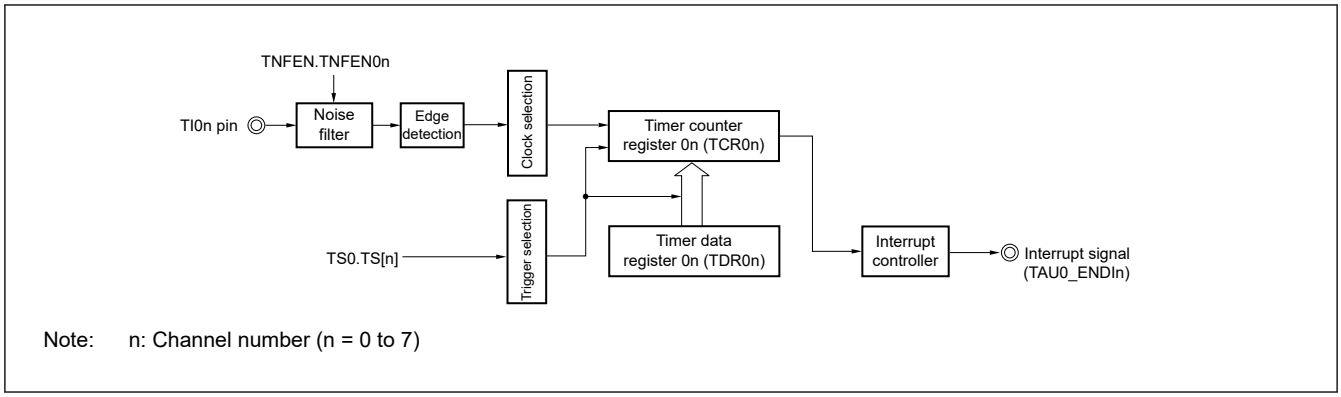
The TCR0n register counts down each time the valid input edge of the TI0n pin has been detected. When TCR0n = 0x0000, the TCR0n register loads the value of the TDR0n register again, and outputs TAU0_ENDIn.

After that, the above operation is repeated.

An irregular waveform that depends on external events is output from the TO0n pin. Stop the output by setting the TOE[n] bit of timer output enable register 0 (TOE0) to 0.

The TDR0n register can be rewritten at any time. The new value of the TDR0n register becomes valid during the next count period.

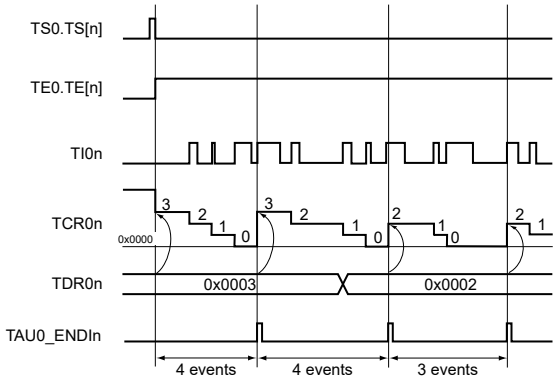
Figure 18.39 shows a block diagram of the operation as an external event counter.



Note: n: Channel number (n = 0 to 7)

Figure 18.39 Block diagram for operation as an external event counter

Figure 18.40 shows an example of basic timing during operation as an external event counter.



Note: n: Channel number (n = 0 to 7)

Figure 18.40 Example of basic timing during operation as an external event counter

Table 18.19 to Table 18.24 show register settings and procedure for operation when the external event counter.

Table 18.19 Example of TMR0n settings in external event counter mode (1 of 2)

Bit	Symbol	Set value	Function
0	OPIRQ	0	Setting of operation when counting is started 0: Neither generates TAU0_ENDIn nor inverts timer output when counting is started
3:1	MD[2:0]	011b	Operation mode of channel n 0 1 1: Event count mode
5:4	—	00b	Fixed to 0
7:6	CIS[1:0]	00b to 10b	Selection of TI0n pin input edge 0 0: Detects falling edge 0 1: Detects rising edge 1 0: Detects both edges Others: Setting prohibited
10:8	STS[2:0]	000b	Start trigger selection 0 0 0: Selects only software start
11	— (n = 0, 5, 7)	0	Fixed to 0 (channels 0, 5, 7)
	SPLIT (n = 1, 3)	1/0	Setting of SPLIT bit (channels 1, 3) 0: 16-bit timer mode 1: 8-bit timer mode
	MASTER (n = 2, 4, 6)	0	Setting of MASTER bit (channels 2, 4, 6) 0: Independent channel operation function
12	CCS	1	Count clock selection 1: Selects the TI0n pin input valid edge
13	—	0	Fixed to 0

Table 18.19 Example of TMR0n settings in external event counter mode (2 of 2)

Bit	Symbol	Set value	Function
15:14	CKS[1:0]	00b to 11b	Selection of the operating clock (f_{MCK}) 0 0: Selects CK00 as the operating clock for channel n 0 1: Selects CK02 as the operating clock (this can only be selected for channels 1 and 3) 1 0: Selects CK01 as the operating clock for channel n 1 1: Selects CK03 as the operating clock (this can only be selected for channels 1 and 3)

Table 18.20 Example of TO0 settings in external event counter mode

Bit	Symbol	Set value	Function
n	TO[n]	0	Timer output of channel n 0: Outputs 0 from TO0n

Table 18.21 Example of TOE0 settings in external event counter mode

Bit	Symbol	Set value	Function
n	TOE[n]	0	Enabling or disabling timer output for channel n 0: Stops the TO0n output operation by counting operation

Table 18.22 Example of TOL0 settings in external event counter mode

Bit	Symbol	Set value	Function
n	— (n = 0)	0	Fixed to 0 (channels 0)
	TOL[n − 1] (n = 1 to 7)		Control of timer output of channel n (channels 1 to 7) 0: Set this bit to 0 when TOM0.TOM[n − 1] = 0 (master channel output mode)

Table 18.23 Example of TOM0 settings in external event counter mode

Bit	Symbol	Set value	Function
n	— (n = 0)	0	Fixed to 0 (channels 0)
	TOM[n − 1] (n = 1 to 7)		Control of timer output mode of channel n (channels 1 to 7) 0: Sets master channel output mode

Table 18.24 Procedure for operations when the external event counter function is to be used (1 of 2)

	Step	Software operation		Hardware state
TAU default setting	<1>	Sets timer clock select register 0 (TPS0). Determines clock frequencies of CK00 to CK03.		—
Channel default setting	<2>	Sets the corresponding bit of the TAU noise filter enable register (TNFEN) to 0 (off) or 1 (on). Sets timer mode register 0n (TMR0n) (determines operation mode of channel). Sets number of counts to timer data register 0n (TDR0n). Clears the TOE[n] bit of timer output enable register 0 (TOE0) to 0.		Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	<3>	Sets the TS0.TS[n] bit to 1. The TS0.TS[n] bit automatically returns to 0 because it is a trigger bit.	→	TE0.TE[n] = 1 and count operation starts. Value of the TDR0n register is loaded to timer counter register 0n (TCR0n) and detection of the TI0n pin input edge is awaited.

Table 18.24 Procedure for operations when the external event counter function is to be used (2 of 2)

	Step	Software operation		Hardware state
During operation	<4>	Set value of the TDR0n register can be changed. The TCR0n register can always be read. The TSR0n register is not used. Set values of the TMR0n register, TOM0.TOM[n − 1], TOL0.TOL[n − 1], TO0.TO[n], and TOE0.TOE[n] bits cannot be changed.		Counter (TCR0n) counts down each time input edge of the TI0n pin has been detected. When count value reaches 0x0000, the value of the TDR0n register is loaded to the TCR0n register again, and the count operation is continued. By detecting TCR0n = 0x0000, the TAU0_ENDIn output is generated. After that, the above operation is repeated.
Operation stop	<5>	The TT0.TT[n] bit is set to 1. The TT0.TT[n] bit automatically returns to 0 because it is a trigger bit. To resume operation, go to step <3>.	→	TE0.TE[n] = 0, and count operation stops. The TCR0n register holds count value and stops.

Note: n: Channel number (n = 0 to 7)

18.7.3 Operation as a Frequency Divider (Channel 0 only)

The timer array unit can be used as a frequency divider that divides a clock input to the TI00 pin and outputs the result from the TO00 pin.

The divided clock frequency output from TO00 can be calculated by the following expression.

- When rising edge/falling edge is selected:
  Divided clock frequency = Input clock frequency / {(Set value of TDR00 + 1) × 2}
- When both edges are selected:
  Divided clock frequency ≈ Input clock frequency / (Set value of TDR00 + 1)

Timer counter register 00 (TCR00) operates as a down counter in the interval timer mode.

After the channel start trigger bit (TS[0]) of timer channel start register 0 (TS0) is set to 1, the TCR00 register loads the value of timer data register 00 (TDR00) when the TI00 valid edge is detected.

If the OPIRQ bit of timer mode register 00 (TMR00) is 0 at this time, TAU0_ENDI0 is not output and the output on TO00 is not toggled. If the OPIRQ bit of timer mode register 00 (TMR00) is 1, TAU0_ENDI0 is output and the output on TO00 is toggled.

After that, the TCR00 register counts down at the valid edge of the TI00 pin. When TCR00 = 0x0000, it toggles the output on TO00. At the same time, the TCR00 register loads the value of the TDR00 register again, and continues counting.

If detection of both the edges of the TI00 pin is selected, the duty factor error of the input clock affects the divided clock period of the TO00 output.

The period of the TO00 output clock includes a sampling error of one period of the operation clock.

Clock period of TO00 output = Ideal TO00 output clock period ± Operation clock period (error)

The TDR00 register can be rewritten at any time. The new value of the TDR00 register becomes valid during the next count period.

Figure 18.41 shows a block diagram for operation as a frequency divider.

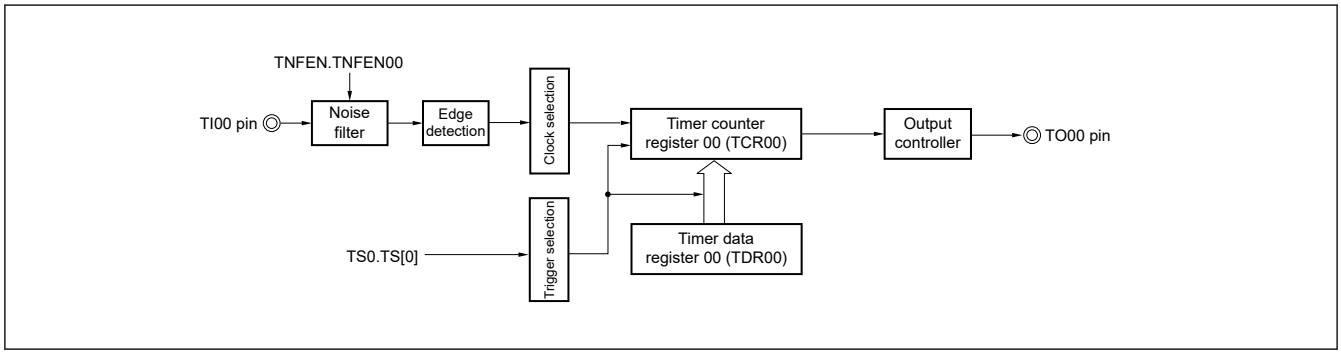


Figure 18.41 Block diagram for operation as a frequency divider

Figure 18.42 shows an example of basic timing during operation as a frequency divider (TMR00.OPIRQ = 1).

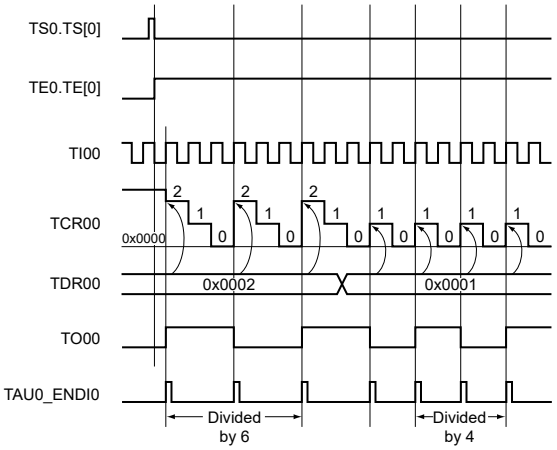


Figure 18.42 Example of basic timing during operation as a frequency divider (TMR00.OPIRQ = 1)

Table 18.25 to Table 18.30 show register settings and procedure for operation when a frequency divider.

Table 18.25 Example of TMR00 settings for operation as a frequency divider

Bit	Symbol	Set value	Function
0	OPIRQ	1/0	Setting of operation when counting is started 0: Neither generates TAU0_ENDI0 nor inverts timer output when counting is started 1: Generates TAU0_ENDI0 and inverts timer output when counting is started
3:1	MD[2:0]	000b	Operation mode of channel 0 0 0 0: Interval timer mode
5:4	—	00b	Fixed to 0
7:6	CIS[1:0]	00b to 10b	Selection of TI0n pin input edge 0 0: Detects falling edge 0 1: Detects rising edge 1 0: Detects both edges Others: Setting prohibited
10:8	STS[2:0]	000b	Start trigger selection 0 0 0: Selects only software start
11	—	0	Fixed to 0
12	CCS	1	Count clock selection 1: Selects the TI00 pin input valid edge
13	—	0	Fixed to 0
15:14	CKS[1:0]	00b or 10b	Selection of the operating clock (f_{MCK}) 0 0: Selects CK00 as the operating clock for channel 0 1 0: Selects CK01 as the operating clock for channel 0

Table 18.26 Example of TO0 settings for operation as a frequency divider

Bit	Symbol	Set value	Function
0	TO[0]	1/0	Timer output of channel 0 0: Outputs 0 from TO00 1: Outputs 1 from TO00

Table 18.27 Example of TOE0 settings for operation as a frequency divider

Bit	Symbol	Set value	Function
0	TOE[0]	1/0	Enabling or disabling timer output for channel 0 0: Stops the TO00 output operation by counting operation 1: Enables the TO00 output operation by counting operation

Table 18.28 Example of TOL0 settings for operation as a frequency divider

Bit	Symbol	Set value	Function
0	—	0	Fixed to 0 (channels 0)

Table 18.29 Example of TOM0 settings for operation as a frequency divider

Bit	Symbol	Set value	Function
0	—	0	Fixed to 0 (channels 0)

Table 18.30 Procedure for operations when the frequency divider function is to be used (1 of 2)

	Step	Software operation		Hardware state
TAU default setting	<1>	Sets timer clock select register 0 (TPS0). Determines clock frequencies of CK00 to CK03.		
Channel default setting	<2>	Sets the corresponding bit of the TAU noise filter enable register (TNFEN) to 0 (off) or 1 (on). Sets timer mode register 00 (TMR00) (determines operation mode of channel and selects the detection edge). Sets interval (period) value to timer data register 00 (TDR00).		Channel stops operating. (Clock is supplied and some power is consumed.)
	<3>	Sets the TO0.TO[0] bit and determines default level of the TO00 output.	→	The TO00 default setting level is output when the corresponding bit of the port direction register (PDR) is in the output mode.
		Sets the TOE0.TOE[0] bit to 1 and enables operation of TO00.	→	TO00 does not change because channel stops operating.
		Sets the corresponding bit of the port direction register (PDR) to 1.	→	The TO00 pin outputs the TO00 set level.
Operation start	<4>	Sets the TOE0.TOE[0] bit to 1 (only when operation is resumed). Sets the TS0.TS[0] bit to 1. The TS0.TS[0] bit automatically returns to 0 because it is a trigger bit.	→	TE0.TE[0] = 1, and count operation starts. Value of the TDR00 register is loaded to timer counter register 00 (TCR00). TAU0_ENDI0 is generated and TO00 performs toggle operation if the OPIRQ bit of the TMR00 register is 1.
During operation	<5>	Set value of the TDR00 register can be changed. The TCR00 register can always be read. The TSR00 register is not used. Set values of the TO0 and TOE0 registers can be changed. Set values of the TMR00 register cannot be changed.		Counter (TCR00) counts down. When count value reaches 0x0000, the value of the TDR00 register is loaded to the TCR00 register again, and the count operation is continued. By detecting TCR00 = 0x0000, TAU0_ENDI0 is generated and TO00 performs toggle operation. After that, the above operation is repeated.
Operation stop	<6>	The TT0.TT[0] bit is set to 1. The TT0.TT[0] bit automatically returns to 0 because it is a trigger bit.	→	TE0.TE[0] = 0, and count operation stops. The TCR00 register holds count value and stops. The TO00 output is not initialized and retains its current state.
	<7>	The TOE0.TOE[0] bit is cleared to 0 and value is set to the TO0.TO[0] bit. To resume operation, go to step <4>. To terminate the operation, go to step <8>	→	The TO00 pin outputs the TO00 set level.

Table 18.30 Procedure for operations when the frequency divider function is to be used (2 of 2)

	Step	Software operation		Hardware state
TAU stop	<8>	To hold the TO00 pin output level Sets the PSEL[4:0] bits to 00000b after the value to be held is set to the corresponding bit of the port output data register (PODR). When holding the TO00 pin output level is not necessary, setting not required.	→	The TO00 pin output level is held by port function.

18.7.4 Operation for Input Pulse Interval Measurement

The count value can be captured at the TI0n valid edge and the interval of the pulse input to TI0n can be measured. In addition, the count value can be captured by using software operation (TS0.TS[n] = 1) as a capture trigger while the TE0.TE[n] bit is set to 1.

The pulse interval can be calculated by the following expression.

TI0n input pulse interval = Period of count clock × ((0x10000 × TSR0n.OVF) + (Captured value of TDR0n + 1))

Note: The TI0n pin input is sampled using the operating clock selected with the CKS[1:0] bits of timer mode register 0n (TMR0n), so an error of up to one operating clock cycle occurs.

Timer counter register 0n (TCR0n) operates as an up counter in the capture mode.

When the channel start trigger bit (TS[n]) of timer channel start register 0 (TS0) is set to 1, the TCR0n register counts up from 0x0000 in synchronization with the count clock.

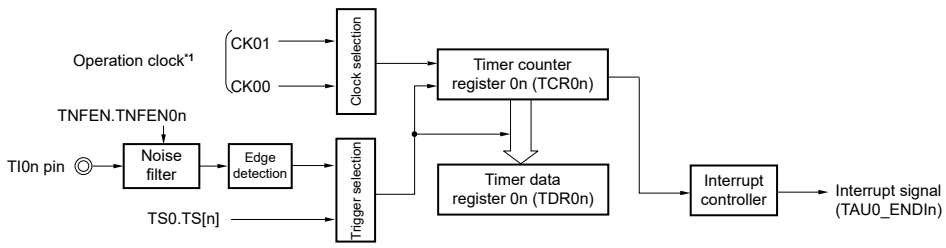
When the TI0n pin input valid edge is detected, the count value of the TCR0n register is transferred (captured) to timer data register 0n (TDR0n) and, at the same time, the TCR0n register is cleared to 0x0000, and the TAU0_ENDIn is output. If the counter overflows at this time, the OVF bit of timer status register 0n (TSR0n) is set to 1. If the counter does not overflow, the TSR0n.OVF bit is cleared. After that, the above operation is repeated.

As soon as the count value has been captured to the TDR0n register, the OVF bit of the TSR0n register is updated depending on whether the counter overflows during the measurement period. Therefore, the overflow state of the captured value can be checked.

If the counter reaches a full count for two or more periods, it is judged to be an overflow occurrence, and the OVF bit of the TSR0n register is set to 1. However, a normal interval value cannot be measured for the OVF bit, if two or more overflows occur.

Set the STS[2:0] bits of the TMR0n register to 001b to use the valid edges of TI0n as a start trigger and a capture trigger.

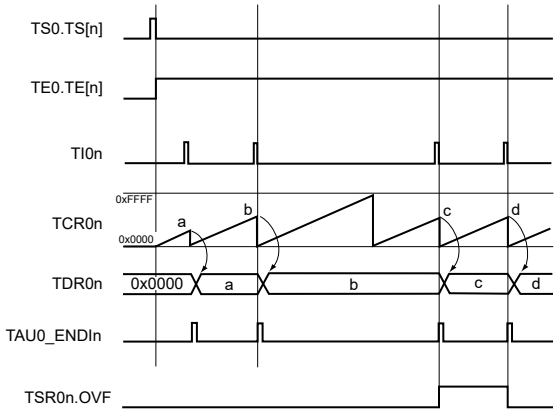
Figure 18.43 shows a block diagram for operation for input pulse interval measurement.



Note: n: Channel number (n = 0 to 7)
Note 1. For channels 1 and 3, the clock can be selected as CK00, CK01, CK02, or CK03.

Figure 18.43 Block diagram for operation for input pulse interval measurement

Figure 18.44 shows an example of basic timing during operation for input pulse interval measurement (TMR0n.OPIRQ = 0).



Note: n: Channel number (n = 0 to 7)

Figure 18.44 Example of basic timing during operation for input pulse interval measurement (TMR0n.OPIRQ = 0)

Table 18.31 to Table 18.36 show register settings and procedure for operation for input pulse interval measurement.

Table 18.31 Example of TMR0n settings for operation for input pulse interval measurement (1 of 2)

Bit	Symbol	Set value	Function
0	OPIRQ	1/0	Setting of operation when counting is started 0: Neither generates TAU0_ENDIn nor inverts timer output when counting is started 1: Generates TAU0_ENDIn and inverts timer output when counting is started
3:1	MD[2:0]	010b	Operation mode of channel n 0 1 0: Capture mode
5:4	—	00b	Fixed to 0
7:6	CIS[1:0]	00b to 10b	Selection of TI0n pin input edge 0 0: Detects falling edge 0 1: Detects rising edge 1 0: Detects both edges Others: Setting prohibited
10:8	STS[2:0]	001b	Start trigger selection 0 0 1: Selects the TI0n pin input valid edge
11	— (n = 0, 5, 7)	0	Fixed to 0 (channels 0, 5, 7)
	SPLIT (n = 1, 3)	0	Setting of SPLIT bit (channels 1, 3) 0: 16-bit timer mode
	MASTER (n = 2, 4, 6)	0	Setting of MASTER bit (channels 2, 4, 6) 0: Independent channel operation function
12	CCS	0	Count clock selection 0: Selects operation clock (f_{MCK})
13	—	0	Fixed to 0

Table 18.31 Example of TMR0n settings for operation for input pulse interval measurement (2 of 2)

Bit	Symbol	Set value	Function
15:14	CKS[1:0]	00b to 11b	Selection of the operating clock (f_{MCK}) 0 0: Selects CK00 as the operating clock for channel n 0 1: Selects CK02 as the operating clock (this can only be selected for channels 1 and 3) 1 0: Selects CK01 as the operating clock for channel n 1 1: Selects CK03 as the operating clock (this can only be selected for channels 1 and 3)

Table 18.32 Example of TO0 settings for operation for input pulse interval measurement

Bit	Symbol	Set value	Function
n	TO[n]	0	Timer output of channel n 0: Outputs 0 from TO0n

Table 18.33 Example of TOE0 settings for operation for input pulse interval measurement

Bit	Symbol	Set value	Function
n	TOE[n]	0	Enabling or disabling timer output for channel n 0: Stops the TO0n output operation by counting operation

Table 18.34 Example of TOL0 settings for operation for input pulse interval measurement

Bit	Symbol	Set value	Function
n	— (n = 0)	0	Fixed to 0 (channels 0)
	TOL[n − 1] (n = 1 to 7)		Control of timer output of channel n (channels 1 to 7) 0: Set this bit to 0 when TOM0.TOM[n − 1] = 0 (master channel output mode)

Table 18.35 Example of TOM0 settings for operation for input pulse interval measurement

Bit	Symbol	Set value	Function
n	— (n = 0)	0	Fixed to 0 (channels 0)
	TOM[n − 1] (n = 1 to 7)		Control of timer output mode of channel n (channels 1 to 7) 0: Sets master channel output mode

Table 18.36 Procedure for operations when the input pulse interval measurement function is to be used (1 of 2)

	Step	Software operation		Hardware state
TAU default setting	<1>	Sets timer clock select register 0 (TPS0). Determines clock frequencies of CK00 to CK03.		—
Channel default setting	<2>	Sets the corresponding bit of the TAU noise filter enable register (TNFEN) to 0 (off) or 1 (on). Sets timer mode register 0n (TMR0n) (determines operation mode of channel).		Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	<3>	Sets TS0.TS[n] bit to 1. The TS0.TS[n] bit automatically returns to 0 because it is a trigger bit.	→	TE0.TE[n] = 1 and count operation starts. Timer counter register 0n (TCR0n) is cleared to 0x0000. When the OPIRQ bit of the TMR0n register is 1, TAU0_ENDIn is generated.

Table 18.36 Procedure for operations when the input pulse interval measurement function is to be used (2 of 2)

	Step	Software operation		Hardware state
During operation	<4>	Set values of only the CIS[1:0] bits of the TMR0n register can be changed. The TDR0n register can always be read. The TCR0n register can always be read. The TSR0n register can always be read. Set values of the TOM0.TOM[n − 1], TOL0.TOL[n − 1], TO0.TO[n], and TOE0.TOE[n] bits cannot be changed.		Counter (TCR0n) counts up from 0x0000. When the valid edge of the TI0n pin input is detected or the TS0.TS[n] bit is set to 1, the count value is transferred (captured) to timer data register 0n (TDR0n). At the same time, the TCR0n register is cleared to 0x0000, and the TAU0_ENDIn signal is generated. If an overflow occurs at this time, the OVF bit of timer status register 0n (TSR0n) is set; if an overflow does not occur, the TSR0n.OVF bit is cleared. After that, the above operation is repeated.
Operation stop	<5>	The TT0.TT[n] bit is set to 1. The TT0.TT[n] bit automatically returns to 0 because it is a trigger bit. To resume operation, go to step <3>.	→	TE0.TE[n] = 0, and count operation stops. The TCR0n register holds count value and stops. The OVF bit of the TSR0n register is also held.

Note: n: Channel number (n = 0 to 7)

18.7.5 Operation for Input Signal High- or Low-Level Width Measurement

Note: When using a channel to implement the LIN-bus, set the ISC1 bit of the input switch control register (ISC) to 1. In the following descriptions, read TI0n as RxD2.

By starting counting at one edge of the TI0n pin input and capturing the number of counts at another edge, the signal width (high-level width or low-level width) of TI0n can be measured. The signal width of TI0n can be calculated by the following expression.

Signal width of TI0n input = Period of count clock × ((0x10000 × TSR0n.OVF) + (Captured value of TDR0n + 1))

Note: The TI0n pin input is sampled using the operating clock selected with the CKS[1:0] bits of timer mode register 0n (TMR0n), so an error equivalent to one operation clock occurs.

Timer counter register 0n (TCR0n) operates as an up counter in the capture & one-count mode.

When the channel start trigger bit (TS[n]) of timer channel start register 0 (TS0) is set to 1, the TE0.TE[n] bit is set to 1 and the TI0n pin start edge detection wait state is set.

When the TI0n pin input start edge (rising edge of the TI0n pin input when the high-level width is to be measured) is detected, the counter counts up from 0x0000 in synchronization with the count clock. When the valid capture edge (falling edge of the TI0n pin input when the high-level width is to be measured) is detected later, the count value is transferred to timer data register 0n (TDR0n) and, at the same time, TAU0_ENDIn is output. If the counter overflows at this time, the OVF bit of timer status register 0n (TSR0n) is set to 1. If the counter does not overflow, the TSR0n.OVF bit is cleared. The TCR0n register stops at the value "value transferred to the TDR0n register + 1", and the TI0n pin start edge detection wait state is set. After that, the above operation is repeated.

As soon as the count value has been captured to the TDR0n register, the OVF bit of the TSR0n register is updated depending on whether the counter overflows during the measurement period. Therefore, the overflow state of the captured value can be checked.

If the counter reaches a full count for two or more periods, it is determined to be an overflow occurrence, and the OVF bit of the TSR0n register is set to 1. However, a normal interval value cannot be measured for the OVF bit, if two or more overflows occur.

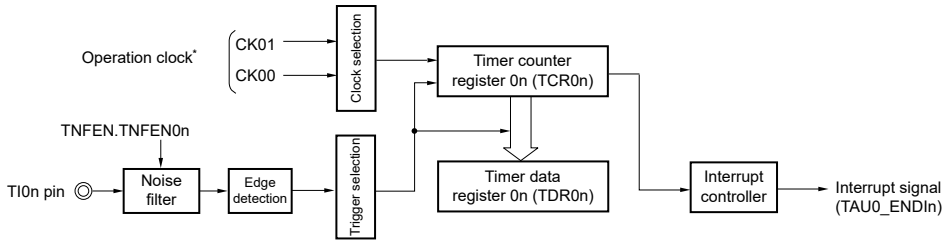
Whether the high-level width or low-level width of the TI0n pin is to be measured can be selected by using the CIS[1:0] bits of the TMR0n register.

Because this function is used to measure the signal width of the TI0n pin input, the TS0.TS[n] bit cannot be set to 1 while the TE0.TE[n] bit is 1.

CIS[1:0] of TMR0n register = 10b: Low-level width is measured.

CIS[1:0] of TMR0n register = 11b: High-level width is measured.

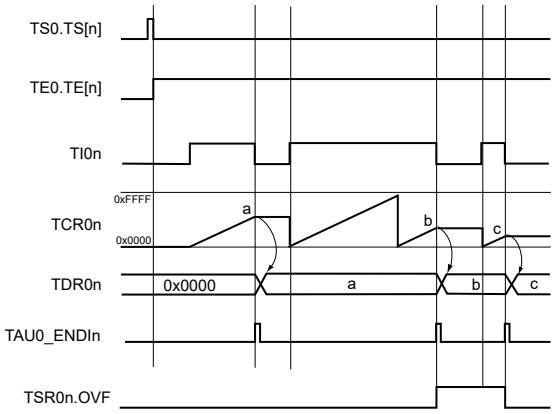
Figure 18.45 shows a block diagram for operation of input signal high- or low-level width measurement.



Note: n: Channel number (n = 0 to 7)
Note 1. For channels 1 and 3, the clock can be selected as CK00, CK01, CK02, or CK03.

Figure 18.45 Block diagram for operation of input signal high- or low-level width measurement

Figure 18.46 shows an example of basic timing during operation for input signal high- or low-level width measurement.



Note: n: Channel number (n = 0 to 7)

Figure 18.46 Example of basic timing during operation for input signal high- or low-level width measurement

Table 18.37 to Table 18.42 show register settings and procedure for operation for input signal high- or low-level width measurement.

Table 18.37 Example of TMR0n settings for operation for input signal high- or low-level width measurement (1 of 2)

Bit	Symbol	Set value	Function
0	OPIRQ	0	Setting of operation when counting is started 0: Neither generates TAU0_ENDIn nor inverts timer output when counting is started
3:1	MD[2:0]	110b	Operation mode of channel n 1 1 0: Capture & one-count
5:4	—	00b	Fixed to 0
7:6	CIS[1:0]	10b to 11b	Selection of TI0n pin input valid edge 1 0: Both edges (to measure low-level width) 1 1: Both edges (to measure high-level width) Others: Setting prohibited
10:8	STS[2:0]	010b	Start trigger selection 0 1 0: Selects the TI0n pin input valid edge

Table 18.37 Example of TMR0n settings for operation for input signal high- or low-level width measurement (2 of 2)

Bit	Symbol	Set value	Function
11	— (n = 0, 5, 7)	0	Fixed to 0 (channels 0, 5, 7)
	SPLIT (n = 1, 3)	0	Setting of SPLIT bit (channels 1, 3) 0: 16-bit timer mode
	MASTER (n = 2, 4, 6)	0	Setting of MASTER bit (channels 2, 4, 6) 0: Independent channel operation function
12	CCS	0	Count clock selection 0: Selects operation clock (f_{MCK})
13	—	0	Fixed to 0.
15:14	CKS[1:0]	00b to 11b	Selection of the operating clock (f_{MCK}) 0 0: Selects CK00 as the operating clock for channel n 0 1: Selects CK02 as the operating clock (this can only be selected for channels 1 and 3) 1 0: Selects CK01 as the operating clock for channel n 1 1: Selects CK03 as the operating clock (this can only be selected for channels 1 and 3)

Table 18.38 Example of TO0 settings for operation of input signal high- or low-level width measurement

Bit	Symbol	Set value	Function
n	TO[n]	0	Timer output of channel n 0: Outputs 0 from TO0n

Table 18.39 Example of TOE0 settings for operation of input signal high- or low-level width measurement

Bit	Symbol	Set value	Function
n	TOE[n]	0	Enabling or disabling timer output for channel n 0: Stops the TO0n output operation by counting operation

Table 18.40 Example of TOL0 settings for operation of input signal high- or low-level width measurement

Bit	Symbol	Set value	Function
n	— (n = 0)	0	Fixed to 0 (channels 0)
	TOL[n − 1] (n = 1 to 7)		Control of timer output of channel n (channels 1 to 7) 0: Sets this bit to 0 when TOM0.TOM[n − 1] = 0 (master channel output mode)

Table 18.41 Example of TOM0 settings for operation of input signal high- or low-level width measurement

Bit	Symbol	Set value	Function
n	— (n = 0)	0	Fixed to 0 (channels 0)
	TOM[n − 1] (n = 1 to 7)		Control of timer output mode of channel n (channels 1 to 7) 0: Sets master channel output mode

Table 18.42 Procedure for operations when the input signal high- or low-level width measurement function is to be used (1 of 2)

	Step	Software operation		Hardware state
TAU default setting	<1>	Sets timer clock select register 0 (TPS0). Determines clock frequencies of CK00 to CK03.		—

Table 18.42 Procedure for operations when the input signal high- or low-level width measurement function is to be used (2 of 2)

	Step	Software operation		Hardware state
Channel default setting	<2>	Sets the corresponding bit of the TAU noise filter enable register (TNFEN) to 0 (off) or 1 (on). Sets timer mode register 0n (TMR0n) (determines operation mode of channel). Clears the TOE0.TOE[n] bit to 0 and stops operation of TO0n.		Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	<3>	Sets TS0.TS[n] bit to 1. The TS0.TS[n] bit automatically returns to 0 because it is a trigger bit.	→	TE0.TE[n] = 1, and the TI0n pin start edge detection wait state is set.
	<4>	Detects the TI0n pin input count start valid edge.	→	Clears timer counter register 0n (TCR0n) to 0x0000 and starts counting up.
During operation	<5>	Set value of the TDR0n register can always be read. The TCR0n register can always be read. The TSR0n register can always be read. Set values of the TMR0n register, TOM0.TOM[n − 1], TOL0.TOL[n − 1], TO0.TO[n], and TOE0.TOE[n] bits cannot be changed.		When the TI0n pin start edge is detected, the counter (TCR0n) counts up from 0x0000. If a capture edge of the TI0n pin is detected, the count value is transferred to timer data register 0n (TDR0n) and TAU0_ENDIn is generated. If an overflow occurs at this time, the OVF bit of timer status register 0n (TSR0n) is set; if an overflow does not occur, the TSR0n.OVF bit is cleared. The TCR0n register stops the count operation until the next TI0n pin start edge is detected.
Operation stop	<6>	The TT0.TT[n] bit is set to 1. The TT0.TT[n] bit automatically returns to 0 because it is a trigger bit. To resume operation, go to step <3>.	→	TE0.TE[n] = 0, and count operation stops. The TCR0n register holds count value and stops. The OVF bit of the TSR0n register is also held.

Note: n: Channel number (n = 0 to 7)

18.7.6 Operation as a Delay Counter

It is possible to start counting down when the valid edge of the TI0n pin input is detected (an external event), and then generate TAU0_ENDIn (a timer interrupt) after any specified interval.

It is also possible to start counting down and generate TAU0_ENDIn (timer interrupt) at any interval by setting TS0.TS[n] to 1 by software while TE0.TE[n] = 1.

The interrupt generation period can be calculated by the following expression.

Generation period of TAU0_ENDIn (timer interrupt) = Period of count clock × (Set value of TDR0n + 1)

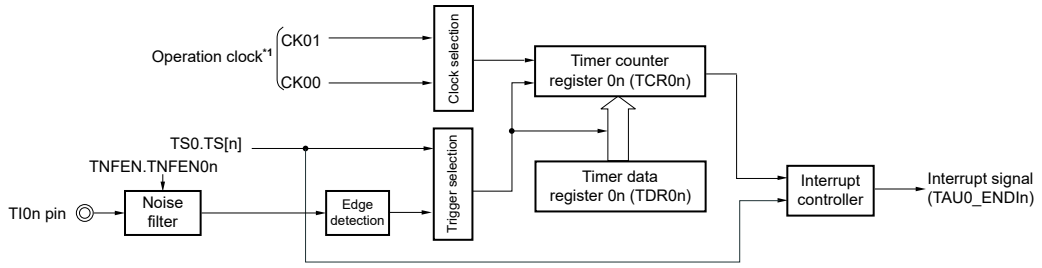
Timer counter register 0n (TCR0n) operates as a down counter in the one-count mode.

When the channel start trigger bit (TS[n], TSH1, TSH3) of timer channel start register 0 (TS0) is set to 1, the TE0.TE[n], TEH1, and TEH3 bits are set to 1 and the TI0n pin input valid edge detection wait state is set.

Timer counter register 0n (TCR0n) starts operating upon TI0n pin input valid edge detection and loads the value of timer data register 0n (TDR0n). The TCR0n register counts down from the value of the TDR0n register it has loaded, in synchronization with the count clock. When TCR0n = 0x0000, it outputs TAU0_ENDIn and stops counting until the next TI0n pin input valid edge is detected.

The TDR0n register can be rewritten at any time. The new value of the TDR0n register becomes valid from the next period.

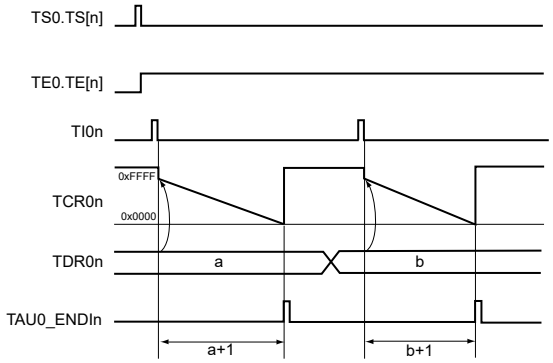
Figure 18.47 shows a block diagram for operation as a delay counter.



Note: n: Channel number (n = 0 to 7)
Note 1. For channels 1 and 3, the clock can be selected as CK00, CK01, CK02, or CK03.

Figure 18.47 Block diagram for operation as a delay counter

Figure 18.48 shows an example of basic timing during operation as a delay counter.



Note: n: Channel number (n = 0 to 7)

Figure 18.48 Example of basic timing during operation as a delay counter

Table 18.43 to Table 18.48 show register settings and procedure for operation as a delay counter.

Table 18.43 Example of TMR0n settings for operation as a delay counter (1 of 2)

Bit	Symbol	Set value	Function
0	OPIRQ	1/0	Start trigger during operation 0: Trigger input is invalid 1: Trigger input is valid
3:1	MD[2:0]	100b	Operation mode of channel n 1 0 0: One-count mode
5:4	—	00b	Fixed to 0
7:6	CIS[1:0]	00b to 10b	Selection of TI0n pin input edge 0 0: Detects falling edge 0 1: Detects rising edge 1 0: Detects both edges Others: Setting prohibited
10:8	STS[2:0]	001b	Start trigger selection 0 0 1: Selects the TI0n pin input valid edge

Table 18.43 Example of TMR0n settings for operation as a delay counter (2 of 2)

Bit	Symbol	Set value	Function
11	— (n = 0, 5, 7)	0	Fixed to 0 (channels 0, 5, 7)
	SPLIT (n = 1, 3)	1/0	Setting of SPLIT bit (channels 1, 3) 0: 16-bit timer mode 1: 8-bit timer mode
	MASTER (n = 2, 4, 6)	0	Setting of MASTER bit (channels 2, 4, 6) 0: Independent channel operation function
12	CCS	0	Count clock selection 0: Selects operation clock (f_{MCK})
13	—	0	Fixed to 0
15:14	CKS[1:0]	00b to 11b	Selection of the operating clock (f_{MCK}) 0 0: Selects CK00 as the operating clock for channel n 0 1: Selects CK02 as the operating clock (this can only be selected for channels 1 and 3) 1 0: Selects CK01 as the operating clock for channel n 1 1: Selects CK03 as the operating clock (this can only be selected for channels 1 and 3)

Table 18.44 Example of TO0 settings for operation as a delay counter

Bit	Symbol	Set value	Function
n	TO[n]	0	Timer output of channel n 0: Outputs 0 from TO0n

Table 18.45 Example of TOE0 settings for operation as a delay counter

Bit	Symbol	Set value	Function
n	TOE[n]	0	Enabling or disabling timer output for channel n 0: Stops the TO0n output operation by counting operation

Table 18.46 Example of TOL0 settings for operation as a delay counter

Bit	Symbol	Set value	Function
n	— (n = 0)	0	Fixed to 0 (channels 0)
	TOL[n − 1] (n = 1 to 7)		Control of timer output of channel n (channels 1 to 7) 0: Sets this bit to 0 when TOM0.TOM[n − 1] = 0 (master channel output mode)

Table 18.47 Example of TOM0 settings for operation as a delay counter

Bit	Symbol	Set value	Function
n	— (n = 0)	0	Fixed to 0 (channels 0)
	TOM[n − 1] (n = 1 to 7)		Control of timer output mode of channel n (channels 1 to 7) 0: Sets master channel output mode

Table 18.48 Procedure for operations when the delay counter function is to be used (1 of 2)

	Step	Software operation		Hardware state
TAU default setting	<1>	Sets timer clock select register 0 (TPS0). Determines clock frequencies of CK00 to CK03.		—

Table 18.48 Procedure for operations when the delay counter function is to be used (2 of 2)

	Step	Software operation		Hardware state
Channel default setting	<2>	Sets the corresponding bit of the TAU noise filter enable register (TNFEN) to 0 (off) or 1 (on). Sets timer mode register 0n (TMR0n) (determines operation mode of channel). TAU0_ENDIn output delay is set to timer data register 0n (TDR0n). Clears the TOE0.TOE[n] bit to 0 and stops operation of TO0n.		Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	<3>	Sets TS0.TS[n] bit to 1. The TS0.TS[n] bit automatically returns to 0 because it is a trigger bit.	→	TE0.TE[n] = 1, and the start trigger detection (the valid edge of the TI0n pin input is detected or the TS0.TS[n] bit is set to 1) wait state is set.
	<4>	The counter starts counting down by the next start trigger detection. ● Detects the TI0n pin input valid edge. ● Sets the TS0.TS[n] bit to 1 by the software.	→	Value of the TDR0n register is loaded to the timer counter register 0n (TCR0n).
During operation	<5>	Set value of the TDR0n register can be changed. The TCR0n register can always be read. The TSR0n register is not used.		The counter (TCR0n) counts down. When the count value of TCR0n reaches 0x0000, the TAU0_ENDIn output is generated, and the count operation stops until the next start trigger detection (the valid edge of the TI0n pin input is detected or the TS0.TS[n] bit is set to 1).
Operation stop	<6>	The TT0.TT[n] bit is set to 1. The TT0.TT[n] bit automatically returns to 0 because it is a trigger bit. To resume operation, go to step <3>.	→	TE0.TE[n] = 0, and count operation stops. The TCR0n register holds count value and stops.

Note: n: Channel number (n = 0 to 7)

18.8 Simultaneous Channel Operation Function of Timer Array Unit

18.8.1 Operation for the One-shot Pulse Output Function

By using two channels as a set, a one-shot pulse having any delay pulse width can be generated from the signal input to the TI0n pin.

The delay time and pulse width can be calculated by the following expressions.

Delay time = {Set value of TDR0n (master) + 2} × Count clock period

Pulse width = {Set value of TDR0p (slave)} × Count clock period

The master channel operates in the one-count mode and counts the delays. Timer counter register 0n (TCR0n) of the master channel starts operating upon start trigger detection and loads the value of timer data register 0n (TDR0n).

The TCR0n register counts down from the value of the TDR0n register it has loaded, in synchronization with the count clock. When TCR0n = 0x0000, it outputs TAU0_ENDIn and stops counting until the next start trigger is detected.

The slave channel operates in the one-count mode and counts the pulse width. The TCR0p register of the slave channel starts operation using TAU0_ENDIn of the master channel as a start trigger, and loads the value of the TDR0p register. The TCR0p register counts down from the value of the TDR0p register it has loaded, in synchronization with the count value. When count value = 0x0000, it outputs TAU0_ENDIp and stops counting until the next start trigger (TAU0_ENDIn of the master channel) is detected. The output level of TO0p becomes active one count clock after generation of TAU0_ENDIn from the master channel, and inactive when TCR0p = 0x0000.

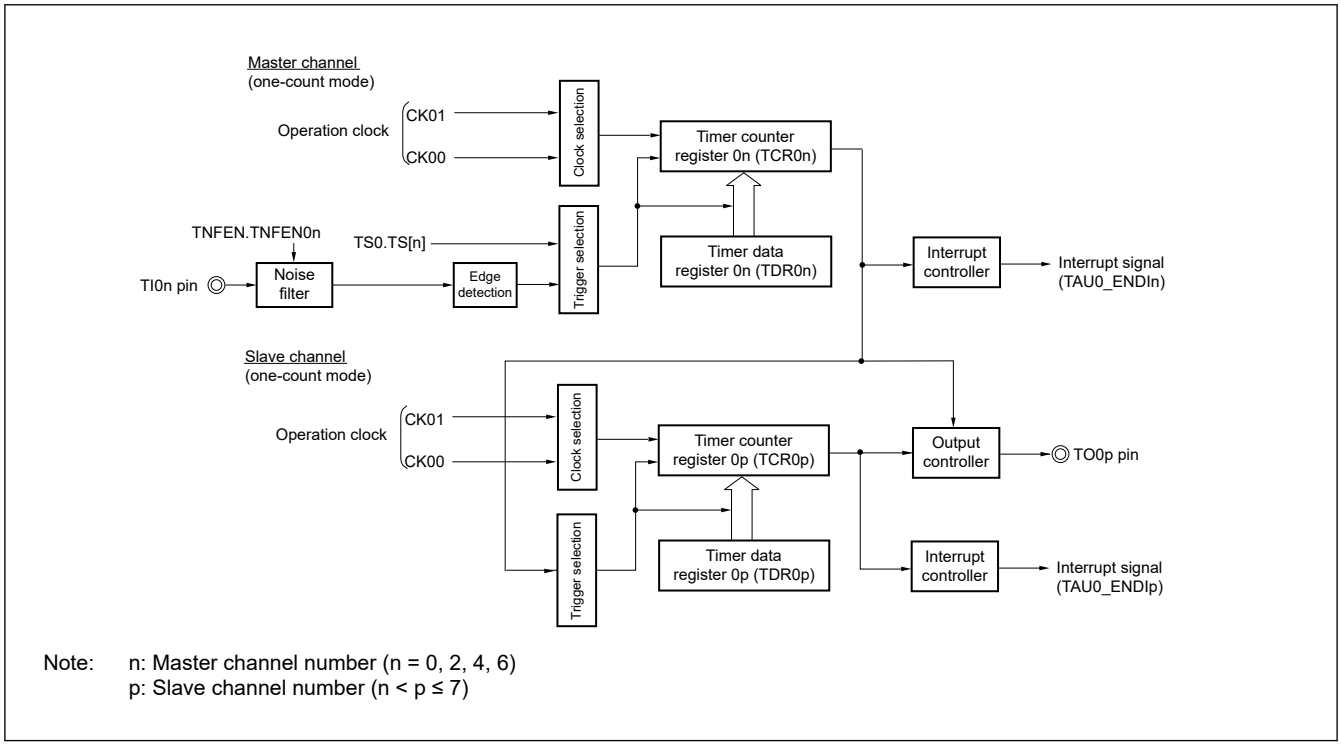
Instead of using the TI0n pin input, a one-shot pulse can also be output using the software operation (TS0.TS[n] = 1) as a start trigger.

Note: The timing of loading of timer data register 0n (TDR0n) of the master channel is different from that of the TDR0p register of the slave channel. If the TDR0n and TDR0p registers are rewritten during operation, therefore, an illegal waveform is output. Rewrite the TDR0n register after TAU0_ENDIn is generated and the TDR0p register after TAU0_ENDIp is generated.

Note: n: Master channel number (n = 0, 2, 4, 6)

p: Slave channel number (n < p ≤ 7)

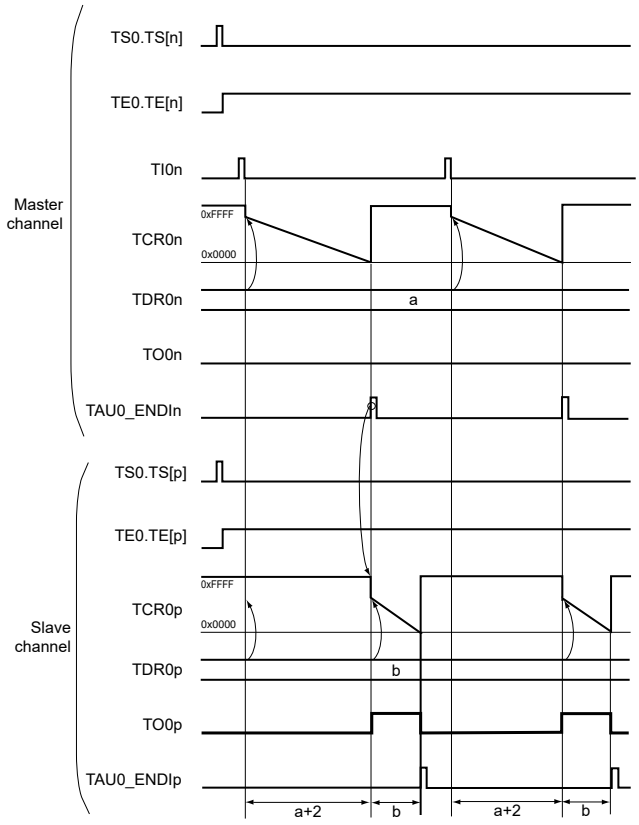
Figure 18.49 shows a block diagram for operation for the one-shot pulse output function.



Note: n: Master channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

Figure 18.49 Block diagram for operation for the one-shot pulse output function

Figure 18.50 shows an example of basic timing during operation for the one-shot pulse output function.



Note: n: Master channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

Figure 18.50 Example of basic timing during operation for the one-shot pulse output function

Table 18.49 to Table 18.53 show register settings for the master channel when the one-shot pulse output function is to be used.

Table 18.49 Example of TMR0n settings for the master channel when the one-shot pulse output function is to be used (1 of 2)

Bit	Symbol	Set value	Function
0	OPIRQ	0	Start trigger during operation 0: Trigger input is invalid
3:1	MD[2:0]	100b	Operation mode of channel n 1 0 0: One-count mode
5:4	—	00b	Fixed to 0
7:6	CIS[1:0]	00b to 10b	Selection of TI0n pin input edge 0 0: Detects falling edge 0 1: Detects rising edge 1 0: Detects both edges Others: Setting prohibited
10:8	STS[2:0]	001b	Start trigger selection 0 0 1: Selects the TI0n pin input valid edge

Table 18.49 Example of TMR0n settings for the master channel when the one-shot pulse output function is to be used (2 of 2)

Bit	Symbol	Set value	Function
11	— (n = 0)	0	Fixed to 0 (channels 0)
	MASTER (n = 2, 4, 6)	1	Setting of MASTER bit (channels 2, 4, 6) 1: Master channel
12	CCS	0	Count clock selection 0: Selects operation clock (f_{MCK}).
13	—	0	Fixed to 0.
15:14	CKS[1:0]	00b or 10b	Selection of the operating clock (f_{MCK}) 0 0: Selects CK00 as the operating clock for channel n 1 0: Selects CK01 as the operating clock for channel n

Table 18.50 Example of TO0 settings for the master channel when the one-shot pulse output function is to be used

Bit	Symbol	Set value	Function
n	TO[n]	0	Timer output of channel n 0: Outputs 0 from TO0n

Table 18.51 Example of TOE0 settings for the master channel when the one-shot pulse output function is to be used

Bit	Symbol	Set value	Function
n	TOE[n]	0	Enabling or disabling timer output for channel n 0: Stops the TO0n output operation by counting operation.

Table 18.52 Example of TOL0 settings for the master channel when the one-shot pulse output function is to be used

Bit	Symbol	Set value	Function
n	— (n = 0)	0	Fixed to 0 (channels 0)
	TOL[n − 1] (n = 2, 4, 6)		Control of timer output of channel n (channels 2, 4, 6) 0: Set this bit to 0 when TOM0.TOM[n-1] = 0 (master channel output mode)

Table 18.53 Example of TOM0 settings for the master channel when the one-shot pulse output function is to be used

Bit	Symbol	Set value	Function
n	— (n = 0)	0	Fixed to 0 (channels 0)
	TOM[n − 1] (n = 2, 4, 6)		Control of timer output mode of channel n (channels 2, 4, 6) 0: Sets master channel output mode

Table 18.54 to Table 18.58 show register settings for the slave channel when the one-shot pulse output function is to be used.

Table 18.54 Example of TMR0p settings for the slave channel when the one-shot pulse output function is to be used (1 of 2)

Bit	Symbol	Set value	Function
0	OPIRQ	0	Start trigger during operation 0: Trigger input is invalid.
3:1	MD[2:0]	100b	Operation mode of channel p 1 0 0: One-count mode

Table 18.54 Example of TMR0p settings for the slave channel when the one-shot pulse output function is to be used (2 of 2)

Bit	Symbol	Set value	Function
5:4	—	00b	Fixed to 0
7:6	CIS[1:0]	00b	Selection of TI0p pin input edge 0 0: Set to 00b because the TI0p input pin is not to be used
10:8	STS[2:0]	100b	Start trigger selection 1 0 0: Selects TAU0_ENDIn of master channel
11	— (p = 5, 7)	0	Fixed to 0 (channels 5, 7)
	SPLIT (p = 1, 3)	0	Setting of SPLIT bit (channels 1, 3) 0: 16-bit timer mode
	MASTER (p = 2, 4, 6)	0	Setting of MASTER bit (channels 2, 4, 6) 0: Slave channel operation function
12	CCS	0	Count clock selection 0: Selects operation clock (f_{MCK})
13	—	0	Fixed to 0
15:14	CKS[1:0]	00b or 10b	Selection of the operating clock (f_{MCK}) (make the same setting as master channel.) 0 0: Selects CK00 as the operating clock for channel p 1 0: Selects CK01 as the operating clock for channel p

Table 18.55 Example of TO0 settings for the slave channel when the one-shot pulse output function is to be used

Bit	Symbol	Set value	Function
p	TO[p]	1/0	Timer output of channel p 0: Outputs 0 from TO0p 1: Outputs 1 from TO0p

Table 18.56 Example of TOE0 settings for the slave channel when the one-shot pulse output function is to be used

Bit	Symbol	Set value	Function
p	TOE[p]	1/0	Enabling or disabling timer output for channel p 0: Stops the TO0p output operation by counting operation 1: Enables the TO0p output operation by counting operation

Table 18.57 Example of TOL0 settings for the slave channel when the one-shot pulse output function is to be used

Bit	Symbol	Set value	Function
p	TOL[p − 1]	1/0	Control of timer output of channel p 0: Positive logic output (active-high) 1: Negative logic output (active-low)

Table 18.58 Example of TOM0 settings for the slave channel when the one-shot pulse output function is to be used

Bit	Symbol	Set value	Function
p	TOM[p − 1]	1	Control of timer output mode of channel p (channels 1 to 7) 1: Sets the slave channel output mode

Table 18.59 show procedure for operations when the one-shot pulse output function is to be used.

Table 18.59 Procedure for operations when the one-shot pulse output function is to be used (1 of 2)

	Step	Software operation		Hardware state
TAU default setting	<1>	Sets timer clock select register 0 (TPS0). Determines clock frequencies of CK00 and CK01.		—
Channel default setting	<2>	Sets the corresponding bit of the TAU noise filter enable register (TNFEN) to 1. Sets timer mode registers 0n, 0p (TMR0n, TMR0p) of two channels to be used (determines operation mode of channels). An output delay is set to timer data register 0n (TDR0n) of the master channel, and a pulse width is set to the TDR0p register of the slave channel.		Channel stops operating. (Clock is supplied and some power is consumed.)
	<3>	Sets slave channel. The TOM0.TOM[p − 1] bit of timer output mode register 0 (TOM0) is set to 1 (slave channel output mode). Sets the TOL0.TOL[p − 1] bit.		The TO0p pin goes into Hi-Z output state.
		Sets the TO0.TO[p] bit and determines default level of the TO0p output.	→	The TO0p default setting level is output when the corresponding bit of the port direction register (PDR) is in the output mode.
		Sets the TOE0.TOE[p] bit to 1 and enables operation of TO0p.	→	TO0p does not change because channel stops operating.
		Sets the corresponding bit of the port direction register (PDR) to 1.	→	The TO0p pin outputs the TO0p set level.
Operation start	<4>	Sets the TOE0.TOE[p] bit (slave) to 1 (only when operation is resumed). The TS0.TS[n] (master) and TS0.TS[p] (slave) bits of timer channel start register 0 (TS0) are set to 1 at the same time. The TS0.TS[n], TS[p] bits automatically return to 0 because they are trigger bits.	→	The TE0.TE[n], TE[p] bits are set to 1 and the master channel enters the start trigger detection (the valid edge of the TI0n pin input is detected or the TS0.TS[n] bit of the master channel is set to 1) wait state. Counter stops operating.
	<5>	Count operation of the master channel is started by start trigger detection of the master channel. ● Detects the TI0n pin input valid edge. ● Sets the TS0.TS[n] bit of the master channel to 1 by software[*1].		Master channel starts counting.
During operation	<6>	Set values of only the TMR0n.CIS[1:0] bits of the TMR0n register can be changed. Set values of the TMR0p, TDR0n, TDR0p registers, TOM0.TOM[n − 1], TOM[p − 1], TOL0.TOL[n − 1], and TOL[p − 1] bits cannot be changed. The TCR0n and TCR0p registers can always be read. The TSR0n and TSR0p registers are not used. Set values of the TO0 and TOE0 registers by slave channel can be changed.		Master channel loads the value of the TDR0n register to timer counter register 0n (TCR0n) by the start trigger detection (the valid edge of the TI0n pin input is detected or the TS0.TS[n] bit of the master channel is set to 1), and the counter starts counting down. When the count value reaches TCR0n = 0x0000, the TAU0_ENDIn output is generated, and the counter stops until the next start trigger detection. The slave channel, triggered by TAU0_ENDIn of the master channel, loads the value of the TDR0p register to the TCR0p register, and the counter starts counting down. The output level of TO0p becomes active one count clock after generation of TAU0_ENDIn from the master channel. It becomes inactive when TCR0p = 0x0000, and the counting operation is stopped. After that, the above operation is repeated.
Operation stop	<7>	The TT0.TT[n] (master) and TT[p] (slave) bits are set to 1 at the same time. The TT0.TT[n], TT[p] bits automatically return to 0 because they are trigger bits.	→	TE0.TE[n], TE[p] = 0, and count operation stops. The TCR0n and TCR0p registers hold count value and stop. The TO0p output is not initialized and retains its current state.
	<8>	The TOE0.TOE[p] bit of slave channel is cleared to 0 and value is set to the TO0.TO[p] bit. To resume operation, go to step <4>. To terminate the operation, go to step <9>	→	The TO0p pin outputs the TO0p set level.

Table 18.59 Procedure for operations when the one-shot pulse output function is to be used (2 of 2)

	Step	Software operation		Hardware state
TAU stop	<9>	To hold the TO0p pin output level Sets the PSEL[4:0] bits to 00000b after the value to be held is set to the corresponding bit of the port output data register (PODR). When holding the TO0p pin output level is not necessary, setting is not required.	→	The TO0p pin output level is held by port function.

Note: n: Master channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)
Note 1. Do not set the TS0.TS[n] bit of the slave channel to 1.

18.8.2 Operation for the PWM Function

Two channels can be used as a set to generate a pulse of any period and duty factor.

The period and duty factor of the output pulse can be calculated by the following expressions.

Pulse period = {Set value of TDR0n (master) + 1} × Count clock period

Duty factor [%] = {Set value of TDR0p (slave)} / {Set value of TDR0n (master) + 1} × 100

0% output: Set value of TDR0p (slave) = 0x0000

100% output: Set value of TDR0p (slave) ≥ {Set value of TDR0n (master) + 1}

Note: Although the duty factor exceeds 100% if the set value of TDR0p (slave) > (set value of TDR0n (master) + 1), it totals to 100% output.

The master channel operates in the interval timer mode. If the channel start trigger bit (TS[n]) of timer channel start register 0 (TS0) is set to 1, an interrupt (TAU0_ENDIn) is output, the value set to timer data register 0n (TDR0n) is loaded to timer counter register 0n (TCR0n), and the counter counts down in synchronization with the count clock. When the counter reaches 0x0000, TAU0_ENDIn is output, the value of the TDR0n register is loaded again to the TCR0n register, and the counter counts down. This operation is repeated until the channel stop trigger bit (TT[n]) of timer channel stop register 0 (TT0) is set to 1.

If two channels are used to output a PWM waveform, the period until the master channel counts down to 0x0000 is the PWM output (TO0p) cycle.

The slave channel operates in one-count mode. By using TAU0_ENDIn from the master channel as a start trigger, the TCR0p register loads the value of the TDR0p register and the counter counts down to 0x0000. When the counter reaches 0x0000, it outputs TAU0_ENDIp and waits until the next start trigger (TAU0_ENDIn from the master channel) is generated.

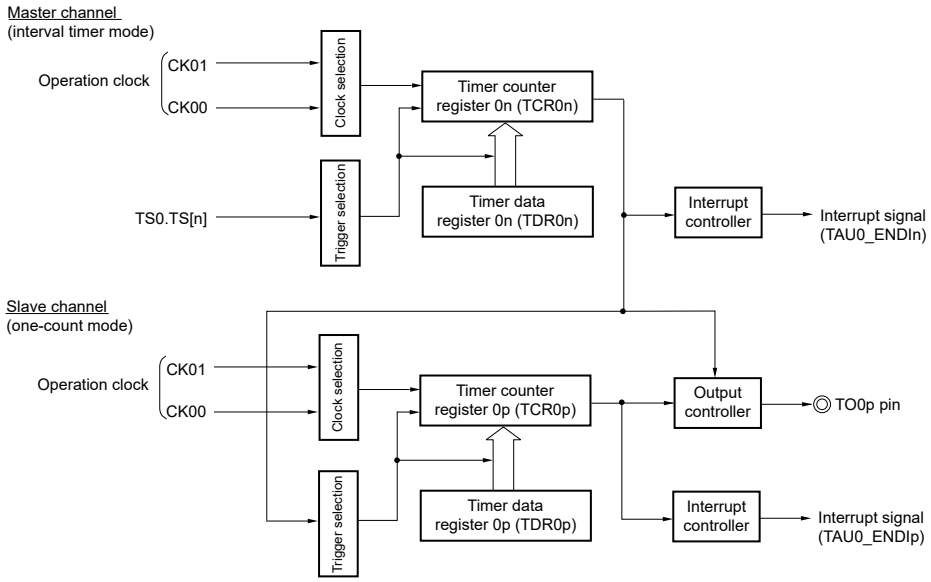
If two channels are used to output a PWM waveform, the period until the slave channel counts down to 0x0000 is the PWM output (TO0p) duty.

PWM output (TO0p) goes to the active level one clock after the master channel generates TAU0_ENDIn and goes to the inactive level when the TCR0p register of the slave channel becomes 0x0000.

Note: To rewrite both timer data register 0n (TDR0n) of the master channel and the TDR0p register of the slave channel, a write access is necessary two times. The timing at which the values of the TDR0n and TDR0p registers are loaded to the TCR0n and TCR0p registers is upon occurrence of TAU0_ENDIn of the master channel. Thus, when rewriting is performed split before and after occurrence of TAU0_ENDIn of the master channel, the TO0p pin cannot output the expected waveform. To rewrite both the TDR0n register of the master and the TDR0p register of the slave, therefore, be sure to rewrite both the registers immediately after TAU0_ENDIn is generated from the master channel.

Note: n: Master channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

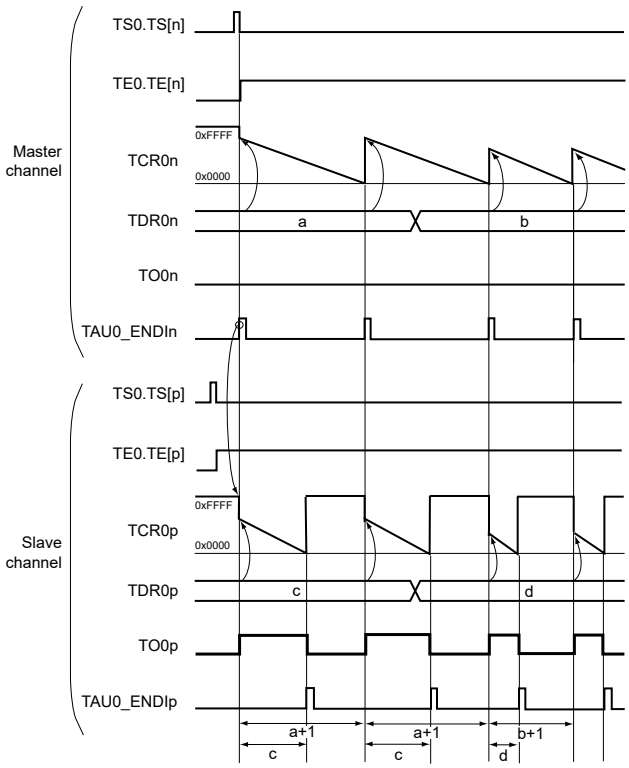
Figure 18.51 shows a block diagram for operation for the PWM function.



Note: n: Master channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

Figure 18.51 Block diagram for operation for the PWM function

Figure 18.52 shows an example of basic timing during operation for the PWM function.



Note: n: Master channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

Figure 18.52 Example of basic timing during operation for the PWM function

Table 18.60 to Table 18.64 show register settings for the master channel when the PWM function is to be used.

Table 18.60 Example of TMR0n settings for the master channel when the PWM function is to be used

Bit	Symbol	Set value	Function
0	OPIRQ	1	Setting of operation when counting is started 1: Generates TAU0_ENDIn when counting is started
3:1	MD[2:0]	000b	Operation mode of channel n 0 0 0: Interval timer
5:4	—	00b	Fixed to 0
7:6	CIS[1:0]	00b	Selection of TI0n pin input edge 0 0: Set to 00b because the TI0n input pin is not to be used
10:8	STS[2:0]	000b	Start trigger selection 0 0 0: Selects only software start
11	— (n = 0)	0	Fixed to 0 (channels 0)
	MASTER (n = 2, 4, 6)	1	Setting of MASTER bit (channels 2, 4, 6) 1: Master channel
12	CCS	0	Count clock selection 0: Selects operation clock (f_{MCK})
13	—	0	Fixed to 0
15:14	CKS[1:0]	00b or 10b	Selection of the operating clock (f_{MCK}) 0 0: Selects CK00 as the operating clock for channel n 1 0: Selects CK01 as the operating clock for channel n

Table 18.61 Example of TO0 settings for the master channel when the PWM function is to be used

Bit	Symbol	Set value	Function
n	TO[n]	0	Timer output of channel n 0: Outputs 0 from TO0n

Table 18.62 Example of TOE0 settings for the master channel when the PWM function is to be used

Bit	Symbol	Set value	Function
n	TOE[n]	0	Enabling or disabling timer output for channel n 0: Stops the TO0n output operation by counting operation.

Table 18.63 Example of TOL0 settings for the master channel when the PWM function is to be used

Bit	Symbol	Set value	Function
n	— (n = 0)	0	Fixed to 0 (channels 0)
	TOL[n − 1] (n = 2, 4, 6)		Control of timer output of channel n (channels 2, 4, 6) 0: Set this bit to 0 when TOM0.TOM[n − 1] = 0 (master channel output mode)

Table 18.64 Example of TOM0 settings for the master channel when the PWM function is to be used

Bit	Symbol	Set value	Function
n	— (n = 0)	0	Fixed to 0 (channels 0)
	TOM[n − 1] (n = 2, 4, 6)		Control of timer output mode of channel n (channels 2, 4, 6) 0: Sets master channel output mode.

Table 18.65 to Table 18.69 show register settings for the slave channel when the PWM function is to be used.

Table 18.65 Example of TMR0p settings for the slave channel when the PWM function is to be used

Bit	Symbol	Set value	Function
0	OPIRQ	1	Start trigger during operation 1: Trigger input is valid
3:1	MD[2:0]	100b	Operation mode of channel p 1 0 0: One-count mode
5:4	—	00b	Fixed to 0
7:6	CIS[1:0]	00b	Selection of TI0p pin input valid edge 0 0: Set to 00b because the TI0p input pin is not to be used
10:8	STS[2:0]	100b	Start trigger selection 1 0 0: Selects TAU0_ENDIn of master channel
11	— (p = 5, 7)	0	Fixed to 0 (channels 5, 7)
	SPLIT (p = 1, 3)	0	Setting of SPLIT bit (channels 1, 3) 0: 16-bit timer mode
	MASTER (p = 2, 4, 6)	0	Setting of MASTER bit (channels 2, 4, 6) 0: Slave channel
12	CCS	0	Count clock selection 0: Selects operation clock (f_{MCK})
13	—	0	Fixed to 0
15:14	CKS[1:0]	00b or 10b	Selection of the operating clock (f_{MCK}) (Make the same setting as master channel.) 0 0: Selects CK00 as the operating clock for channel p 1 0: Selects CK01 as the operating clock for channel p

Table 18.66 Example of TO0 settings for the slave channel when the PWM function is to be used

Bit	Symbol	Set value	Function
p	TO[p]	1/0	Timer output of channel p 0: Outputs 0 from TO0p 1: Outputs 1 from TO0p

Table 18.67 Example of TOE0 settings for the slave channel when the PWM function is to be used

Bit	Symbol	Set value	Function
p	TOE[p]	1/0	Enabling or disabling timer output for channel p 0: Stops the TO0p output operation by counting operation. 1: Enables the TO0p output operation by counting operation.

Table 18.68 Example of TOL0 settings for the slave channel when the PWM function is to be used

Bit	Symbol	Set value	Function
p	TOL[p − 1]	1/0	Control of timer output of channel p 0: Positive logic output (active-high) 1: Negative logic output (active-low)

Table 18.69 Example of TOM0 settings for the slave channel when the PWM function is to be used

Bit	Symbol	Set value	Function
p	TOM[p − 1]	1	Control of timer output mode of channel p (channels 1 to 7) 1: Sets the slave channel output mode.

Note: n: Master channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

Table 18.70 show procedure for operations when the PWM function is to be used.

Table 18.70 Procedure for operations when the PWM function is to be used

	Step	Software operation		Hardware state
TAU default setting	<1>	Sets timer clock select register 0 (TPS0). Determines clock frequencies of CK00 and CK01.		
Channel default setting	<2>	Sets timer mode registers 0n, 0p (TMR0n, TMR0p) of two channels to be used (determines operation mode of channels). An interval (period) value is set to timer data register 0n (TDR0n) of the master channel, and a duty factor is set to the TDR0p register of the slave channel.		Channel stops operating. (Clock is supplied and some power is consumed.)
	<3>	Sets slave channel. The TOM0.TOM[p − 1] bit of timer output mode register 0 (TOM0) is set to 1 (slave channel output mode). Sets the TOL0.TOL[p − 1] bit.		The TO0p pin goes into Hi-Z output state.
		Sets the TO0.TO[p] bit and determines default level of the TO0p output.	→	The TO0p default setting level is output when the corresponding bit of the port direction register (PDR) is in the output mode.
		Sets the TOE0.TOE[p] bit to 1 and enables operation of TO0p.	→	TO0p does not change because channel stops operating.
		Sets the corresponding bit of the port direction register (PDR) to 1.	→	The TO0p pin outputs the TO0p set level.
Operation start	<4>	Sets the TOE0.TOE[p] bit (slave) to 1 (only when operation is resumed). The TS[n] (master) and TS[p] (slave) bits of timer channel start register 0 (TS0) are set to 1 at the same time. The TS0.TS[n], TS[p] bits automatically return to 0 because they are trigger bits.	→	TE0.TE[n] = 1, TE0.TE[p] = 1 When the master channel starts counting, TAU0_ENDIn is generated. Triggered by this interrupt, the slave channel also starts counting.
During operation	<5>	Set values of the TMR0n and TMR0p registers, TOM0.TOM[n − 1], TOM[p − 1] and TOL0.TOL[n − 1], TOL[p − 1] bits cannot be changed. Set values of the TDR0n and TDR0p registers can be changed after TAU0_ENDIn of the master channel is generated. The TCR0n and TCR0p registers can always be read. The TSR0n and TSR0p registers are not used.		The counter of the master channel loads the TDR0n register value to timer counter register 0n (TCR0n), and counts down. When the count value reaches TCR0n = 0x0000, TAU0_ENDIn output is generated. At the same time, the value of the TDR0n register is loaded to the TCR0n register, and the counter starts counting down again. At the slave channel, the value of the TDR0p register is loaded to the TCR0p register, triggered by TAU0_ENDIn of the master channel, and the counter starts counting down. The output level of TO0p becomes active one count clock after generation of the TAU0_ENDIn output from the master channel. It becomes inactive when TCR0p = 0x0000, and the counting operation is stopped. After that, the above operation is repeated.
Operation stop	<6>	The TT0.TT[n] (master) and TT[p] (slave) bits are set to 1 at the same time. The TT0.TT[n], TT[p] bits automatically return to 0 because they are trigger bits.	→	TE0.TE[n], TE[p] = 0, and count operation stops. The TCR0n and TCR0p registers hold count value and stop. The TO0p output is not initialized and retains its current state.
	<7>	The TOE0.TOE[p] bit of slave channel is cleared to 0 and value is set to the TO0.TO[p] bit. To resume operation, go to step <4>. To terminate the operation, go to step <8>	→	The TO0p pin outputs the TO0p set level.
TAU stop	<8>	To hold the TO0p pin output level Sets the PSEL[4:0] bits to 00000b after the value to be held is set to the corresponding bit of the port output data register (PODR). When holding the TO0p pin output level is not necessary, setting is not required.	→	The TO0p pin output level is held by port function.

Note: n: Channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

18.8.3 Operation for the Multiple PWM Output Function

By extending the PWM function and using multiple slave channels, many PWM waveforms with different duty values can be output.

For example, when using two slave channels, the period and duty factor of an output pulse can be calculated by the following expressions.

Pulse period = {Set value of TDR0n (master) + 1} × Count clock period

Duty factor 1 [%] = {Set value of TDR0p (slave 1)} / {Set value of TDR0n (master) + 1} × 100

Duty factor 2 [%] = {Set value of TDR0q (slave 2)} / {Set value of TDR0n (master) + 1} × 100

Note: Although the duty factor exceeds 100% if the set value of TDR0p (slave 1) > {set value of TDR0n (master) + 1} or if the {set value of TDR0q (slave 2)} > {set value of TDR0n (master) + 1}, it is summarized into 100% output.

Timer counter register 0n (TCR0n) of the master channel operates in the interval timer mode and counts the periods. The TCR0p register of the slave channel 1 operates in one-count mode, counts the duty factor, and outputs a PWM waveform from the TO0p pin. The TCR0p register loads the value of timer data register 0p (TDR0p), using TAU0_ENDIn of the master channel as a start trigger, and starts counting down. When TCR0p = 0x0000, TCR0p outputs TAU0_ENDIp and stops counting until the next start trigger (TAU0_ENDIn of the master channel) has been input. The output level of TO0p becomes active one count clock after generation of TAU0_ENDIn from the master channel, and inactive when TCR0p = 0x0000.

In the same way as the TCR0p register of the slave channel 1, the TCR0q register of the slave channel 2 operates in one-count mode, counts the duty factor, and outputs a PWM waveform from the TO0q pin. The TCR0q register loads the value of the TDR0q register, using TAU0_ENDIn of the master channel as a start trigger, and starts counting down.

When TCR0q = 0x0000, the TCR0q register outputs TAU0_ENDIq and stops counting until the next start trigger (TAU0_ENDIn of the master channel) has been input. The output level of TO0q becomes active one count clock after generation of TAU0_ENDIn from the master channel, and inactive when TCR0q = 0x0000.

When channel 0 is used as the master channel as above, up to seven types of PWM signals can be output at the same time.

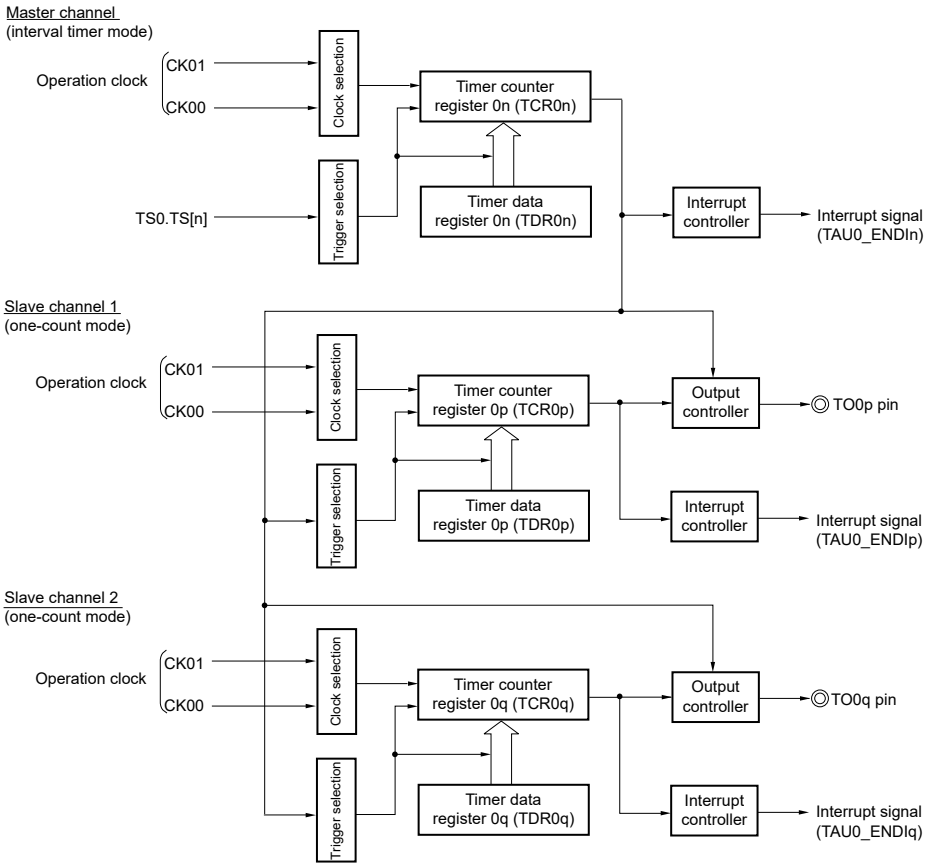
Note: To rewrite both timer data register 0n (TDR0n) of the master channel and the TDR0p register of the slave channel 1, write access is necessary at least twice. Since the values of the TDR0n and TDR0p registers are loaded to the TCR0n and TCR0p registers after TAU0_ENDIn is generated from the master channel, if rewriting is performed separately before and after generation of TAU0_ENDIn from the master channel, the TO0p pin cannot output the expected waveform. To rewrite both the TDR0n register of the master and the TDR0p register of the slave, be sure to rewrite both the registers immediately after TAU0_ENDIn is generated from the master channel (this applies also to the TDR0q register of the slave channel 2).

Note: n: Master channel number (n = 0, 2, 4)

p: Slave channel number, q: Slave channel number

n < p < q ≤ 7 (Where p and q are integers greater than n)

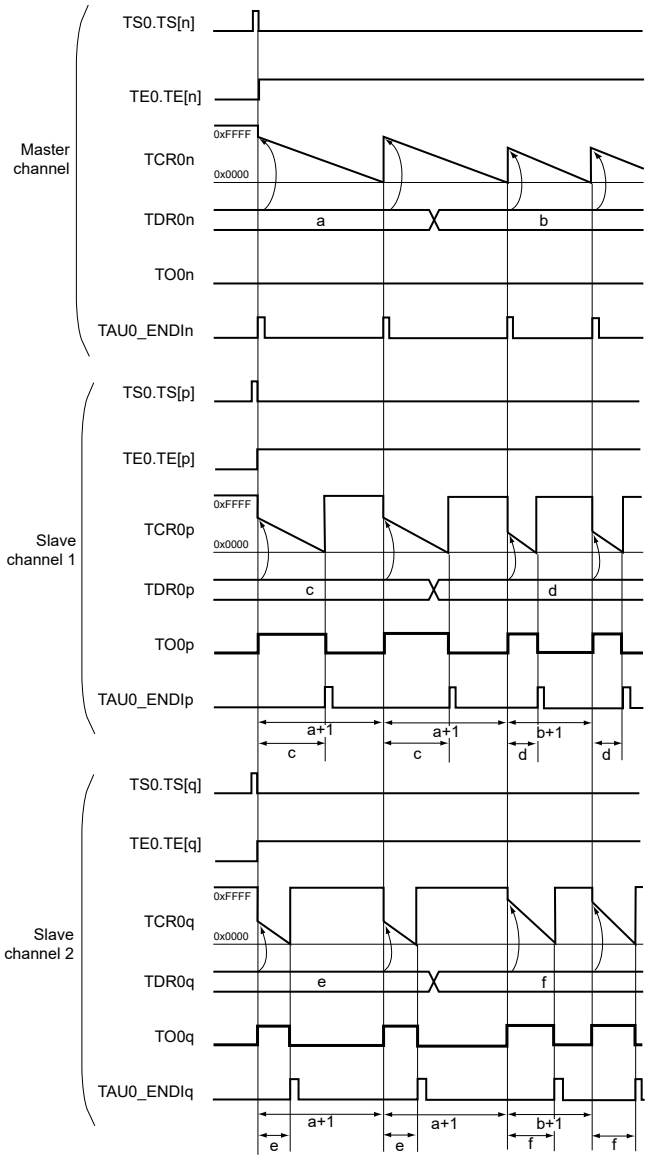
Figure 18.53 shows a block diagram for operation for the multiple PWM output function (for two types of PWM output).



Note: n: Master channel number (n = 0, 2, 4)
p: Slave channel number, q: Slave channel number
n < p < q ≤ 7 (Where p and q are integers greater than n)

Figure 18.53 Block diagram for the multiple PWM output function (for two types of PWM output)

Figure 18.54 shows an example of basic timing during operation for the multiple PWM output function (for two types of PWM output).



Note: n: Master channel number (n = 0, 2, 4)
p: Slave channel number, q: Slave channel number
n < p < q ≤ 7 (Where p and q are integers greater than n)

Figure 18.54 Example of basic timing during operation for the multiple PWM output function (for two types of PWM output)

Table 18.71 to Table 18.75 show register settings for the master channel when the multiple PWM output function is to be used.

Table 18.71 Example of TMR0n settings for the master channel when the multiple PWM output function is to be used (1 of 2)

Bit	Symbol	Set value	Function
0	OPIRQ	1	Setting of operation when counting is started 1: Generates TAU0_ENDIn when counting is started
3:1	MD[2:0]	000b	Operation mode of channel n 0 0 0: Interval timer
5:4	—	00b	Fixed to 0

Table 18.71 Example of TMR0n settings for the master channel when the multiple PWM output function is to be used (2 of 2)

Bit	Symbol	Set value	Function
7:6	CIS[1:0]	00b	Selection of TI0n pin input edge 0 0: Set to 00b because the TI0n input pin is not to be used
10:8	STS[2:0]	000b	Start trigger selection 0 0 0: Selects only software start
11	— (n = 0)	0	Fixed to 0 (channels 0)
	MASTER (n = 2, 4)	1	Setting of MASTER bit (channels 2, 4) 1: Master channel
12	CCS	0	Count clock selection 0: Selects operation clock (f_{MCK})
13	—	0	Fixed to 0.
15:14	CKS[1:0]	00b or 10b	Selection of the operating clock (f_{MCK}) 0 0: Selects CK00 as the operating clock for channel n 1 0: Selects CK01 as the operating clock for channel n

Table 18.72 Example of TO0 settings for the master channel when the multiple PWM output function is to be used

Bit	Symbol	Set value	Function
n	TO[n]	0	Timer output of channel n 0: Outputs 0 from TO0n

Table 18.73 Example of TOE0 settings for the master channel when the multiple PWM output function is to be used

Bit	Symbol	Set value	Function
n	TOE[n]	0	Enabling or disabling timer output for channel n 0: Stops the TO0n output operation by counting operation.

Table 18.74 Example of TOL0 settings for the master channel when the multiple PWM output function is to be used

Bit	Symbol	Set value	Function
n	— (n = 0)	0	Fixed to 0 (channels 0)
	TOL[n − 1] (n = 2, 4)		Control of timer output of channel n (channels 2, 4) 0: Set this bit to 0 when TOM0.TOM[n − 1] = 0 (master channel output mode).

Table 18.75 Example of TOM0 settings for the master channel when the multiple PWM output function is to be used

Bit	Symbol	Set value	Function
n	— (n = 0)	0	Fixed to 0 (channels 0)
	TOM[n − 1] (n = 2, 4)		Control of timer output mode of channel n (channels 2, 4) 0: Sets master channel output mode.

Table 18.76 to Table 18.81 show register settings for the slave channel when the multiple PWM output function is to be used (for two types of PWM output).

Table 18.76 Example of TMR0p settings for the slave channel when the multiple PWM output function is to be used (for two types of PWM output)

Bit	Symbol	Set value	Function
0	OPIRQ	1	Start trigger during operation 1: Trigger input is valid
3:1	MD[2:0]	100b	Operation mode of channel p 1 0 0: One-count mode
5:4	—	00b	Fixed to 0
7:6	CIS[1:0]	00b	Selection of TI0p pin input edge 0 0: Set to 00b because the TI0p input pins are not to be used
10:8	STS[2:0]	100b	Start trigger selection 1 0 0: Selects TAU0_ENDIn of master channel
11	— (p = 5)	0	Fixed to 0 (channel 5)
	SPLIT (p = 1, 3)	0	Setting of SPLIT bit (channels 1, 3) 0: 16-bit timer mode
	MASTER (p = 2, 4, 6)	0	Setting of MASTER bit (channels 2, 4, 6) 0: Slave channel
12	CCS	0	Count clock selection 0: Selects operation clock (f_{MCK})
13	—	0	Fixed to 0.
15:14	CKS[1:0]	00b or 10b	Selection of the operating clock (f_{MCK}) (make the same setting as master channel.) 0 0: Selects CK00 as the operating clock for channel p 1 0: Selects CK01 as the operating clock for channel p

Table 18.77 Example of TMR0q settings for the slave channel when the multiple PWM output function is to be used (for two types of PWM output) (1 of 2)

Bit	Symbol	Set value	Function
0	OPIRQ	1	Start trigger during operation 1: Trigger input is valid
3:1	MD[2:0]	100b	Operation mode of channel q 1 0 0: One-count mode
5:4	—	00b	Fixed to 0
7:6	CIS[1:0]	00b	Selection of TI0p pin input edge 0 0: Set to 00b because the TI0q input pins are not to be used
10:8	STS[2:0]	100b	Start trigger selection 1 0 0: Selects TAU0_ENDIn of master channel
11	— (q = 5, 7)	0	Fixed to 0 (channels 5, 7)
	SPLIT (q = 3)	0	Setting of SPLIT bit (channel 3) 0: 16-bit timer mode
	MASTER (q = 2, 4, 6)	0	Setting of MASTER bit (channels 2, 4, 6) 0: Slave channel
12	CCS	0	Count clock selection 0: Selects operation clock (f_{MCK})
13	—	0	Fixed to 0.

Table 18.77 Example of TMR0q settings for the slave channel when the multiple PWM output function is to be used (for two types of PWM output) (2 of 2)

Bit	Symbol	Set value	Function
15:14	CKS[1:0]	00b or 10b	Selection of the operating clock (f_{MCK}) (make the same setting as master channel.) 0 0: Selects CK00 as the operating clock for channel q 1 0: Selects CK01 as the operating clock for channel q

Table 18.78 Example of TO0 settings for the slave channel when the multiple PWM output function is to be used (for two types of PWM output)

Bit	Symbol	Set value	Function
p	TO[p]	1/0	Timer output of channel p 0: Outputs 0 from TO0p. 1: Outputs 1 from TO0p.
q	TO[q]	1/0	Timer output of channel q 0: Outputs 0 from TO0q. 1: Outputs 1 from TO0q.

Table 18.79 Example of TOE0 settings for the slave channel when the multiple PWM output function is to be used (for two types of PWM output)

Bit	Symbol	Set value	Function
p	TOE[p]	1/0	Enabling or disabling timer output for channel p 0: Stops the TO0p output operation by counting operation. 1: Enables the TO0p output operation by counting operation.
q	TOE[q]	1/0	Enabling or disabling timer output for channel q 0: Stops the TO0q output operation by counting operation. 1: Enables the TO0q output operation by counting operation.

Table 18.80 Example of TOL0 settings for the slave channel when the multiple PWM output function is to be used (for two types of PWM output)

Bit	Symbol	Set value	Function
p	TOL[p − 1]	1/0	Control of timer output of channel p 0: Positive logic output (active-high) 1: Negative logic output (active-low)
q	TOL[q − 1]	1/0	Control of timer output of channel q 0: Positive logic output (active-high) 1: Negative logic output (active-low)

Table 18.81 Example of TOM0 settings for the slave channel when the multiple PWM output function is to be used (for two types of PWM output)

Bit	Symbol	Set value	Function
p	TOM[p − 1]	1	Control of timer output mode of channel p (channels 1 to 6) 1: Sets the slave channel output mode
q	TOM[q − 1]	1	Control of timer output mode of channel q (channels 2 to 7) 1: Sets the slave channel output mode

Table 18.82 show procedure for operations when the multiple PWM output function is to be used.

Table 18.82 Procedure for operations when the multiple PWM output function is to be used (for two types of PWM output) (1 of 2)

	Step	Software operation		Hardware state
TAU default setting	<1>	Sets timer clock select register 0 (TPS0). Determines clock frequencies of CK00 and CK01.		—
Channel default setting	<2>	Sets timer mode registers 0n, 0p, 0q (TMR0n, TMR0p, TMR0q) of each channel to be used (determines operation mode of channels). An interval (period) value is set to timer data register 0n (TDR0n) of the master channel, and a duty factor is set to the TDR0p and TDR0q registers of the slave channels.		Channel stops operating. (Clock is supplied and some power is consumed.)
	<3>	Sets slave channels. The TOM[p − 1], TOM[q − 1] bits of timer output mode register 0 (TOM0) are set to 1 (slave channel output mode).		The TO0p and TO0q pins go into Hi-Z output state.
		Sets the TOL0.TOL[p − 1], TOL[q − 1] bits. Sets the TO0.TO[p], TO[q] bits and determines default level of the TO0p and TO0q outputs.	→	The TO0p and TO0q default setting levels are output when the corresponding bit of the port direction register (PDR) is in the output mode.
		Sets the TOE0.TOE[p], TOE[q] bits to 1 and enables operation of TO0p and TO0q.	→	TO0p and TO0q do not change because channels stop operating.
		Sets the corresponding bit of the port direction register (PDR) to 1.	→	The TO0p and TO0q pins output the TO0p and TO0q set levels.
Operation start	<4>	(Sets the TOE0.TOE[p], TOE[q] (slave) bits to 1 only when resuming operation.) The TS[n] bit (master), and TS[p], TS[q] (slave) bits of timer channel start register 0 (TS0) are set to 1 at the same time. The TS0.TS[n], TS[p], TS[q] bits automatically return to 0 because they are trigger bits.	→	TE0.TE[n], TE[p], TE[q] = 1 When the master channel starts counting, TAU0_ENDIn is generated. Triggered by this interrupt, the slave channel also starts counting.
During operation	<5>	Set values of the TMR0n, TMR0p, TMR0q registers, TOM0.TOM[n − 1], TOM[p − 1], TOM[q − 1] and TOL0.TOL[n − 1], TOL[p − 1], TOL[q − 1] bits cannot be changed. Set values of the TDR0n, TDR0p, and TDR0q registers can be changed after TAU0_ENDIn of the master channel is generated. The TCR0n, TCR0p, and TCR0q registers can always be read. The TSR0n, TSR0p, and TSR0q registers are not used.		The counter of the master channel loads the TDR0n register value to timer counter register 0n (TCR0n) and counts down. When the count value reaches TCR0n = 0x0000, TAU0_ENDIn output is generated. At the same time, the value of the TDR0n register is loaded to the TCR0n register, and the counter starts counting down again. At the slave channel 1, the values of the TDR0p register are transferred to the TCR0p register, triggered by TAU0_ENDIn of the master channel, and the counter starts counting down. The output levels of TO0p become active one count clock after generation of the TAU0_ENDIn output from the master channel. It becomes inactive when TCR0p = 0x0000, and the counting operation is stopped. At the slave channel 2, the values of the TDR0q register are transferred to TCR0q register, triggered by TAU0_ENDIn of the master channel, and the counter starts counting down. The output levels of TO0q become active one count clock after generation of the TAU0_ENDIn output from the master channel. It becomes inactive when TCR0q = 0x0000, and the counting operation is stopped. After that, the above operation is repeated.
Operation stop	<6>	The TT0.TT[n] bit (master), TT[p], and TT[q] (slave) bits are set to 1 at the same time. The TT0.TT[n], TT[p], TT[q] bits automatically return to 0 because they are trigger bits.	→	TE0.TE[n], TE[p], TE[q] = 0, and count operation stops. The TCR0n, TCR0p, and TCR0q registers hold count value and stop. The TO0p and TO0q outputs are not initialized and retain their current states.
	<7>	The TOE0.TOE[p], TOE[q] bits of slave channels are cleared to 0 and value is set to the TO0.TO[p], TO[q] bits. To resume operation, go to step <4>. To terminate the operation, go to step <8>	→	The TO0p and TO0q pins output the TO0p and TO0q set levels.

Table 18.82 Procedure for operations when the multiple PWM output function is to be used (for two types of PWM output) (2 of 2)

	Step	Software operation		Hardware state
TAU stop	<8>	To hold the TO0p and TO0q pin output levels Sets the PSEL[4:0] bits to 00000b after the value to be held is set to the corresponding bit of the port output data register (PODR). When holding the TO0p and TO0q pin output levels are not necessary, setting is not required	→	The TO0p and TO0q pin output levels are held by port function.

Note: n: Master channel number (n = 0, 2, 4)
p: Slave channel number, q: Slave channel number
n < p < q ≤ 7 (Where p and q are consecutive integers greater than n)

18.9 Usage Notes

18.9.1 Cautions when Using Timer Output

Pins may be assigned multiplexed timer output and other alternate functions. The assignment depends on the product. If you intend to use a timer output, set the outputs from all other multiplexed pin functions to their initial values.

For details, see section 16, I/O Ports.

19. 32-bit Interval Timer (TML32)

19.1 Overview

The 32-bit interval timer is made up of four 8-bit interval timers (referred to as channels 0 to 3). Each is capable of operating independently and in that case, they all have the same functions. Two 8-bit interval timer channels can be connected to operate as a 16-bit interval timer. Four 8-bit interval timer channels can be connected to operate as a 32-bit interval timer.

The 32-bit interval timer operates with the TML32MCLK, TML32LCLK/TML32SCLK, TML32HCLK, or TML32MOCLK or the event input from the ELC, which is asynchronous with the CPU operation.

The 32-bit interval timer has the following modes:

- 8-bit counter mode — The 8-bit timers are usable as four 8-bit interval timers that generate interrupts (TML32_OUTI) at fixed intervals. Figure 19.1 shows a four 8-bit interval timer function.
- 16-bit counter mode — The 8-bit timers are usable as two 16-bit interval timers that generate interrupts (TML32_OUTI) at fixed intervals. Figure 19.2 shows a two 16-bit interval timer function.
- 16-bit capture mode — The 8-bit timers are usable as two 16-bit interval timers that generate interrupts (TML32_OUTI) at fixed intervals. The value of one of the 16-bit interval timers can also be stored in the capture register in response to a selected capture trigger. Figure 19.3 shows a 16-bit interval timer and 16-bit capture function.
- 32-bit counter mode — The 8-bit timers are usable as a 32-bit interval timer that generates interrupts (TML32_OUTI) at fixed intervals. Figure 19.4 shows a 32-bit interval timer function.

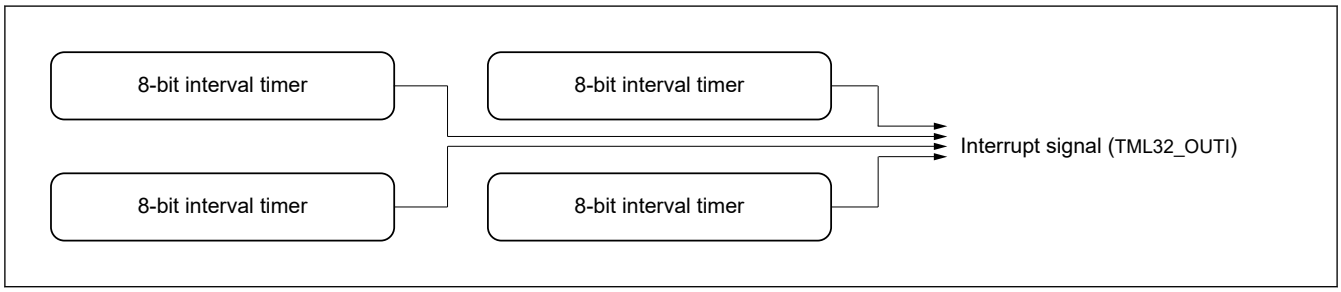


Figure 19.1 Image of four 8-bit interval timer function

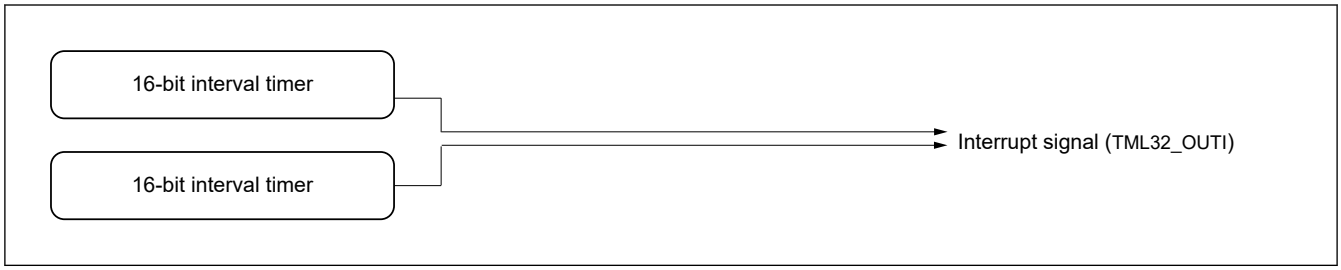


Figure 19.2 Image of two 16-bit interval timer function

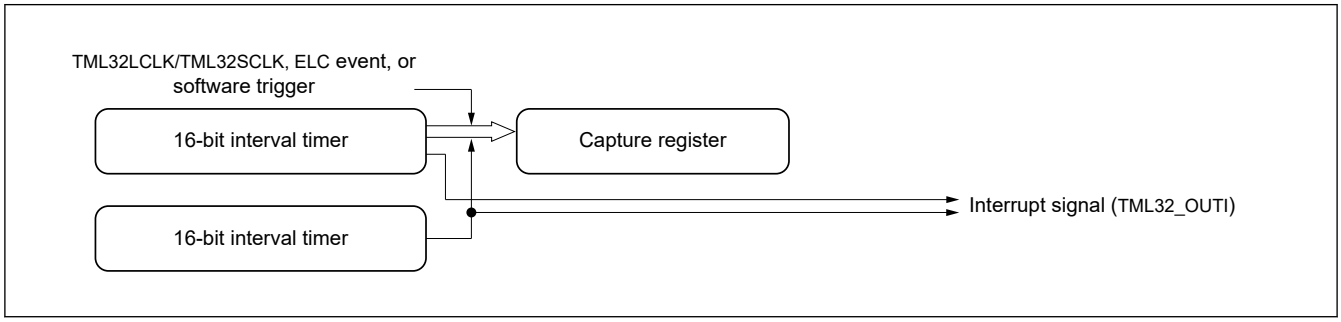


Figure 19.3 Image of the 16-bit interval timer and 16-bit capture function

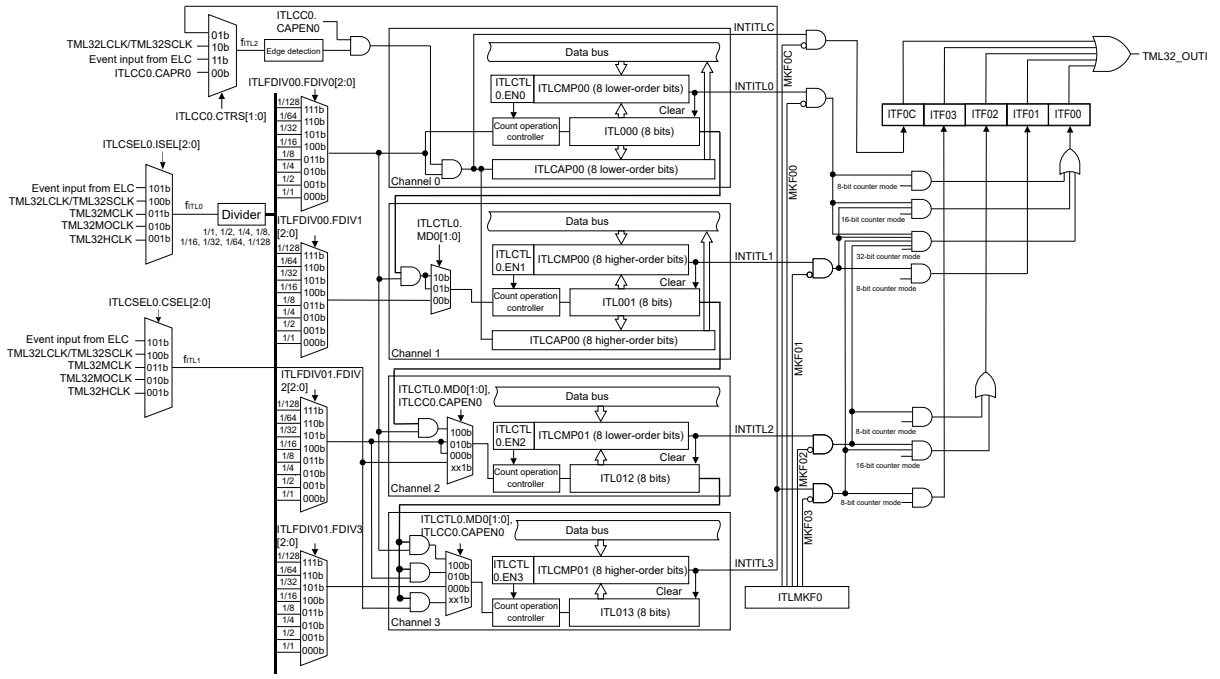
Figure 19.4 Image of a 32-bit interval timer function

Table 19.1 lists the 32-bit interval timer functions and Figure 19.5 shows a block diagram of the 32-bit interval timer.

Table 19.1 Specifications of 32-bit interval timer operations

Item	Description
Count source (operating clock)	• TML32MCLK • TML32LCLK/TML32SCLK • TML32HCLK • TML32MOCLK • Event input from the ELC
Capture clock (selectable sources for counting by the timer which can generate a capture trigger)	• TML32MCLK • TML32LCLK/TML32SCLK • TML32HCLK • TML32MOCLK • Event input from the ELC
Frequency division ratio	• 1/1, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128
Operating mode	• 8-bit counter mode Channels 0 to 3 independently operate as 8-bit counters. • 16-bit counter mode The combinations of channels 0 and 1 and channels 2 and 3 are cascade-connectable to operate as two 16-bit counters. • 32-bit counter mode Channels 0 to 3 are connected to operate as a 32-bit counter. • 16-bit capture mode Channels 0 and 1 are connected to operate as a 16-bit counter using the count source, channels 2 and 3 are connected to operate as a 16-bit counter using the capture clock, and the connected counters are used for capture operation.
Interrupt	• Five interrupt sources are integrated into one interrupt signal and output as the TML32_OUTI signal. – Output when the counter value in any of channels 0 to 3 matches the compare value. – Output when the capturing of the counter value is completed in capture mode.

Note:
- TML32MCLK: TML32 external clock
- TML32SCLK: TML32 sub clock
- TML32HCLK: TML32 HOCO clock
- TML32MOCLK: TML32 MOCO clock
- TML32LCLK: TML32 LOCO clock



Note: ● TML32MCLK: TML32 external clock
● TML32SCLK: TML32 sub clock
● TML32HCLK: TML32 HOCO clock
● TML32MOCLK: TML32 MOCO clock
● TML32LCLK: TML32 LOCO clock

Note: ITL000, ITL001, ITL012, ITL013: 8-bit counters

Note: In 16-bit counter mode, the counters in channels 0 and 1 are connected (ITL000 + ITL001) and the counters in channels 2 and 3 are connected (ITL012 + ITL013).
In 32-bit counter mode, the counters in channels 0 to 3 are connected (ITL000 + ITL001 + ITL012 + ITL013).

Figure 19.5 Block diagram of 32-bit interval timer

19.2 Register Descriptions

19.2.1 ITLCMP0n : Interval Timer Compare Registers 0n (n = 0, 1)

Base address: TML32 = 0x4009_2200

Offset address: 0x00 + 0x2 × n

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CMP16[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	CMP16[15:0]	16-bit timer setting compare data for unit n	R/W*1

Note 1. Write to the ITLCMP00 register while the ITLCTL0.EN0 bit is 0 in 16-bit counter mode or while both the ITLCTL0.EN0 bit and the ITLCTL0.EN1 bit are 0 in 8-bit counter mode.
Write to the ITLCMP01 register while the ITLCTL0.EN2 bit is 0 in 16-bit counter mode or while the ITLCTL0.EN0 bit is 0 in 32-bit counter mode or while both the ITLCTL0.EN2 bit and the ITLCTL0.EN3 bit are 0 in 8-bit counter mode.

These are compare value registers used in 8-bit, 16-bit, or 32-bit counter mode.

These registers can be set by 16-bit access only.

A value from 0x0001 to 0xFFFF can be specified. Setting these registers to 0x0000 is prohibited.

These registers hold values to be compared with the ITL000 + ITL001 + ITL012 + ITL013 counter value (in 32-bit counter mode), the ITL000 + ITL001 or ITL012 + ITL013 counter value (in 16-bit counter mode), and the ITL000 to ITL013 counter values (in 8-bit counter mode).

When the ITLCTL0.MD0[1:0] bits are set to 10b, these registers are used as compare registers in 32-bit counter mode. Specify the upper 16-bit compare value in the ITLCMP01 register and the lower 16-bit compare value in the ITLCMP00 register.

When the ITLCTL0.MD0[1:0] bits are set to 00b, these registers are used as compare registers in 8-bit counter mode. Specify these registers as follow:

- The ITLCMP00.CMP16[7:0] bits for channel 0
- The ITLCMP00.CMP16[15:8] bits for channel 1
- The ITLCMP01.CMP16[7:0] bits for channel 2
- The ITLCMP01.CMP16[15:8] bits for channel 3

When ITLCMP00.CMP16[7:0] or ITLCMP01.CMP16[7:0] is written by 8-bit access, ITLCMP00.CMP16[15:8] or ITLCMP01.CMP16[15:8] is also overwritten.

19.2.2 ITLCAP00 : Interval Timer Capture Register 00

Base address: TML32 = 0x4009_2200

Offset address: 0x04

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	n/a	16-bit capture result data for unit 0	R

This register holds 16-bit captured values when the interval timers are operating in 16-bit capture mode.

The values of the 16-bit counters (ITL000 + ITL001) are stored in the ITLCAP00 register in response to the capture trigger selected in the ITLCC0 register when the CAPEN0 bit in the ITLCC0 register is 1.

When an interrupt on compare match with the ITLCMP01 register is to be used, select the counter clock in the ITLCSEL0 register and set the comparison value in the ITLCMP01 register.

19.2.3 ITLCTL0 : Interval Timer Control Register

Base address: TML32 = 0x4009_2200

Offset address: 0x06

Bit position:	7	6	5	4	3	2	1	0
Bit field:	MD0[1:0]		—	—	EN3	EN2	EN1	EN0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	EN0	8-bit counter mode: ITL000 count enable*1 16-bit counter mode: ITL000 + ITL001 count enable*1 32-bit counter mode: ITL000 + ITL001 + ITL012 + ITL013 count enable*1 0: Counting stops 1: Counting begins	R/W
1	EN1	8-bit counter mode: ITL001 count enable*1 0: Counting stops 1: Counting begins	R/W

Bit	Symbol	Function	R/W
2	EN2	8-bit counter mode: ITL012 count enable[*1] 16-bit counter mode: ITL012 + ITL013 count enable[*1] 0: Counting stops 1: Counting begins	R/W
3	EN3	8-bit counter mode: ITL013 count enable[*1] 0: Counting stops 1: Counting begins	R/W
5:4	—	These bits are read as 0. The write value should be 0.	R/W
7:6	MD0[1:0]	Selection of 8-bit, 16-bit, or 32-bit counter mode[*2] 0 0: The interval timer operates in 8-bit counter mode. 0 1: The interval timer operates in 16-bit counter mode (channel 0 is connected with channel 1 and channel 2 is connected with channel 3). 1 0: The interval timer operates in 32-bit counter mode (channels 0 to 3 are connected). 1 1: Setting prohibited.	R/W

Note 1. When one of the EN3 to EN0 bits is cleared to 0, the corresponding counter is cleared to 0 without synchronization with the counter clock.
Note 2. To change the timer mode, be sure to write to the MD0[1:0] bits only while the EN0, EN1, EN2, and EN3 bits are all 0.

This register is used to start or stop counting by the interval timer and to select 8-bit, 16-bit, or 32-bit counter mode.

EN0 bit (8-bit counter mode: ITL000 count enable, 16-bit counter mode: ITL000 + ITL001 count enable, 32-bit counter mode: ITL000 + ITL001 + ITL012 + ITL013 count enable)

In 8-bit counter mode, writing 1 to this bit starts up-counting in the ITL000 counter and writing 0 stops it.

In 16-bit counter mode, writing 1 to this bit starts up-counting in the ITL000 + ITL001 counter and writing 0 stops it.

In 32-bit counter mode, writing 1 to this bit starts up-counting in the ITL000 + ITL001 + ITL012 + ITL013 counter and writing 0 stops it.

EN1 bit (8-bit counter mode: ITL001 count enable)

In 8-bit counter mode, writing 1 to this bit starts up-counting in the ITL001 counter and writing 0 stops it.

In 16-bit counter mode, set this bit to 0.

In 32-bit counter mode, set this bit to 0.

EN2 bit (8-bit counter mode: ITL012 count enable, 16-bit counter mode: ITL012 + ITL013 count enable)

In 8-bit counter mode, writing 1 to this bit starts up-counting in the ITL012 counter and writing 0 stops it.

In 16-bit counter mode, writing 1 to this bit starts up-counting in the ITL012 + ITL013 counter and writing 0 stops it.

In 32-bit counter mode, set this bit to 0.

EN3 bit (8-bit counter mode: ITL013 count enable)

In 8-bit counter mode, writing 1 to this bit starts up-counting in the ITL013 counter and writing 0 stops it.

In 16-bit counter mode, set this bit to 0.

In 32-bit counter mode, set this bit to 0.

MD0[1:0] bits (Selection of 8-bit, 16-bit, or 32-bit counter mode)

Table 19.2 lists the target counters that can be enabled in the MD0[1:0] bits and EN0 to EN3 bit settings.

Table 19.2 Target counter setting (1 of 2)

Mode	MD0[1:0]	EN3	EN2	EN1	EN0	Target counter
8-bit mode	00b	—	—	—	✓	ITL000
		—	—	✓	—	ITL001
		—	✓	—	—	ITL012
		✓	—	—	—	ITL013

Table 19.2 Target counter setting (2 of 2)

Mode	MD0[1:0]	EN3	EN2	EN1	EN0	Target counter
16-bit mode	01b	Always set to 0.	—	Always set to 0.	✓	ITL000 + ITL001
		Always set to 0.	✓	Always set to 0.	—	ITL012 + ITL013
32-bit mode	10b	Always set to 0.	Always set to 0.	Always set to 0.	✓	ITL000 + ITL001 + ITL012 + ITL013

Note: ✓: Enables counting in the target counter.
Note: In 8-bit counter mode, two or more bits of EN3 to EN0 can be set to 1 or 0 at the same time.
Note: In 16-bit counter mode, the EN2 and EN0 bits can be set to 1 or 0 at the same time.

19.2.4 ITLCSEL0 : Interval Timer Clock Select Register 0

Base address: TML32 = 0x4009_2200

Offset address: 0x07

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

Bit	Symbol	Function	R/W
2:0	ISEL[2:0]	Selection of interval timer count clock (f_{ITL0})*1 0 0 0: Counting stops 0 0 1: TML32HCLK 0 1 0: TML32MOCLK 0 1 1: TML32MCLK 1 0 0: TML32LCLK/TML32SCLK 1 0 1: Event input from the ELC Others: Setting prohibited	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
6:4	CSEL[2:0]	Selection of interval timer count clock for capturing (f_{ITL1})*1 0 0 0: Counting stops 0 0 1: TML32HCLK 0 1 0: TML32MOCLK 0 1 1: TML32MCLK 1 0 0: TML32LCLK/TML32SCLK 1 0 1: Event input from the ELC Others: Setting prohibited	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

Note:
- TML32MCLK: TML32 external clock
- TML32SCLK: TML32 sub clock
- TML32HCLK: TML32 HOCO clock
- TML32MOCLK: TML32 MOCO clock
- TML32LCLK: TML32 LOCO clock

Note 1. Be sure to write to the CSEL[2:0] bits and ISEL[2:0] bits only while the ITLCTL0.EN3 to ITLCTL0.EN0 bits are all 0.

This register is used to select the count source for the interval timer.

19.2.5 ITLFDIV00 : Interval Timer Frequency Division Register 0

Base address: TML32 = 0x4009_2200

Offset address: 0x08

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	FDIV1[2:0]			—	FDIV0[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	FDIV0[2:0]	8-bit counter mode: Counter clock for ITL000*1 16-bit counter mode: Counter clock for ITL000 + ITL001*1 32-bit counter mode: Counter clock for ITL000 + ITL001 + ITL012 + ITL013*1 0 0 0: f_{ITL0} 0 0 1: $f_{ITL0}/2$ 0 1 0: $f_{ITL0}/4$ 0 1 1: $f_{ITL0}/8$ 1 0 0: $f_{ITL0}/16$ 1 0 1: $f_{ITL0}/32$ 1 1 0: $f_{ITL0}/64$ 1 1 1: $f_{ITL0}/128$	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
6:4	FDIV1[2:0]	8-bit counter mode: Counter clock for ITL001*2 0 0 0: f_{ITL0} 0 0 1: $f_{ITL0}/2$ 0 1 0: $f_{ITL0}/4$ 0 1 1: $f_{ITL0}/8$ 1 0 0: $f_{ITL0}/16$ 1 0 1: $f_{ITL0}/32$ 1 1 0: $f_{ITL0}/64$ 1 1 1: $f_{ITL0}/128$	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

Note 1. Be sure to write to the FDIV0[2:0] bits only while the ITLCTL0.EN0 bit is 0.
Note 2. In 8-bit counter mode, be sure to write to the FDIV1[2:0] bits only while the ITLCTL0.EN1 bit is 0.

This register is used to select the counter clock for the interval timer.

FDIV0[2:0] bits (8-bit counter mode: Counter clock for ITL000, 16-bit counter mode: Counter clock for ITL000 + ITL001, 32-bit counter mode: Counter clock for ITL000 + ITL001 + ITL012 + ITL013)

In 8-bit counter mode, ITL000 counts cycles of the counter clock specified in the FDIV0[2:0] bits.

In 16-bit counter mode, ITL000 + ITL001 counts cycles of the counter clock specified in the FDIV0[2:0] bits.
In 32-bit counter mode, ITL000 + ITL001 + ITL012 + ITL013 counts cycles of the counter clock specified in the FDIV0[2:0] bits.

FDIV1[2:0] bits (8-bit counter mode: Counter clock for ITL001)

In 8-bit counter mode, ITL001 counts cycles of the counter clock specified in the FDIV1[2:0] bits.

In 16-bit counter mode, set these bits to 000b.

In 32-bit counter mode, set these bits to 000b.

19.2.6 ITLFDIV01 : Interval Timer Frequency Division Register 1

Base address: TML32 = 0x4009_2200

Offset address: 0x09

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	FDIV3[2:0]			—	FDIV2[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	FDIV2[2:0]	8-bit counter mode: Counter clock for ITL012[*1] 16-bit counter mode: Counter clock for ITL012 + ITL013[*1] 0 0 0: f_{ITL0} 0 0 1: $f_{ITL0}/2$ 0 1 0: $f_{ITL0}/4$ 0 1 1: $f_{ITL0}/8$ 1 0 0: $f_{ITL0}/16$ 1 0 1: $f_{ITL0}/32$ 1 1 0: $f_{ITL0}/64$ 1 1 1: $f_{ITL0}/128$	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
6:4	FDIV3[2:0]	8-bit counter mode: Counter clock for ITL013[*2] 0 0 0: f_{ITL0} 0 0 1: $f_{ITL0}/2$ 0 1 0: $f_{ITL0}/4$ 0 1 1: $f_{ITL0}/8$ 1 0 0: $f_{ITL0}/16$ 1 0 1: $f_{ITL0}/32$ 1 1 0: $f_{ITL0}/64$ 1 1 1: $f_{ITL0}/128$	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

Note 1. In 8-bit or 16-bit counter mode, be sure to write to the FDIV2[2:0] bits only while the ITLCTL0.EN2 bit is 0.
Note 2. In 8-bit counter mode, be sure to write to the FDIV3[2:0] bits only while the ITLCTL0.EN3 bit is 0.

This register is used to select the counter clock for the interval timer.

FDIV2[2:0] bits (8-bit counter mode: Counter clock for ITL012, 16-bit counter mode: Counter clock for ITL012 + ITL013)

In 8-bit counter mode, ITL012 counts cycles of the counter clock specified in the FDIV2[2:0] bits.

In 16-bit counter mode, ITL012 + ITL013 counts cycles of the counter clock specified in the FDIV2[2:0] bits.
In 32-bit counter mode, these bits are not used; write 000b to them.

FDIV3[2:0] bits (8-bit counter mode: Counter clock for ITL013)

In 8-bit counter mode, ITL013 counts cycles of the counter clock specified in the FDIV3[2:0] bits.

In 16-bit counter mode, set these bits to 000b.

In 32-bit counter mode, set these bits to 000b.

19.2.7 ITLCC0 : Interval Timer Capture Control Register 0

Base address: TML32 = 0x4009_2200

Offset address: 0x0A

Bit position:	7	6	5	4	3	2	1	0
Bit field:	CAPEN0	CAPF0CR	CAPF0	CAPR0	CAPC0CR	—	CTRS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	CTRS[1:0]	Selection of capture trigger[*1] [*2] 0 0: Software trigger 0 1: Interrupt on compare match with ITLCMP01[*3] 1 0: TML32LCLK/TML32SCLK (rising edge) 1 1: Event input from ELC (rising edge)	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
3	CAPC0CR	Selection of capture counter clearing after capturing*4 0: The capture counter value is held after the completion of capturing 1: The capture counter value is cleared after the completion of capturing	R/W
4	CAPR0	Software capture trigger*2 *5 0: Trigger operation does not proceed 1: A software trigger for capturing is generated	R/W
5	CAPF0	Capture completion flag 0: Capturing has not been completed. 1: Capturing has been completed. This flag is set to 1 after a capture trigger selected in the CTRS[1:0] bits is generated and the captured data is stored in ITLCAP00. Writing 1 to the CAPF0CR bit clears this flag to 0.	R
6	CAPF0CR	Capture completion flag clear*6 0: The value of the capture completion flag CAPF0 is held 1: The value of the capture completion flag CAPF0 is cleared	R/W
7	CAPEN0	Capture enable*7 0: Capturing is disabled 1: Capturing is enabled	R/W

Note 1. Be sure to write to the CTRS[1:0] bits only while the ITLCTL0.EN3 to ITLCTL0.EN0 bits are all 0s.
Note 2. In the capture operation, the interval at which the capture trigger is generated should be two or more cycles of the counter clock.
Note 3. When the interrupt on compare match with the ITLCMP01 register is selected as a capture trigger, the ITLS0.ITF02 and ITF0C flags are set on capture of the counter value. When only using the capture detection flag, set the ITLMKF0 register to mask the compare match detection flag for channel 2.
Note 4. Be sure to write to the CAPC0CR bit only while the ITLCTL0.EN3 to EN0 bits are all 0s.
Note 5. The CAPR0 bit is always read as 0.
Note 6. The CAPF0CR bit is always read as 0.
Note 7. Be sure to write to the CAPEN0 bit only while the ITLCTL0.EN3 to EN0 bits are all 0s.

This register is used to enable or disable the capture function of the interval timer, specify whether to hold or clear the capture completion flag, set up the software trigger, and select the capture trigger.

19.2.8 ITLS0 : Interval Timer Status Register

Base address: TML32 = 0x4009_2200

Offset address: 0x0B

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	ITF0C	ITF03	ITF02	ITF01	ITF00
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ITF00	Compare match detection flag for channel 0 0: A compare match signal has not been detected in channel 0. 1: A compare match signal has been detected in channel 0.	R/W
1	ITF01	Compare match detection flag for channel 1 0: A compare match signal has not been detected in channel 1. 1: A compare match signal has been detected in channel 1.	R/W
2	ITF02	Compare match detection flag for channel 2 0: A compare match signal has not been detected in channel 2. 1: A compare match signal has been detected in channel 2.	R/W
3	ITF03	Compare match detection flag for channel 3 0: A compare match signal has not been detected in channel 3. 1: A compare match signal has been detected in channel 3.	R/W
4	ITF0C	Capture detection flag 0: Completion of capturing has not been detected. 1: Completion of capturing has been detected.	R/W

Bit	Symbol	Function	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: Writing 1 to each bit is ignored. To clear the ITF0C or ITF0i bit (i = 0, 1, 2, 3), write 0 to the desired bit and 1 to the other bits.
Note: If clearing any of the ITF0C, ITF03, ITF02, ITF01, ITF00 flag bits to 0 does not lead to the value of the ITLS0 register becoming 0x00, an interrupt request (TML32_OUTI) is generated.
Note: To clear a flag bit in the ITLS0 register to 0, only write 0 to a bit that has the setting 1. This is because writing 0 to a bit that has the setting 0 may make detecting a compare match signal or capture detection signal generated at the same time as the writing of 0 impossible. For example, when the ITF01 flag bit is set to 1, write 00011101b to the ITLS0 register to clear the ITF01 flag bit.

This is a status register for the interval timer.

When the value of the ITL0mn counter (mn = 00, 01, 12, 13) matches the value specified in the ITLCMP00 and ITLCMP01 registers, the compare-match flag for the corresponding channel is set.

When a capture trigger is generated while the CAPEN0 bit in the ITLCC0 register is 1, the capture detection flag is set after the value of the ITL0mn counter is stored in the ITLCAP00 register.

The values of the ITF0C and ITF03 to ITF00 bits in this register are ORed and output as the TML32_OUTI interrupt signal. Table 19.3 shows the conditions for setting the status flags in each timer mode selected by the ITLCTL0.MD0[1:0] bits.

Table 19.3 Conditions for setting the status flags in each timer mode

Mode	ITLCTL0.MD0[1:0]	ITLCC0.CAPEN0	Status flag	Conditions for setting status flag
8-bit mode	00b	x	ITF00	The next rising edge of the counter clock following a match between the ITLCMP00 (8 lower-order bits) and ITL000 values
		x	ITF01	The next rising edge of the counter clock following a match between the ITLCMP00 (8 higher-order bits) and ITL001 values
		x	ITF02	The next rising edge of the counter clock following a match between the ITLCMP01 (8 lower-order bits) and ITL012 values
		x	ITF03	The next rising edge of the counter clock following a match between the ITLCMP01 (8 higher-order bits) and ITL013 values
16-bit mode	01b	x	ITF00	The next rising edge of the counter clock following a match between the ITLCMP00 and ITL000 + ITL001 values
		x	ITF02	The next rising edge of the counter clock following a match between the ITLCMP01 and ITL012 + ITL013 values
		1	ITF0C	The ITL000 + ITL001 value is stored in ITLCAP00 after a capture trigger is generated.
32-bit mode	10b	—	ITF00	The next rising edge of the counter clock following a match between the ITLCMP00 + ITLCMP01 and ITL000 + ITL001 + ITL012 + ITL013 values

19.2.9 ITLMKF0 : Interval Timer Match Detection Mask Register

Base address: TML32 = 0x4009_2200

Offset address: 0x0C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	MKF0C	MKF03	MKF02	MKF01	MKF00
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MKF00	Mask for compare match status flag for channel 0*1 0: ITLS0.ITF00 is not masked 1: ITLS0.ITF00 is masked	R/W
1	MKF01	Mask for compare match status flag for channel 1*1 0: ITLS0.ITF01 is not masked 1: ITLS0.ITF01 is masked	R/W
2	MKF02	Mask for compare match status flag for channel 2*1 0: ITLS0.ITF02 is not masked 1: ITLS0.ITF02 is masked	R/W
3	MKF03	Mask for compare match status flag for channel 3*1 0: ITLS0.ITF03 is not masked 1: ITLS0.ITF03 is masked	R/W
4	MKF0C	Mask for capture detection status flag*1 0: ITLS0.ITF0C is not masked 1: ITLS0.ITF0C is masked	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Each bit in this register corresponds to a bit in the ITLS0 register. If the bits in this register are set to 1, the associated bits in the ITLS0 register are masked and the setting is prohibited. This in turn prevents software detection of compare matches and completion of capture. When compare match for any of channels 0 to 3 is to be used, be sure to set the bit corresponding to the given status flag to 0 so that the flag is not masked. For the state of completion of capture, on the other hand, the CAPF0 flag in the interval timer capture control register 0 (ITLCC0) can be used to detect this even when the MKF0C bit is set to 1 to mask the ITLS0.ITF0C flag.

This register is used to enable or disable setting of each valid bit in the interval timer status register (ITLS0) to 1.

Setting an MKF0C or MKF0i (i = 0 to 3) bit to 1 masks the corresponding status flag among ITF0C and ITF0i (i = 0 to 3), after which the given flag is not set to 1 even if a compare match with a compare register or capture completion is detected. Since the status flag is not set to 1, masking also prevents generation of the interval detection interrupt (TML32_OUTI).

19.3 Operation

19.3.1 Counter Mode Settings

The 32-bit interval timer has three counter modes: 8-bit counter mode, 16-bit counter mode, and 32-bit counter mode. Table 19.4 to Table 19.6 show the registers and settings for use in 8-bit counter mode, 16-bit counter mode, and 32-bit counter mode, respectively.

Table 19.4 Registers and settings used in 8-bit counter mode (1 of 2)

Register name (symbol)	Bit	Setting
Interval timer compare registers 00 (ITLCMP00)	CMP16[7:0]	Specify 8-bit compare values for channels 0.
Interval timer compare registers 00 (ITLCMP00)	CMP16[15:8]	Specify 8-bit compare values for channels 1.
Interval timer compare registers 01 (ITLCMP01)	CMP16[7:0]	Specify 8-bit compare values for channels 2.
Interval timer compare registers 01 (ITLCMP01)	CMP16[15:8]	Specify 8-bit compare values for channels 3.
Interval timer control register 0 (ITLCTL0)	EN0	Specify whether to start or stop counting in channel 0.
	EN1	Specify whether to start or stop counting in channel 1.
	EN2	Specify whether to start or stop counting in channel 2.
	EN3	Specify whether to start or stop counting in channel 3.
	MD0[1:0]	Set to 00b.
Interval timer frequency division registers 0 (ITLFDIV00)	FDIV0[2:0]	Select the count clock for channel 0.
	FDIV1[2:0]	Select the count clock for channel 1.
Interval timer frequency division registers 1 (ITLFDIV01)	FDIV2[2:0]	Select the count clock for channel 2.
	FDIV3[2:0]	Select the count clock for channel 3.

Table 19.4 Registers and settings used in 8-bit counter mode (2 of 2)

Register name (symbol)	Bit	Setting
Interval timer clock select register 0 (ITLCSEL0)	ISEL[2:0]	Select the count clock for the interval timer.
	CSEL[2:0]	Set to 000b.
Interval timer capture control register 0 (ITLCC0)	Bits 7 to 0	Set to 0.

Table 19.5 Registers and settings used in 16-bit counter mode

Register name (symbol)	Bit	Setting
Interval timer compare registers 00 (ITLCMP00)	CMP16[15:0]	Specify 16-bit compare values for channels 0 and 1
Interval timer compare registers 01 (ITLCMP01)	CMP16[15:0]	Specify 16-bit compare values for channels 2 and 3.
Interval timer control register 0 (ITLCTL0)	EN0	Specify whether to start or stop counting in channels 0 and 1.
	EN1	Set to 0.
	EN2	Specify whether to start or stop counting in channels 2 and 3.
	EN3	Set to 0.
	MD0[1:0]	Set to 01b.
Interval timer frequency division registers 0 (ITLFDIV00)	FDIV0[2:0]	Select the count clock for channels 0 and 1.
	FDIV1[2:0]	Set to 000b.
Interval timer frequency division registers 1 (ITLFDIV01)	FDIV2[2:0]	Select the count clock for channels 2 and 3.
	FDIV3[2:0]	Set to 000b.
Interval timer clock select register 0 (ITLCSEL0)	ISEL[2:0]	Select the count clock for the interval timer.
	CSEL[2:0]	Set to 000b.
Interval timer capture control register 0 (ITLCC0)	Bits 7 to 0	Set to 0.

Table 19.6 Registers and settings used in 32-bit counter mode

Register name (symbol)	Bit	Setting
Interval timer compare registers 00 (ITLCMP00)	CMP16[15:0]	Specify a compare value in 32-bit counter mode. Specify the lower 16 bits of the compare value in channels 0 and 1 (ITLCMP00).
Interval timer compare registers 01 (ITLCMP01)	CMP16[15:0]	Specify a compare value in 32-bit counter mode. Specify the upper 16 bits of the compare value in channels 2 and 3 (ITLCMP01).
Interval timer control register 0 (ITLCTL0)	EN0	Specify whether to start or stop counting in channels 0 to 3.
	EN1	Set to 0.
	EN2	Set to 0.
	EN3	Set to 0.
	MD0[1:0]	Set to 10b.
Interval timer frequency division registers 0 (ITLFDIV00)	FDIV0[2:0]	Select the count clock for channels 0 to 3.
	FDIV1[2:0]	Set to 000b.
Interval timer frequency division registers 1 (ITLFDIV01)	FDIV2[2:0]	Set to 000b.
	FDIV3[2:0]	Set to 000b.
Interval timer clock select register 0 (ITLCSEL0)	ISEL[2:0]	Select the count clock for the interval timer.
	CSEL[2:0]	Set to 000b.
Interval timer capture control register 0 (ITLCC0)	Bits 7 to 0	Set to 0.

19.3.2 Capture Mode Settings

When the 16-bit capture mode is to be used for channels 0 and 1, the counter value is stored in interval timer capture register 00 (ITLCAP00) in response to a selected capture trigger.

Table 19.7 shows the registers and settings for use in 16-bit capture mode.

Table 19.7 Registers and settings used in 16-bit capture mode

Register name (symbol)	Bit	Setting
Interval timer compare register 00 (ITLCMP00)	CMP16[15:0]	Specify 16-bit compare values for channels 0 and 1.
Interval timer compare register 01 (ITLCMP01)*1	CMP16[15:0]	Specify 16-bit compare values for channels 2 and 3.
Interval timer control register 0 (ITLCTL0)	EN0	Specify whether to start or stop counting in channels 0 and 1.
	EN1	Set to 0.
	EN2	Specify whether to start or stop counting in channels 2 and 3.
	EN3	Set to 0.
	MD0[1:0]	Set to 01b.
Interval timer frequency division registers 0 (ITLFDIV00)	FDIV0[2:0]	Select the count clock for channel 0.
	FDIV1[2:0]	Set to 000b.
Interval timer frequency division registers 1 (ITLFDIV01)	FDIV2[2:0]	Set to 000b.
	FDIV3[2:0]	Set to 000b.
Interval timer clock select register 0 (ITLCSEL0)	ISEL[2:0]	Select the count clock for the interval timer in channels 0 and 1.
	CSEL[2:0]	Select the count clock for the interval timer for capturing in channels 2 and 3.
Interval timer capture control register 0 (ITLCC0)	CAPEN0	Set to 1.
	CAPC0CR	Specify whether to clear or hold the counter value in channels 0 and 1 after the completion of capturing.
	CTRS0[1:0]	Select a capture trigger.

Note 1. Channels 2 and 3 can only be used in 16-bit counter mode when an interrupt on compare match with ITLCMP01 is not to be used as a capture trigger.

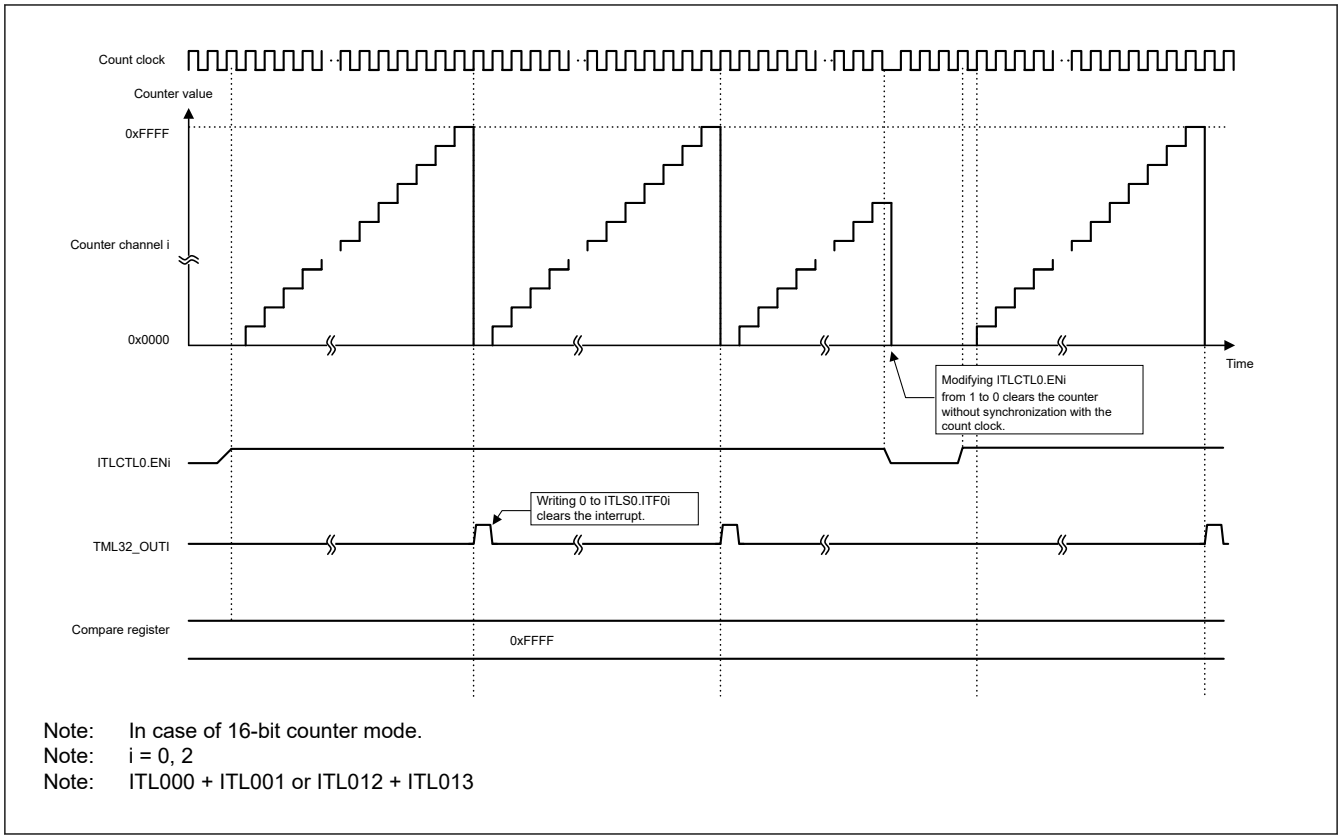
19.3.3 Timer Operation

The ITL0mn counter counts up cycles of the counting clock specified in the interval timer frequency division registers (ITLFDIV00 and ITLFDIV01). An interrupt request signal (TML32_OUTI) is generated on the counting of the next clock cycle after the value of the counter matches the comparison value. The interrupt request signal (TML32_OUTI) remains high until the value of the ITLS0 register becomes 0x00.

While the interrupt request signal (TML32_OUTI) is high, the generation of an additional interrupt request (TML32_OUTI) does not proceed even if a compare match or capture completion is detected for an operating channel.

Clearing the ITLCTL0.EN0 to EN3 bits to 0 clears the counter value.

Figure 19.6 shows an example of timer operation.



Note: In case of 16-bit counter mode.
Note: i = 0, 2
Note: ITL000 + ITL001 or ITL012 + ITL013

Figure 19.6 Example of timer operation

19.3.4 Capture Operation

When the setting of the CAPEN0 bit in the interval timer capture control register 0 (ITLCC0) is 1, the values in the 16-bit counters (ITL000 + ITL001) are stored in interval timer capture register 00 (ITLCAP00) in response to the capture trigger specified in the ITLCC0 register.

The capture trigger is selectable from among the interrupt on compare match with the ITLCMP01 register, TML32LCLK/TML32SCLK, an event input from the ELC, and a software trigger (setting the ITLCC0.CAPR0 bit to 1). To use the interrupt on compare match with the ITLCMP01 register as the capture trigger, set interval timer clock select register 0 (ITLCSEL0) to select the clock for counting, and set interval timer compare register 01 (ITLCMP01) to specify the comparison value. When using TML32LCLK/TML32SCLK, an event input from the ELC, or a software trigger (setting the ITLCC0.CAPR0 bit to 1) as a capture trigger, channels 2 and 3 can be used in 16-bit counter mode.

After a capture trigger is input and the counter value is stored in the interval timer capture register, the interrupt request signal (TML32_OUTI) is output, the capture completion flag (ITLCC0.CAPF0) and capture detection flag (ITLS0.ITF0C) are set to 1, and the flag values are retained until they are explicitly cleared *1. The ITLCC0.CAPF0 flag can be cleared by setting the ITLCC0.CAPF0CR bit to 1. The ITF0C flag in the ITLS0 register can be cleared by writing 0 to it. Since capture operations operate with the counter clock, the interval at which the capture trigger is generated should be at least 5 cycles of the counter clock. If a capture trigger is generated again within 2 cycles of the counter clock after an earlier capture trigger was generated, the ITLCC0.CAPF0 bit may not be set.

Note 1. If the value of the ITLS0 register is other than 0x00, interrupt operation does not proceed even when the capture detection flag (ITLS0.ITF0C) is set to 1 because the interrupt request signal (TML32_OUTI) is kept at the high level.

Figure 19.7 shows an example of capture operation.

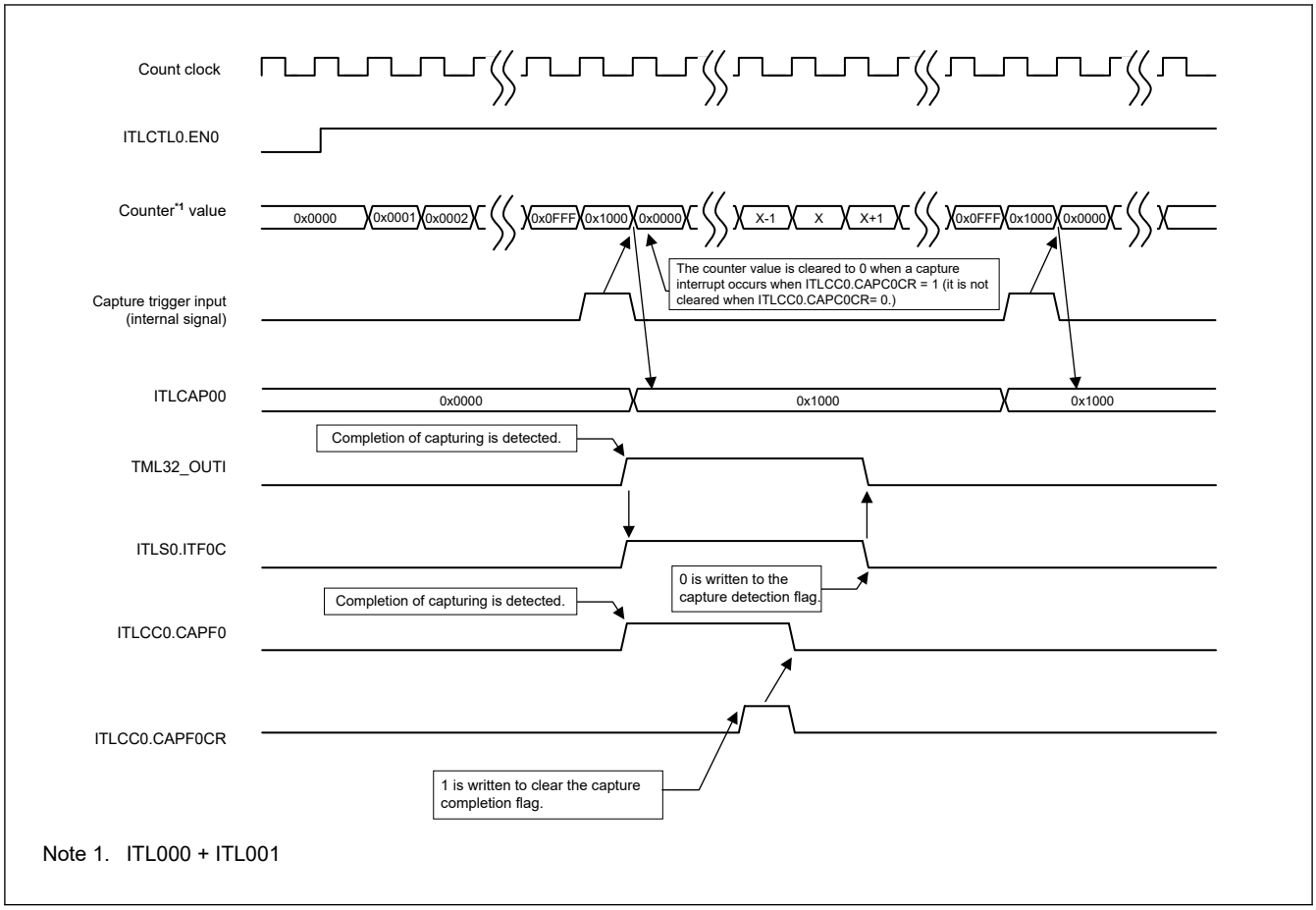


Figure 19.7 Example of capture operation

When the counter value matches the comparison value while the CAPC0CR bit in the ITLCC0 register is set to 1 (mode where the capture counter value is cleared following the completion of capture), counting of the next clock cycle clears the counter value. The counter value is not cleared in this way if the ITLCC0.CAPC0CR bit is set to 0 (mode where the capture counter retains its value after the completion of capture).

19.3.5 Interrupt

Table 19.8 shows the interrupt sources in 8-bit, 16-bit, and 32-bit counter modes.

The ITF00 to ITF03 and ITF0C bits are interrupt status flags in the ITLS0 register. When any of the interrupt status flag is set, an interrupt request is output as the TML32_OUTI signal.

Table 19.8 Interrupt sources in 8-bit, 16-bit, and 32-bit counter modes

Interrupt source	Interrupt condition in 8-bit counter mode	Interrupt condition in 16-bit counter mode	Interrupt condition in 32-bit counter mode
ITLS0.ITF00	Next rising edge of the counter clock after a compare match in channel 0	Next rising edge of the counter clock after a compare match in channels 0 and 1	Next rising edge of the counter clock after a compare match
ITLS0.ITF01	Next rising edge of the counter clock after a compare match in channel 1	Not generated	Not generated
ITLS0.ITF02	Next rising edge of the counter clock after a compare match in channel 2	Next rising edge of the counter clock after a compare match in channels 2 and 3	Not generated
ITLS0.ITF03	Next rising edge of the counter clock after a compare match in channel 3	Not generated	Not generated
ITLS0.ITF0C	Not generated; this is the case when the setting of the ITLCC0 register is 0x00.	Timing of storing the counter value in the capture register after a capture trigger is input.	Not generated; this is the case when the setting of the ITLCC0 register is 0x00.

If the value of the ITLS0 register is other than 0x00, the interrupt request signal (TML32_OUTI) is kept at the high level. Accordingly, the generation of an additional interrupt request (TML32_OUTI) does not proceed, even when a compare match or completion of capture is detected for an operating channel.

However, if the value of the ITLS0 register is not 0x00 after any bit in the ITLS0 register is set to 0, a low-level pulse signal is output on the TML32_OUTI pin. Accordingly, clearing a status flag in the ITLS0 register to 0 during interrupt processing or other processing enables the detection of an interrupt in response to another status bit having the setting 1. Figure 19.8 shows the relationship between clearing of the detection flags and the interval detection interrupt signal.

The following describes the operation shown in Figure 19.8.

When a compare match in channel 1 is detected while the value of the ITLS0 register is 0x00, the ITF01 flag is set to 1 and the interval detection interrupt signal (TML32_OUTI) is driven high. While the interval detection interrupt signal (TML32_OUTI) is kept at the high level, the generation of an additional interrupt request (TML32_OUTI) does not proceed even when a compare match or completion of capture is detected for an operating channel.

Note that if another detection flag is set to 1 immediately before clearing the ITLS0.ITF0x (x = 0, 1, 2, 3, C) flag bit to 0, the output of the TML32_OUTI temporarily goes to the low level after clearing of the given ITLS0.ITF0x flag bit.

<1> The ITLS0.ITF01 flag is set to 1 in response to a compare match in channel 1 and the interval detection interrupt signal (TML32_OUTI) are driven high. The interval detection interrupt processing is executed.

<2> Check which detection flag in the ITLS0 register is set to 1 from within the interval detection interrupt processing. In the case shown in Figure 19.8, the ITLS0.ITF01 and ITF00 flags being set to 1 can be confirmed.

<3> To clear the ITLS0.ITF01 and ITF00 flags detected in step 2, write 00011100b to the ITLS0 register so that its value becomes 0x00.*1

<4> The respective processing sequences in response to the ITLS0.ITF01 and ITF00 flags being set to 1 are then executed.*1

Note 1. Missing an interrupt source can also be prevented by repeating the processing for clearing an interrupt source per flag.

<5> The ITLS0.ITF01 flag is set to 1 in response to a further compare match in channel 1 and the interval detection interrupt signal (TML32_OUTI) is driven high. The interval detection interrupt processing is executed.

<6> Check which detection flag in the ITLS0 register is set to 1 from within the interval detection interrupt processing. In the case shown in Figure 19.8, the ITLS0.ITF01 flag being set to 1 can be confirmed.

<7> To clear the ITLS0.ITF01 flag detected in step 6, write 00011101b to the ITLS0 register so that its value becomes 0x00. Though the ITLS0.ITF00 flag is also set to 1 in response to the compare match in channel 0 at this time, the ITLS0.ITF00 flag is not cleared because the processing for the flag does not proceed.

<8> As the ITLS0.ITF00 flag is set to 1 at the time the ITLS0.ITF01 flag is cleared to 0 in step 7, the TML32_OUTI signal is temporarily driven low.

<9> The processing in response to the ITLS0.ITF01 flag being set to 1 is then executed.

<10> Check which detection flag in the ITLS0 register is set to 1 from within the interval detection interrupt processing. In the case shown in Figure 19.8, the ITLS0.ITF00 flag being set to 1 can be confirmed.

<11> To clear the ITLS0.ITF00 flag detected in step 10, write 00011110b to the ITLS0 register so that its value becomes 0x00.

<12> The processing in response to the ITLS0.ITF00 flag being set to 1 is then executed.

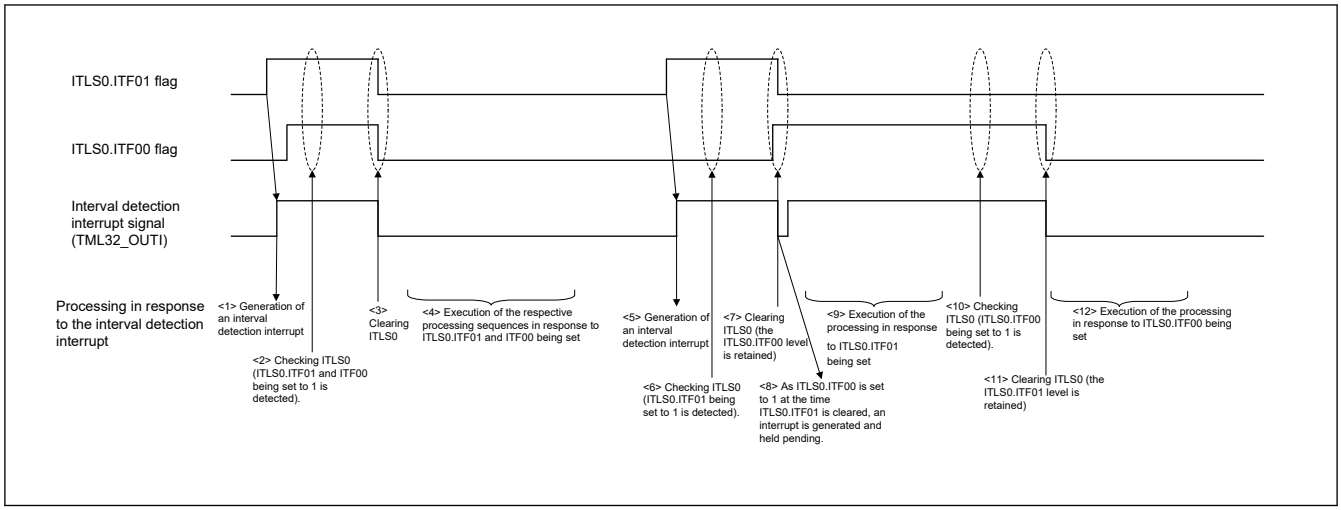


Figure 19.8 Example of clearing the detected flags

19.3.6 Interval Timer Setting Procedures

Table 19.9 shows the procedure for setting up the 32-bit interval timer.

Table 19.9 Procedure for starting the 32-bit interval timer

Step		Process	Detail
Starting the 32-bit interval timer	<1>	Start operation.	—
	<2>	Select 8-bit, 16-bit, or 32-bit counter mode.	Set up the ITLCTL0.MD0[1:0] bits.
	<3>	Select the count clock for the interval timer. Select the frequency division ratio for the count source. Specify a compare value.	Set up the ITLCSEL0.ISEL[2:0] bits. Set up the ITLFDIV0n register. Set up the ITLCMP0n register.
	<4>	When using the capture function ● Enable capturing. ● Clear the capture completion flag. ● Select the count clock for the capture timer. ● Specify the clearing of the counter values in channels 0 and 1 after completion of capturing. ● Select the capture trigger.	Set the ITLCC0.CAPEN0 bit. Clear the ITLCC0.CAPF0CR bit. Set up the ITLCSEL0.CSEL[2:0] bits. Clear the ITLCC0.CAPC0CR bit. Set up the ITLCC0.CTRS[1:0] bits.
	<5>	When using an interrupt*1 ● Clear the ITLS0.ITF0i interrupt status flags. ● Set up masks for the ITLS0.ITF0i status flags.	Clear the ITLCC0.CAPF0CR bit. Set up the ITLCSEL0.CSEL[2:0] bits.
	<6>	Start the 32-bit interval timer.	Set the ITLCTL0.ENi bit.
	<7>	Set the ITLCC0.CAPR0 bit to 1 when using the software capture trigger.	Set the ITLCC0.CAPR0 bit.
	<8>	Wait for an interrupt.	—

Note: n = 0, 1, i = 0 to 3

Note 1. When using this timer as an interval timer, do not mask the interrupts. When selecting a compare match in channels 2 and 3 as the capture trigger in 16-bit counter mode, set the ITLMKF0.MKF02 bit to 1 to specify a mask.

Table 19.10 shows the procedure for stopping the 32-bit interval timer.

Table 19.10 Procedure for stopping the 32-bit interval timer

Step		Process	Detail
Stopping the 32-bit interval timer	<1>	Starting to stop the counter.	—
	<2>	Set up masks for the ITLS0.ITF0i status flags. Clear the ITLS0.ITF0i interrupt status flags.	Set the ITLMKF0.MKF0i bits. Clear the ITLS0.ITF0i bits.
	<3>	When the capture function is in use ● Set up a mask for the ITLS0.ITF0C status flag ● Clear the ITLS0.ITF0C interrupt status flag.	Set the ITLMKF0.MKF0C bit. Clear the ITLS0.ITF0C bit.
	<4>	Stop the 32-bit interval timer. Counting stops after one cycle of the source clock.	Clear the ITLCTL0.ENi bit.
	<5>	Completion of stopping the counter.	—

Note: i = 0 to 3

Table 19.11 shows the procedure for changing the operating mode of the 32-bit interval timer.

Table 19.11 Procedure for changing the operating mode of the 32-bit interval timer

Step		Process	Detail
Changing the operating mode of the 32-bit interval timer	<1>	Starting to change the operating mode.	—
	<2>	Set up masks for the ITLS0.ITF0i status flags. Clear the ITLS0.ITF0i interrupt status flags.	Set the ITLMKF0.MKF0i bits. Clear the ITLS0.ITF0i bits.
	<3>	When the capture function is in use ● Set up a mask for the ITLS0.ITF0C status flag ● Clear the ITF0C interrupt status flag.	Set the ITLMKF0.MKF0C bit. Clear the ITLS0.ITF0C bit.
	<4>	Disable all counters in the 32-bit interval timer.	Clear the ITLCTL0.EN0 to EN3 bits.
	<5>	Wait for at least one cycle of the count source until the timer is stopped.	Wait for stopping.
	<6>	Change the operating mode of the 32-bit interval timer. (see Table 19.9)	Make the setting to change the operating mode.
	<7>	Completion of changing the operating mode.	—

Note: i = 0 to 3

Table 19.12 shows the procedure for starting event input from the ELC.

Table 19.12 Procedure for starting event input from the ELC

Step		Process	Detail
Starting event Input from the ELC	<1>	Start of the procedure for starting event input from the ELC.	—
	<2>	Select the 32-bit interval timer as the destination of output.	Use the ELSR19[*1] register. Set the appropriate ELSR19.ELS[7:0] bits for the 32-bit interval timer to be linked.
	<3>	Set up the ELCR[*1] register to enable the output.	Set the ELCR.ELCON bit to 1 to enable linkage of all events.
	<4>	Specify the operating mode of the event generation source.	See Table 19.9. Use the CSEL[2:0] or ISEL[2:0] bits in the ITLCSEL0 register or the CTRS[1:0] bits in the ITLCC0 register to select the event input from the ELC for the count source or capture trigger as desired.
	<5>	Specify the operating mode of the 32-bit interval timer.	Wait for stopping.
	<6>	Start the operation of the event generation source.	—
	<7>	Completion of the procedure for starting event input from the ELC.	—

Note 1. For details, see section 15, Event Link Controller (ELC).

Table 19.13 shows the procedure for stopping event input from the ELC.

Table 19.13 Procedure for stopping event input from the ELC

Step		Process	Detail
Stopping event Input from the ELC	<1>	Start of the procedure for stopping event input from the ELC.	—
	<2>	Stop the operation of the event generation source.	—
	<3>	Stop the 32-bit interval timer.	See Table 19.10.
	<4>	Set up the ELSR19 register[*1] to disable the output. (Optional: set up the ELCR[*1] register to disable all event linkage).	Set the ELSR19.ELS[7:0] bits to 0.
	<5>	Completion of the procedure for stopping event input from the ELC.	—

Note 1. For details, see section 15, Event Link Controller (ELC).

19.3.7 Snooze Mode Function

When using TML32 to trigger ADC in Snooze mode by way of ELC, DTC must be set in repeat transfer mode to clear TML32 level interrupt.

Figure 19.9 shows an example setting for using TML32 to trigger ADC in Snooze mode with TML32 interrupt cleared by DTC.

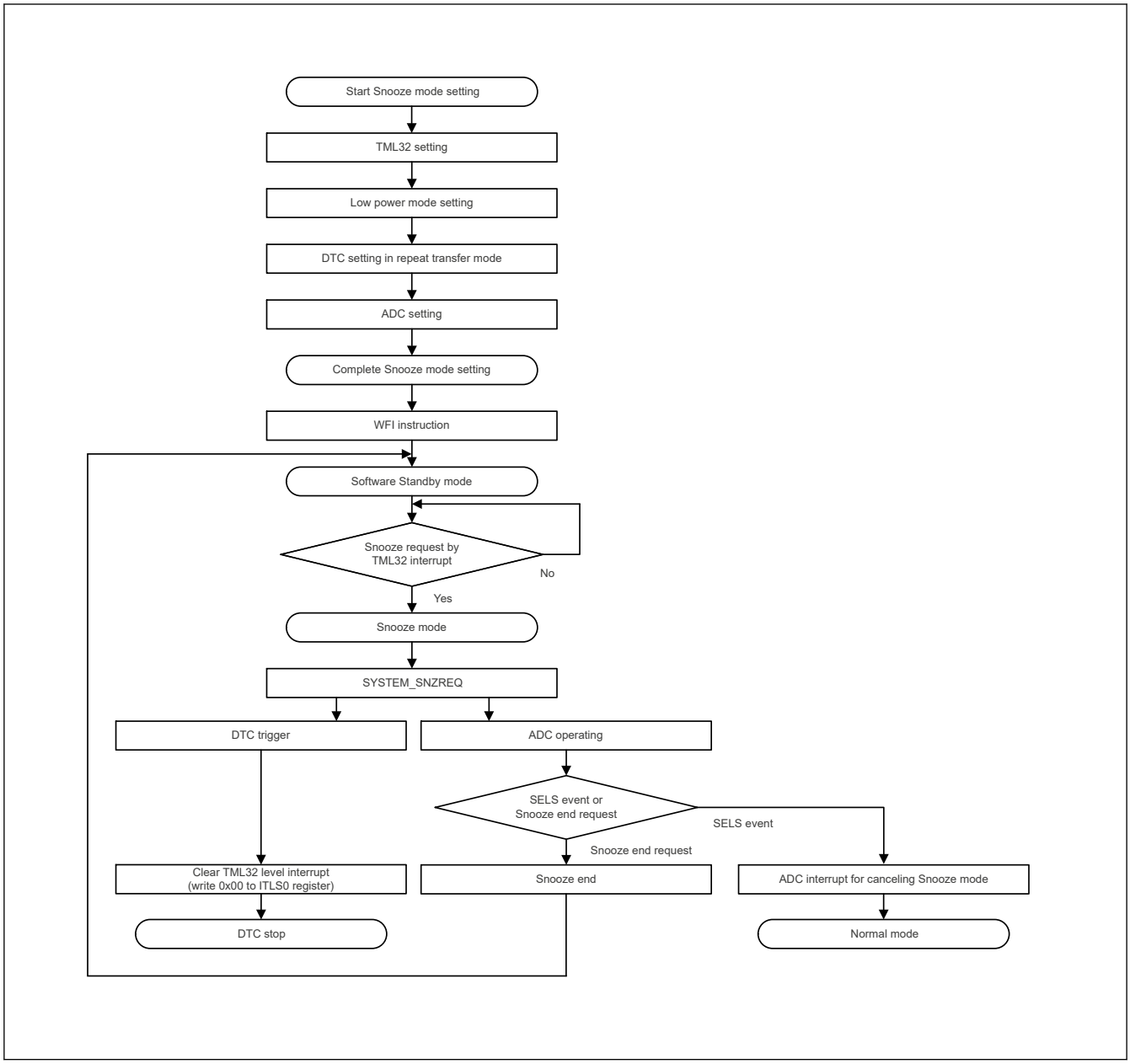


Figure 19.9 Example setting for using TML32 to trigger ADC in Snooze mode with TML32 interrupt cleared by DTC

Figure 19.10 shows an example setting for DTC in repeat transfer mode.

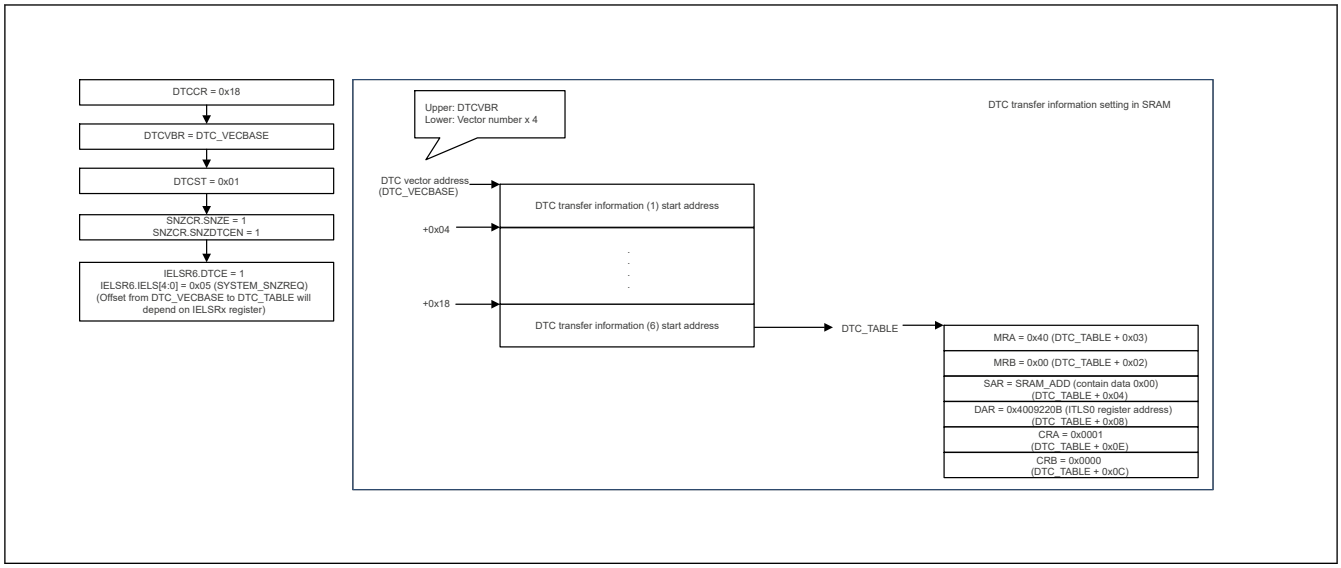


Figure 19.10 Example setting for DTC in repeat transfer mode

20. Realtime Clock (RTC)

20.1 Overview

Table 20.1 shows the specifications for RTC.

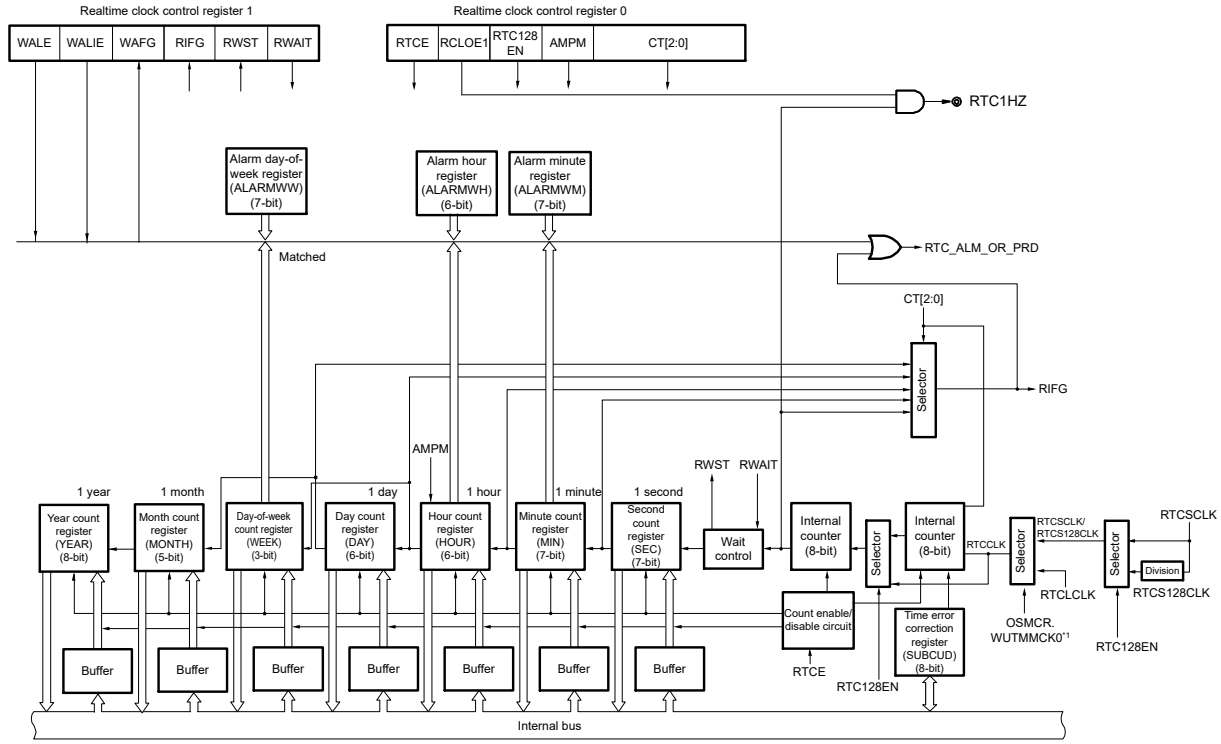
Table 20.1 RTC specifications

Item	Description
Count mode	Calendar count mode
Count source	RTCSCLK or RTCS128CLK/RTCLCLK selectable
Calendar functions	Year, month, date, day of week, hour, minute, and second are counted for up to 99 years
Interrupts (RTC_ALM_OR_PRD)	• Fixed-cycle interrupt – Period selectable from among 0.5 of a second, 1 second, 1 minute, 1 hour, 1 day, or 1 month • Alarm interrupt – Alarm set by day of week, hour, and minute
Pin output function	1 Hz clock output

Note:
- RTCSCLK: RTC sub clock
- RTCS128CLK: RTC sub 128 Hz clock
- RTCLCLK: RTC LOCO clock

The realtime clock interrupt signal (RTC_ALM_OR_PRD) can be used to wake up the MCU from the Software Standby mode or to trigger transitions to the Snooze mode. Additionally, it can wake up the MCU from the Snooze mode.

Figure 20.1 shows a block diagram of the realtime clock.



Note: • RTCSCLK: RTC sub clock
• RTCS128CLK: RTC sub 128 Hz clock
• RTCLCLK: RTC LOCO clock

Note: The count of years, months, weeks, days, hours, minutes, and seconds can only proceed when the RTC sub clock (RTCSCLK = 32.768 kHz) is selected as the operating clock of the realtime clock (RTCCLK).
When the RTC LOCO clock (RTCLCLK = 32.768 kHz) is selected, only the fixed-cycle interrupt is available.

Note 1. For details, see section 8, Clock Generation Circuit.

Figure 20.1 Block diagram of the realtime clock

20.2 Register Descriptions

Table 20.2 shows the register states depending on reset sources.

Table 20.2 Register states depending on reset sources

Reset source	System-related registers*1	Calendar-related registers*2
POR	Reset	Not reset
External reset	Retained	Retained
WDT	Retained	Retained
IWDT	Retained	Retained
LVD	Retained	Retained
Other internal reset sources	Retained	Retained

Note 1. RTCC0, RTCC1, and SUBCUD
Note 2. SEC, MIN, HOUR, DAY, WEEK, MONTH, YEAR, ALARMWM, ALARMWH, and ALARMWW

Assertion of the reset signal does not reset the SEC, MIN, HOUR, DAY, WEEK, MONTH, YEAR, ALARMWM, ALARMWH, or ALARMWW register. Initialize all the registers after power on.

20.2.1 RTCC0 : Realtime Clock Control Register 0

Base address: RTC = 0x4009_2000

Offset address: 0x0B

Bit position:	7	6	5	4	3	2	1	0
Bit field:	RTCE	—	RCLOE1	RTC128EN	AMPM	CT[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	CT[2:0]	Fixed-cycle interrupt (RTC_ALM_OR_PRD) selection 0 0 0: Does not use fixed-cycle interrupt 0 0 1: Once per 0.5 s (synchronized with second count up) 0 1 0: Once per 1 s (same time as second count up) 0 1 1: Once per 1 m (second 00 of every minute) 1 0 0: Once per 1 hour (minute 00 and second 00 of every hour) 1 0 1: Once per 1 day (hour 00, minute 00, and second 00 of every day) Others: Once per 1 month (Day 1, hour 00 a.m., minute 00, and second 00 of every month)	R/W
3	AMPM	Selection of 12- or 24-hour system 0: 12-hour system (a.m. and p.m. are displayed.) 1: 24-hour system	R/W
4	RTC128EN	Selection of the operating clock for the realtime clock (RTCSCLK/RTCS128CLK) 0: RTCSCLK (32.768 kHz) 1: RTCS128CLK (128 Hz)	R/W
5	RCLOE1	RTC1HZ pin output control 0: Disables output of the RTC1HZ pin (1 Hz) 1: Enables output of the RTC1HZ pin (1 Hz)	R/W
6	—	This bit is read as 0. The write value should be 0.*1	R/W
7	RTCE*2	Realtime clock operation control 0: Stops counter operation 1: Starts counter operation	R/W

Note: Do not change the value of the RCLOE1 bit when RTCE is 1.
Note: 1 Hz is not output even if RCLOE1 is set to 1 when RTCE is 0.
Note:
- RTCSCLK: RTC sub clock
- RTCS128CLK: RTC sub 128 Hz clock

Note 1. Be sure to clear bit [6] to 0.
Note 2. To shift to Software Standby mode immediately after setting the RTCE bit to 1, follow the procedure in Figure 20.3.

The RTCC0 register is used to start or stop the realtime clock operation, control the RTC1HZ pin, and set a 12- or 24-hour system and the fixed-cycle interrupt.

CT[2:0] bits (Fixed-cycle interrupt (RTC_ALM_OR_PRD) selection)

To change the values of the CT[2:0] bits while counting is in progress (RTCE = 1), rewrite the values of the CT[2:0] bits after disabling interrupt processing of RTC_ALM_OR_PRD. Furthermore, after rewriting the values of the CT[2:0] bits, enable interrupt processing after clearing the RTCC1.RIFG flag.

AMPM bit (Selection of 12- or 24-hour system)

- Rewrite the AMPM bit value after setting the RWAIT bit of the realtime clock control register 1 (RTCC1) to 1. If the AMPM bit value is changed, the values of the hour count register (HOUR) change according to the specified time system.
- Table 20.3 shows the time (hour) digits indicated according to the setting of this bit.

RTC128EN bit (Selection of the operating clock for the realtime clock (RTCSCLK/RTCS128CLK))

- Setting this bit to 1 enables the realtime clock to operate with the 128-Hz clock for lower-power operation.
- Time error correction cannot be used when the setting of this bit is 1.

- The WUTMMCK0 bit in the OSMCR register should be set to 0 when setting this bit to 1. For details, see section 8, Clock Generation Circuit.

RCLOE1 bit (RTC1HZ pin output control)

This bit is used for RTC1HZ pin output control.

RTCE bit (Realtime clock operation control)

This bit is used for realtime clock operation control.

20.2.2 RTCC1 : Realtime Clock Control Register 1

Base address: RTC = 0x4009_2000

Offset address: 0x0C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	WALE	WALIE	—	WAFG	RIFG	—	RWST	RWAIT
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RWAIT	Wait control of realtime clock 0: Counting proceeds 1: Stops the SEC to YEAR counters. Counter values are readable and writable.	R/W
1	RWST	Wait status flag of realtime clock 0: Counting is in progress 1: Counter values are readable and writable	R
2	—	This bit is read as 0. The write value should be 0.	R/W
3	RIFG	Fixed-cycle interrupt status flag 0: Fixed-cycle interrupt is not generated 1: Fixed-cycle interrupt is generated	R/W
4	WAFG	Alarm detection status flag 0: Alarm mismatch 1: Detection of matching of alarm	R/W
5	—	This bit is read as 0. The write value should be 0.	R/W
6	WALIE	Control of alarm interrupt (RTC_ALM_OR_PRD) 0: Does not generate interrupt on matching of alarm 1: Generates interrupt on matching of alarm	R/W
7	WALE	Alarm operation control 0: Match operation is invalid 1: Match operation is valid	R/W

Note: To prevent the RIFG and WAFG flags from being cleared during writing, disable writing by setting 1 to the corresponding bit.
Note: Fixed-cycle interrupts and alarm match interrupts use the same interrupt source (RTC_ALM_OR_PRD). When using these two types of interrupts at the same time, which interrupt occurred can be judged by checking the fixed-cycle interrupt status flag (RIFG) and the alarm detection status flag (WAFG) upon RTC_ALM_OR_PRD occurrence.
Note: The internal counter (16 bits) is cleared when the second count register (SEC) is written.

The RTCC1 register is used to control the alarm interrupt and the wait time of the counter.

RWAIT bit (Wait control of realtime clock)

This bit controls the operation of the counter.

Be sure to write 1 to this bit to read or write the counter value.

So that the 16-bit internal counter continues to run, return the value of this bit to 0 on completion of reading or writing within one second. When reading or writing to the counter is required while generation of the alarm interrupt is enabled, first set the RTCC0.CT[2:0] bits to 010b (generating the constant-period interrupt once per 1 second). Then, complete the processing from setting the RWAIT bit to 1 to setting it to 0 before generation of the next constant-period interrupt.

After setting this bit to 1, it takes up to one cycle of RTCCLK until the counter value can be actually read or written (RWST = 1).*1 *2

When the internal counter (16 bits) overflows while the setting of this bit is 1, an indicator of the counter having overflowed is retained after RWAIT has become 0, after which counting up continues.

Note that, when the second count register has been written to, the overflow is not retained.

Note 1. When the RWAIT bit is set to 1 within one cycle of RTCCLK after setting the RTCC0.RTCE bit to 1, the setting of the RWST bit actually becoming 1 may take up to two cycles of the operating clock (RTCCLK).
Note 2. When the RWAIT bit is set to 1 within one cycle of RTCCLK after release from Sleep mode, Software Standby mode, or Snooze mode, the setting of the RWST bit actually becoming 1 may take up to two cycles of the operating clock (RTCCLK).

RWST flag (Wait status flag of realtime clock)

This flag indicates whether the setting of the RWAIT bit is valid.

Before reading or writing the counter value, confirm that the value of this flag is 1.

RIFG flag (Fixed-cycle interrupt status flag)

This flag indicates the status of generation of the fixed-cycle interrupt. When the fixed-cycle interrupt is generated, it is set to 1. This flag is cleared when 0 is written to it. Writing 1 to it is invalid.

WAFG flag (Alarm detection status flag)

This flag indicates detection of matching with the alarm. It is only valid when WALE is 1 and is set to 1 one cycle of RTCCLK after matching of the alarm is detected. This flag is cleared when 0 is written to it. Writing 1 to it is invalid.

WALIE bit (Control of alarm interrupt (RTC_ALM_OR_PRD))

This bit is used for control of alarm interrupt (RTC_ALM_OR_PRD).

WALE bit (Alarm operation control)

When setting a value to the WALE bit while counting is in progress (RTCC0.RTCE = 1) and WALIE = 1, rewrite the WALE bit after disabling interrupt processing of RTC_ALM_OR_PRD.

Furthermore, clear the WAFG flag after rewriting the WALE bit. When setting any of the alarm-related registers (WALIE flag of realtime clock control register 1 (RTCC1), the alarm minute register (ALARMWM), the alarm hour register (ALARMWH), and the alarm day-of-week register (ALARMWW)), set the WALE bit to 0 to disable matching.

20.2.3 SEC : Second Count Register

Base address: RTC = 0x4009_2000

Offset address: 0x00

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	SEC10[2:0]			SEC1[3:0]			
Value after reset:	0	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
3:0	SEC1[3:0]	1-second count Counts from 0 to 9 every second. When a carry is generated, 1 is added to the tens place.	R/W
6:4	SEC10[2:0]	10-second count Counts from 0 to 5 for 60-second counting.	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

Note: When reading from or writing to this register while the counter is in operation (RTCC0.RTCE = 1), follow the procedures described in section 20.3.3. Reading from and Writing to the Counters of the Realtime Clock.
Note: The internal counter (16 bits) is cleared when the second count register (SEC) is written.

The SEC register takes a value of 0 to 59 (decimal) and indicates the count value of seconds. This counter is incremented each time the internal counter (16-bit) overflows. When data is written to this register, it is written to a buffer and then to the

counter up to two cycles of RTCCLK later. Set a decimal value of 00 to 59 to this register in BCD code. This register is not initialized by a reset signal.

20.2.4 MIN : Minute Count Register

Base address: RTC = 0x4009_2000

Offset address: 0x01

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	MIN10[2:0]			MIN1[3:0]			
Value after reset:	0	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
3:0	MIN1[3:0]	1-minute count Counts from 0 to 9 every minute. When a carry is generated, 1 is added to the tens place.	R/W
6:4	MIN10[2:0]	10-minute count Counts from 0 to 5 for 60-minute counting.	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

Note: When reading from or writing to this register while the counter is in operation (RTCC0.RTCE = 1), follow the procedures described in section 20.3.3. Reading from and Writing to the Counters of the Realtime Clock.

The MIN register takes a value of 0 to 59 (decimal) and indicates the count value of minutes. This counter is incremented each time the second counter overflows. When data is written to this register, it is written to a buffer and then to the counter up to two cycles of RTCCLK later. Even if the second count register overflows while this register is being written, this register ignores the overflow and is set to the value written. Set a decimal value of 00 to 59 to this register in BCD code. This register is not initialized by a reset signal.

20.2.5 HOUR : Hour Count Register

Base address: RTC = 0x4009_2000

Offset address: 0x02

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	HOUR10[1:0]		HOUR1[3:0]			
Value after reset:	0	0	x	x	x	x	x	x

Bit	Symbol	Function	R/W
3:0	HOUR1[3:0]	1-hour count Counts from 0 to 9 per hour. When a carry is generated, 1 is added to the tens place.	R/W
5:4	HOUR10[1:0]	10-hour count Counts from 0 to 3 once per carry from the ones place.	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W

Note: The HOUR10[1] bit indicates AM (0)/PM (1) if RTCC0.AMPM = 0 (if the 12-hour system is selected).

Note: When reading from or writing to this register while the counter is in operation (RTCC0.RTCE = 1), follow the procedures described in section 20.3.3. Reading from and Writing to the Counters of the Realtime Clock.

The HOUR register takes a value of 00 to 23 or 01 to 12 and 21 to 32 (decimal) and indicates the count value of hours. This counter is incremented each time the minute counter overflows. When data is written to this register, it is written to a buffer and then to the counter up to two cycles of RTCCLK later. Even if the minute count register overflows while this register is being written, this register ignores the overflow and is set to the value written. Specify a decimal value of 00 to 23, 01 to 12, or 21 to 32 by using BCD code according to the time system specified using the AMPM bit of the realtime clock control register 0 (RTCC0). If the RTCC0.AMPM bit value is changed, the values of the HOUR register change according to the specified time system. This register is not initialized by a reset signal.

Table 20.3 shows the relationship between the setting value of the RTCC0.AMPM bit, the hour count register (HOUR) value, and time.

Table 20.3 Displayed time digits

24-hour display (RTCC0.AMPM = 1)		12-hour display (RTCC0.AMPM = 0)	
Time	**HOUR register**	**Time**	**HOUR register**
0	0x00	12 a.m.	0x12
1	0x01	1 a.m.	0x01
2	0x02	2 a.m.	0x02
3	0x03	3 a.m.	0x03
4	0x04	4 a.m.	0x04
5	0x05	5 a.m.	0x05
6	0x06	6 a.m.	0x06
7	0x07	7 a.m.	0x07
8	0x08	8 a.m.	0x08
9	0x09	9 a.m.	0x09
10	0x10	10 a.m.	0x10
11	0x11	11 a.m.	0x11
12	0x12	12 p.m.	0x32
13	0x13	1 p.m.	0x21
14	0x14	2 p.m.	0x22
15	0x15	3 p.m.	0x23
16	0x16	4 p.m.	0x24
17	0x17	5 p.m.	0x25
18	0x18	6 p.m.	0x26
19	0x19	7 p.m.	0x27
20	0x20	8 p.m.	0x28
21	0x21	9 p.m.	0x29
22	0x22	10 p.m.	0x30
23	0x23	11 p.m.	0x31

The HOUR register value is set to 12-hour display when the RTCC0.AMPM bit is 0 and to 24-hour display when the RTCC0.AMPM bit is 1. In 12-hour display, the HOUR10[1] bit displays 0 for AM and 1 for PM.

20.2.6 DAY : Day Count Register

Base address: RTC = 0x4009_2000

Offset address: 0x04

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	DAY10[1:0]		DAY1[3:0]			
Value after reset:	0	0	x	x	x	x	x	x

Bit	Symbol	Function	R/W
3:0	DAY1[3:0]	1-day count Counts from 0 to 9 per day. When a carry is generated, 1 is added to the tens place.	R/W
5:4	DAY10[1:0]	10-day count Counts from 0 to 3 once per carry from the ones place.	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W

Note: When reading from or writing to this register while the counter is in operation (RTCC0.RTCE = 1), follow the procedures described in section 20.3.3. Reading from and Writing to the Counters of the Realtime Clock.

The DAY register takes a value of 1 to 31 (decimal) and indicates the count value of days.

This counter is incremented each time the hour counter overflows. Counting by the date counter proceeds as shown below.

- 01 to 31 (January, March, May, July, August, October, December)
- 01 to 30 (April, June, September, November)
- 01 to 29 (February, leap year)
- 01 to 28 (February, normal year)

When data is written to this register, it is written to a buffer and then to the counter up to two cycles of RTCCLK later. Even if the hour count register overflows while this register is being written, this register ignores the overflow and is set to the value written. Set a decimal value of 01 to 31 to this register in BCD code.

This register is not initialized by a reset signal.

20.2.7 WEEK : Day-of-Week Count Register

Base address: RTC = 0x4009_2000

Offset address: 0x03

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

Bit	Symbol	Function	R/W
2:0	WEEK[2:0]	Day-of-Week Counting 0 0 0: Sunday 0 0 1: Monday 0 1 0: Tuesday 0 1 1: Wednesday 1 0 0: Thursday 1 0 1: Friday 1 1 0: Saturday Others: Setting prohibited	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: The value corresponding to the month count register (MONTH) or the day count register (DAY) is not stored in the day-of-week count register (WEEK) automatically. Set the day of the week count register at each setting.

Note: When reading from or writing to this register while the counter is in operation (RTCC0.RTCE = 1), follow the procedures described in section 20.3.3. Reading from and Writing to the Counters of the Realtime Clock.

The WEEK register takes a value of 0 to 6 (decimal) and indicates the count value of days of the week. This counter is incremented in synchronization with the date counter. When data is written to this register, it is written to a buffer and then to the counter up to two cycles of RTCCLK later. Set a decimal value of 00 to 06 to this register in BCD code. This register is not initialized by a reset signal.

20.2.8 MONTH : Month Count Register

Base address: RTC = 0x4009_2000

Offset address: 0x05

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	MONTH10	MONTH1[3:0]			
Value after reset:	0	0	0	x	x	x	x	x

Bit	Symbol	Function	R/W
3:0	MONTH1[3:0]	1-month count Counts from 0 to 9 once per month. When a carry is generated, 1 is added to the tens place.	R/W
4	MONTH10	10-month count Counts from 0 to 1 once per carry from the ones place.	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: When reading from or writing to this register while the counter is in operation (RTCC0.RTCE = 1), follow the procedures described in section 20.3.3. Reading from and Writing to the Counters of the Realtime Clock.

The MONTH register takes a value of 1 to 12 (decimal) and indicates the count value of months.

This counter is incremented each time the day counter overflows. When data is written to this register, it is written to a buffer and then to the counter up to two cycles of RTCCLK later. Even if the day count register overflows while this register is being written, this register ignores the overflow and is set to the value written. Set a decimal value of 01 to 12 to this register in BCD code. This register is not initialized by a reset signal.

20.2.9 YEAR : Year Count Register

Base address: RTC = 0x4009_2000

Offset address: 0x06

Bit position:	7	6	5	4	3	2	1	0
Bit field:	YEAR10[3:0]				YEAR1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
3:0	YEAR1[3:0]	1-year count Counts from 0 to 9 once per year. When a carry is generated, 1 is added to the tens place.	R/W
7:4	YEAR10[3:0]	10-year count Counts from 0 to 1 once per carry from the ones place.	R/W

Note: When reading from or writing to this register while the counter is in operation (RTCC0.RTCE = 1), follow the procedures described in section 20.3.3. Reading from and Writing to the Counters of the Realtime Clock.

The YEAR register takes a value of 0 to 99 (decimal) and indicates the value of the counter of years. This counter is incremented each time the month count register (MONTH) overflows.

Values 00, 04, 08, …, 92, and 96 indicate a leap year. When data is written to this register, it is written to a buffer and then to the counter up to two cycles of RTCCLK later. Even if the month count register overflows while this register is being written, this register ignores the overflow and is set to the value written. Set a decimal value of 00 to 99 to this register in BCD code. This register is not initialized by a reset signal.

20.2.10 SUBCUD : Time Error Correction Register

Base address: RTC = 0x4009_2000

Offset address: 0x07

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DEV	F6	F[5:0]					
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	F[5:0]	Adjustment Value	R/W
6	F6	Setting of time error correction value 0: Increases by (F[5:0] – 1) × 2 1: Decreases by (/F[5:0] + 1) × 2	R/W

Bit	Symbol	Function	R/W
7	DEV	Setting of time error correction timing 0: Corrects time error when the second digits are at 00, 20, or 40 (every 20 seconds) 1: Corrects time error only when the second digits are at 00 (every 60 seconds)	R/W

This register is used to correct the time with high accuracy when it is running slow or fast by adjusting the value that is considered an overflow from the internal counter (16 bits) to the second count register (SEC) (reference value: 0x7FFF).

Note: Time error correction cannot be used in the 128-Hz operating mode (RTCC0.RTC128EN = 1). It can only proceed if the setting of the RTCC0.RTC128EN bit is 0.

F[5:0] bits (Adjustment Value)

These bits specify the adjustment value from the prescaler.

F6 bit (Setting of time error correction value)

When (F6, F[5:0]) = *00000*b, the time error is not corrected. * is 0 or 1.

/F[5:0] are the inverted values of the corresponding bits (000011b when 111100b).

Range of correction value:

- (when F6 = 0) 2, 4, 6, 8, …, 120, 122, 124
- (when F6 = 1) –2, –4, –6, –8, …, –120, –122, –124

DEV bit (Setting of time error correction timing)

Writing to the SUBCUD register at the following timing is prohibited.

- When DEV = 0 is set: For a period of SEC = 0x00, 0x20, 0x40
- When DEV = 1 is set: For a period of SEC = 0x00

The range of value that can be corrected by using the time error correction register (SUBCUD) is shown in Table 20.4.

Table 20.4 Correction range using time error correction register (SUBCUD)

	DEV = 0 (correction every 20 seconds)	DEV = 1 (correction every 60 seconds)
Correctable range	–189.2 ppm to 189.2 ppm	–63.1 ppm to 63.1 ppm
Maximum excludes quantization error	±1.53 ppm	±0.51 ppm
Minimum resolution	±3.05 ppm	±1.02 ppm

Note: If a correctable range is –63.1 ppm or lower and 63.1 ppm or higher, set DEV to 0.

20.2.11 ALARMWM : Alarm Minute Register

Base address: RTC = 0x4009_2000

Offset address: 0x08

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	WM10[2:0]			WM1[3:0]			
Value after reset:	0	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
3:0	WM1[3:0]	1-digit minute setting Value for the ones place of minutes.	R/W
6:4	WM10[2:0]	10-digit minute setting Value for the tens place of minutes.	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

This register is used to set minutes of alarm. This register is not initialized by a reset signal.

Note: Set a decimal value of 00 to 59 to this register in BCD code. If a value outside the range is set, the alarm is not detected.

20.2.12 ALARMWH : Alarm Hour Register

Base address: RTC = 0x4009_2000

Offset address: 0x09

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	WH10[1:0]		WH1[3:0]			
Value after reset:	0	0	x	x	x	x	x	x

Bit	Symbol	Function	R/W
3:0	WH1[3:0]	1-digit hour setting Value for the ones place of hours.	R/W
5:4	WH10[1:0]	10-digit hour setting Value for the tens place of hours.	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W

Note: The WH10[1] bit indicates AM (0)/PM (1) if RTCC0.AMPM = 0 (if the 12-hour system is selected).

This register is used to set hours of alarm. This register is not initialized by a reset signal.

Note: Set a decimal value of 00 to 23, 01 to 12, or 21 to 32 to this register in BCD code. If a value outside the range is set, the alarm is not detected.

20.2.13 ALARMWW : Alarm Day-of-Week Register

Base address: RTC = 0x4009_2000

Offset address: 0x0A

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	WW6	WW5	WW4	WW3	WW2	WW1	WW0
Value after reset:	0	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
0	WW0	Alarm enabled setting "Sunday" 0: Disable alarm settings for that day of the week 1: Enable alarm settings for that day of the week	R/W
1	WW1	Alarm enabled setting "Monday" 0: Disable alarm settings for that day of the week 1: Enable alarm settings for that day of the week	R/W
2	WW2	Alarm enabled setting "Tuesday" 0: Disable alarm settings for that day of the week 1: Enable alarm settings for that day of the week	R/W
3	WW3	Alarm enabled setting "Wednesday" 0: Disable alarm settings for that day of the week 1: Enable alarm settings for that day of the week	R/W
4	WW4	Alarm enabled setting "Thursday" 0: Disable alarm settings for that day of the week 1: Enable alarm settings for that day of the week	R/W
5	WW5	Alarm enabled setting "Friday" 0: Disable alarm settings for that day of the week 1: Enable alarm settings for that day of the week	R/W

Bit	Symbol	Function	R/W
6	WW6	Alarm enabled setting "Saturday" 0: Disable alarm settings for that day of the week 1: Enable alarm settings for that day of the week	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

This register is used to set days of the week of alarm. This register is not initialized by a reset signal.

Table 20.5 shows an example of setting the alarm.

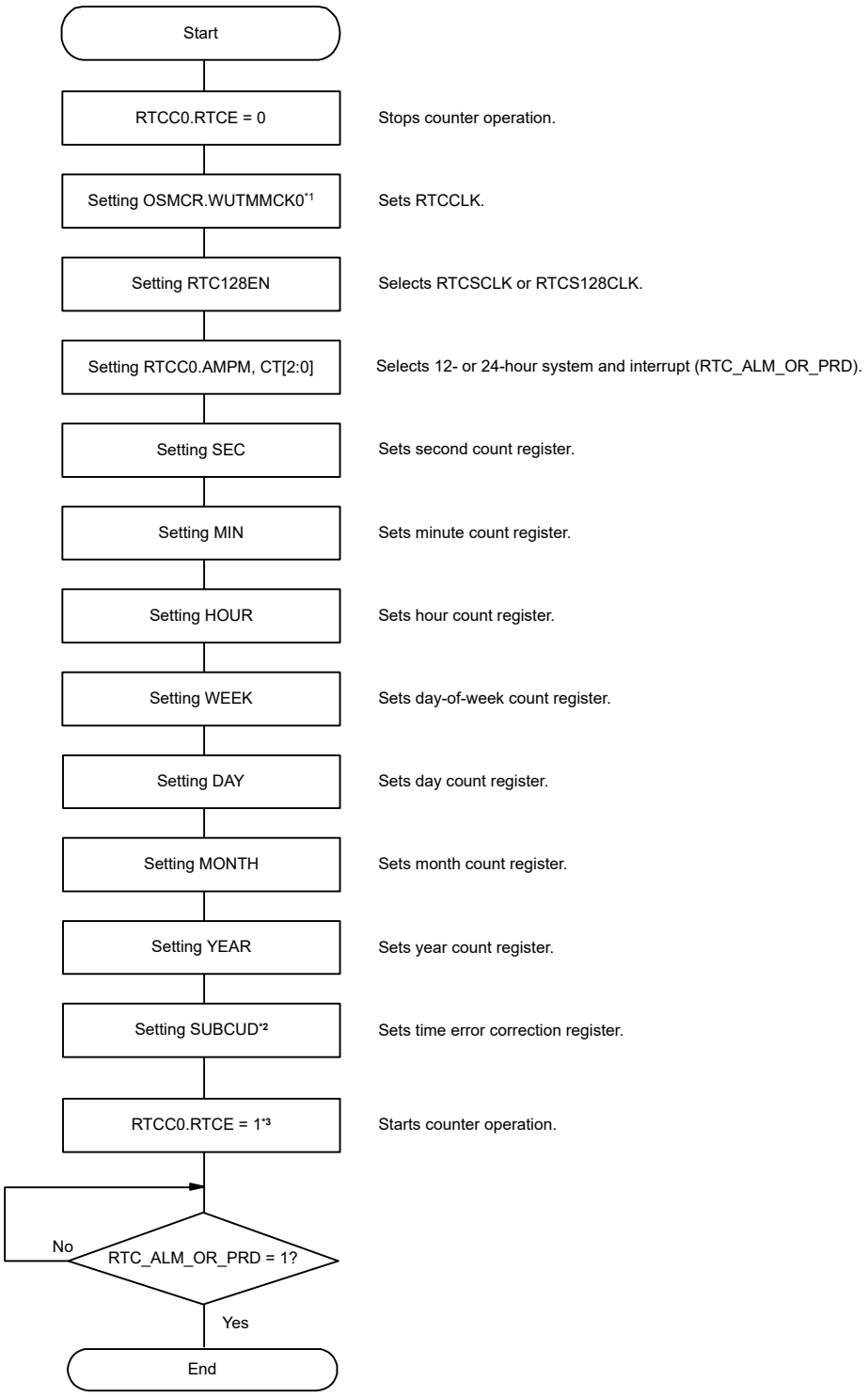
Table 20.5 Example of setting the alarm

Time of Alarm	Day of week							12-hour display				24-hour display			
	Sunday WW0	Monday WW1	Tuesday WW2	Wednesday WW3	Thursday WW4	Friday WW5	Saturday WW6	Hour 10	Hour 1	Minute 10	Minute 1	Hour 10	Hour 1	Minute 10	Minute 1
Every day, 0:00 a.m.	1	1	1	1	1	1	1	1	2	0	0	0	0	0	0
Every day, 1:30 a.m.	1	1	1	1	1	1	1	0	1	3	0	0	1	3	0
Every day, 11:59 a.m.	1	1	1	1	1	1	1	1	1	5	9	1	1	5	9
Monday through Friday, 0:00p.m.	0	1	1	1	1	1	0	3	2	0	0	1	2	0	0
Sunday, 1:30 p.m.	1	0	0	0	0	0	0	2	1	3	0	1	3	3	0
Monday, Wednesday, Friday, 11:59 p.m.	0	1	0	1	0	1	0	3	1	5	9	2	3	5	9

20.3 Operation

20.3.1 Starting the Realtime Clock Operation

Figure 20.2 shows the procedure for starting realtime clock operation.



Note: ● RTCSCLK: RTC sub clock
● RTCS128CLK: RTC sub 128 Hz clock

Note 1. For details, see section 8, Clock Generation Circuit.

Note 2. Set up the SUBCUD register only if the time error must be corrected. For details about how to calculate the correction value, see section 20.3.6. Example of Time Error Correction by the Realtime Clock.
Time error correction cannot be used while the setting of the RTCC0.RTC128EN bit is 1.

Note 3. Confirm the procedure described in section 20.3.2. Shifting to Sleep or Software Standby Mode after Starting Operation when shifting to Sleep or Software Standby mode without waiting for RTC_ALM_OR_PRD = 1 after RTCC0.RTCE = 1.

Figure 20.2 Procedure for starting the realtime clock operation

20.3.2 Shifting to Sleep or Software Standby Mode after Starting Operation

Take either of the steps listed below when shifting to Sleep or Software Standby mode immediately after setting the RTCC0.RTCE bit to 1. Note that any of these steps is not required when shifting to the Sleep or Software Standby mode after the RTC_ALM_OR_PRD interrupt has occurred.

- Transition to Sleep or Software Standby mode when at least two counter clock cycles (RTCCLK) have elapsed after setting the RTCC0.RTCE bit to 1 (see Figure 20.3, Example 1).
- After setting the RTCC0.RTCE bit to 1 and then setting the RTCC1.RWAIT bit to 1, poll the RTCC1.RWST bit to check if it has become 1 yet. After setting the RTCC1.RWAIT bit to 0 and polling the RTCC1.RWST bit to check if it has become 0 yet, a transition to Sleep or Software Standby mode will proceed (see Figure 20.3, Example 2).

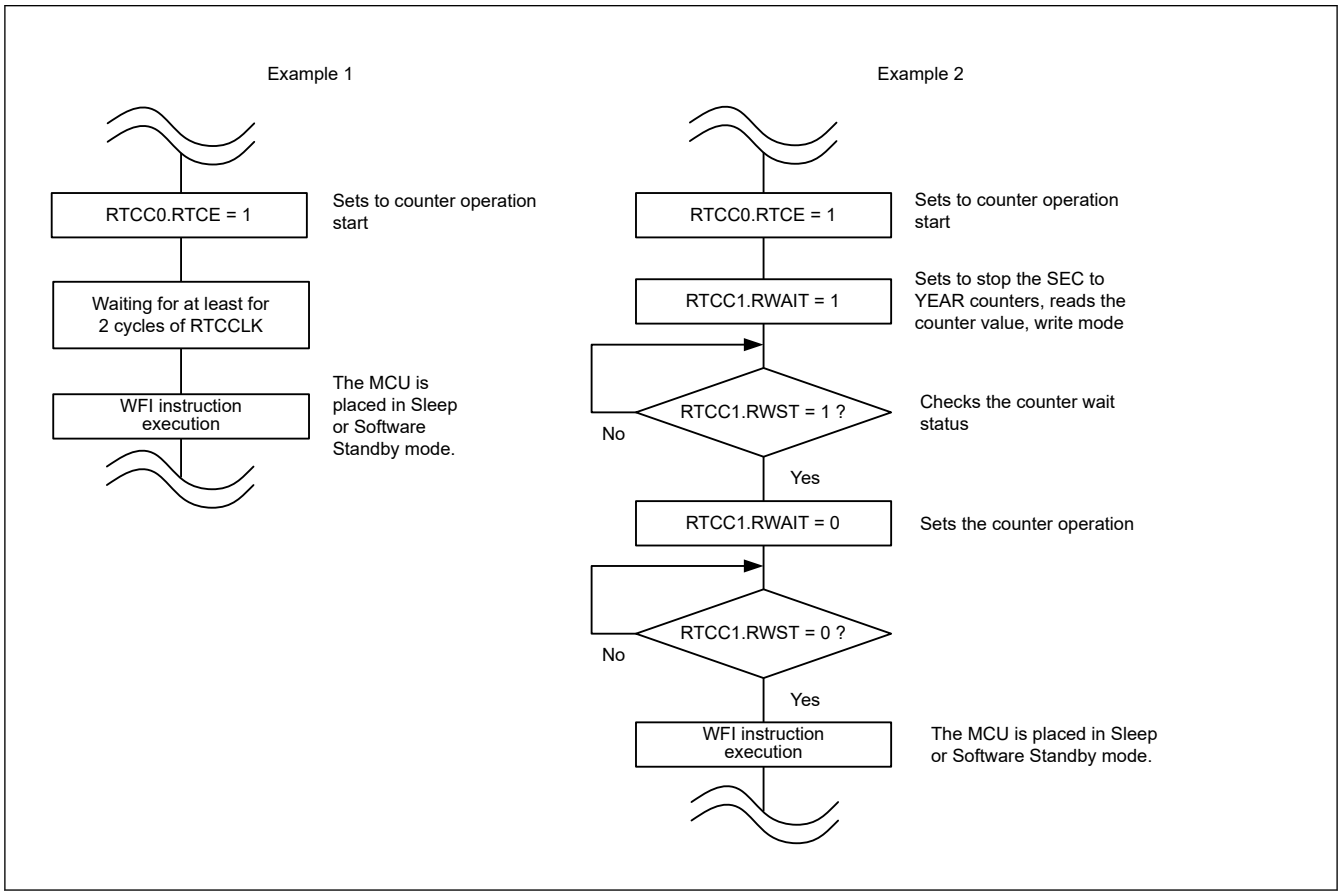


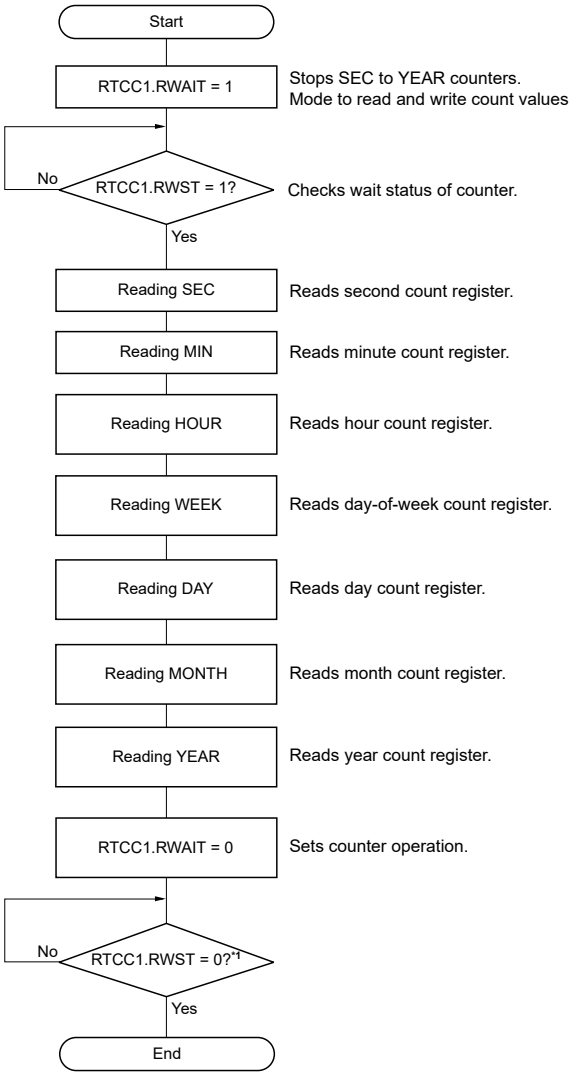
Figure 20.3 Procedure for shifting to Sleep or Software Standby mode after setting RTCC0.RTCE bit to 1

20.3.3 Reading from and Writing to the Counters of the Realtime Clock

Read or write the counter after setting 1 to the RTCC1.RWAIT bit first.

Set the RTCC1.RWAIT bit to 0 after completion of reading or writing the counter.

Figure 20.4 shows the procedure for reading realtime clock.



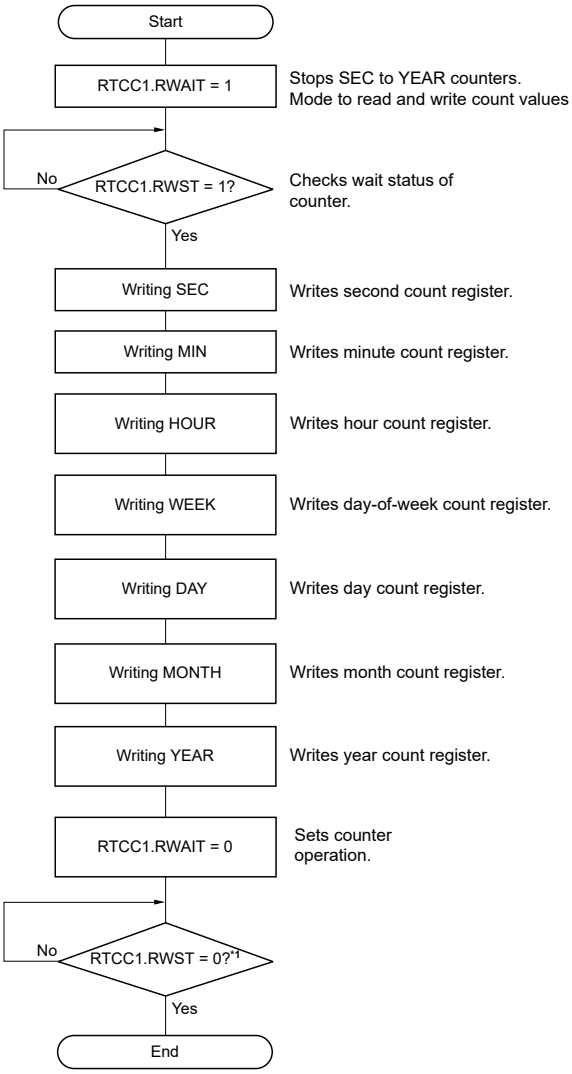
Note: Complete the series of process of setting the RTCC1.RWAIT bit to 1 to clearing the RTCC1.RWAIT bit to 0 within 1 second. When reading or writing to the counter is required while generation of the alarm interrupt is enabled, first set the RTCC0.CT[2:0] bits to 010b (generating the constant-period interrupt once per 1 second). Then, complete the processing from setting the RWAIT bit to 1 to setting it to 0 before generation of the next constant-period interrupt.

Note: The second count register (SEC), minute count register (MIN), hour count register (HOUR), day-of-week count register (WEEK), day count register (DAY), month count register (MONTH), and year count register (YEAR) may be read in any sequence. All the registers do not have to read and only some registers may be read.

Note 1. Be sure to confirm that RTCC1.RWST = 0 before setting Software Standby mode.

Figure 20.4 Procedure for reading realtime clock

Figure 20.5 shows the procedure for writing realtime clock.



Note: Complete the series of operations of setting the RTCC1.RWAIT bit to 1 to clearing the RTCC1.RWAIT bit to 0 within 1 second. When reading or writing to the counter is required while generation of the alarm interrupt is enabled, first set the RTCC0.CT[2:0] bits to 010b (generating the constant-period interrupt once per 1 second). Then, complete the processing from setting the RWAIT bit to 1 to setting it to 0 before generation of the next constant-period interrupt.

Note: When changing the values of the SEC, MIN, HOUR, WEEK, DAY, MONTH, and YEAR register while the counting is in progress (RTCC0.RTCE = 1), rewrite the values of each register after disabling interrupt processing of RTC_ALM_OR_PRD. Furthermore, clear the RTCC1.WAFG and RTCC1.RIFG flags after rewriting each register.

Note: The second count register (SEC), minute count register (MIN), hour count register (HOUR), day-of-week count register (WEEK), day count register (DAY), month count register (MONTH), and year count register (YEAR) may be written in any sequence. All the registers do not have to be set and only some registers may be written.

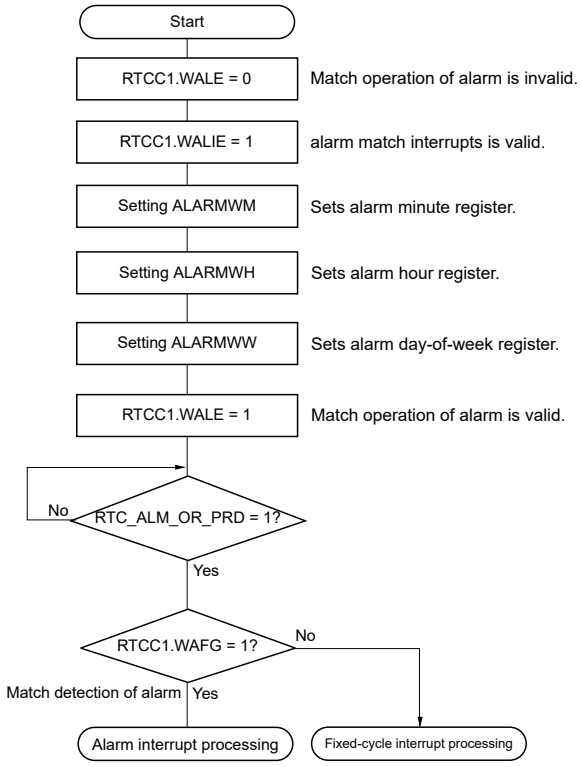
Note 1. Be sure to confirm that RTCC1.RWST = 0 before setting Software Standby mode.

Figure 20.5 Procedure for writing realtime clock

20.3.4 Setting Alarm by the Realtime Clock

Set time of alarm after setting 0 to the RTCC1.WALE bit (alarm operation invalid.) first.

Figure 20.6 shows the alarm processing procedure.



Note: The alarm minute register (ALARMWM), alarm hour register (ALARMWH), and alarm day-of-week register (ALARMWW) may be written in any sequence.

Note: Fixed-cycle interrupts and alarm match interrupts use the same interrupt source (RTC_ALM_OR_PRD). To use these two types of interrupts at the same time, the source of the interrupt can be identified by checking the fixed-cycle interrupt status flag (RTCC1.RIFG) and the alarm detection status flag (RTCC1.WAFG) when an RTC_ALM_OR_PRD is generated.

Figure 20.6 Alarm processing procedure

20.3.5 1 Hz Output by the Realtime Clock

Table 20.6 shows the 1 Hz output setting procedure.

Table 20.6 1 Hz output setting procedure

Step		Process	Detail
1 Hz output setting procedure	<1>	1 Hz output setting start.	—
	<2>	Stop counter operation.	RTCC0.RTCE bit to 0.
	<3>	Setting port.	See section 16, I/O Ports.
	<4>	Enable output of the RTC1HZ pin.	RTCC0.RCLOE1 bit to 1.
	<5>	Start counter operation.	Set RTCC0.RTCE bit to 1.
	<6>	Output start from RTC1HZ pin.	—

20.3.6 Example of Time Error Correction by the Realtime Clock

Time can be corrected with high accuracy when it is slow or fast, by setting a value to the time error correction register.

(1) Example of calculating the correction value

The correction value used when correcting the count value of the internal counter (16 bits) is calculated by using the following expression.

Set the SUBCUD.DEV bit to 0 when the correction range is –63.1 ppm or less, or 63.1 ppm or more.

(When SUBCUD.DEV = 0)

Correction value[*1] [*2]= Number of correction counts in 1 minute ÷ 3 = (Oscillation frequency[*3] ÷ Target frequency[*4] – 1) × 32768 × 60 ÷ 3

(When SUBCUD.DEV = 1)

Correction value[*1] [*2] = Number of correction counts in 1 minute = (Oscillation frequency[*3] ÷ Target frequency[*4] – 1) × 32768 × 60

Note 1. The correction value is the time error correction value calculated by using the F6 and F[5:0] bits of the time error correction register (SUBCUD).

(When SUBCUD.F6 = 0) Correction value = (F[5:0] – 1) × 2

(When SUBCUD.F6 = 1) Correction value = –(/F[5:0] + 1) × 2

(When SUBCUD.F6, F[5:0] = *00000*b) time error correction is not performed. "*" is 0 or 1.

/F[5:0] are bit-inverted values (000011b when 111100b).

Note 2. The correction value is 2, 4, 6, 8, … 120, 122, 124 or –2, –4, –6, –8, … –120, –122, –124.

Note 3. The oscillation frequency is a value of the count clock (RTCCLK). It can be calculated from the output frequency of the RTC1HZ pin × 32768 when the time error correction register is set to its initial value (0x00).

Note 4. The target frequency is the frequency resulting after correction performed by using the time error correction register.

(2) Correction example

Example of correcting from 32767.4 Hz to 32768 Hz (32767.4 Hz + 18.3 ppm)

[Measuring the oscillation frequency]

To measure the oscillation frequency [*1] of each product, a signal at about 1 Hz can be output from the RTC1HZ pin when the clock error correction register (SUBCUD) is set to its initial value (0x00).

Note 1. See section 20.3.5. 1 Hz Output by the Realtime Clock for the setting procedure for output of about 1 Hz from the RTC1HZ pin.

[Calculating the correction value]

When the output frequency from the RTC1HZ pin is 0.9999817 Hz:

Oscillation frequency = 32768 × 0.9999817 ≈ 32767.4 Hz

Given that the target frequency is 32768 Hz (32767.4 Hz + 18.3 ppm), set the SUBCUD.DEV bit to 0. Accordingly, the expression for calculating the correction value when the SUBCUD.DEV bit is 1 is applicable.

Correction value

= Error for correction of counting of 1 minute

= (Oscillation frequency ÷ Target frequency –1) × 32768 × 60

= (32767.4 ÷ 32768 – 1) × 32768 × 60

= –36

[Calculating the values to be set to (SUBCUD.F6, F[5:0])]

When the correction value is –36:

If the correction value is 0 or less (the clock is running fast), set the SUBCUD.F6 bit to 0. Calculate the SUBCUD.F[5:0] bit from the correction value.

–(/F[5:0] + 1) × 2 = –36

/F[5:0] = 17

/F[5:0] = 010001b

F[5:0] = 101110b

Consequently, when correcting from 32767.4 Hz to 32768 Hz (32767.4 Hz + 18.3 ppm), setting the correction register such that the SUBCUD.DEV bit is 1 and the correction value is –36 ((SUBCUD.F6, F[5:0]) = 1101110b) results in the desired frequency of 32768 Hz (error of 0 ppm).

Figure 20.7 shows the operation for correction when the value of (SUBCUD.DEV, F6, F[5:0]) is 11101110b.

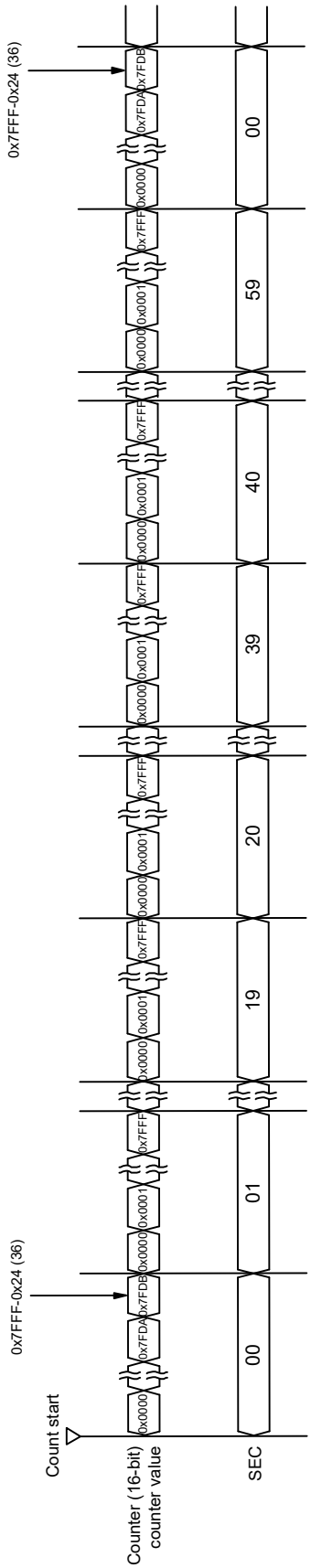


Figure 20.7 Operation for correction when the value of (SUBCUD.DEV, F6, F[5:0]) = 11101110b

21. Watchdog Timer (WDT)

21.1 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 21.1 lists the WDT specifications and Figure 21.1 shows a block diagram.

Table 21.1 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
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

Note 1. Satisfy the frequency of the peripheral module clock (PCLKB) ≥ 4 × (the frequency of the count clock source after division).

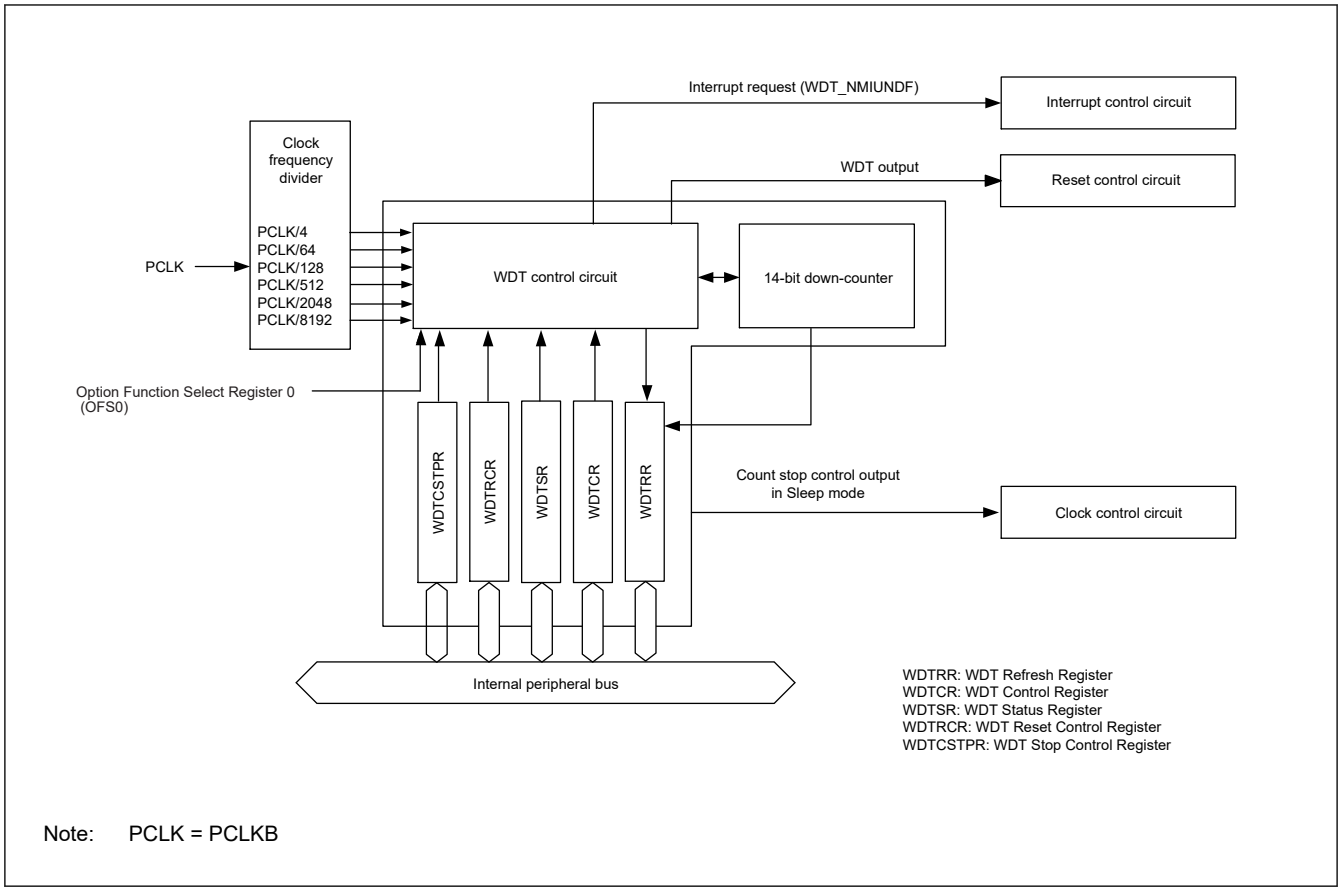


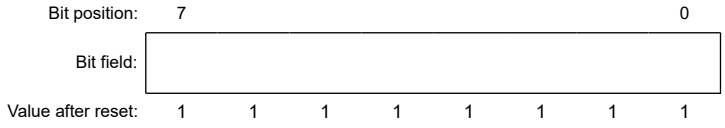
Figure 21.1 WDT block diagram

21.2 Register Descriptions

21.2.1 WDTRR : WDT Refresh Register

Base address: WDT = 0x4004_4200

Offset address: 0x00



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 21.3.3. Refresh Operation.

21.2.2 WDTCR : WDT Control Register

Base address: WDT = 0x4004_4200

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 21.3.2. 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 21.3.8. 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 21.2 lists the relationship between the CKS[3:0] and TOPS[1:0] bit settings, the timeout period, and the number of PCLKB cycles.

Table 21.2 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)

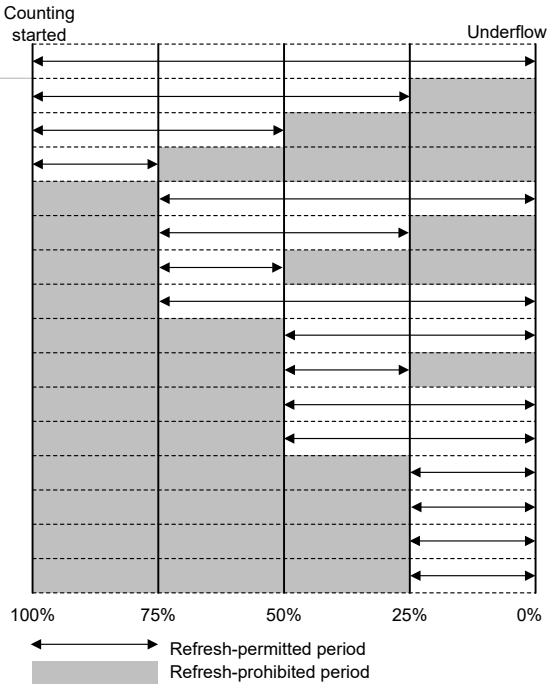
The RPSS[1:0] bits specify the window start position that indicates the refresh-permitted period. 100%, 75%, 50%, or 25% of the timeout period can be selected for the window start position. Set the window start position to a value greater than the value for the window end position. If the window start 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 21.3 lists the counter values for the window start and end positions, and Figure 21.2 shows the refresh-permitted period set in the RPSS[1:0], RPES[1:0], and TOPS[1:0] bits.

Table 21.3 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 21.2 RPSS[1:0] and RPES[1:0] bits setting and refresh-permitted period

21.2.3 WDTSR : WDT Status Register

Base address: WDT = 0x4004_4200

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

21.2.4 WDTRCR : WDT Reset Control Register

Base address: WDT = 0x4004_4200

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 21.3.2. 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 21.3.8. Association between Option Function Select Register 0 (OFS0) and WDT Registers.

21.2.5 WDTCSTPR : WDT Count Stop Control Register

Base address: WDT = 0x4004_4200

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 21.3.2. 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 21.3.8. 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.

21.2.6 Option Function Select Register 0 (OFS0)

For information on the OFS0 register, see section 21.3.8. Association between Option Function Select Register 0 (OFS0) and WDT Registers.

21.3 Operation

21.3.1 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.

21.3.1.1 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:

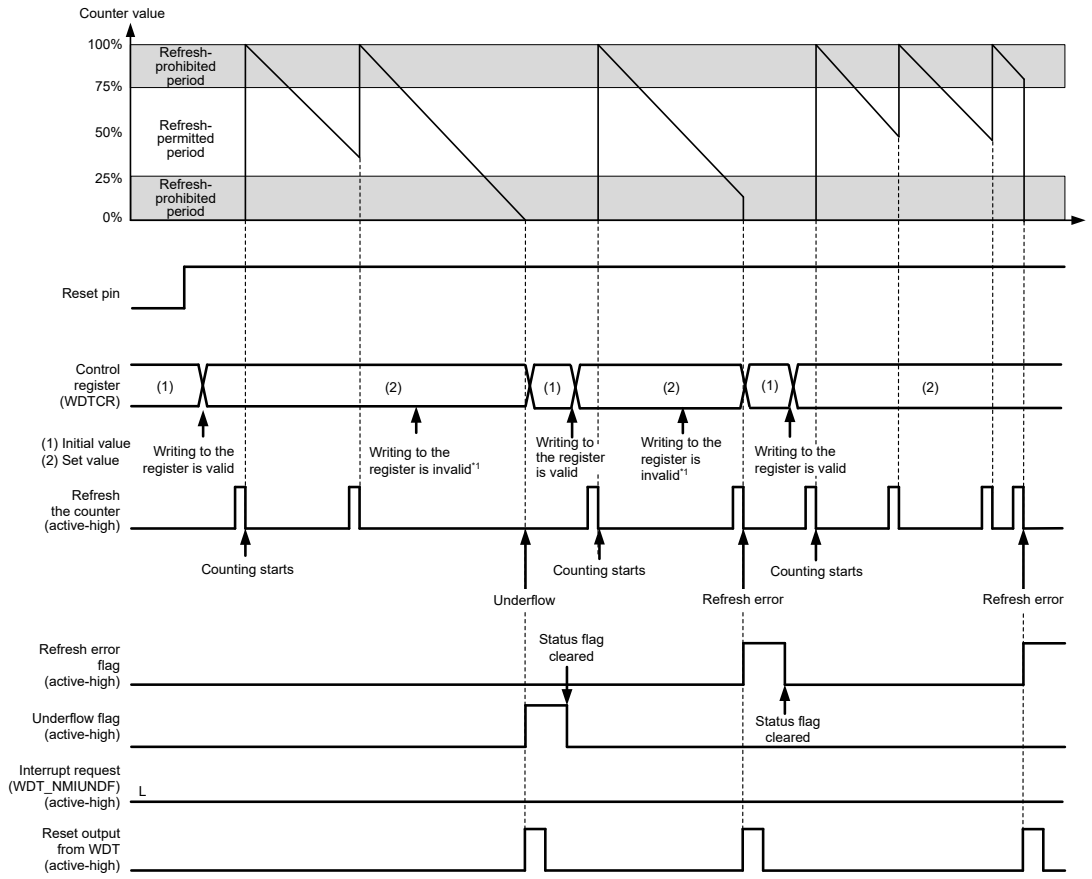
- Clock division ratio in the WDTCR register
- Window start and end positions in the WDTCR register
- Timeout period in the WDTCR register
- Reset output or interrupt request output in the WDTRCR register
- Counter stop control during transitions to Sleep mode in the WDTCSTPR register

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 21.3 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 21.3.2. Controlling Writes to the WDTCR, WDTRCR, and WDTCSTPR Registers.
Reset pin = RES

Figure 21.3 Operation example in register start mode

21.3.1.2 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

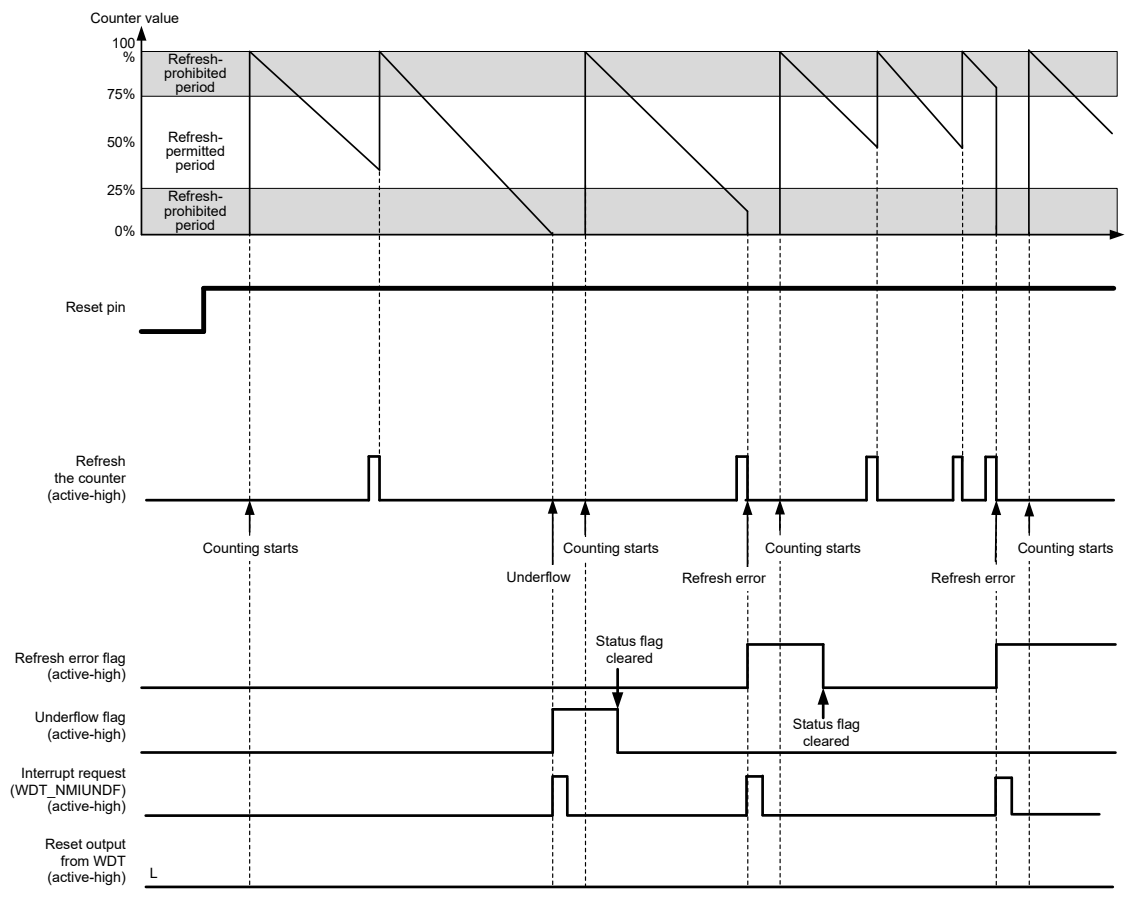
request/interrupt request (WDT_NMIUNDF) as long as the counting continues. However, if the down-counter underflows because refreshing of the down-counter is not possible due to a runaway program or if a refresh error occurs due to refreshing outside the refresh-permitted period, the WDT outputs the reset 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 21.4 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 21.4 Operation example in auto start mode

21.3.2 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 21.5 shows control waveforms produced in response to writing to the WDTCR.

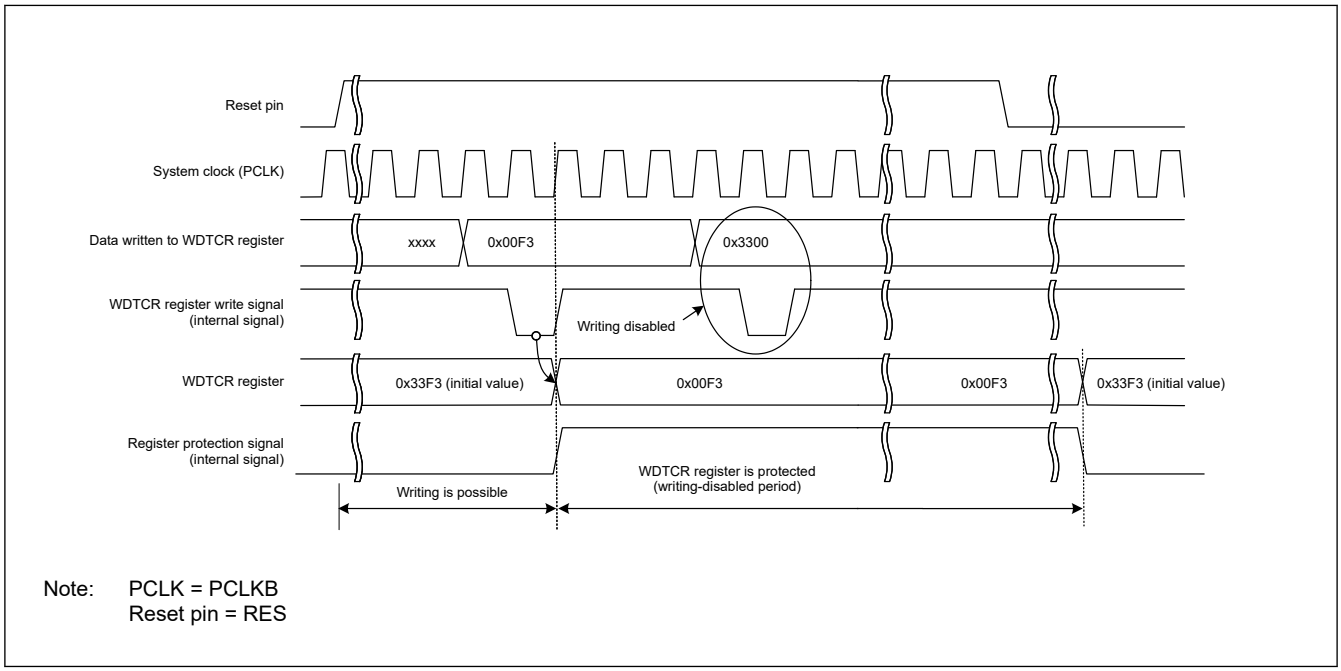


Figure 21.5 Control waveforms produced in response to writes to the WDTCR register

21.3.3 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]

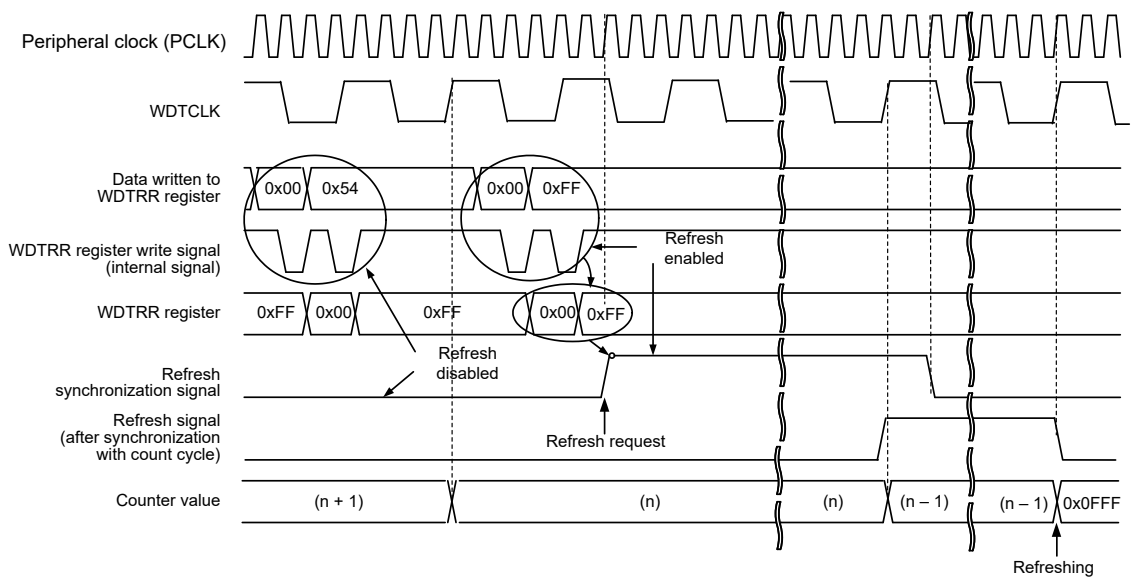
- 0x00 → 0xFF
- 0x00 ((n–1)th time) → 0x00 (nth time) → 0xFF
- 0x00 → access to another register or read from WDTRR → 0xFF

[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 21.6 shows the WDT refresh-operation waveforms when the clock division ratio is PCLKB/64.



Note: PCLK = PCLKB

Figure 21.6 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.

21.3.4 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 21.2.3. WDTSR : WDT Status Register.

21.3.5 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.

21.3.6 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).

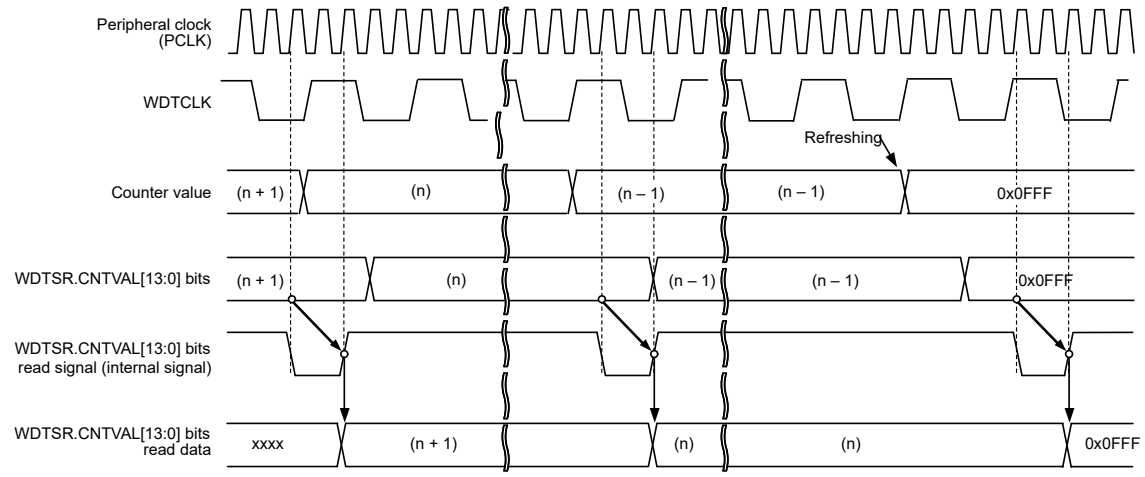
Table 21.4 WDT interrupt source

Name	Interrupt source	Interrupt to CPU	Start DTC
WDT_NMIUNDF	• Down-counter underflow • Refresh error	Possible	Not possible

21.3.7 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 21.7 shows the processing for reading the WDT down-counter value when the clock division ratio is PCLKB/64.



Note: PCLK = PCLKB

Figure 21.7 Processing for reading WDT down-counter value when WDTCR.CKS[3:0] = 0x4 and WDTCR.TOPS[1:0] = 01b

21.3.8 Association between Option Function Select Register 0 (OFS0) and WDT Registers

Table 21.5 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 21.5 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

21.4 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 settings of the WDTRCR.RSTIRQS bit in register start mode or the OFS0.WDTRSTIRQS bit in 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 15, Event Link Controller (ELC).

21.5 Usage Notes

21.5.1 ICU Event Link Setting Register n (IELSRn) Setting

Setting 0x06 to ICU Event Link Setting Register n (ICU.IELSRn) is prohibited when WDT reset interrupt request selection resets (OFS0.WDTRSTIRQS = 1 or WDTRCR.RSTIRQS = 1) or when enabling event link operation (ELSRn.ELS[7:0] = 0x12).

22. Independent Watchdog Timer (IWDT)

22.1 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 22.1 lists the IWDT specifications and Figure 22.1 shows a block diagram.

Table 22.1 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
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).

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.

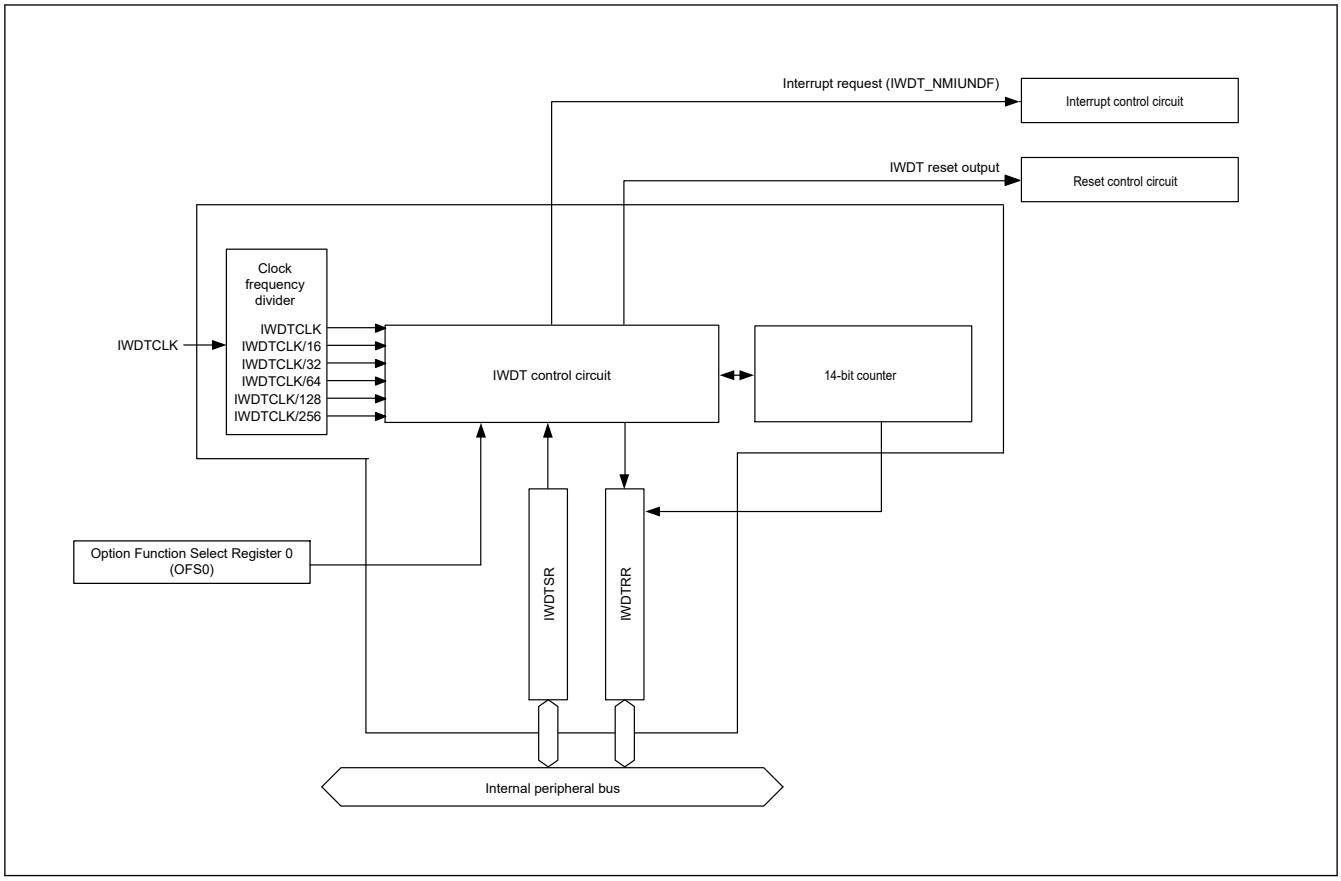


Figure 22.1 IWDT block diagram

22.2 Register Descriptions

22.2.1 IWDTRR : IWDT Refresh Register

Base address: IWDT = 0x4004_4400

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 22.3.2. Refresh Operation.

22.2.2 IWDTSR : IWDT Status Register

Base address: IWDT = 0x4004_4400

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.

22.2.3 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 22.2 lists the relationship between the IWDTCKS[3:0] and IWDTTOPS[1:0] bit settings, the timeout period, and the number of IWDTCLK cycles.

Table 22.2 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	b3	b2			
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)

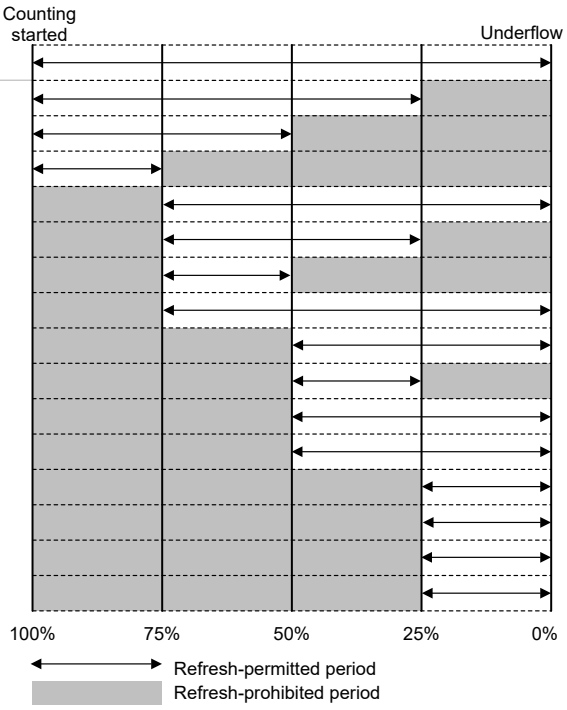
The IWDTRPSS[1:0] bits specify the window start position that indicates the refresh-permitted period. 100%, 75%, 50%, or 25% of the timeout period can be selected for the window start position. Set the window start position to a value greater than the window end position. If the window start position 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 22.3 lists the counter values for the window start and end positions, and Figure 22.2 shows the refresh-permitted period set in the IWDTRPSS[1:0], IWDTRPES[1:0], and IWDTTOPS[1:0] bits.

Table 22.3 Relationship between the timeout period and window start and end counter values

IWDTTOPS[1:0] bits		Timeout period		Window start and end counter value			
b3	b2	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 22.2 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.

22.3 Operation

22.3.1 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)

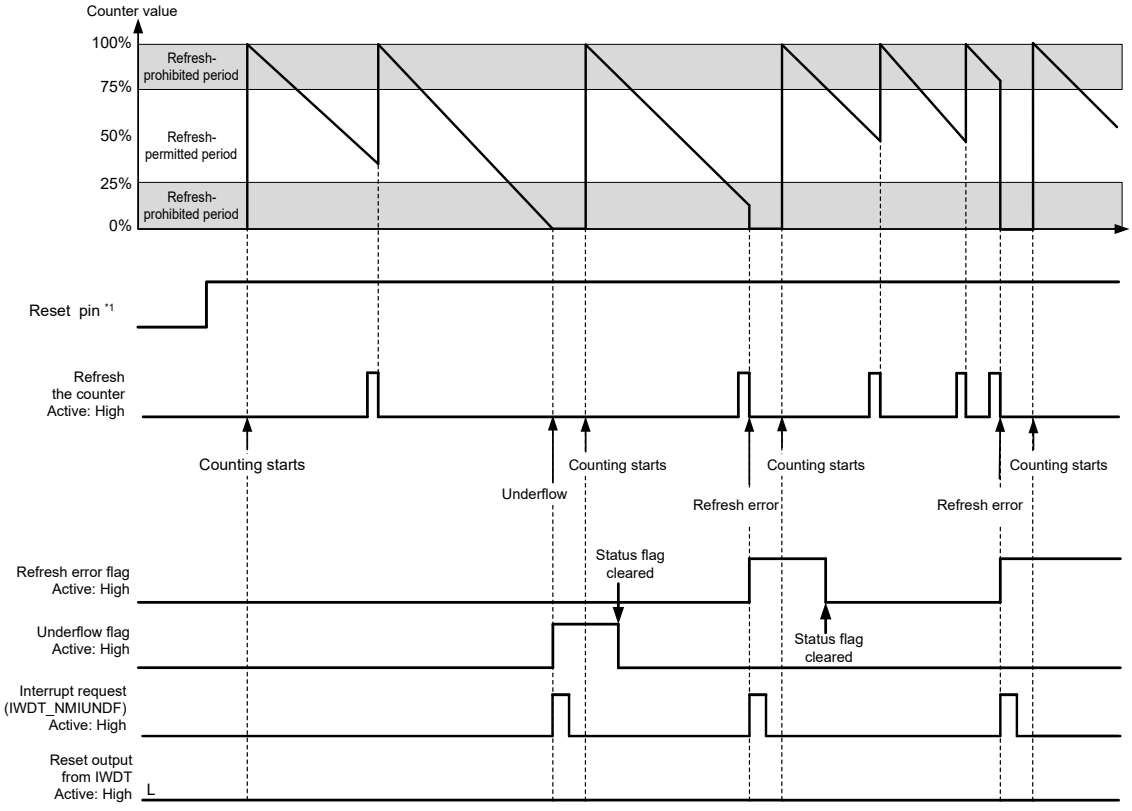
When the reset state is released, the counter automatically starts counting down from the value selected in the IWDT Timeout Period Select bits (OFS0.IWDTTOPS[1:0]).

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 22.3 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 22.3 Operation example in auto start mode

22.3.2 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

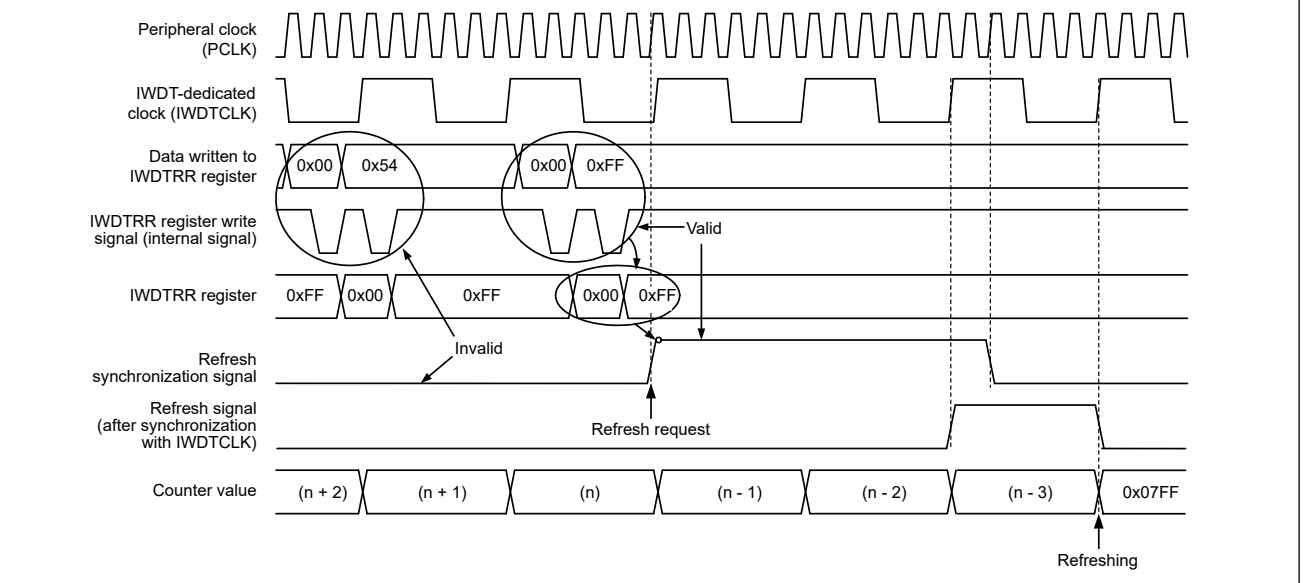
- 0x00 → 0x54 (a value other than 0xFF)
- 0x00 → 0xAA (0x00 and a value other than 0xFF) → 0xFF.

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 0x01FF, even if 0x00 is written to IWDTRR before 0x01FF is reached at (0x0202, for example), refreshing occurs if 0xFF is written to IWDTRR after the value of the IWDTSR.CNTVAL[13:0] bits reaches 0x01FF
- When the window end position is set to 0x01FF, refreshing occurs if 0x0203 (4 count cycles before 0x01FF) 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 22.4 shows the IWDT refresh-operation waveforms when PCLKB > IWDTCLK and the clock division ratio is IWDTCLK.



Note: PCLK = PCLKB

Figure 22.4 IWDT refresh operation waveforms when OFS0.IWDTCKS[3:0] = 0000b, OFS0.IWDTTOPS[1:0] = 11b

22.3.3 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 22.2.2. IWDTSR : IWDT Status Register.

22.3.4 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.

22.3.5 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 22.4 IWDT interrupt source

Name	Interrupt source	Interrupt to CPU	Start DTC
IWDT_NMIUNDF	• Down-counter underflow • Refresh error	Possible	Not possible

22.3.6 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 22.5 shows the processing for reading the IWDT counter value when PCLKB > IWDTCLK and the clock division ratio is IWDTCLK.

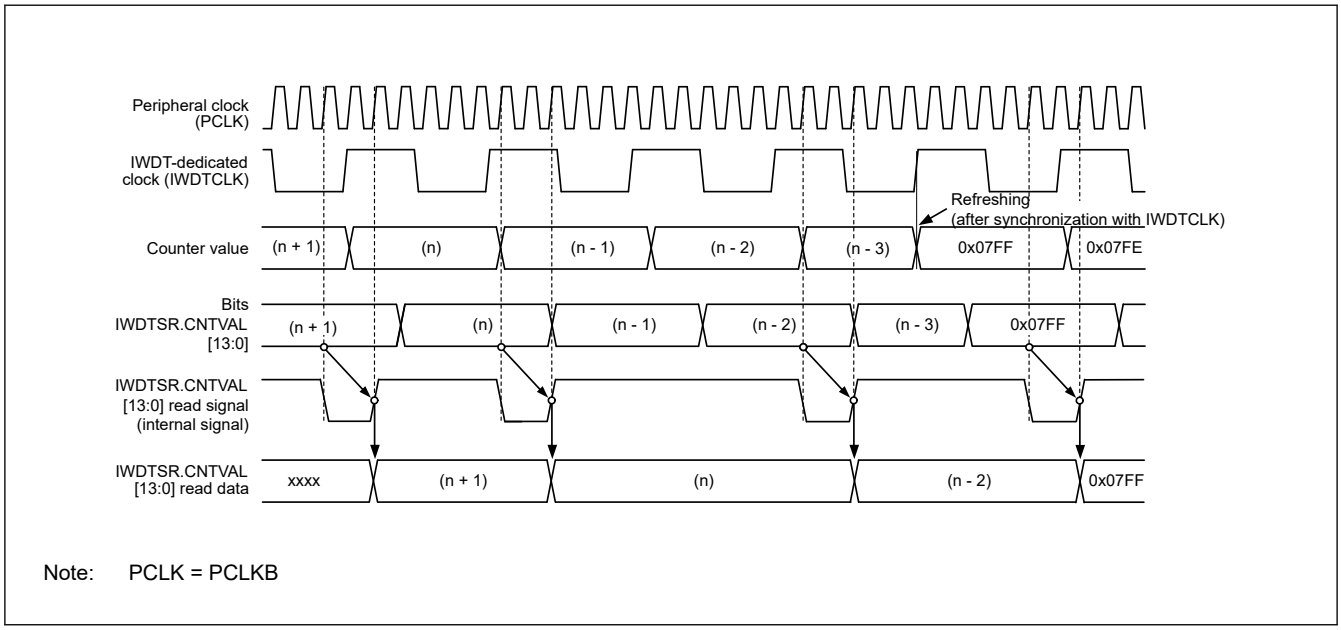


Figure 22.5 Processing for reading IWDT counter value when OFS0.IWDTCKS[3:0] = 0000b, OFS0.IWDTTOPS[1:0] = 11b

22.4 Usage Notes

22.4.1 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.

22.4.2 Clock Division Ratio Setting

Satisfy the frequency of the peripheral module clock (PCLKB) ≥ 4 × (the frequency of the count clock source after division).

22.4.3 Constraints on the ICU Event Link Setting Register n (IELSRn) Setting

Setting 0x03 to ICU Event Link Setting Register n (IELSRn.IELS[4:0]) is prohibited when enabling the IWDT reset assertion (OFS0.IWDTRSTIRQS = 1).

23. Serial Array Unit (SAU)

23.1 Overview

A single serial array unit has up to four serial channels. Each channel can achieve 3-wire serial (simplified SPI), UART, and simplified I^2C communication.

Function assignment of each channel supported by the MCU is as shown in Table 23.1 to Table 23.3.

Table 23.1 Function assignment for 16-pin products

Unit	Channel	Used as simplified SPI	Used as UART	Used as simplified I_2C
0	0	SPI00	UART0	IIC00
	1	—		—
	2	—	UART1	—
	3	—		—

Table 23.2 Function assignment for 24- and 32-pin products

Unit	Channel	Used as simplified SPI	Used as UART	Used as simplified I^2C
0	0	SPI00	UART0	IIC00
	1	—		—
	2	—	UART1	—
	3	SPI11		IIC11
1	0	SPI20	UART2 (supporting LIN-bus)	IIC20
	1	—		—

Table 23.3 Function assignment for 48-pin products

Unit	Channel	Used as simplified SPI	Used as UART	Used as simplified I^2C
0	0	SPI00	UART0	IIC00
	1	SPI01		IIC01
	2	SPI10	UART1	IIC10
	3	SPI11		IIC11
1	0	SPI20	UART2 (supporting LIN-bus)	IIC20
	1	SPI21		IIC21

When UART0 is used for channels 0 and 1 of the unit 0, SPI00 and SPI01 cannot be used, but SPI10, UART1, or IIC10 in channels 2 and 3 can be used.

Note: Most of the following descriptions in this section use the units and channels of the 48-pin products as an example.

Each serial interface supported by the MCU has the following features:

- Simplified SPI
- UART
- Simplified I^2C.

23.1.1 Simplified SPI

Data is transmitted or received in synchronization with the serial clock (SCK) output from the master.

3-wire serial communication is clocked communication performed by using three communication lines: one for the serial clock (SCK), one for transmitting serial data (SO), one for receiving serial data (SI).

For details about the settings, see section 23.5. Operation of Simplified SPI.

[Data transmission and reception]

- Data length of 7 or 8 bits
- Phase control of transmit and receive data
- MSB- or LSB-first selectable

[Clock control]

- Master or slave selection
- Phase control of I/O clock
- Setting of transfer period by prescaler and internal counter of each channel
- Maximum transfer rate *1
  During master communication: Max. PCLKB/2 (SPI00 only), Max. PCLKB/4
  During slave communication: Max. $f_{MCK}/6$

[Interrupt function]

- Transfer end interrupt or buffer empty interrupt

[Error detection flag]

- Overrun error

In addition, simplified SPIs of following channels support the Snooze mode. In the Snooze mode, data can be received without CPU processing upon detecting SCK input in the Software Standby mode. The Snooze mode is only available in the following simplified SPIs, which support asynchronous reception.

- SPI00 and SPI20

Note 1. Set up the transfer rate within a range satisfying the SCK cycle time (t_{KCY}). For details, see section 37, Electrical Characteristics.

Note: Use a general-purpose port pin to send a chip select signal when required.

23.1.2 UART

This is a start-stop synchronization communication function using two lines: serial data transmission (TxD) and serial data reception (RxD) lines. By using these two communication lines, each data frame, which consist of a start bit, data, parity bit, and stop bit, is transferred asynchronously (using the internal baud rate) between the microcontroller and the other communication party. Full-duplex UART communication can be performed by using a channel dedicated to transmission (even-numbered channel) and a channel dedicated to reception (odd-numbered channel). The LIN-bus can be implemented by using timer array unit with an external interrupt (IRQ2).

For details about the settings, see section 23.6. Operation of UART Communication .

[Data Transmission and reception]

- Data length of 7, 8, or 9 bits*1
- MSB or LSB first selectable
- Level setting of transmit and receive data and select of reverse
- Parity bit appending and parity check functions
- Stop bit appending

[Interrupt function]

- Transfer end interrupt and buffer empty interrupt
- Error interrupt in case of framing error, parity error, or overrun error

[Error detection flag]

- Framing error, parity error, or overrun error

In addition, UART reception of following channels supports the Snooze mode. In the Snooze mode, data can be received without CPU processing upon detecting RxD input in the Software Standby mode. The Snooze mode is only available in the following UARTs, which support the reception baud rate adjustment function.

- UART0 and UART2

The LIN-bus is accepted in UART2 (channels 0 and 1 of unit 1).

[LIN-bus function][*2]

- Wakeup signal detection
- Break field (BF) detection
- Sync field measurement, baud rate calculation

Note 1. Only the following UARTs support the 9-bit data length:
- 16-, 24-, and 32-pin products: UART0
- 48-pin products: UART0 and UART2

Note 2. Using the external interrupt (IRQ2) and timer array unit (channel 7).

23.1.3 Simplified I2C

This is a clocked communication function to communicate with two or more devices by using two lines: serial clock (SCL) and serial data (SDA). This simplified I2C is designed for single communication with a device such as EEPROM, flash memory, or A/D converter, and therefore, it functions only as a master.

Make sure by using software, as well as operating the control registers, that the AC specifications of the start and stop conditions are observed.

For details about the settings, see section 23.8. Operation of Simplified I2C Communication.

[Data transmission and reception]

- Master transmission, master reception (only master function with a single master)
- ACK output function[*1] and ACK detection function
- Data length of 8 bits
  (When an address is transmitted, the address is specified by the higher 7 bits, and the least significant bit is used for R/W control.)
- Manual generation of start condition and stop condition

[Interrupt function]

- Transfer end interrupt

[Error detection flag]

- ACK error, or overrun error

[Functions not supported by simplified I2C]

- Slave transmission, slave reception
- Arbitration loss detection function
- Clock stretch detection

Note 1. When receiving the last data, ACK is not output if 0 is written to the SOE bit (serial output enable register m (SOEm)) and serial communication data output is stopped. See (2) Processing flow for details.

Note: To use an I2C bus of full function, see section 24, I2C Bus Interface (IICA).

Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 11

23.2 Configuration of Serial Array Unit

The serial array unit includes the registers, and input and output pins shown in Table 23.4.

Table 23.4 Configuration of serial array

Item	Configuration
Shift register	8 or 9 bits*1
Buffer register	Lower 8 bits or 9 bits of serial data register mn (SDRmn)*1
Serial clock I/O	SCK00, SCK01, SCK10, SCK11, SCK20, SCK21 pins (for simplified SPI) SCL00, SCL01, SCL10, SCL11, SCL20, SCL21 pins (for simplified I^2C)
Serial data input	SI00, SI01, SI10, SI11, SI20, SI21 pins (for simplified SPI) RxD0, RxD1 pins (for UART), RxD2 pin (for UART supporting LIN-bus)
Serial data output	SO00, SO01, SO10, SO11, SO20, SO21 pins (for simplified SPI) TxD0, TxD1 pins (for UART), TxD2 pin (for UART supporting LIN-bus)
Serial data I/O	SDA00, SDA01, SDA10, SDA11, SDA20, SDA21 pins (for simplified I^2C)
Control registers	<Registers of unit setting block> ● Serial clock select register m (SPSm) ● Serial channel enable status register m (SEm) ● Serial channel start register m (SSm) ● Serial channel stop register m (STm) ● Serial output enable register m (SOEm) ● Serial output register m (SOm) ● Serial output level register m (SOLm) ● Serial standby control register m (SSCm) ● Input switch control register (ISC) ● SAU Noise filter enable register (SNFEN)
	<Registers of each channel> ● Serial data register mn (SDRmn) ● Serial mode register mn (SMRmn) ● Serial communication operation setting register mn (SCRmn) ● Serial status register mn (SSRmn) ● Serial flag clear trigger register mn (SIRmn)
	● UART loopback select register (ULBS)

Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: SPI number (p = 00, 01, 10, 11, 20, 21)
q: UART number (q = 0 to 2), r: IIC number (r = 00, 01, 10, 11, 20, 21), mn = 00 to 03, 10 to 11

Note 1. The number of bits used as the shift register and buffer register differs depending on the unit and channel.
- mn = 00, 01, 10, 11: lower 9 bits
- Other than above: lower 8 bits

Figure 23.1 shows the block diagram of serial array unit 0.

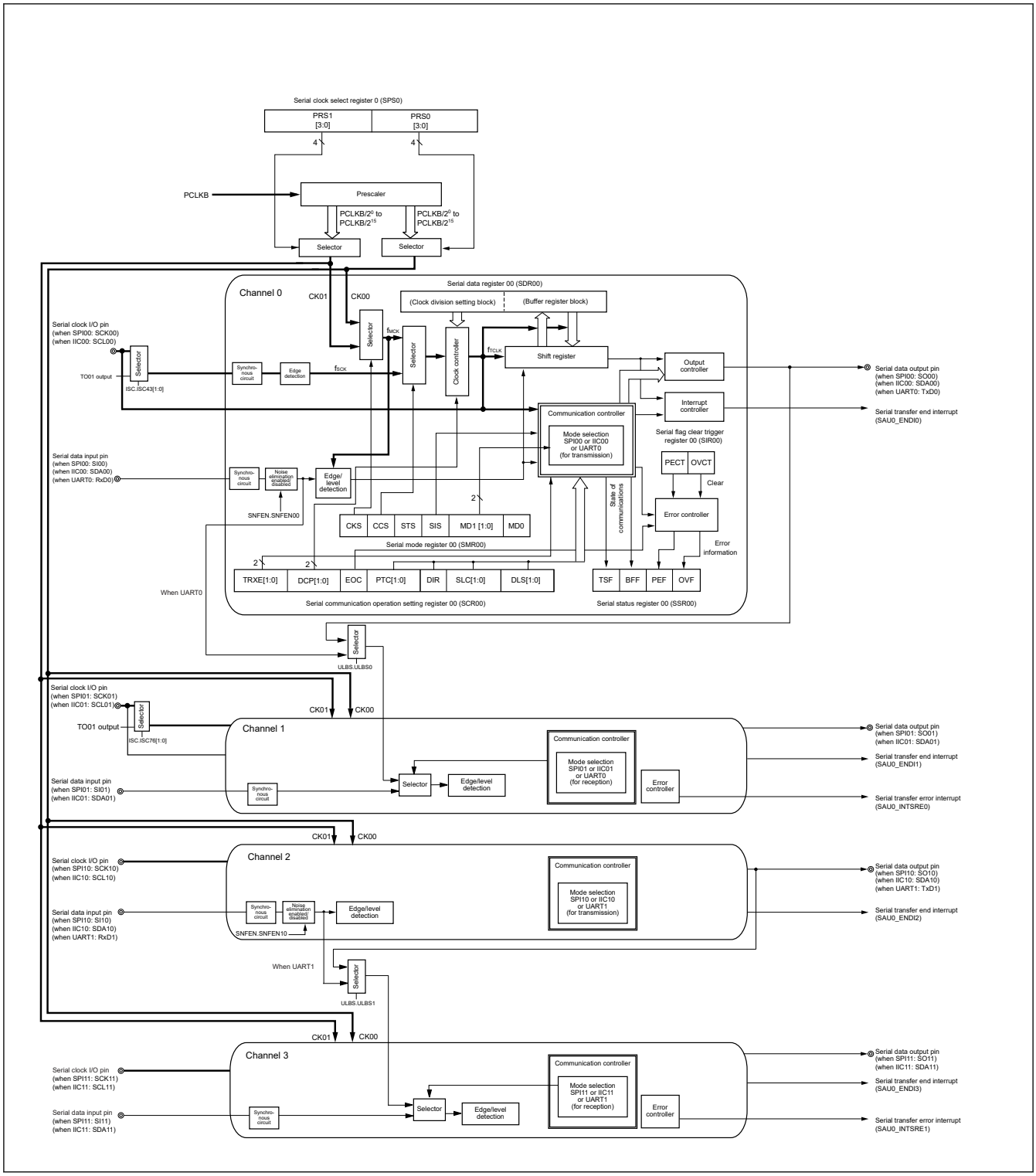


Figure 23.1 Block diagram of serial array unit 0

Figure 23.2 shows the block diagram of serial array unit 1.

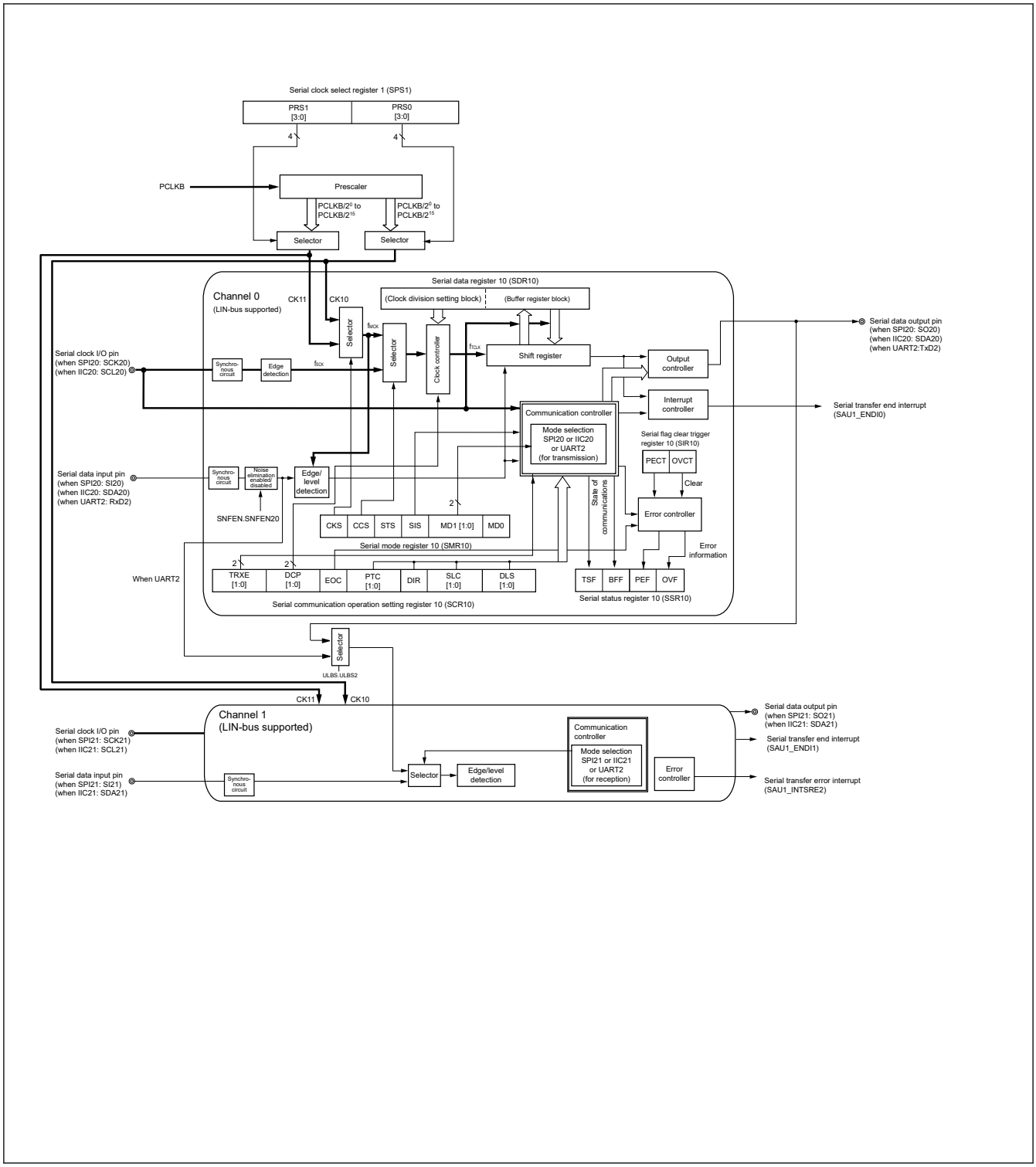


Figure 23.2 Block diagram of serial array unit 1

23.3 Register Descriptions

23.3.1 SPSm : Serial Clock Select Register m (m = 0, 1)

Base address: SAUm = 0x4009_4000 + 0x0100 × m

Offset address: 0x00A6[*1]

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	PRS1[3:0]				PRS0[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	PRS0[3:0]	Selection of operation clock (CKm0)[*1] 0x0: PCLKB 0x1: PCLKB/2 0x2: PCLKB/2^2 0x3: PCLKB/2^3 0x4: PCLKB/2^4 0x5: PCLKB/2^5 0x6: PCLKB/2^6 0x7: PCLKB/2^7 0x8: PCLKB/2^8 0x9: PCLKB/2^9 0xA: PCLKB/2^{10} 0xB: PCLKB/2^{11} 0xC: PCLKB/2^{12} 0xD: PCLKB/2^{13} 0xE: PCLKB/2^{14} 0xF: PCLKB/2^{15}	R/W
7:4	PRS1[3:0]	Selection of operation clock (CKm1)[*1] 0x0: PCLKB 0x1: PCLKB/2 0x2: PCLKB/2^2 0x3: PCLKB/2^3 0x4: PCLKB/2^4 0x5: PCLKB/2^5 0x6: PCLKB/2^6 0x7: PCLKB/2^7 0x8: PCLKB/2^8 0x9: PCLKB/2^9 0xA: PCLKB/2^{10} 0xB: PCLKB/2^{11} 0xC: PCLKB/2^{12} 0xD: PCLKB/2^{13} 0xE: PCLKB/2^{14} 0xF: PCLKB/2^{15}	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. When changing the clock selected for PCLKB, do so after having stopped (serial channel stop register m (STm) = 0x000F) the operation of the serial array unit (SAU).

The SPSm register is used to select two types of operation clocks (CKm0, CKm1) that are commonly supplied to each channel. CKm1 is selected by the PRS1[3:0] bits, and CKm0 is selected by the PRS0[3:0] bits. Rewriting the SPSm register is prohibited when the register is in operation (when SEm.SE[n] = 1).

The input sources that can be selected with the PRS0[3:0] and PRS1[3:0] bits are shown in Table 23.5.

Table 23.5 Selection of operation clock (PRSk[3:0](k = 0, 1))

PRSk[3:0]		Selection of operation clock (CKmk) (k = 0, 1)					
		PCLKB = 2 MHz	PCLKB = 5 MHz	PCLKB = 10 MHz	PCLKB = 20 MHz	PCLKB = 32 MHz	PCLKB = 48 MHz
0x0	PCLKB	2 MHz	5 MHz	10 MHz	20 MHz	32 MHz	Setting prohibited
0x1	PCLKB/2	1 MHz	2.5 MHz	5 MHz	10 MHz	16 MHz	24 MHz
0x2	PCLKB/2^2	500 kHz	1.25 MHz	2.5 MHz	5 MHz	8 MHz	12 MHz
0x3	PCLKB/2^3	250 kHz	625 kHz	1.25 MHz	2.5 MHz	4 MHz	6 MHz
0x4	PCLKB/2^4	125 kHz	313 kHz	625 kHz	1.25 MHz	2 MHz	3 MHz
0x5	PCLKB/2^5	62.5 kHz	156 kHz	313 kHz	625 kHz	1 MHz	1.5 MHz
0x6	PCLKB/2^6	31.3 kHz	78.1 kHz	156 kHz	313 kHz	500 kHz	750 kHz
0x7	PCLKB/2^7	15.6 kHz	39.1 kHz	78.1 kHz	156 kHz	250 kHz	375 kHz
0x8	PCLKB/2^8	7.81 kHz	19.5 kHz	39.1 kHz	78.1 kHz	125 kHz	187 kHz
0x9	PCLKB/2^9	3.91 kHz	9.77 kHz	19.5 kHz	39.1 kHz	62.5 kHz	93.8 kHz
0xA	PCLKB/2^{10}	1.95 kHz	4.88 kHz	9.77 kHz	19.5 kHz	31.3 kHz	46.9 kHz
0xB	PCLKB/2^{11}	977 Hz	2.44 kHz	4.88 kHz	9.77 kHz	15.6 kHz	23.4 kHz
0xC	PCLKB/2^{12}	488 Hz	1.22 kHz	2.44 kHz	4.88 kHz	7.81 kHz	11.7 kHz
0xD	PCLKB/2^{13}	244 Hz	610 Hz	1.22 kHz	2.44 kHz	3.91 kHz	5.86 kHz
0xE	PCLKB/2^{14}	122 Hz	305 Hz	610 Hz	1.22 kHz	1.95 kHz	2.93 kHz
0xF	PCLKB/2^{15}	61.0 Hz	153 Hz	305 Hz	610 Hz	977 Hz	1.46 kHz

23.3.2 SMRmn : Serial Mode Register mn (mn = 00, 02, 10)

Base address: SAUm = 0x4009_4000 + 0x0100 × m

Offset address: 0x0090 + 0x02 × n

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CKS	CCS	—	—	—	—	—	—	—	—	—	—	—	MD1[1:0]		MD0
Value after reset:	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MD0	Selection of channel n interrupt source 0: Transfer end interrupt 1: Buffer empty interrupt (occurs when data is transferred from the SDRmn register to the shift register.)	R/W
2:1	MD1[1:0]	Setting of channel n operation mode 0 0: Simplified SPI mode 0 1: UART mode 1 0: Simplified I^2C mode 1 1: Setting prohibited	R/W
4:3	—	These bits are read as 0. The write value should be 0.	R/W
5	—	This bit is read as 1. The write value should be 1.	R/W
13:6	—	These bits are read as 0. The write value should be 0.	R/W
14	CCS	Selection of transfer clock (f_{TCLK}) of channel n 0: Divided operation clock f_{MCK} specified by the CKS bit 1: Clock input f_{SCK} from the SCKp pin (slave transfer in simplified SPI mode)	R/W

Bit	Symbol	Function	R/W
15	CKS	Selection of operation clock (f_{MCK}) of channel n 0: Operation clock CKm0 set by the SPSm register 1: Operation clock CKm1 set by the SPSm register	R/W

The SMRmn register is used to set an operation mode of channel n. It is also used to select an operation clock (f_{MCK}), specify whether the serial clock (f_{SCK}) may be input or not, set a start trigger, the operating mode (as simplified SPI, UART or simplified I^2C), and an interrupt source. This register is also used to invert the level of the receive data only in the UART mode.

Rewriting the SMRmn register is prohibited when the register is in operation (when SEm.SE[n] = 1). However, the MD0 bit can be rewritten during operation.

MD0 bit (Selection of channel n interrupt source)

For continuous transmission, set this bit to 1 and write the next transmit data when SDRmn data has run out.

MD1[1:0] bits (Setting of channel n operation mode)

The MD1[1:0] bits are used for setting of channel n operation mode.

CCS bit (Selection of transfer clock (f_{TCLK}) of channel n)

Transfer clock (f_{TCLK}) is used for the shift register, communication controller, output controller, interrupt controller, and error controller. When CCS = 0, the division ratio of operation clock (f_{MCK}) is set by the STCLK[6:0] bits of the SDRmn register.

CKS bit (Selection of operation clock (f_{MCK}) of channel n)

Operation clock (f_{MCK}) is used by the edge detector. In addition, depending on the setting of the CCS bit and the STCLK[6:0] bits of the SDRmn register, a transfer clock (f_{TCLK}) is generated.

23.3.3 SMRmn : Serial Mode Register mn (mn = 01, 03, 11)

Base address: SAUm = 0x4009_4000 + 0x0100 × m

Offset address: 0x0090 + 0x02 × n

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CKS	CCS	—	—	—	—	—	STS	—	SIS0	—	—	—	MD1[1:0]		MD0
Value after reset:	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MD0	Selection of channel n interrupt source 0: Transfer end interrupt 1: Buffer empty interrupt (occurs when data is transferred from the SDRmn register to the shift register.)	R/W
2:1	MD1[1:0]	Setting of channel n operation mode 0 0: Simplified SPI mode 0 1: UART mode 1 0: Simplified I^2C mode 1 1: Setting prohibited	R/W
4:3	—	These bits are read as 0. The write value should be 0.	R/W
5	—	This bit is read as 1. The write value should be 1.	R/W
6	SIS0	Controls inversion of level of channel n receive data in UART mode 0: Falling edge is detected as the start bit. The input communication data is captured as is. 1: Rising edge is detected as the start bit. The input communication data is inverted and captured.	R/W

Bit	Symbol	Function	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
8	STS	Selection of start trigger source 0: Only software trigger is valid (selected for simplified SPI, UART transmission, and simplified I^2C) 1: Valid edge of the RxDq pin (selected for UART reception)	R/W
13:9	—	These bits are read as 0. The write value should be 0.	R/W
14	CCS	Selection of transfer clock (f_{TCLK}) of channel n 0: Divided operation clock f_{MCK} specified by the CKS bit 1: Clock input f_{SCK} from the SCKp pin (slave transfer in simplified SPI mode)	R/W
15	CKS	Selection of operation clock (f_{MCK}) of channel n 0: Operation clock CKm0 set by the SPSm register 1: Operation clock CKm1 set by the SPSm register	R/W

The SMRmn register is used to set an operation mode of channel n. It is also used to select an operation clock (f_{MCK}), specify whether the serial clock (f_{SCK}) may be input or not, set a start trigger, the operating mode (as simplified SPI, UART, or simplified I^2C), and an interrupt source. This register is also used to invert the level of the receive data only in the UART mode.

Rewriting the SMRmn register is prohibited when the register is in operation (when SEm.SE[n] = 1). However, the MD0 bit can be rewritten during operation.

MD0 bit (Selection of channel n interrupt source)

For continuous transmission, set this bit to 1 and write the next transmit data when SDRmn data has run out.

MD1[1:0] bits (Setting of channel n operation mode)

The MD1[1:0] bits are used for setting of channel n operation mode.

SIS0 bit (Controls inversion of level of channel n receive data in UART mode)

The SIS0 bit is used for control inversion of the level of channel n receive data in UART mode.

STS bit (Selection of start trigger source)

Transfer is started when the above source is satisfied after 1 is set to the SSm register.

CCS bit (Selection of transfer clock (f_{TCLK}) of channel n)

Transfer clock (f_{TCLK}) is used for the shift register, communication controller, output controller, interrupt controller, and error controller. When CCS = 0, the division ratio of operation clock (f_{MCK}) is set by the STCLK[6:0] bits of the SDRmn register.

CKS bit (Selection of operation clock (f_{MCK}) of channel n)

Operation clock (f_{MCK}) is used by the edge detector. In addition, depending on the setting of the CCS bit and the STCLK[6:0] bits of the SDRmn register, a transfer clock (f_{TCLK}) is generated.

23.3.4 SCRm0 : Serial Communication Operation Setting Register m0 (m = 0, 1)

Base address: SAUm = 0x4009_4000 + 0x0100 × m

Offset address: 0x0098

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TRXE[1:0]		DCP[1:0]		—	—	PTC[1:0]		DIR	—	SLC[1:0]		—	—	DLS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	1

Bit	Symbol	Function	R/W
1:0	DLS[1:0]	Setting of data length in simplified SPI and UART modes 0 0: Setting prohibited 0 1: 9-bit data length (stored in the DAT[8:0] bits of the SDRm0 register) (settable in UART mode only) 1 0: 7-bit data length (stored in the DAT[6:0] bits of the SDRm0 register) 1 1: 8-bit data length (stored in the DAT[7:0] bits of the SDRm0 register)	R/W
2	—	This bit is read as 1. The write value should be 1.	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
5:4	SLC[1:0]	Setting of stop bit in UART mode 0 0: No stop bit 0 1: Stop bit length = 1 bit 1 0: Stop bit length = 2 bits 1 1: Setting prohibited	R/W
6	—	This bit is read as 0. The write value should be 0.	R/W
7	DIR	Selection of data transfer sequence in simplified SPI and UART modes 0: Inputs or outputs data with MSB first 1: Inputs or outputs data with LSB first	R/W
9:8	PTC[1:0]	Setting of parity bit in UART mode 0 0: Transmission: Does not output the parity bit Reception: Receives without parity 0 1: Transmission: Outputs 0 parity[*1] Reception: No parity determination 1 0: Transmission: Outputs even parity Reception: Determines as even parity 1 1: Transmission: Outputs odd parity Reception: Determines as odd parity	R/W
11:10	—	These bits are read as 0. The write value should be 0.	R/W
13:12	DCP[1:0]	Selection of data and clock phase in simplified SPI mode 0 0: Type1 (SCK: inverted, Input timing: rising edge) 0 1: Type2 (SCK: non-inverted, Input timing: falling edge) 1 0: Type3 (SCK: inverted, Input timing: falling edge) 1 1: Type4 (SCK: non-inverted, Input timing: rising edge)	R/W
15:14	TRXE[1:0]	Setting of channel 0 operation mode 0 0: Disable communication 0 1: Reception only 1 0: Transmission only 1 1: Transmission and reception	R/W

Note 1. 0 is always added regardless of the data contents.

The SCRm0 register is used to set a data transmission and reception mode, phase of data and clock, whether an error signal is to be masked or not, parity bit, start bit, stop bit, and data length.

Rewriting the SCRm0 register is prohibited when the register is in operation (when SEm.SE[0] = 1).

DLS[1:0] bits (Setting of data length in simplified SPI and UART modes)

Be sure to set DLS[1:0] = 11b in the simplified I^2C mode.

SLC[1:0] bits (Setting of stop bit in UART mode)

When the transfer end interrupt is selected, the interrupt is generated when all stop bits have been completely transferred.

Set 1 bit (SLC[1:0]= 01b) during UART reception and in the simplified I^2C mode. Set no stop bit (SLC[1:0] = 00b) in the simplified SPI mode.

Set 1 bit (SLC[1:0] = 01b) or 2 bits (SLC[1:0] = 10b) during UART transmission.

DIR bit (Selection of data transfer sequence in simplified SPI and UART modes)

Be sure to clear DIR = 0 in the simplified I^2C mode.

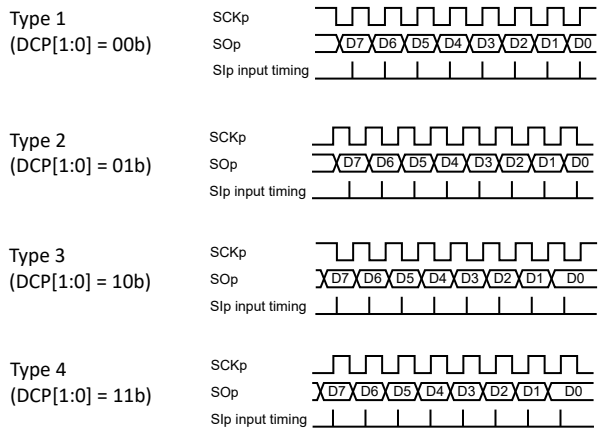
PTC[1:0] bits (Setting of parity bit in UART mode)

Be sure to set PTC[1:0] = 00b in the simplified SPI mode and simplified I^2C mode

DCP[1:0] bits (Selection of data and clock phase in simplified SPI mode)

Be sure to set DCP[1:0] = 00b in the UART mode and simplified I^2C mode.

Figure 23.3 shows the data and clock phase in simplified SPI mode.



Note: p: SPI number (p = 00, 01, 10, 11, 20, 21)

Figure 23.3 Data and clock phase in simplified SPI mode

TRXE[1:0] bits (Setting of channel 0 operation mode)

The TRXE[1:0] bits are used for setting of channel 0 operation mode.

23.3.5 SCRm1 : Serial Communication Operation Setting Register m1 (m = 0, 1)

Base address: SAUm = 0x4009_4000 + 0x0100 × m

Offset address: 0x009A

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TRXE[1:0]		DCP[1:0]		—	EOC	PTC[1:0]		DIR	—	—	SLC	—	—	DLS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	1

Bit	Symbol	Function	R/W
1:0	DLS[1:0]	Setting of data length in simplified SPI and UART modes 0 0: Setting prohibited 0 1: 9-bit data length (stored in the DAT[8:0] bits of the SDRm1 register) (settable in UART mode only) 1 0: 7-bit data length (stored in the DAT[6:0] bits of the SDRm1 register) 1 1: 8-bit data length (stored in the DAT[7:0] bits of the SDRm1 register)	R/W
2	—	This bit is read as 1. The write value should be 1.	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	SLC	Setting of stop bit in UART mode 0: No stop bit 1: Stop bit length = 1 bit	R/W
6:5	—	This bit is read as 0. The write value should be 0.	R/W
7	DIR	Selection of data transfer sequence in simplified SPI and UART modes 0: Inputs or outputs data with MSB first 1: Inputs or outputs data with LSB first	R/W

Bit	Symbol	Function	R/W
9:8	PTC[1:0]	Setting of parity bit in UART mode 0 0: Transmission: Does not output the parity bit Reception: Receives without parity 0 1: Transmission: Outputs 0 parity*1 Reception: No parity judgment 1 0: Transmission: Outputs even parity Reception: Determines as even parity 1 1: Transmission: Outputs odd parity Reception: Determines as odd parity	R/W
10	EOC	Mask control of error interrupt signal SAU0_INTSRE0 (m = 0), SAU1_INTSRE2 (m = 1) 0: Disables generation of error interrupt SAU0_INTSRE0 (m = 0), SAU1_INTSRE2 (m = 1) (SAUm_ENDI1 is generated) 1: Enables generation of error interrupt SAU0_INTSRE0 (m = 0), SAU1_INTSRE2 (m = 1) (SAUm_ENDI1 is not generated if an error occurs)	R/W
11	—	This bit is read as 1. The write value should be 1.	R/W
13:12	DCP[1:0]	Selection of data and clock phase in simplified SPI mode 0 0: Type1 (SCK: inverted, Input timing: rising edge) 0 1: Type2 (SCK: non-inverted, Input timing: falling edge) 1 0: Type3 (SCK: inverted, Input timing: falling edge) 1 1: Type4 (SCK: non-inverted, Input timing: rising edge)	R/W
15:14	TRXE[1:0]	Setting of channel 1 operation mode 0 0: Disable communication 0 1: Reception only 1 0: Transmission only 1 1: Transmission and reception	R/W

Note 1. 0 is always added regardless of the data contents.

The SCRm1 register is used to set a data transmission and reception mode, phase of data and clock, whether an error signal is to be masked or not, parity bit, start bit, stop bit, and data length.

Rewriting the SCRm1 register is prohibited when the register is in operation (when SEm.SE[1] = 1).

DLS[1:0] bit (Setting of data length in simplified SPI and UART modes)

Be sure to set DLS[1:0] = 11b in the simplified I^2C mode.

SLC bit (Setting of stop bit in UART mode)

When the transfer end interrupt is selected, the interrupt is generated when all stop bits have been completely transferred.

Set 1 bit (SLC = 1) during UART reception and in the simplified I^2C mode. Set no stop bit (SLC = 0) in the simplified SPI mode.

Set 1 bit (SLC = 0) during UART transmission.

DIR bit (Selection of data transfer sequence in simplified SPI and UART modes)

Be sure to clear DIR = 0 in the simplified I^2C mode.

PTC[1:0] bit (Setting of parity bit in UART mode)

Be sure to set PTC[1:0] = 00b in the simplified SPI mode and simplified I^2C mode.

EOC bit (Mask control of error interrupt signal SAU0_INTSRE0 (m = 0), SAU1_INTSRE2 (m = 1))

Set EOC = 0 in the simplified SPI mode, simplified I^2C mode, and during UART transmission.*1

DCP[1:0] bit (Selection of data and clock phase in simplified SPI mode)

See Figure 23.3.

Be sure to set DCP[1:0] = 00b in the UART mode and simplified I^2C mode.

TRXE[1:0] bit (Setting of channel 1 operation mode)

The TRXE[1:0] bits are used for setting of channel 1 operation mode.

Note 1. When using SPI01 not with SCR01.EOC = 0, error interrupt SAU0_INTSRE0 may be generated.
When using SPI21 not with SCR11.EOC = 0, error interrupt SAU1_INTSRE2 may be generated.

23.3.6 SCR02 : Serial Communication Operation Setting Register 02

Base address: SAU0 = 0x4009_4000

Offset address: 0x009C

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TRXE[1:0]		DCP[1:0]		—	—	PTC[1:0]		DIR	—	SLC[1:0]		—	—	—	DLS
Value after reset:	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	1

Bit	Symbol	Function	R/W
0	DLS	Setting of data length in simplified SPI and UART modes 0: 7-bit data length (stored in the DAT[6:0] bits of the SDR02 register) 1: 8-bit data length (stored in the DAT[7:0] bits of the SDR02 register)	R/W
2:1	—	These bits are read as 1. The write value should be 1.	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
5:4	SLC[1:0]	Setting of stop bit in UART mode 0 0: No stop bit 0 1: Stop bit length = 1 bit 1 0: Stop bit length = 2 bits 1 1: Setting prohibited	R/W
6	—	This bit is read as 0. The write value should be 0.	R/W
7	DIR	Selection of data transfer sequence in simplified SPI and UART modes 0: Inputs or outputs data with MSB first 1: Inputs or outputs data with LSB first	R/W
9:8	PTC[1:0]	Setting of parity bit in UART mode 0 0: Transmission: Does not output the parity bit Reception: Receives without parity 0 1: Transmission: Outputs 0 parity*1 Reception: No parity judgment 1 0: Transmission: Outputs even parity Reception: Determines as even parity 1 1: Transmission: Outputs odd parity Reception: Determines as odd parity	R/W
11:10	—	These bits are read as 0. The write value should be 0.	R/W
13:12	DCP[1:0]	Selection of data and clock phase in simplified SPI mode 0 0: Type1 (SCK: inverted, Input timing: rising edge) 0 1: Type2 (SCK: non-inverted, Input timing: falling edge) 1 0: Type3 (SCK: inverted, Input timing: falling edge) 1 1: Type4 (SCK: non-inverted, Input timing: rising edge)	R/W
15:14	TRXE[1:0]	Setting of channel 2 operation mode 0 0: Disables communication 0 1: Reception only 1 0: Transmission only 1 1: Transmission and reception	R/W

Note 1. 0 is always added regardless of the data contents.

The SCR02 register is used to set a data transmission and reception mode, phase of data and clock, whether an error signal is to be masked or not, parity bit, start bit, stop bit, and data length.

Rewriting the SCR02 register is prohibited when the register is in operation (when SE0.SE[2] = 1).

DLS bit (Setting of data length in simplified SPI and UART modes)

Be sure to set DLS = 1 in the simplified I^2C mode.

SLC[1:0] bits (Setting of stop bit in UART mode)

When the transfer end interrupt is selected, the interrupt is generated when all stop bits have been completely transferred.

Set 1 bit (SLC[1:0] = 01b) during UART reception and in the simplified I^2C mode. Set no stop bit (SLC[1:0] = 00b) in the simplified SPI mode.

Set 1 bit (SLC[1:0] = 01b) or 2 bits (SLC[1:0] = 10b) during UART transmission.

DIR bit (Selection of data transfer sequence in simplified SPI and UART modes)

Be sure to clear DIR = 0 in the simplified I^2C mode.

PTC[1:0] bits (Setting of parity bit in UART mode)

Be sure to set PTC[1:0] = 00b in the simplified SPI mode and simplified I^2C mode.

DCP[1:0] bits (Selection of data and clock phase in simplified SPI mode)

See Figure 23.3.

Be sure to set DCP[1:0] = 00b in the UART mode and simplified I^2C mode.

TRXE[1:0] bits (Setting of channel 2 operation mode)

The TRXE[1:0] bits are used for setting of channel 2 operation mode.

23.3.7 SCR03 : Serial Communication Operation Setting Register 03

Base address: SAU0 = 0x4009_4000

Offset address: 0x009E

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TRXE[1:0]		DCP[1:0]		—	EOC	PTC[1:0]		DIR	—	—	SLC	—	—	—	DLS
Value after reset:	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	1

Bit	Symbol	Function	R/W
0	DLS	Setting of data length in simplified SPI and UART modes 0: 7-bit data length (stored in the DAT[6:0] bits of the SDR03 register) 1: 8-bit data length (stored in the DAT[7:0] bits of the SDR03 register)	R/W
2:1	—	These bits are read as 1. The write value should be 1.	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	SLC	Setting of stop bit in UART mode 0: No stop bit 1: Stop bit length = 1 bit	R/W
6:5	—	These bits are read as 0. The write value should be 0.	R/W
7	DIR	Selection of data transfer sequence in simplified SPI and UART modes 0: Inputs or outputs data with MSB first 1: Inputs or outputs data with LSB first	R/W
9:8	PTC[1:0]	Setting of parity bit in UART mode 0 0: Transmission: Does not output the parity bit Reception: Receives without parity 0 1: Transmission: Outputs 0 parity*1 Reception: No parity determination 1 0: Transmission: Outputs even parity Reception: Determines as even parity 1 1: Transmission: Outputs odd parity Reception: Determines as odd parity	R/W

Bit	Symbol	Function	R/W
10	EOC	Mask control of error interrupt signal SAU0_INTSRE1 0: Disables generation of error interrupt SAU0_INTSRE1 (SAU0_ENDI3 is generated) 1: Enables generation of error interrupt SAU0_INTSRE1 (SAU0_ENDI3 is not generated if an error occurs)	R/W
11	—	This bit is read as 0. The write value should be 0.	R/W
13:12	DCP[1:0]	Selection of data and clock phase in simplified SPI mode 0 0: Type1 (SCK: inverted, Input timing: rising edge) 0 1: Type2 (SCK: non-inverted, Input timing: falling edge) 1 0: Type3 (SCK: inverted, Input timing: falling edge) 1 1: Type4 (SCK: non-inverted, Input timing: rising edge)	R/W
15:14	TRXE[1:0]	Setting of operation mode of channel 3 0 0: Disable communication 0 1: Reception only 1 0: Transmission only 1 1: Transmission and reception	R/W

Note 1. 0 is always added regardless of the data contents.

The SCR03 register is used to set a data transmission and reception mode, phase of data and clock, whether an error signal is to be masked or not, parity bit, start bit, stop bit, and data length.

Rewriting the SCR03 register is prohibited when the register is in operation (when SE0.SE[3] = 1).

DLS bit (Setting of data length in simplified SPI and UART modes)

Be sure to set DLS = 1 in the simplified I^2C mode.

SLC bit (Setting of stop bit in UART mode)

When the transfer end interrupt is selected, the interrupt is generated when all stop bits have been completely transferred.

Set 1 bit (SLC = 1) during UART reception and in the simplified I^2C mode. Set no stop bit (SLC = 0) in the simplified SPI mode.

Set 1 bit (SLC = 0) during UART transmission.

DIR bit (Selection of data transfer sequence in simplified SPI and UART modes)

Be sure to clear DIR = 0 in the simplified I^2C mode.

PTC[1:0] bits (Setting of parity bit in UART mode)

Be sure to set PTC[1:0] = 00b in the simplified SPI mode and simplified I^2C mode.

EOC bit (Mask control of error interrupt signal SAU0_INTSRE1)

Set EOC = 0 in the simplified SPI mode, simplified I^2C mode, and during UART transmission.*1

DCP[1:0] bits (Selection of data and clock phase in simplified SPI mode)

See Figure 23.3.

Be sure to set DCP[1:0] = 00b in the UART mode and simplified I^2C mode.

TRXE[1:0] bits (Setting of operation mode of channel 3)

The TRXE[1:0] bits are used for setting of channel 3 operation mode.

Note 1. When using SPI11 not with EOC = 0, error interrupt SAU0_INTSRE1 may be generated.

23.3.8 SDRmn : Serial Data Register mn (mn = 00, 01, 02, 03, 10, 11)

Base address: SAUm = 0x4009_4000 + 0x0100 × m

Offset address: 0x0000 + 0x02 × n

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	STCLK[6:0]							DAT[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	DAT[8:0]	Data buffer for transmit and receive	R/W*1
15:9	STCLK[6:0]*2 *3	Transfer clock setting by dividing the operation clock 0x00: f_{MCK} /2 0x01: f_{MCK} /4 0x02: f_{MCK} /6 0x03: f_{MCK} /8 ⋮ ⋮ 0x7C: f_{MCK} /250 0x7D: f_{MCK} /252 0x7E: f_{MCK} /254 0x7F: f_{MCK} /256	R/W

Note 1. The SDR02.DAT[8] bit and the SDR03.DAT[8] bit are reserved bits. These bits are read as 0. The write value should be 0.
Note 2. Setting STCLK[6:0] = 0x00 or 0x01 is prohibited when UART is used.
Note 3. Setting STCLK[6:0] = 0x00 is prohibited when simplified I^2C is used. Set STCLK[6:0] = 0x01 or greater.

The SDRmn register is the transmit and receive data register (16 bits) of unit m and channel n.

The DAT[8:0] bits of SDR00, SDR01, SDR10, and SDR11 or the DAT[7:0] bits of SDR02 and SDR03 registers, function as a transmit and receive buffer register, and the STCLK[6:0] bits are used as a register that sets the division ratio of the operation clock (f_{MCK}).

If the CCS bit of serial mode register mn (SMRmn) is cleared to 0, the clock set by dividing the operation clock by the STCLK[6:0] bits is used as the transfer clock.

If the CCS bit of serial mode register mn (SMRmn) is set to 1, set the STCLK[6:0] bits of SDR00, SDR01, SDR10, and SDR11 to 0x00. The input clock f_{SCK} (slave transfer in simplified SPI mode) from the SCKp pin is used as the transfer clock.

The DAT[7:0] or DAT[8:0] bits function as a transmit and receive buffer register. During reception, the parallel data converted by the shift register is stored in the DAT[7:0] or DAT[8:0] bits, and during transmission, the data to be transmitted to the shift register is set to the DAT[7:0] or DAT[8:0] bits.

The data stored in the DAT[7:0] or DAT[8:0] bits is as follows, depending on the setting of the DLS[1:0] bits of serial communication operation setting register mn (SCRmn), regardless of the output sequence of the data.

- 7-bit data length (stored in the DAT[6:0] bits)
- 8-bit data length (stored in the DAT[7:0] bits)
- 9-bit data length (stored in the DAT[8:0] bits)*1

The SDRmn register can be read or written in 16-bit access. However, the STCLK[6:0] bits can only be written or read when the operation is stopped (SEm.SE[n] = 0). During operation (SEm.SE[n] = 1), a value is written only to the DAT[7:0] or DAT[8:0] bits. When the SDRmn register is read during operation, the STCLK[6:0] bits are always read as 0.

Note 1. Only the following UARTs support the 9-bit data length:
- 16-, 24-, and 32-pin products: UART0
- 48-pin products: UART0 and UART2

Note: p: SPI number (p = 00, 01, 10, 11, 20, 21)

23.3.9 SIRmn : Serial Flag Clear Trigger Register mn (mn = 00, 02, 10)

Base address: SAUm = 0x4009_4000 + 0x0100 × m

Offset address: 0x0088 + 0x02 × n

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PECT	OVCT
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	OVCT	Clear trigger of overrun error flag of channel n 0: Not cleared 1: Clears the OVF flag of the SSRmn register to 0	R/W
1	PECT	Clear trigger of parity error flag of channel n 0: Not cleared 1: Clears the PEF flag of the SSRmn register to 0.	R/W
15:2	—	These bits are read as 0. The write value should be 0.	R/W

The SIRmn register is a trigger register that is used to clear each error flag of channel n.

When each bit (PECT, OVCT) is set to 1, the corresponding flag (PEF, OVF) of serial status register mn (SSRmn) is cleared to 0. Because the SIRmn register is a trigger register, it is cleared immediately when the corresponding bit of the SSRmn register is cleared. When the SIRmn register is read, 0x0000 is always read.

23.3.10 SIRmn : Serial Flag Clear Trigger Register mn (mn = 01, 03, 11)

Base address: SAUm = 0x4009_4000 + 0x0100 × m

Offset address: 0x0088 + 0x02 × n

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	FECT	PECT	OVCT
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	OVCT	Clear trigger of overrun error flag of channel n 0: Not cleared 1: Clears the OVF flag of the SSRmn register to 0	R/W
1	PECT	Clear trigger of parity error flag of channel n 0: Not cleared 1: Clears the PEF flag of the SSRmn register to 0	R/W
2	FECT	Clear trigger of framing error flag of channel n 0: Not cleared 1: Clears the FEF flag of the SSRmn register to 0	R/W
15:3	—	These bits are read as 0. The write value should be 0.	R/W

The SIRmn register is a trigger register that is used to clear each error flag of channel n.

When each bit (FECT, PECT, OVCT) is set to 1, the corresponding flag (FEF, PEF, OVF) of serial status register mn (SSRmn) is cleared to 0. Because the SIRmn register is a trigger register, it is cleared immediately when the corresponding bit of the SSRmn register is cleared. When the SIRmn register is read, 0x0000 is always read.

23.3.11 SSRmn : Serial Status Register mn (mn = 00, 02, 10)

Base address: SAUm = 0x4009_4000 + 0x0100 × m

Offset address: 0x0080 + 0x02 × n

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	TSF	BFF	—	—	—	PEF	OVF
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	OVF	Overrun error detection flag of channel n 0: No error occurs 1: An error occurs	R
1	PEF	Parity or ACK error detection flag of channel n 0: No error occurs 1: Parity error occurs (during UART reception) or ACK is not detected (during I^2C transmission)	R
4:2	—	These bits are read as 0.	R
5	BFF	Flag indicating the state of the buffer register for channel n 0: Valid data is not stored in the SDRmn register 1: Valid data is stored in the SDRmn register	R
6	TSF	Flag indicating the state of communications for channel n 0: Communication is stopped or suspended 1: Communication is in progress	R
15:7	—	These bits are read as 0.	R

Note: When the simplified SPI is handling reception in the Snooze mode (SSCm.SWC = 1), the OVF flag and the BFF flag do not change.
Note: If data is written to the SDRmn register when BFF = 1, the transmit or receive data stored in the register is discarded and an overrun error (OVF = 1) is detected.

The SSRmn register indicates the state of communications and occurrence of errors for channel n. The errors indicated by this register are a framing error, parity error, and overrun error.

OVF flag (Overrun error detection flag of channel n)

<Clearing condition>

- 1 is written to the OVCT bit of the SIRmn register.

<Setting condition>

- Even though receive data is stored in the SDRmn register, that data is not read and transmit data or the next receive data is written while the TRXE[0] bit of the SCRmn register is set to 1 (reception or transmission and reception mode in each communication mode).
- Transmit data is not ready for slave transmission or transmission and reception in simplified SPI mode.

PEF flag (Parity or ACK error detection flag of channel n)

<Clearing condition>

- 1 is written to the PECT bit of the SIRmn register.

<Setting condition>

- The parity of the transmit data and the parity bit do not match when UART reception ends (parity error).
- No ACK signal is returned from the slave at the ACK reception timing during I^2C transmission (ACK is not detected).

BFF flag (Flag indicating the state of the buffer register for channel n)

<Clearing condition>

- Transferring transmit data from the SDRmn register to the shift register ends during transmission.

- Reading receive data from the SDRmn register ends during reception.
- The ST[n] bit of the STm register is set to 1 (communication is stopped) or the SS[n] bit of the SSm register is set to 1 (communication is enabled).

<Setting condition>

- Transmit data is written to the SDRmn register while the TRXE[1] bit of the SCRmn register is set to 1 (transmission or transmission and reception mode in each communication mode).
- Receive data is stored in the SDRmn register while the TRXE[0] bit of the SCRmn register is set to 1 (reception or transmission and reception mode in each communication mode).
- A reception error occurs.

TSF flag (Flag indicating the state of communications for channel n)

<Clearing condition>

- The ST[n] bit of the STm register is set to 1 (communication is stopped) or the SS[n] bit of the SSm register is set to 1 (communication is suspended).
- Communication ends.

<Setting condition>

- Communication starts.

23.3.12 SSRmn : Serial Status Register mn (mn = 01, 03, 11)

Base address: SAUm = 0x4009_4000 + 0x0100 × m

Offset address: 0x0080 + 0x02 × n

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	TSF	BFF	—	—	FEF	PEF	OVF
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	OVF	Overrun error detection flag of channel n 0: No error occurs 1: An error occurs	R
1	PEF	Parity or ACK error detection flag of channel n 0: No error occurs 1: Parity error occurs (during UART reception) or ACK is not detected (during I^2C transmission)	R
2	FEF	Framing error detection flag of channel n 0: No error occurs 1: An error occurs (during UART reception)	R
4:3	—	These bits are read as 0.	R
5	BFF	Flag indicating the state of the buffer register for channel n 0: Valid data is not stored in the SDRmn register 1: Valid data is stored in the SDRmn register	R
6	TSF	Flag indicating the state of communications for channel n 0: Communication is stopped or suspended 1: Communication is in progress	R
15:7	—	These bits are read as 0. The write value should be 0.	R

Note: If data is written to the SDRmn register when BFF = 1, the transmit or receive data stored in the register is discarded and an overrun error (OVF = 1) is detected.

The SSRmn register indicates the state of communications and occurrence of errors for channel n. The errors indicated by this register are a framing error, parity error, and overrun error.

OVF flag (Overrun error detection flag of channel n)

<Clearing condition>

- 1 is written to the OVCT bit of the SIRmn register.

<Setting condition>

- Even though receive data is stored in the SDRmn register, that data is not read and transmit data or the next receive data is written while the TRXE[0] bit of the SCRmn register is set to 1 (reception or transmission and reception mode in each communication mode).
- Transmit data is not ready for slave transmission or transmission and reception in simplified SPI mode.

PEF flag (Parity or ACK error detection flag of channel n)

<Clearing condition>

- • 1 is written to the PECT bit of the SIRmn register.

<Setting condition>

- The parity of the transmit data and the parity bit do not match when UART reception ends (parity error).
- No ACK signal is returned from the slave at the ACK reception timing during I^2C transmission (ACK is not detected).

FEF flag (Framing error detection flag of channel n)

<Clearing condition>

- 1 is written to the FECT bit of the SIRmn register.

<Setting condition>

- A stop bit is not detected when UART reception ends.

BFF flag (Flag indicating the state of the buffer register for channel n)

<Clearing condition>

- Transferring transmit data from the SDRmn register to the shift register ends during transmission.
- Reading receive data from the SDRmn register ends during reception.
- The ST[n] bit of the STm register is set to 1 (communication is stopped) or the SS[n] bit of the SSm register is set to 1 (communication is enabled).

<Setting condition>

- Transmit data is written to the SDRmn register while the TRXE[1] bit of the SCRmn register is set to 1 (transmission or transmission and reception mode in each communication mode).
- Receive data is stored in the SDRmn register while the TRXE[0] bit of the SCRmn register is set to 1 (reception or transmission and reception mode in each communication mode).
- A reception error occurs.

TSF flag (Flag indicating the state of communications for channel n)

<Clearing condition>

- The ST[n] bit of the STm register is set to 1 (communication is stopped) or the SS[n] bit of the SSm register is set to 1 (communication is suspended).
- Communication ends.

<Setting condition>

- Communication starts.

23.3.13 SS0 : Serial Channel Start Register 0

Base address: SAU0 = 0x4009_4000

Offset address: 0x00A2

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	SS[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	SS[3:0]	Operation start trigger of channel n 0: No trigger operation 1: Set the SE0.SE[n] bit to 1 to place the channel in communications waiting state*1	R/W
15:4	—	These bits are read as 0. The write value should be 0.*2	R/W

Note: For the UART reception, set the TRXE[0] bit of SCR0n register to 1, and then be sure to set the SS[n] bit to 1 after at least 4 f_{MCK} clock cycles have elapsed.

Note 1. Setting an SS[n] bit to 1 during communications stops communications through channel n and places the channel in the waiting state. At this time, the values of the control registers and shift register, the states of the SCKp and SOp pins, and the values of the SSR0n.FEF, PEF, and OVF flags are retained.

Note 2. Be sure to clear bits [15:4] to 0.

The SS0 register is a trigger register that is used to enable starting communication or count by each channel of serial array unit 0.

When 1 is written to the SS[n] bit, the corresponding bit (SE[n]) of serial channel enable status register 0 (SE0) is set to 1 (operation is enabled). Because the SS[n] bit is a trigger bit, it is cleared immediately when SE0.SE[n] = 1. When the SS0 register is read, 0x0000 is always read.

Note: p: SPI number (p = 00, 01, 10, 11), n: Channel number (n = 0 to 3)

23.3.14 SS1 : Serial Channel Start Register 1

Base address: SAU1 = 0x4009_4100

Offset address: 0x00A2

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SS[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	SS[1:0]	Operation start trigger of channel n 0: No trigger operation 1: Set the SE1.SE[n] bit to 1 to place the channel in communications waiting state*1	R/W
15:2	—	These bits are read as 0. The write value should be 0.*2	R/W

Note: For the UART reception, set the TRXE[0] bit of SCR1n register to 1, and then be sure to set the SS[n] bit to 1 after at least 4 f_{MCK} clock cycles have elapsed.

Note 1. Setting an SS[n] bit to 1 during communications stops communications through channel n and places the channel in the waiting state. At this time, the values of the control registers and shift register, the states of the SCKp and SOp pins, and the values of the SSR1n.FEF, PEF, and OVF flags are retained.

Note 2. Be sure to clear bits [15:2] to 0.

The SS1 register is a trigger register that is used to enable starting communication or count by each channel of serial array unit 1.

When 1 is written to the SS[n] bit, the corresponding bit (SE[n]) of serial channel enable status register 1 (SE1) is set to 1 (operation is enabled). Because the SS[n] bit is a trigger bit, it is cleared immediately when SE1.SE[n] = 1. When the SS1 register is read, 0x0000 is always read.

Note: p: SPI number (p = 20, 21), n: Channel number (n = 0, 1)

23.3.15 ST0 : Serial Channel Stop Register 0

Base address: SAU0 = 0x4009_4000

Offset address: 0x00A4

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	ST[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	ST[3:0]	Operation stop trigger of channel n 0: No trigger operation 1: Clears the SE0.SE[n] bit to 0 and stops the communication operation*1	R/W
15:4	—	These bits are read as 0. The write value should be 0.*2	R/W

Note 1. The values of the control registers and shift register, the states of the SCKp and SOp pins, and the values of the SSR0n.FEF, PEF, and OVF flags are retained.
Note 2. Be sure to clear bits [15:4] to 0.

The ST0 register is a trigger register that is used to enable stopping communication or count by each channel of serial array unit 0.

When 1 is written to the ST[n] bit, the corresponding bit (SE[n]) of serial channel enable status register 0 (SE0) is cleared to 0 (operation is stopped). Because the ST[n] bit is a trigger bit, it is cleared immediately when SE0.SE[n] = 0. When the ST0 register is read, 0x0000 is always read.

Note: p: SPI number (p = 00, 01, 10, 11), n: Channel number (n = 0 to 3)

23.3.16 ST1 : Serial Channel Stop Register 1

Base address: SAU1 = 0x4009_4100

Offset address: 0x00A4

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ST[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	ST[1:0]	Operation stop trigger of channel n 0: No trigger operation 1: Clears the SE1.SE[n] bit to 0 and stops the communication operation*1	R/W
15:2	—	These bits are read as 0. The write value should be 0.*2	R/W

Note 1. The values of the control registers and shift register, the states of the SCKp and SOp pins, and the values of the SSR1n.FEF, PEF, and OVF flags are retained.
Note 2. Be sure to clear bits [15:2] to 0.

The ST1 register is a trigger register that is used to enable stopping communication or count by each channel of serial array unit 1.

When 1 is written to the ST[n] bit, the corresponding bit (SE[n]) of serial channel enable status register 1 (SE1) is cleared to 0 (operation is stopped). Because the ST[n] bit is a trigger bit, it is cleared immediately when SE1.SE[n] = 0. When the ST1 register is read, 0x0000 is always read.

Note: p: SPI number (p = 20, 21), n: Channel number (n = 0, 1)

23.3.17 SE0 : Serial Channel Enable Status Register 0

Base address: SAU0 = 0x4009_4000

Offset address: 0x00A0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	SE[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	SE[3:0]	Indication of whether operation of channel n is enabled or stopped. 0: Operation stops 1: Operation is enabled	R
15:4	—	These bits are read as 0.	R

The SE0 register indicates whether data transmission and reception operation of each channel of serial array unit 0 is enabled or stopped. When 1 is written to a bit of serial channel start register 0 (SS0), the corresponding bit of this register is set to 1. When 1 is written to a bit of serial channel stop register 0 (ST0), the corresponding bit is cleared to 0.

For channel n whose operation is enabled, the value of the CKO[n] bit of serial output register 0 (SO0) to be described later cannot be rewritten by software, and a value reflected by a communication operation is output from the serial clock pin.

For channel n whose operation is stopped, the value of the CKO[n] bit of the SO0 register can be set by software and is output from the serial clock pin. In this way, any waveform, such as that of a start condition or stop condition, can be created by software.

Note: n: Channel number (n = 0 to 3)

23.3.18 SE1 : Serial Channel Enable Status Register 1

Base address: SAU1 = 0x4009_4100

Offset address: 0x00A0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SE[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	SE[1:0]	Indication of whether operation of channel n is enabled or stopped. 0: Operation stops 1: Operation is enabled	R
15:2	—	These bits are read as 0.	R

The SE1 register indicates whether data transmission and reception operation of each channel of serial array unit 1 is enabled or stopped. When 1 is written to a bit of serial channel start register 1 (SS1), the corresponding bit of this register is set to 1. When 1 is written to a bit of serial channel stop register 1 (ST1), the corresponding bit is cleared to 0.

For channel n whose operation is enabled, the value of the CKO[n] bit of serial output register 1 (SO1) to be described later cannot be rewritten by software, and a value reflected by a communication operation is output from the serial clock pin.

For channel n whose operation is stopped, the value of the CKO[n] bit of the SO1 register can be set by software and is output from the serial clock pin. In this way, any waveform, such as that of a start condition or stop condition, can be created by software.

Note: n: Channel number (n = 0, 1)

23.3.19 SOE0 : Serial Output Enable Register 0

Base address: SAU0 = 0x4009_4000

Offset address: 0x00AA

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	SOE[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	SOE[3:0]	Serial output enable or stop of channel n 0: Stops output by serial communication operation 1: Enables output by serial communication operation	R/W
15:4	—	These bits are read as 0. The write value should be 0.*1	R/W

Note 1. Be sure to clear bits [15:4] to 0.

The SOE0 register is used to enable or stop output of the serial communication operation of each channel of serial array unit 0.

For channel n whose serial output is enabled, the value of the SO[n] bit of serial output register 0 (SO0) to be described later cannot be rewritten by software, and a value reflected by a communication operation is output from the serial data output pin.

For channel n, whose serial output is stopped, the SO[n] bit value of the SO0 register can be set by software, and that value can be output from the serial data output pin. In this way, any waveform, such as that of a start condition or stop condition, can be created by software.

Note: n: Channel number (n = 0 to 3)

23.3.20 SOE1 : Serial Output Enable Register 1

Base address: SAU1 = 0x4009_4100

Offset address: 0x00AA

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SOE[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	SOE[1:0]	Serial output enable or stop of channel n 0: Stops output by serial communication operation 1: Enables output by serial communication operation	R/W
15:2	—	These bits are read as 0. The write value should be 0.*1	R/W

Note 1. Be sure to clear bits [15:2] to 0.

The SOE1 register is used to enable or stop output of the serial communication operation of each channel of serial array unit 1.

For channel n whose serial output is enabled, the value of the SO[n] bit of serial output register 1 (SO1) to be described later cannot be rewritten by software, and a value reflected by a communication operation is output from the serial data output pin.

For channel n, whose serial output is stopped, the SO[n] bit value of the SO1 register can be set by software, and that value can be output from the serial data output pin. In this way, any waveform, such as that of a start condition or stop condition, can be created by software.

Note: n: Channel number (n = 0, 1)

23.3.21 SO0 : Serial Output Register 0

Base address: SAU0 = 0x4009_4000

Offset address: 0x00A8

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

Bit	Symbol	Function	R/W
3:0	SO[3:0]	Serial data output of channel n 0: Serial data output value is 0 1: Serial data output value is 1	R/W
7:4	—	These bits are read as 0. The write value should be 0.*1	R/W
11:8	CKO[3:0]	Serial clock output of channel n 0: Serial clock output value is 0 1: Serial clock output value is 1	R/W
15:12	—	These bits are read as 0. The write value should be 0.*1	R/W

Note 1. Be sure to clear bits [15:12] and bits [7:4] to 0.

The SO0 register is a buffer register for serial output of each channel of serial array unit 0.

The value of the SO[n] bit is output from the serial data output pin of channel n.

The value of the CKO[n] bit is output from the serial clock output pin of channel n.

The SO[n] bit can be rewritten by software only when serial output is disabled (SOE0.SOE[n] = 0). When serial output is enabled (SOE0.SOE[n] = 1), rewriting by software is ignored, and the value of the register can be changed only by a serial communication operation.

The CKO[n] bit can be rewritten by software only when the channel operation is stopped (SE0.SE[n] = 0). While channel operation is enabled (SE0.SE[n] = 1), rewriting by software is ignored, and the value of the CKO[n] bit can be changed only by a serial communication operation.

To use the pin for serial interface as a port function pin, set the corresponding CKO[n] and SO[n] bits to 1.

Note: n: Channel number (n = 0 to 3)

23.3.22 SO1 : Serial Output Register 1

Base address: SAU1 = 0x4009_4100

Offset address: 0x00A8

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

Bit	Symbol	Function	R/W
1:0	SO[1:0]	Serial data output of channel n 0: Serial data output value is 0 1: Serial data output value is 1	R/W
3:2	—	These bits are read as 1. The write value should be 1.*1	R/W
7:4	—	These bits are read as 0. The write value should be 0.*1	R/W
9:8	CKO[1:0]	Serial clock output of channel n 0: Serial clock output value is 0 1: Serial clock output value is 1	R/W

Bit	Symbol	Function	R/W
11:10	—	These bits are read as 1. The write value should be 1.*1	R/W
15:12	—	These bits are read as 0. The write value should be 0.*1	R/W

Note 1. Be sure to clear bits [15:12] and bits [7:4] to 0 and set bits [11:10] and bits [3:2] to 1.

The SO1 register is a buffer register for serial output of each channel of serial array unit 1.

The value of the SO[n] bit is output from the serial data output pin of channel n.

The value of the CKO[n] bit is output from the serial clock output pin of channel n.

The SO[n] bit can be rewritten by software only when serial output is disabled (SOE1.SOE[n] = 0). When serial output is enabled (SOE1.SOE[n] = 1), rewriting by software is ignored, and the value of the register can be changed only by a serial communication operation.

The CKO[n] bit can be rewritten by software only when the channel operation is stopped (SE1.SE[n] = 0). While channel operation is enabled (SE1.SE[n] = 1), rewriting by software is ignored, and the value of the CKO[n] bit can be changed only by a serial communication operation.

To use the pin for serial interface as a port function pin, set the corresponding CKO[n] and SO[n] bits to 1.

Note: n: Channel number (n = 0, 1)

23.3.23 SOL0 : Serial Output Level Register 0

Base address: SAU0 = 0x4009_4000

Offset address: 0x00B4

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	SOL2	—	SOL0
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	SOL0	Selects inversion of the level of the transmit data of channel 0 in UART mode 0: Communication data is output as is 1: Communication data is inverted and output	R/W
1	—	This bit is read as 0. The write value should be 0.*1	R/W
2	SOL2	Selects inversion of the level of the transmit data of channel 2 in UART mode 0: Communication data is output as is 1: Communication data is inverted and output	R/W
15:3	—	These bits are read as 0. The write value should be 0.*1	R/W

Note 1. Be sure to clear bits [15:3] and bit [1] to 0.

The SOL0 register is used to set inversion of the data output level of each channel of serial array unit 0.

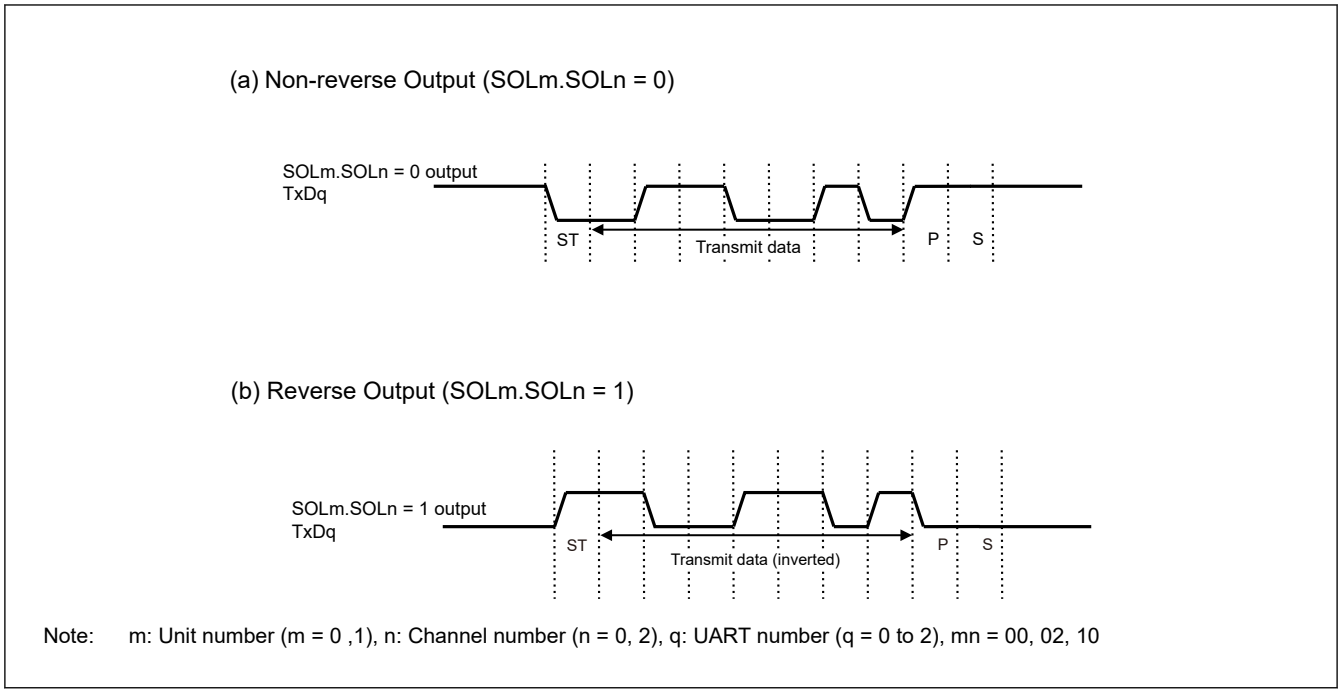
This register can be set only in the UART mode. Be sure to set 0 for the bit corresponding the channel used in the simplified SPI mode or simplified I^2C mode.

Inverting channel n by using this register is reflected on pin output only when serial output is enabled (SOE0.SOE[n] = 1).

When serial output is disabled (SOE0.SOE[n] = 0), the value of the SO0.SO[n] bit is output as is.

Rewriting the SOL0 register is prohibited when the register is in operation (when SE0.SE[n] = 1).

Figure 23.4 shows examples in which the level of transmit data is reversed during UART transmission.



Note: m: Unit number (m = 0 ,1), n: Channel number (n = 0, 2), q: UART number (q = 0 to 2), mn = 00, 02, 10

Figure 23.4 Examples of reverse transmit data

23.3.24 SOL1 : Serial Output Level Register 1

Base address: SAU1 = 0x4009_4100

Offset address: 0x00B4

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SOL0
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	SOL0	Selects inversion of the level of the transmit data of channel 0 in UART mode 0: Communication data is output as is 1: Communication data is inverted and output	R/W
15:1	—	These bits are read as 0. The write value should be 0.*1	R/W

Note 1. Be sure to clear bits [15:1] to 0.

The SOL1 register is used to set inversion for the data output level of channel 0 of serial array unit 1.

This register can be set only in the UART mode. Be sure to set 0 for the bit corresponding the channel used in the simplified SPI mode or simplified I^2C mode.

Inverting channel n by using this register is reflected on pin output only when serial output is enabled (SOE1.SOE[0] = 1).

When serial output is disabled (SOE1.SOE[0] = 0), the value of the SO1.SO[0] bit is output as is.

Rewriting the SOL1 register is prohibited when the register is in operation (when SE1.SE[0] = 1).

Figure 23.4 shows examples in which the level of transmit data is reversed during UART transmission.

23.3.25 SSCm : Serial Standby Control Register m (m = 0, 1)

Base address: SAUm = 0x4009_4000 + 0x0100 × m

Offset address: 0x00B8

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SSEC	SWC
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	SWC	Setting of the Snooze mode 0: Do not use the Snooze mode function 1: Use the Snooze mode function	R/W
1	SSEC	Selection of whether to enable or disable the generation of communication error interrupts in the Snooze mode 0: Enable the generation of error interrupts SAUm_INTSREq 1: Disable the generation of error interrupts SAUm_INTSREq	R/W
15:2	—	These bits are read as 0. The write value should be 0.	R/W

The SSC0 register is used to control the startup of reception (the Snooze mode) while in the Software Standby mode when receiving SPI00 or UART0 serial data.

The SSC1 register is used to control the startup of reception (the Snooze mode) while in the Software Standby mode when receiving SPI20 or UART2 serial data.

Note: The maximum transfer rate in the Snooze mode is as follows.
- When using SPI00, SPI20: Up to 0.5 Mbps
- When using UART0, UART2: Up to 9600 bps

SWC bit (Setting of the Snooze mode)

- When there is a hardware trigger signal in the Software Standby mode, the Software Standby mode is exited, and simplified SPI or UART reception is performed without operating the CPU (the Snooze mode).
- The Snooze mode function can only be specified when the high-speed on-chip oscillator clock (HOCO) or middle-speed on-chip oscillator clock (MOCO) is selected for the CPU and peripheral module clock (PCLKB). If any other clock is selected, specifying this mode is prohibited.
- Even when using Snooze mode, be sure to set the SWC bit to 0 in normal operation mode and change it to 1 just before shifting to Software Standby mode.

Also, be sure to change the SWC bit to 0 after returning from Software Standby mode to normal operation mode.

SSEC bit (Selection of whether to enable or disable the generation of communication error interrupts in the Snooze mode)

- The SSEC bit can be set to 1 or 0 only when both the SWC and SCRmn.EOC bits are set to 1 during UART reception in the Snooze mode. In other cases, clear the SSEC bit to 0.
- Setting SSEC, SWC = 1, 0 is prohibited.

Table 23.6 shows the interrupt in UART reception operation in Snooze mode.

Table 23.6 Interrupt in UART reception operation in Snooze mode

SCRmn.EOC bit	SSEC bit	Reception ended successfully	Reception ended in an error
0	0	SAUm_ENDIn is generated.	SAUm_ENDIn is generated.
0	1	SAUm_ENDIn is generated.	SAUm_ENDIn is generated.
1	0	SAUm_ENDIn is generated.	SAUm_INTSREq is generated.
1	1	SAUm_ENDIn is generated.	No interrupt is generated.

Note: q: UART number (q = 0, 2)

23.3.26 ISC : Input Switch Control Register

Base address: PORGA = 0x4009_1000

Offset address: 0x0003

Bit position:	7	6	5	4	3	2	1	0
Bit field:	ISC76[1:0]		—	ISC43[1:0]		—	ISC1	—
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	ISC1*1	Switching channel 7 input of timer array unit 0: Uses the input signal of the TI07 pin as a timer input (normal operation) 1: Input signal of the RxD2 pin is used as timer input (detects the wakeup signal and measures the low width of the break field and the pulse width of the sync field).	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W
4:3	ISC43[1:0]*2	Switch of the serial clock input source of SPI00 0 0: Input signal of the SCK00 pin (normal operation) 1 0: TO01 output signal Others: Setting prohibited	R/W
5	—	This bit is read as 0. The write value should be 0.	R/W
7:6	ISC76[1:0]*3	Switch of the serial clock input source of SPI01 0 0: Input signal of the SCK01 pin (normal operation) 1 0: TO01 output signal Others: Setting prohibited	R/W

Note 1. When the LIN-bus communication function is used, select the input signal of the RxD2 pin by setting the ISC1 bit to 1.
Note 2. When UART mode or simplified I^2C mode is to be selected for channel 0, set the ISC43[1:0] bits to 0.
Note 3. When UART mode or simplified I^2C mode is to be selected for channel 1, set the ISC76[1:0] bits to 0.

ISC1 bit (Switching channel 7 input of timer array unit)

The ISC1 bit is used to realize a LIN-bus communication operation by UART2 in coordination with the timer array unit. When this bit is set to 1, the input signal of the serial data input (RxD2) pin is selected as a timer input, so that wake up signal can be detected, the low width of the break field, and the pulse width of the sync field can be measured by the timer.

ISC43[1:0] bits (Switch of the serial clock input source of SPI00)

The ISC43[1:0] bits are used to select the serial data input source and serial clock input source of SPI00. These bits can be used to select the SCK00 pin input or TO01 output signal as the serial clock input source of SPI00.

ISC76[1:0] bits (Switch of the serial clock input source of SPI01)

The ISC76[1:0] bits are used to select the serial data input source and serial clock input source of SPI01. These bits can be used to select the SCK01 pin input or TO01 output signal as the serial clock input source of SPI01.

23.3.27 SNFEN : SAU Noise Filter Enable Register

Base address: PORGA = 0x4009_1000

Offset address: 0x0000

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	SNFEN20	—	SNFEN10	—	SNFEN00
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SNFEN00	Use of noise filter of RxD0 pin 0: Noise filter OFF 1: Noise filter ON	R/W
1	—	This bit is read as 0. The write value should be 0.*1	R/W
2	SNFEN10	Use of noise filter of RxD1 pin 0: Noise filter OFF 1: Noise filter ON	R/W
3	—	This bit is read as 0. The write value should be 0.*1	R/W
4	SNFEN20	Use of noise filter of RxD2 pin 0: Noise filter OFF 1: Noise filter ON	R/W
7:5	—	These bits are read as 0. The write value should be 0.*1	R/W

Note 1. Be sure to clear bits [7:5], bit [3] and bit [1] to 0.

The SNFEN register is used to set whether the noise filter can be used for the input signal from the serial data input pin to each channel.

Disable the noise filter of the pin used for simplified SPI or simplified I^2C communication, by clearing the corresponding bit of this register to 0.

Enable the noise filter of the pin used for UART communication, by setting the corresponding bit of this register to 1. When the noise filter is enabled, after synchronization is performed with the operation clock (f_{MCK}) of the target channel, 2-clock match detection is performed. When the noise filter is disabled, only synchronization is performed with the operation clock (f_{MCK}) of the target channel.

SNFEN00 bit (Use of noise filter of RxD0 pin)

Set SNFEN00 to 1 to use the RxD0 pin.

Clear SNFEN00 to 0 to use the other than RxD0 pin.

SNFEN10 bit (Use of noise filter of RxD1 pin)

Set SNFEN10 to 1 to use the RxD1 pin.

Clear SNFEN10 to 0 to use the other than RxD1 pin.

SNFEN20 bit (Use of noise filter of RxD2 pin)

Set SNFEN20 to 1 to use the RxD2 pin.

Clear SNFEN20 to 0 to use the other than RxD2 pin.

23.3.28 ULBS : UART Loopback Select Register

Base address: PORGA = 0x4009_1000

Offset address: 0x0009

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	ULBS5	ULBS4	—	ULBS2	ULBS1	ULBS0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ULBS0	Selection of the UART0 loopback function 0: Inputs the states of the RxD0 pin of serial array unit UART0 to the reception shift register. 1: Loops back output from the transmission shift register to the reception shift register.	R/W

Bit	Symbol	Function	R/W
1	ULBS1	Selection of the UART1 loopback function 0: Inputs the states of the RxD1 pin of serial array unit UART1 to the reception shift register. 1: Loops back output from the transmission shift register to the reception shift register.	R/W
2	ULBS2	Selection of the UART2 loopback function 0: Inputs the states of the RxD2 pin of serial array unit UART2 to the reception shift register. 1: Loops back output from the transmission shift register to the reception shift register.	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	ULBS4	Selection of the UARTA0 loopback function 0: Inputs the states of the RxDA0 pin of serial interface UARTA0 to the reception shift register. 1: Loops back output from the transmission shift register to the reception shift register.	R/W
5	ULBS5	Selection of the UARTA1 loopback function 0: Inputs the states of the RxDA1 pin of serial interface UARTA1 to the reception shift register. 1: Loops back output from the transmission shift register to the reception shift register.	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W

The ULBS register is used to enable the UART loopback function. This register has bits to individually control UART channels. When the bit corresponding to each channel is set to 1, the UART loopback function is selected, and output from the transmission shift register is looped back to the reception shift register.

23.4 Operation Stop Mode

Each serial interface of serial array unit has the operation stop mode.

In this mode, serial communication cannot be executed, thus reducing the power consumption. In addition, the pin for serial interface can be used as port function pins in this mode.

The stopping of the operation by channels is set using each of the following registers.

Table 23.7 to Table 23.10 show each register setting when stopping the operation by channels.

(a) Serial channel stop register m (STm)

The STm register is a trigger register that is used to enable stopping communication or count by each channel.

Table 23.7 Setting of serial channel stop register m (STm) when stopping the operation by channels

Bit	Symbol	Set value	Function
n	ST[n]	1	Operation stop trigger of channel n Because the ST[n] bit is a trigger bit, it is cleared immediately when SEm.SE[n] = 0. 1: Clears the SEm.SE[n] bit to 0 and stops the communication operation

(b) Serial channel enable status register m (SEm)

This register indicates whether data transmission and reception operation of each channel is enabled or stopped.

Table 23.8 Status of serial channel enable status register m (SEm) when stopping the operation by channels

Bit	Symbol	Read value	Function
n	SE[n]	1 or 0	Indication of whether operation of channel n is enabled or stopped. With a channel whose operation is stopped, the value of the CKO[n] bit of the SOm register can be set by software. The SEm is a read-only status register, whose operation is stopped by using the STm register. 0: Operation stops 1: Operation is enabled.

(c) Serial output enable register m (SOEm)

This register is used to enable or stop output of the serial communication operation of each channel.

Table 23.9 Setting of serial output enable register m (SOEm) when stopping the operation by channels

Bit	Symbol	Set value	Function
n	SOE[n]	0	Serial output enable or stop of channel n For channel n, whose serial output is stopped, the SO[n] bit value of the SOm register can be set by software. 0: Stops output by serial communication operation

(d) Serial output register m (SOm)

The SOm register is a buffer register for serial output of each channel.

Table 23.10 Setting of serial output register m (SOm) when stopping the operation by channels

Bit	Symbol	Set value	Function
n	SO[n]	1	Serial data output of channel n When using pins corresponding to each channel as port function pins, set the corresponding SO[n] bits to 1. 1: Serial data output value is 1
n+8	CKO[n]	1	Serial clock output of channel n When using pins corresponding to each channel as port function pins, set the corresponding CKO[n] bits to 1. 1: Serial clock output value is 1

Note: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 11

23.5 Operation of Simplified SPI

This is a clocked communication function that uses three lines: serial clock (SCK) and serial data (SI and SO) lines.

[Data transmission and reception]

- Data length of 7 or 8 bits
- Phase control of transmit and receive data
- MSB- or LSB-first selectable

[Clock control]

- Master or slave selection
- Phase control of I/O clock
- Setting of transfer period by prescaler and internal counter of each channel
- Maximum transfer rate*1
  - During master communication:
    - Max. PCLKB/2 (SPI00 only)

- Max. PCLKB/4

– During slave communication:

- Max. $f_{MCK}/6$

[Interrupt function]

- Transfer end interrupt or buffer empty interrupt

[Error detection flag]

- Overrun error

In addition, simplified SPIs of following channels support the Snooze mode. In the Snooze mode, data can be received without CPU processing upon detecting SCK input in the Software Standby mode. The Snooze mode is only available in SPI00 and SPI20, which support asynchronous reception.

Note 1. Set up the transfer rate within a range satisfying the SCK cycle time (t_{KCY}). For details, see section 37, Electrical Characteristics .

Note: Use a general-purpose port pin to send a chip select signal when required.

The channels supporting simplified SPI are channels 0 to 3 of SAU0 and channels 0 to 1 of SAU1. See Table 23.1 to Table 23.3.

Simplified SPI performs the following seven types of communication operations.

- Master transmission (See section 23.5.1. Master Transmission.)
- Master reception (See section 23.5.2. Master Reception.)
- Master transmission and reception (See section 23.5.3. Master Transmission and Reception.)
- Slave transmission (See section 23.5.4. Slave Transmission.)
- Slave reception (See section 23.5.5. Slave Reception.)
- Slave transmission and reception (See section 23.5.6. Slave Transmission and Reception.)
- Snooze mode function (See section 23.5.7. Snooze Mode Function.)

23.5.1 Master Transmission

Master transmission is when a microcontroller outputs a transfer clock and transmits data to another device.

Table 23.11 shows the specification of master transmission of Simplified SPI.

Table 23.11 Specification of master transmission of simplified SPI (1 of 2)

Simplified SPI	SPI00	SPI01	SPI10	SPI11	SPI20	SPI21
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1
Pins used	SCK00, SO00	SCK01, SO01	SCK10, SO10	SCK11, SO11	SCK20, SO20	SCK21, SO21
Interrupt	SAU0_ENDI0	SAU0_ENDI1	SAU0_ENDI2	SAU0_ENDI3	SAU1_ENDI0	SAU1_ENDI1
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.					
Error detection flag	None					
Transfer data length	7 or 8 bits					
Transfer rate *1	Max. PCLKB /2 [Hz] (SPI00 only), PCLKB /4 [Hz] Min. PCLKB ($2 \times 2^{15} \times 128$) [Hz]					
Data phase	Selectable by the DCP[1] bit of the SCRmn register • DCP[1] = 0: Data output starts from the start of the operation of the serial clock. • DCP[1] = 1: Data output starts half a clock cycle before the start of the serial clock operation.					
Clock phase	Selectable by the DCP[0] bit of the SCRmn register • DCP[0] = 0: Non-reverse • DCP[0] = 1: Reverse					

Table 23.11 Specification of master transmission of simplified SPI (2 of 2)

Simplified SPI	SPI00	SPI01	SPI10	SPI11	SPI20	SPI21
Data direction	MSB or LSB first					

Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 11
Note 1. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics specified in the electrical characteristics. For details, see section 37, Electrical Characteristics .

(1) Register setting

Table 23.12 to Table 23.17 show examples of the register contents for master transmission of simplified SPI.

(a) Serial mode register mn (SMRmn)

Table 23.12 Example of serial mode register mn (SMRmn) contents for master transmission of simplified SPI

Bit	Symbol	Set value	Function
0	MD0	0/1	Interrupt source of channel n 0: Transfer end interrupt 1: Buffer empty interrupt
2:1	MD1[1:0]	00b	Setting of operation mode of channel n 0 0: Simplified SPI mode
5:3	—	100b	Setting disabled (set to the initial value)
6	SIS0	0	Setting is fixed in the simplified SPI mode
7	—	0	Setting disabled (set to the initial value)
8	STS	0	Selection of start trigger source 0: Only software trigger is valid (selected for simplified SPI, UART transmission, and simplified I^2C).
13:9	—	00000b	Setting disabled (set to the initial value)
14	CCS	0	Selection of transfer clock (f_{TCLK}) of channel n 0: Divided operation clock f_{MCK} specified by the CKS bit
15	CKS	0/1	Operation clock (f_{MCK}) of channel n 0: Prescaler output clock CKm0 set by the SPSm register 1: Prescaler output clock CKm1 set by the SPSm register

(b) Serial communication operation setting register mn (SCRmn)

Table 23.13 Example of serial communication operation setting register mn (SCRmn) contents for master transmission of simplified SPI (1 of 2)

Bit	Symbol	Set Value	Function
1:0	DLS[1:0]	10b or 11b	Setting of data length 1 0: 7-bit data length 1 1: 8-bit data length
3:2	—	01b	Setting disabled (set to the initial value)
5:4	SLC[1:0]	00b	Since this bit is dedicated to UART mode, it is fixed in the simplified SPI mode.
6	—	0	Setting disabled (set to the initial value)
7	DIR	0/1	Selection of data transfer sequence in simplified SPI and UART modes 0: Input or output data with MSB first 1: Input or output data with LSB first
9:8	PTC[1:0]	00b	Since this bit is dedicated to UART mode, it is fixed in the simplified SPI mode.

Table 23.13 Example of serial communication operation setting register mn (SCRmn) contents for master transmission of simplified SPI (2 of 2)

Bit	Symbol	Set Value	Function
10	EOC	0	Since this bit is dedicated to UART receive modes, it is fixed in the simplified SPI mode.
11	—	0	Setting disabled (set to the initial value)
13:12	DCP[1:0]	00b to 11b	Selection of data and clock phase in simplified SPI mode For details about the setting, see section 23.3. Register Descriptions.
15:14	TRXE[1:0]	10b	Setting TRXE[1:0] = 10b is fixed in the simplified SPI master transmission mode

(c) Serial data register mn (SDRmn)

Table 23.14 Example of serial data register mn (SDRmn) contents for master transmission of simplified SPI

Bit	Symbol	Set value	Function
7:0	DAT[7:0]	0x00 to 0xFF	Transmit data (Transmit data setting)
8	DAT[8]	0	0 Fixed
15:9	STCLK[6:0]	0x00 to 0x7F	Baud rate setting (Operation clock (f_{MCK}) division setting)

(d) Serial output register m (SOm)

Set only the bit of the target channel.

Table 23.15 Example of serial output register m (SOm) contents for master transmission of simplified SPI

Bit	Symbol	Set value	Function
n	SO[n]	0/1	Serial data output of channel n 0: Serial data output value is 0 1: Serial data output value is 1
n+8	CKO[n]	0/1	Communication starts when a bit is 1 if the clock phase is non reverse (the SCRmn.DCP[0] = 0). If the clock phase is reversed (SCRmn.DCP[0] = 1), communication starts when a bit is 0

(e) Serial output enable register m (SOEm)

Set only the bit of the target channel to 1.

Table 23.16 Example of serial output enable register m (SOEm) contents for master transmission of simplified SPI

Bit	Symbol	Set value	Function
n	SOE[n]	1	Serial output enable or stop of channel n 1: Enable output by serial communication operation.

(f) Serial channel start register m (SSm)

Set only the bit of the target channel to 1.

Table 23.17 Example of serial channel start register m (SSm) contents for master transmission of simplified SPI

Bit	Symbol	Set value	Function
n	SS[n]	1	Operation start trigger of channel n 1: Set the SEm.SE[n] bit to 1 to place the channel in communication waiting state

Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 11

Note: 0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Table 23.18 shows the procedure for initial setting of master transmission.

Table 23.18 Initial setting procedure for master transmission

Step		Process	Detail
Procedure for initial setting of master transmission	<1>	Starting initial setting	—
	<2>	Setting the SPSm register	Set the operation clock.
	<3>	Setting the SMRmn register	Set an operation mode.
	<4>	Setting the SCRmn register	Set a communication format.
	<5>	Setting the SDRmn register	Set a transfer baud rate (setting the transfer clock by dividing the operation clock (f_{MCK})).
	<6>	Setting the SOm register	Set the initial output level of the serial clock (SOm.CKO[n]) and serial data (SOm.SO[n]).
	<7>	Setting of the SOEm register	Set the SOEm.SOE[n] bit to 1 and enable data output of the target channel.
	<8>	Setting port	Enable data output and clock output of the target channel
	<9>	Writing to the SSm register	Set the SSm.SS[n] bit of the target channel to 1 and set SEm.SE[n] = 1 to enable operation.
	<10>	Completing initial setting	Setting of SAU is completed. Write transmit data to the SDRmn.DAT[7:0] bits and start communication.

Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 11

Table 23.19 shows the procedure for stopping master transmission.

Table 23.19 Procedure for stopping master transmission

Step		Process	Detail
Procedure for stopping master transmission	<1>	Starting setting to stop	—
	<2>	Wait until the SSRmn.TSF flag is cleared (optional)	If there is any data being transferred, wait for their completion. If there is an urgent requirement to stop, do not wait.
	<3>	Writing the STm register	Write 1 to the STm.ST[n] bit of the target channel (stopping operation by setting SEm.SE[n] = 0).
	<4>	Changing setting of the SOEm register	Set the SOEm.SOE[n] bit to 0 and stop the output of the target channel.
	<5>	Changing setting of the SOm register (optional)	The levels of the serial clock (SOm.CKO[n]) and serial data (SOm.SO[n]) on the target channel can be changed if required.
	<6>	Stop setting is completed	The master transmission is stopped. Go to the next processing.

Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 11

Table 23.20 shows the procedure for resuming master transmission.

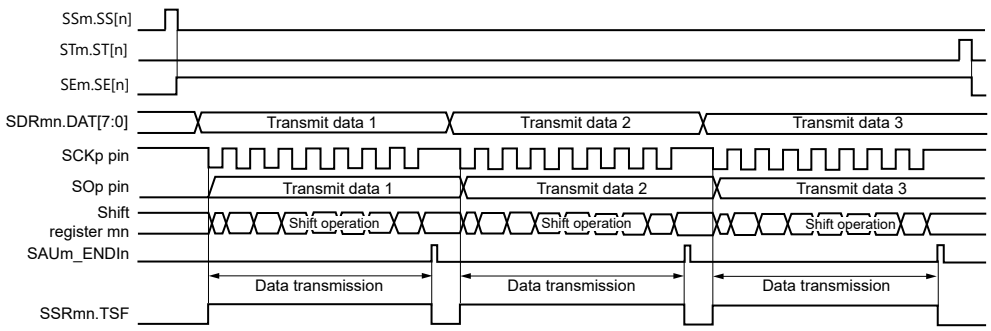
Table 23.20 Procedure for resuming master transmission

Step		Process	Detail
Procedure for resuming master transmission	<1>	Starting setting for resumption	—
	<2>	Wait until Slave is ready	Wait until stop the communication target (slave) or communication operation completed.
	<3>	Port manipulation	Disable data output and clock output of the target channel.
	<4>	Changing setting of the SPSm register (optional)	Reset the register to change the operation clock setting.
	<5>	Changing setting of the SDRmn register (optional)	Reset the register to change the transfer baud rate setting (setting the transfer clock by dividing the operation clock (f_{MCK})).
	<6>	Changing setting of the SMRmn register (optional)	Reset the register to change serial mode register mn (SMRmn) setting.
	<7>	Changing setting of the SCRmn register (optional)	Reset the register to change serial communication operation setting register mn (SCRmn) setting.
	<8>	Changing setting of the SOEm register (optional)	Set the SOEm.SOE[n] bit to 0 to stop output from the target channel.
	<9>	Changing setting of the SOm register (optional)	Set the initial output level of the serial clock (SOm.CKO[n]) and serial data (SOm.SO[n]).
	<10>	Changing setting of the SOEm register	Set the SOEm.SOE[n] bit to 1 and enable output from the target channel.
	<11>	Port manipulation	Enable data output and clock output of the target channel.
	<12>	Writing to the SSm register	Set the SSm.SS[n] bit of the target channel to 1 and set SEm.SE[n] = 1 to enable operation.
	<13>	Completing resumption setting	Setting is completed. Set transmit data to the SDRmn.DAT[7:0] bits and start communication.

Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 11

(3) Processing flow (in single transmission mode)

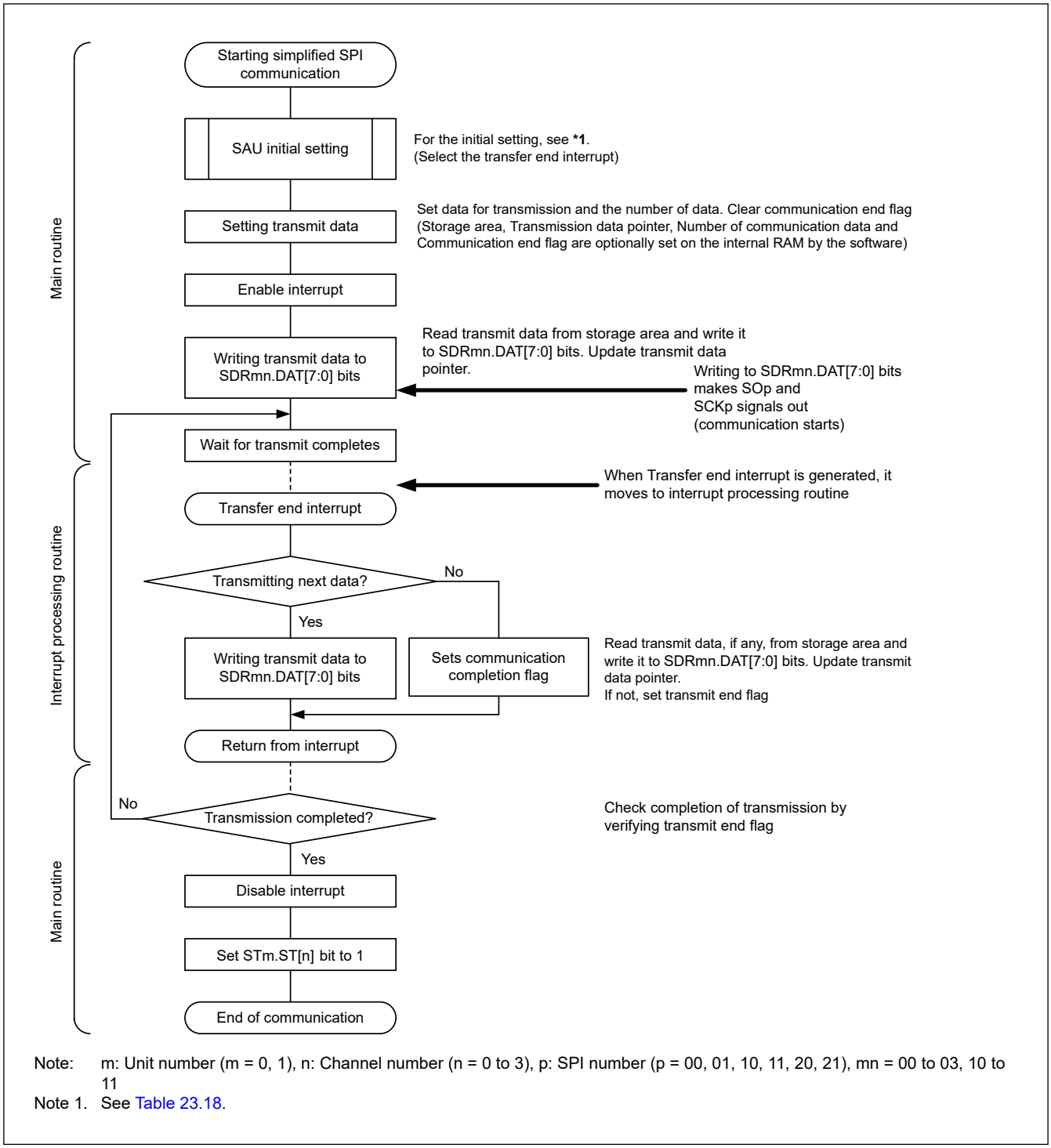
Figure 23.5 shows the timing of master transmission (in single transmission mode) (Type 1: SCRmn.DCP[1:0] = 00b).



Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: SPI number (p = 00, 01, 10, 11, 20, 21), mn = 00 to 03, 10 to 11

Figure 23.5 Timing of master transmission (in single transmission mode) (type 1: SCRmn.DCP[1:0] = 00b)

Figure 23.6 shows the flowchart of master transmission (in single transmission mode).



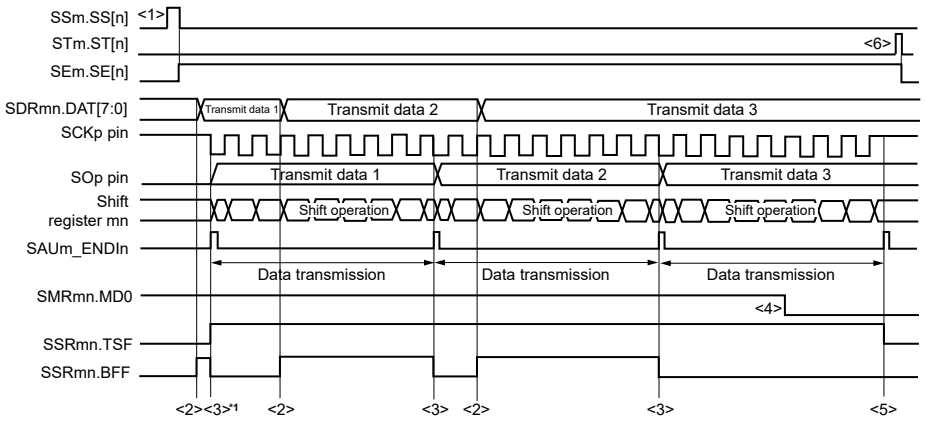
Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: SPI number (p = 00, 01, 10, 11, 20, 21), mn = 00 to 03, 10 to 11

Note 1. See Table 23.18.

Figure 23.6 Flowchart of master transmission (in single transmission mode)

(4) Processing flow (in continuous transmission mode)

Figure 23.7 shows the timing of master transmission (in continuous transmission mode) (Type 1: SCRmn.DCP[1:0] = 00b).



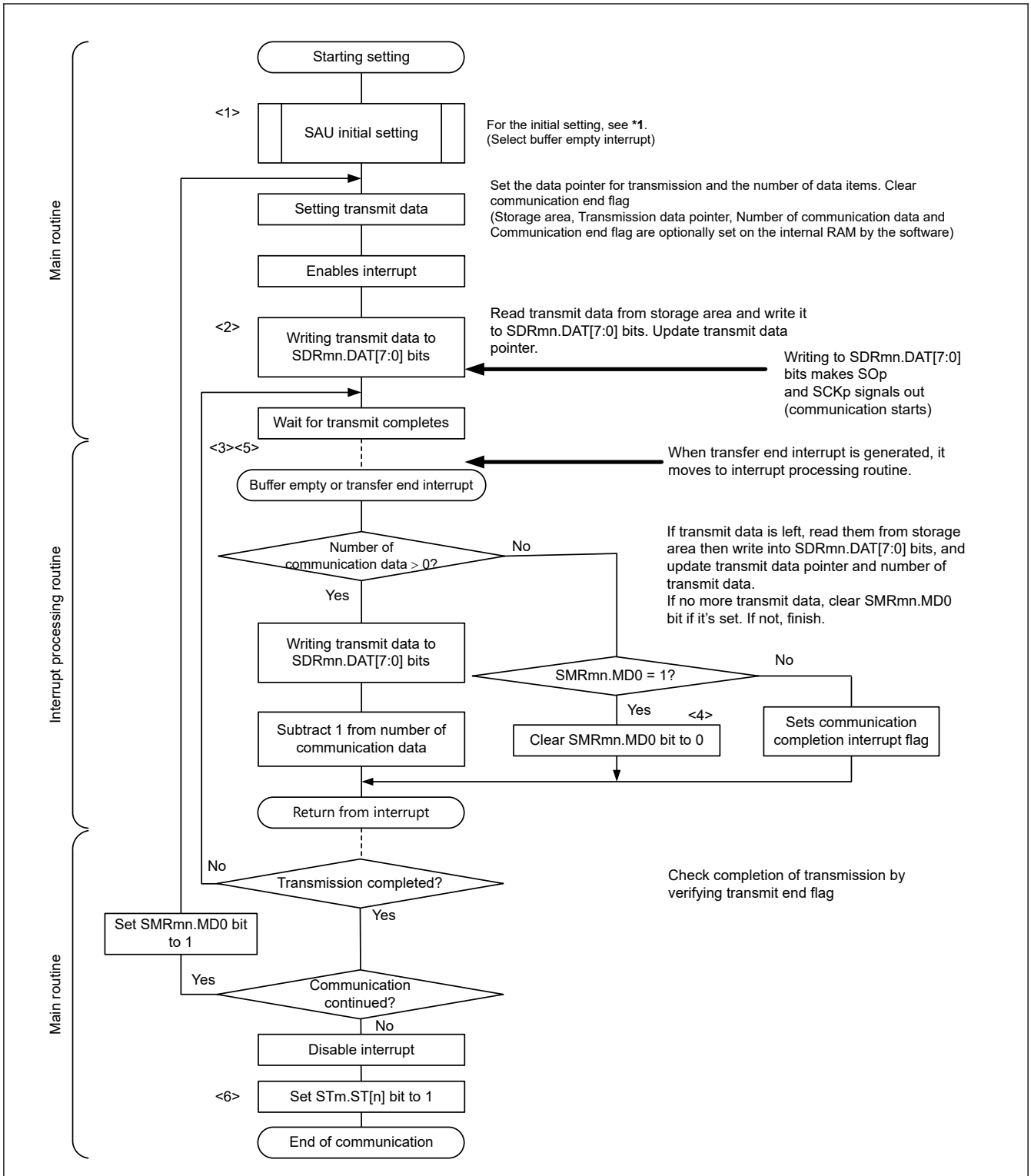
Note: The MD0 bit of serial mode register mn (SMRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last transmit data.

Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: SPI number (p = 00, 01, 10, 11, 20, 21), mn = 00 to 03, 10 to 11

Note 1. If transmit data is written to the SDRmn register while the BFF flag of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

Figure 23.7 Timing of master transmission (in continuous transmission mode) (type 1: SCRmn.DCP[1:0] = 00b)

Figure 23.8 shows the flowchart of master transmission (in continuous transmission mode).



Note: <1> to <6> in the figure correspond to <1> to <6> in Figure 23.7.
Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: SPI number (p = 00, 01, 10, 11, 20, 21), mn = 00 to 03, 10 to 11
Note 1. See Table 23.18.

Figure 23.8 Flowchart of master transmission (in continuous transmission mode)

23.5.2 Master Reception

Master reception is when a microcontroller outputs a transfer clock and receives data from another device.

Table 23.21 shows the specification of master reception of Simplified SPI.

Table 23.21 Specification of master reception of simplified SPI

Simplified SPI	SPI00	SPI01	SPI10	SPI11	SPI20	SPI21
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1
Pins used	SCK00, SI00	SCK01, SI01	SCK10, SI10	SCK11, SI11	SCK20, SI20	SCK21, SI21
Interrupt	SAU0_ENDI0	SAU0_ENDI1	SAU0_ENDI2	SAU0_ENDI3	SAU1_ENDI0	SAU1_ENDI1
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.					
Error detection flag	Overrun error detection flag (SSRmn.OVF) only					
Transfer data length	7 or 8 bits					
Transfer rate*1	Max. PCLKB/2 [Hz] (SPI00 only), PCLKB/4 [Hz] Min. PCLKB/(2 × 2^{15} × 128)[Hz]					
Data phase	Selectable by the DCP[1] bit of the SCRmn register • DCP[1] = 0: Data input starts from the start of the operation of the serial clock. • DCP[1] = 1: Data input starts half a clock cycle before the start of the serial clock operation.					
Clock phase	Selectable by the DCP[0] bit of the SCRmn register • DCP[0] = 0: Non-reverse • DCP[0] = 1: Reverse					
Data direction	MSB or LSB first					

Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 11
Note 1. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics specified in the electrical characteristics. For details, see section 37, Electrical Characteristics.

(1) Register setting

Table 23.22 to Table 23.27 show examples of the register contents for master reception of Simplified SPI.

(a) Serial mode register mn (SMRmn)

Table 23.22 Example of serial mode register mn (SMRmn) contents for master reception of simplified SPI

Bit	Symbol	Set value	Function
0	MD0	0/1	Interrupt source of channel n 0: Transfer end interrupt 1: Buffer empty interrupt
2:1	MD1[1:0]	00b	Setting of operation mode of channel n 0 0: Simplified SPI mode
5:3	—	100b	Setting disabled (set to the initial value)
6	SIS0	0	Setting is fixed in the simplified SPI mode
7	—	0	Setting disabled (set to the initial value)
8	STS	0	Selection of start trigger source 0: Only software trigger is valid (selected for simplified SPI, UART transmission, and simplified I^2C).
13:9	—	00000b	Setting disabled (set to the initial value)
14	CCS	0	Selection of transfer clock (f_{TCLK}) of channel n 0: Divided operation clock f_{MCK} specified by the CKS bit
15	CKS	0/1	Operation clock (f_{MCK}) of channel n 0: Prescaler output clock CKm0 set by the SPSm register 1: Prescaler output clock CKm1 set by the SPSm register

(b) Serial communication operation setting register mn (SCRmn)

Table 23.23 Example of serial communication operation setting register mn (SCRmn) contents for master reception of simplified SPI

Bit	Symbol	Set value	Function
1:0	DLS[1:0]	10b or 11b	Setting of data length 1 0: 7-bit data length 1 1: 8-bit data length
3:2	—	01b	Setting disabled (set to the initial value)
5:4	SLC[1:0]	00b	Since this bit is dedicated to UART mode, it is fixed in the simplified SPI mode.
6	—	0	Setting disabled (set to the initial value)
7	DIR	0/1	Selection of data transfer sequence in simplified SPI and UART modes 0: Input or output data with MSB first 1: Inputs or outputs data with LSB first
9:8	PTC[1:0]	00b	Since this bit is dedicated to UART mode, it is fixed in the simplified SPI mode.
10	EOC	0	Since this bit is dedicated to UART receive modes, it is fixed in the simplified SPI mode.
11	—	0	Setting disabled (set to the initial value)
13:12	DCP[1:0]	00b to 11b	Selection of data and clock phase in simplified SPI mode For details about the setting, see section 23.3. Register Descriptions.
15:14	TRXE[1:0]	01b	Setting TRXE[1:0] = 01b is fixed in the simplified SPI master reception mode

(c) Serial data register mn (SDRmn)

Table 23.24 Example of serial data register mn (SDRmn) contents for master reception of simplified SPI

Bit	Symbol	Set value	Function
7:0	DAT[7:0]	0xFF	Receive data (Write 0xFF as dummy data)
8	DAT[8]	0	0 Fixed
15:9	STCLK[6:0]	0x00 to 0x7F	Baud rate setting (Operation clock (f_{MCK}) division setting)

(d) Serial output register m (SOm)

Set only the bit of the target channel.

Table 23.25 Example of serial output register m (SOm) contents for master reception of simplified SPI

Bit	Symbol	Set value	Function
n	SO[n]	x	Bit that cannot be used in this mode (set to the initial value when not used in any mode)
n+8	CKO[n]	0/1	Communication starts when a bit is 1 if the clock phase is non reverse (SCRmn.DCP[0] = 0). If the clock phase is reversed (SCRmn.DCP[0] = 1), communication starts when a bit is 0

(e) Serial output enable register m (SOEm)

This register is not used in this mode.

Table 23.26 Example of serial output enable register m (SOEm) contents for master reception of simplified SPI

Bit	Symbol	Set value	Function
n	SOE[n]	x	Bit that cannot be used in this mode (set to the initial value when not used in any mode)

(f) Serial channel start register m (SSm)

Set only the bit of the target channel to 1.

Table 23.27 Example of serial channel start register m (SSm) contents for master reception of simplified SPI

Bit	Symbol	Set value	Function
n	SS[n]	1	Operation start trigger of channel n 1: Set the SEm.SE[n] bit to 1 to place the channel in the communications waiting state.

Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 11

Note: ×: Bits not used with serial array units (depending on the settings of other peripheral functions)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Table 23.28 shows the procedure for initial setting of master reception.

Table 23.28 Initial setting procedure for master reception

Step		Process	Detail
Procedure for initial setting for master reception	<1>	Starting initial setting	—
	<2>	Setting the SPSm register	Set the operation clock.
	<3>	Setting the SMRmn register	Set an operation mode, etc.
	<4>	Setting the SCRmn register	Set a communication format.
	<5>	Setting the SDRmn register	Set a transfer baud rate (setting the transfer clock by dividing the operation clock (f_{MCK})).
	<6>	Setting the SOm register	Set the initial output level of the serial clock (SOm.CKO[n]).
	<7>	Setting port	Enable clock output of the target channel.
	<8>	Writing to the SSm register	Set the SSm.SS[n] bit of the target channel to 1 and set SEm.SE[n] = 1 to enable operation.
	<9>	Completing initial setting	Initial setting is completed. Set dummy data to the SDRmn.DAT[7:0] bits and start communication.

Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 11

Table 23.29 shows the procedure for stopping master reception.

Table 23.29 Procedure for stopping master reception

Step		Process	Detail
Procedure for stopping master reception	<1>	Starting setting to stop	—
	<2>	Wait until the SSRmn.TSF flag is cleared (optional)	If there is any data being transferred, wait for their completion. If there is a requirement to stop, do not wait.
	<3>	Writing the STm register	Write 1 to the STm.ST[n] bit of the target channel (stopping operation by setting SEm.SE[n] = 0).
	<4>	Changing setting of the SOEm register	Set the SOEm.SOE[n] bit to 0 and stop the output of the target channel.
	<5>	Changing setting of the SOm register (optional)	The levels of the serial clock (SOm.CKO[n]) and serial data (SOm.SO[n]) on the target channel can be changed if required.
	<6>	Stop setting is completed	After the stop setting is completed, go to the next processing.

Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 11

Table 23.30 shows the procedure for resuming master reception.

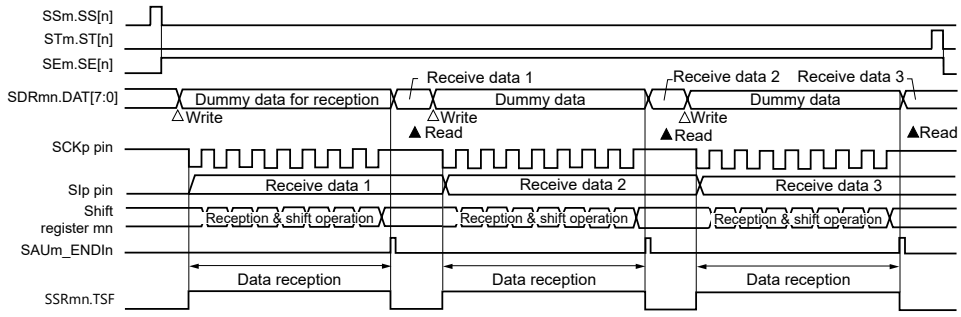
Table 23.30 Procedure for resuming master reception

Step		Process	Detail
Procedure for resuming master reception	<1>	Starting setting for resumption	—
	<2>	Wait until completing slave preparations	Wait until the communication target (slave) stops or communication operation completed.
	<3>	Port manipulation	Disable data output and clock output of the target channel.
	<4>	Changing setting of the SPSm register (optional)	Reset the register to change the operation clock setting.
	<5>	Changing setting of the SDRmn register (optional)	Reset the register to change the transfer baud rate setting (setting the transfer clock by dividing the operation clock (f_{MCK})).
	<6>	Changing setting of the SMRmn register (optional)	Reset the register to change serial mode register mn (SMRmn) setting.
	<7>	Changing setting of the SCRmn register (optional)	Reset the register to change serial communication operation setting register mn (SCRmn) setting.
	<8>	Changing setting of the SOm register (optional)	Set the initial output level of the serial clock (SOm.CKO[n]).
	<9>	Clearing error flag	If the SSRmn.OVF flag remains set, clear this using serial flag clear trigger register mn (SIRmn).
	<10>	Port manipulation	Enable clock output of the target channel.
	<11>	Writing to the SSm register	Set the SSm.SS[n] bit of the target channel to 1 and set SEm.SE[n] = 1 to enable operation.
	<12>	Completing resumption setting	Setting is completed. Set dummy data to the SDRmn.DAT[7:0] bits and start communication.

Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 11

(3) Processing flow (in single reception mode)

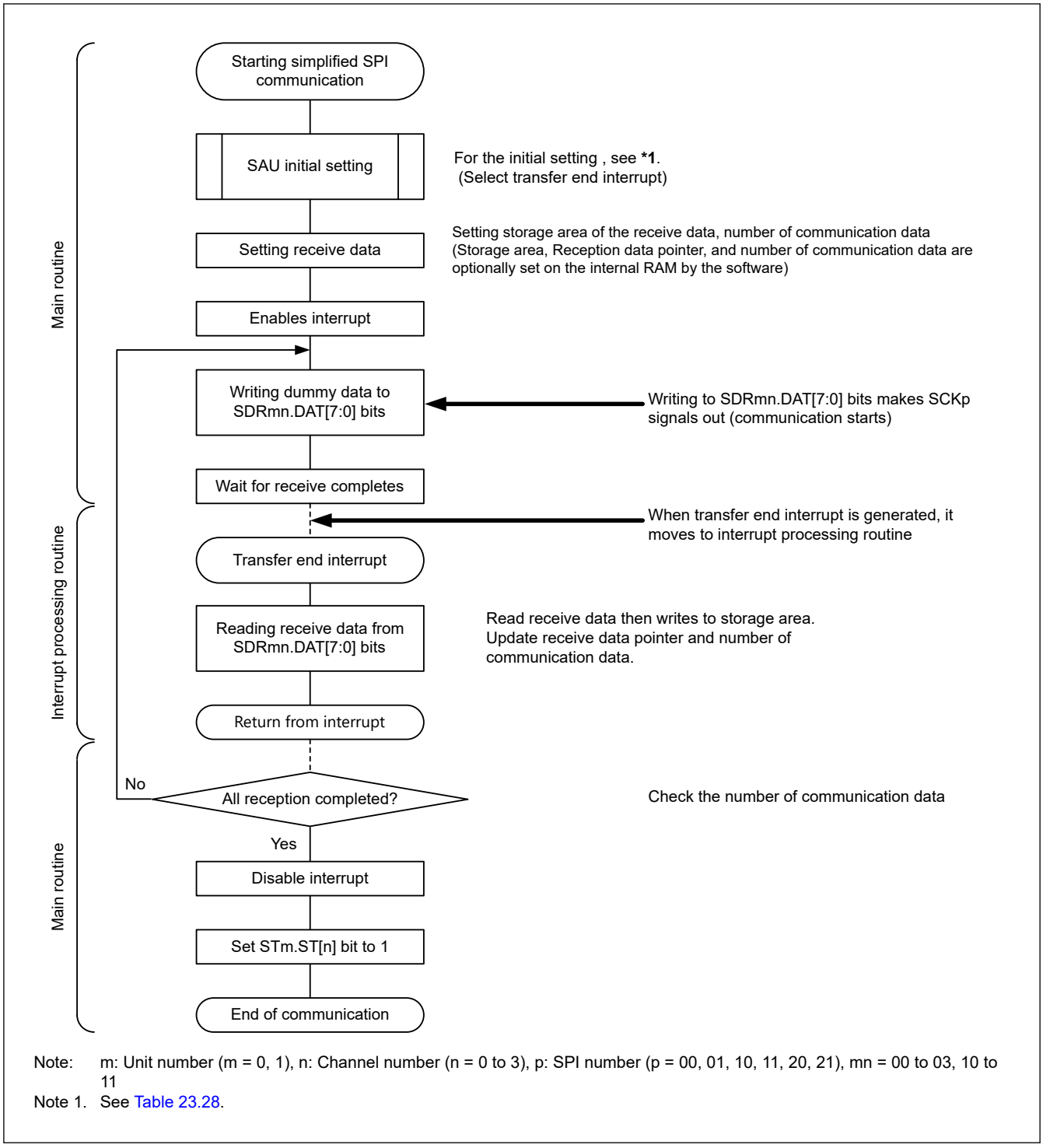
Figure 23.9 shows the timing of master reception (in single reception mode) (Type 1: SCRmn.DCP[1:0] = 00b).



Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: SPI number (p = 00, 01, 10, 11, 20, 21), mn = 00 to 03, 10 to 11

Figure 23.9 Timing of master reception (in single reception mode) (type 1: SCRmn.DCP[1:0] = 00b)

Figure 23.10 shows the flowchart of master reception (in single reception mode).



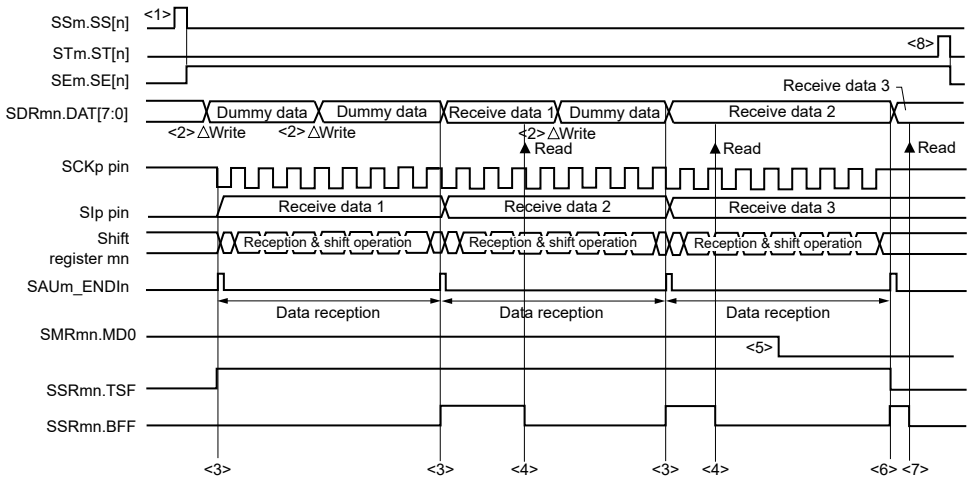
Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: SPI number (p = 00, 01, 10, 11, 20, 21), mn = 00 to 03, 10 to 11

Note 1. See Table 23.28.

Figure 23.10 Flowchart of master reception (in single reception mode)

(4) Processing flow (in continuous reception mode)

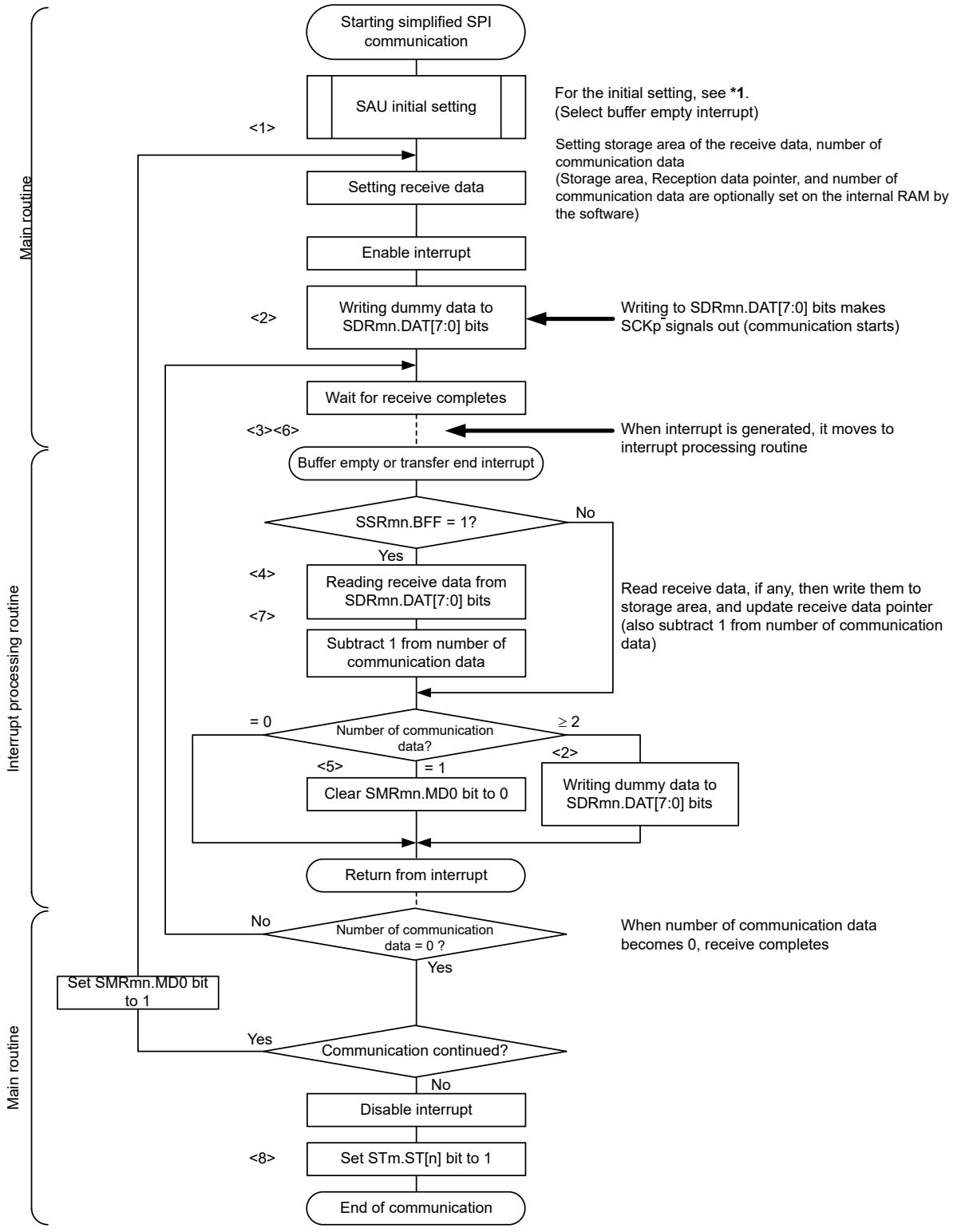
Figure 23.11 shows the timing of master reception (in continuous reception mode) (Type 1: SCRmn.DCP[1:0] = 00b).



Note: The SMRmn.MD0 bit can be rewritten even during operation.
However, rewrite it before receive of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last receive data.
Note: <1> to <8> in the figure correspond to <1> to <8> in Figure 23.12.
Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: SPI number (p = 00, 01, 10, 11, 20, 21), mn = 00 to 03, 10 to 11

Figure 23.11 Timing of master reception (in continuous reception mode) (type 1: DCPmn[1:0] = 00b)

Figure 23.12 shows the flowchart of master reception (in continuous reception mode).



Note: <1> to <8> in the figure correspond to <1> to <8> in Figure 23.11.
Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: SPI number (p = 00, 01, 10, 11, 20, 21), mn = 00 to 03, 10 to 11
Note 1. See Table 23.28.

Figure 23.12 Flowchart of master reception (in continuous reception mode)

23.5.3 Master Transmission and Reception

Master transmission and reception is when a microcontroller outputs a transfer clock and transmits and receives data to and from other device.

Table 23.31 shows the specification for master transmission and reception of Simplified SPI.

Table 23.31 Specification for master transmission and reception of Simplified SPI

Simplified SPI	SPI00	SPI01	SPI10	SPI11	SPI20	SPI21
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1
Pins used	SCK00, SI00, SO00	SCK01, SI01, SO01	SCK10, SI10, SO10	SCK11, SI11, SO11	SCK20, SI20, SO20	SCK21, SI21, SO21
Interrupt	SAU0_ENDI0	SAU0_ENDI1	SAU0_ENDI2	SAU0_ENDI3	SAU1_ENDI0	SAU1_ENDI1
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.					
Error detection flag	Overrun error detection flag (SSRmn.OVF) only					
Transfer data length	7 or 8 bits					
Transfer rate*1	Max. PCLKB /2 [Hz] (SPI00 only), PCLKB /4 [Hz] Min. PCLKB/(2 × 2^{15} × 128)[Hz]					
Data phase	Selectable by the DCP[1] bit of the SCRmn register • DCP[1] = 0: Data I/O starts at the start of the operation of the serial clock. • DCP[1] = 1: Data I/O starts half a clock cycle before the start of the serial clock operation.					
Clock phase	Selectable by the DCP[0] bit of the SCRmn register • DCP[0] = 0: Non-reverse • DCP[0] = 1: Reverse					
Data direction	MSB or LSB first					

Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 11
Note 1. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics specified in the electrical characteristics. For details, see section 37, Electrical Characteristics .

(1) Register setting

Table 23.32 to Table 23.37 show examples of the register contents for master transmission and reception of simplified SPI.

(a) Serial mode register mn (SMRmn)

Table 23.32 Example of serial mode register mn (SMRmn) contents for master transmission and reception of simplified SPI (1 of 2)

Bit	Symbol	Set value	Function
0	MD0	0/1	Interrupt source of channel n 0: Transfer end interrupt 1: Buffer empty interrupt
2:1	MD1[1:0]	00b	Setting of operation mode of channel n 0 0: Simplified SPI mode
5:3	—	100b	Setting disabled (set to the initial value)
6	SIS0	0	Setting is fixed in the simplified SPI mode
7	—	0	Setting disabled (set to the initial value)
8	STS	0	Selection of start trigger source 0: Only software trigger is valid (selected for simplified SPI, UART transmission, and simplified I^2C).
13:9	—	00000b	Setting disabled (set to the initial value)

Table 23.32 Example of serial mode register mn (SMRmn) contents for master transmission and reception of simplified SPI (2 of 2)

Bit	Symbol	Set value	Function
14	CCS	0	Selection of transfer clock (f_{TCLK}) of channel n 0: Divided operation clock f_{MCK} specified by the CKS bit
15	CKS	0/1	Operation clock (f_{MCK}) of channel n 0: Prescaler output clock CKm0 set by the SPSm register 1: Prescaler output clock CKm1 set by the SPSm register

(b) Serial communication operation setting register mn (SCRmn)

Table 23.33 Example of serial communication operation setting register mn (SCRmn) contents for master transmission and reception of simplified SPI

Bit	Symbol	Set value	Function
1:0	DLS[1:0]	10b or 11b	Setting of data length 1 0: 7-bit data length 1 1: 8-bit data length
3:2	—	01b	Setting disabled (set to the initial value)
5:4	SLC[1:0]	00b	Since this bit is dedicated to UART mode, it is fixed in the simplified SPI mode.
6	—	0	Setting disabled (set to the initial value)
7	DIR	0/1	Selection of data transfer sequence in simplified SPI and UART modes 0: Inputs or outputs data with MSB first. 1: Inputs or outputs data with LSB first.
9:8	PTC[1:0]	00b	Since this bit is dedicated to UART mode, it is fixed in the simplified SPI mode.
10	EOC	0	Since this bit is dedicated to UART receive modes, it is fixed in the simplified SPI mode.
11	—	0	Setting disabled (set to the initial value)
13:12	DCP [1:0]	00b to 11b	Selection of data and clock phase in simplified SPI mode For details about the setting, see section 23.3. Register Descriptions.
15:14	TRXE[1:0]	11b	Setting TRXE[1:0] = 11b is fixed in the simplified SPI master transmission and reception mode

(c) Serial data register mn (SDRmn)

Table 23.34 Example of serial data register mn (SDRmn) contents for master transmission and reception of simplified SPI

Bit	Symbol	Set value	Function
7:0	DAT[7:0]	0xFF	Transmit data or Receive data Transmit data setting and write 0xFF as dummy data
8	DAT[8]	0	0 Fixed
15:9	STCLK[6:0]	0x00 to 0x7F	Baud rate setting Operation clock (f_{MCK}) division setting

(d) Serial output register m (SOm)

Set only the bit of the target channel.

Table 23.35 Example of serial output register m (SOm) contents for master transmission and reception of simplified SPI

Bit	Symbol	Set value	Function
n	SO[n]	0/1	Serial data output of channel n 0: Serial data output value is 0 1: Serial data output value is 1
n+8	CKO[n]	0/1	Communication starts when a bit is 1 if the clock phase is non-reversed (SCRmn.DCP[0] = 0). If the clock phase is reversed (SCRmn.DCP[0] = 1), communication starts when a bit is 0

(e) Serial output enable register m (SOEm)

Set only the bit of the target channel to 1.

Table 23.36 Example of serial output enable register m (SOEm) contents for master transmission and reception of simplified SPI

Bit	Symbol	Set value	Function
n	SOE[n]	1	Serial output enable or stop of channel n 1: Enable output by serial communication operation

(f) Serial channel start register m (SSm)

Set only the bit of the target channel to 1.

Table 23.37 Example of serial channel start register m (SSm) contents for master transmission and reception of simplified SPI

Bit	Symbol	Set value	Function
n	SS[n]	1	Operation start trigger of channel n 1: Set the SEm.SE[n] bit to 1 to place the channel in communication waiting state

Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 11

Note: 0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Table 23.38 shows the procedure for initial setting of master transmission and reception.

Table 23.38 Initial setting procedure for master transmission and reception

Step		Process	Detail
Procedure for initial setting of master transmission and reception	<1>	Starting initial setting	—
	<2>	Setting the SPSm register	Set the operation clock.
	<3>	Setting the SMRmn register	Set an operation mode.
	<4>	Setting the SCRmn register	Set a communication format.
	<5>	Setting the SDRmn register	Set a transfer baud rate (setting the transfer clock by dividing the operation clock (f_{MCK})).
	<6>	Setting the SOm register	Set the initial output level of the serial clock (SOm.CKO[n]) and serial data (SOm.SO[n]).
	<7>	Changing setting of the SOEm register	Set the SOEm.SOE[n] bit to 1 and enable data output of the target channel
	<8>	Setting port	Enable data output and clock output of the target channel.
	<9>	Writing to the SSm register	Set the SSm.SS[n] bit of the target channel to 1 and set the SEm.SE[n] bit to 1 to enable operation.
	<10>	Completing initial setting	Initial setting is completed. Set transmit data to the SDRmn.DAT[7:0] bits and start.

Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 11

Table 23.39 shows the procedure for stopping master transmission and reception.

Table 23.39 Procedure for stopping master transmission and reception

Step		Process	Detail
Procedure for stopping master transmission and reception	<1>	Starting setting to stop	—
	<2>	Wait until the SSRmn.TSF flag is cleared (optional)	If there is any data being transferred, wait for their completion. If there is a requirement to stop, do not wait.
	<3>	Writing the STm register	Write 1 to the STm.ST[n] bit of the target channel and set SEm.SE[n] = 0 to stop operation.
	<4>	Changing setting of the SOEm register	Set the SOEm.SOE[n] bit to 0 and stop the output of the target channel.
	<5>	Changing setting of the SOm register (optional)	The levels of the serial clock (SOm.CKO[n]) and serial data (SOm.SO[n]) on the target channel can be changed if required.
	<6>	Stop setting is completed	After the stop setting is completed, go to the next processing.

Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 11

Table 23.40 shows the procedure for resuming master transmission and reception.

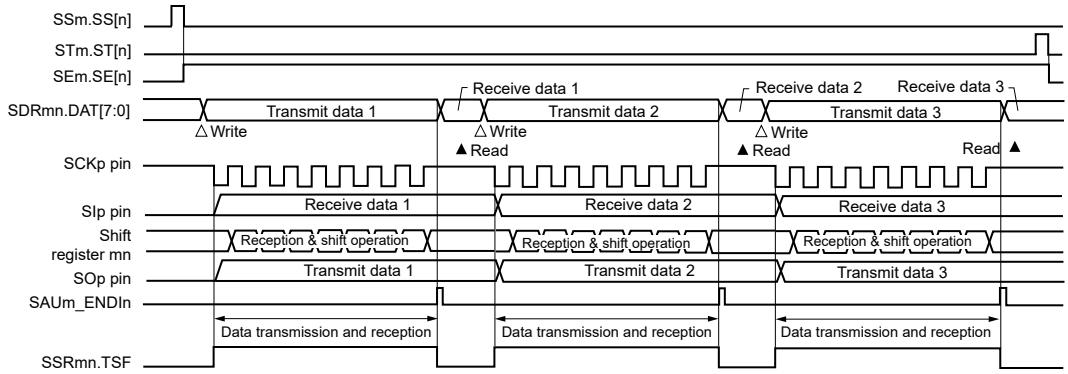
Table 23.40 Procedure for resuming master transmission and reception

Step		Process	Detail
Procedure for Resuming Master Transmission and Reception	<1>	Starting setting for resumption	—
	<2>	Check completing slave preparations	Wait until the communication target (slave) stops or communication operation completed.
	<3>	Port manipulation	Disable data output and clock output of the target channel.
	<4>	Changing setting of the SPSm register (optional)	Reset the register to change the operation clock setting.
	<5>	Changing setting of the SDRmn register (optional)	Reset the register to change the transfer baud rate setting (setting the transfer clock by dividing the operation clock (f_{MCK})).
	<6>	Changing setting of the SMRmn register (optional)	Reset the register to change serial mode register mn (SMRmn) setting.
	<7>	Changing setting of the SCRmn register (optional)	Reset the register to change serial communication operation setting register mn (SCRmn) setting.
	<8>	Clearing error flag (optional)	If the SSRmn.OVF flag remains set, clear this using serial flag clear trigger register mn (SIRmn).
	<9>	Changing setting of the SOEm register (optional)	Set the SOEm.SOE[n] bit to 0 to stop output from the target channel.
	<10>	Changing setting of the SOm register (optional)	Set the initial output level of the serial clock (SOm.CKO[n]) and serial data (SOm.SO[n]).
	<11>	Changing setting of the SOEm register (optional)	Set the SOEm.SOE[n] bit to 1 and enable output from the target channel.
	<12>	Port manipulation	Enable data output and clock output of the target channel.
	<13>	Writing to the SSm register	Set the SSm.SS[n] bit of the target channel to 1 and set the SEm.SE[n] bit to 1 to enable operation.
	<14>	Completing resumption setting	—

Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 11

(3) Processing flow (in single transmission and reception mode)

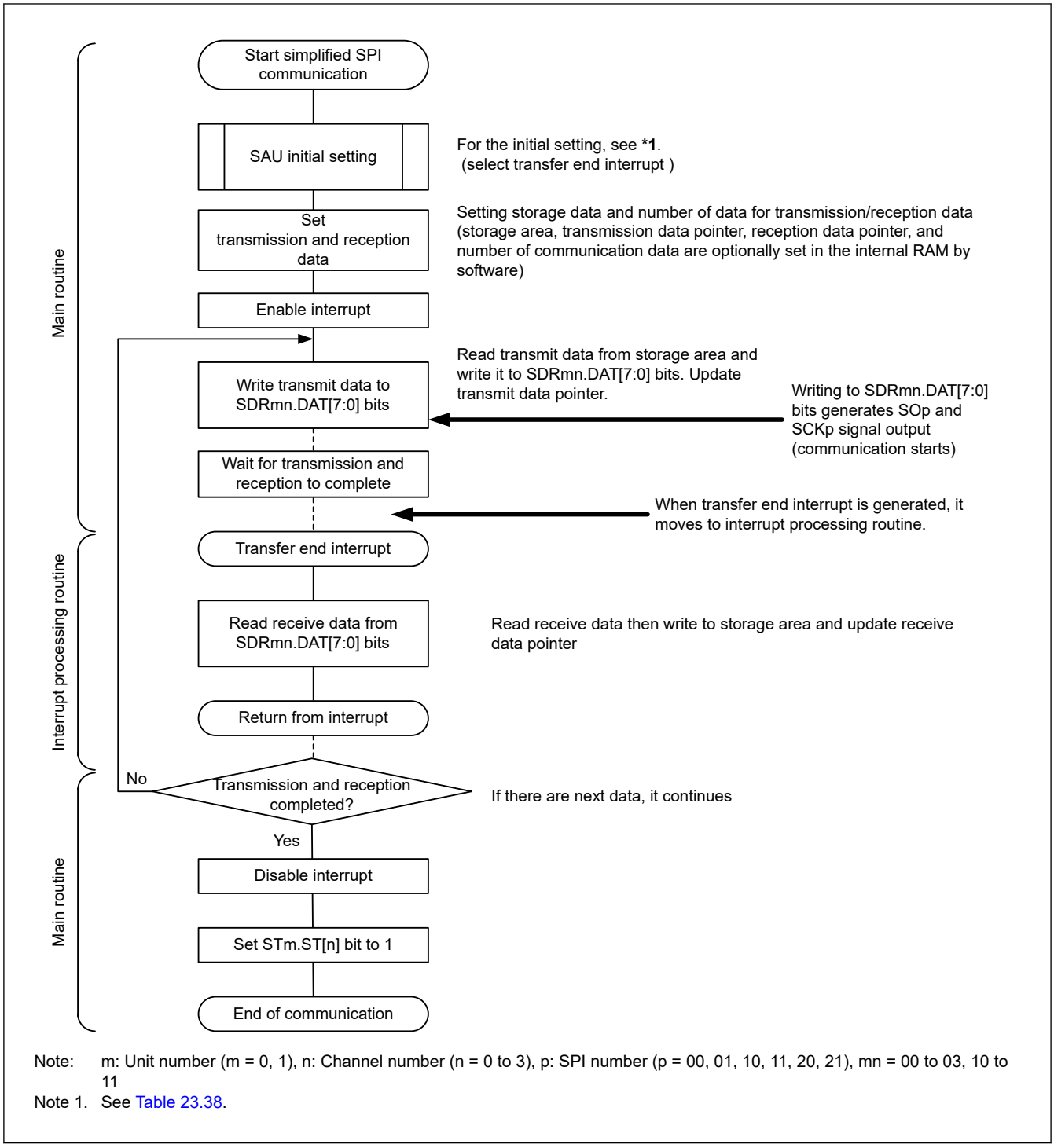
Figure 23.13 shows the timing of master transmission and reception (in single transmission and reception mode) (type 1: SCRmn.DCP[1:0] = 00b).



Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: SPI number (p = 00, 01, 10, 11, 20, 21), mn = 00 to 03, 10 to 11

Figure 23.13 Timing of master transmission and reception (in single transmission and reception mode) (type 1: SCRmn.DCP[1:0] = 00b)

Figure 23.14 shows the flowchart of master transmission and reception (in single transmission and reception mode).



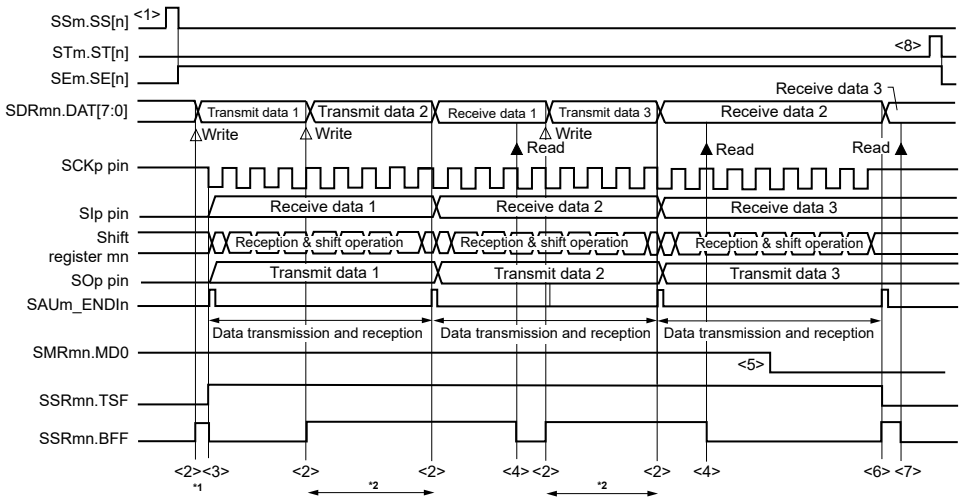
Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: SPI number (p = 00, 01, 10, 11, 20, 21), mn = 00 to 03, 10 to 11

Note 1. See Table 23.38.

Figure 23.14 Flowchart of master transmission and reception (in single transmission and reception mode)

(4) Processing flow (in continuous transmission and reception mode)

Figure 23.15 shows the timing of master transmission and reception (in continuous transmission and reception mode) (type 1: SCRmn.DCP[1:0] = 00b).



Note: The MD0 bit of serial mode register mn (SMRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last transmit data.

Note: <1> to <8> in the figure correspond to <1> to <8> in Figure 23.16.

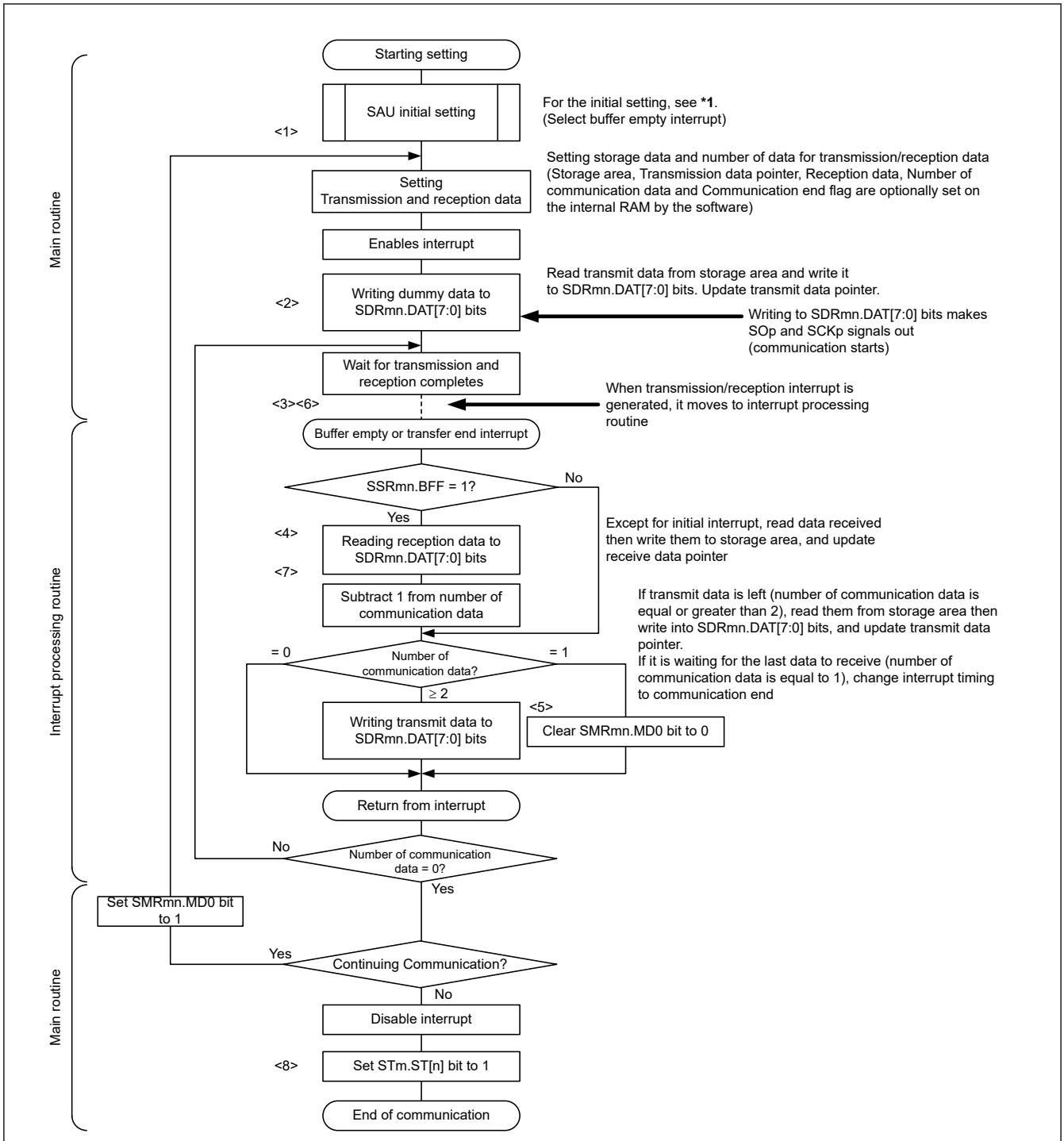
Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: SPI number (p = 00, 01, 10, 11, 20, 21) mn = 00 to 03, 10 to 11

Note 1. If transmit data is written to the SDRmn register while the BFF flag of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

Note 2. The transmit data can be read by reading the SDRmn register during this period. Reading this register does not affect the transfer operation.

Figure 23.15 Timing of master transmission and reception (in continuous transmission and reception mode) (type 1: SCRmn.DCP[1:0] = 00b)

Figure 23.16 shows the flowchart of master transmission and reception (in continuous transmission and reception mode)



Note: <1> to <8> in the figure correspond to <1> to <8> in Figure 23.15
Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: SPI number (p = 00, 01, 10, 11, 20, 21), mn = 00 to 03, 10 to 11
Note 1. See Table 23.38.

Figure 23.16 Flowchart of master transmission and reception (in continuous transmission and reception mode

23.5.4 Slave Transmission

Slave transmission is when a microcontroller transmits data to another device in the state of a transfer clock being input from another device.

Table 23.41 shows the specification of slave transmission of simplified SPI.

Table 23.41 Specification of slave transmission of simplified SPI

Simplified SPI	SPI00	SPI01	SPI10	SPI11	SPI20	SPI21
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1
Pins used	SCK00, SO00	SCK01, SO01	SCK10, SO10	SCK11, SO11	SCK20, SO20	SCK21, SO21
Interrupt	SAU0_ENDI0	SAU0_ENDI1	SAU0_ENDI2	SAU0_ENDI3	SAU1_ENDI0	SAU1_ENDI1
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.					
Error detection flag	Overrun error detection flag (SSRmn.OVF) only					
Transfer data length	7 or 8 bits					
Transfer rate	Max. $f_{MCK}/6$ [Hz]*1 *2					
Data phase	Selectable by the DCP[1] bit of the SCRmn register • DCP[1] = 0: Data output starts from the start of the operation of the serial clock • DCP[1] = 1: Data output starts half a clock cycle before the start of the serial clock operation					
Clock phase	Selectable by the DCP[0] bit of the SCRmn register • DCP[0] = 0: Non-reverse • DCP[0] = 1: Reverse					
Data direction	MSB or LSB first					

Note: f_{MCK}: Operation clock frequency of target channel
f_{SCK}: Serial clock frequency
Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 11
Note 1. Because the external serial clock input to the SCK00, SCK01, SCK10, SCK11, SCK20, and SCK21 pins is sampled internally and used, the fastest transfer rate is $f_{MCK}/6$ [Hz].
Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics specified in the electrical characteristics. For details, see section 37, Electrical Characteristics .

(1) Register setting

Table 23.42 to Table 23.47 show examples of the register contents for slave transmission of simplified SPI.

(a) Serial mode register mn (SMRmn)

Table 23.42 Example of serial mode register mn (SMRmn) contents for slave transmission of simplified SPI (1 of 2)

Bit	Symbol	Set value	Function
0	MD0	0/1	Interrupt source of channel n 0: Transfer end interrupt 1: Buffer empty interrupt
2:1	MD1[1:0]	00b	Setting of operation mode of channel n 0 0: Simplified SPI mode
5:3	—	100b	Setting disabled (set to the initial value)
6	SIS0	0	Setting is fixed in the simplified SPI mode
7	—	0	Setting disabled (set to the initial value)
8	STS	0	Selection of start trigger source 0: Only software trigger is valid (selected for simplified SPI, UART transmission, and simplified I^2C).
13:9	—	00000b	Setting disabled (set to the initial value)
14	CCS	1	Selection of transfer clock (f_{TCLK}) of channel n 1: Clock input f_{SCK} from the SCKp pin (slave transfer in simplified SPI mode)

Table 23.42 Example of serial mode register mn (SMRmn) contents for slave transmission of simplified SPI (2 of 2)

Bit	Symbol	Set value	Function
15	CKS	0/1	Operation clock (f_{MCK}) of channel n 0: Prescaler output clock CKm0 set by the SPSm register 1: Prescaler output clock CKm1 set by the SPSm register

(b) Serial communication operation setting register mn (SCRmn)

Table 23.43 Example of serial communication operation setting register mn (SCRmn) contents for slave transmission of simplified SPI

Bit	Symbol	Set value	Function
1:0	DLS[1:0]	10b or 11b	Setting of data length 1 0: 7-bit data length 1 1: 8-bit data length
3:2	—	01b	Setting disabled (set to the initial value)
5:4	SLC[1:0]	00b	Since this bit is dedicated to UART mode, it is fixed in the simplified SPI mode.
6	—	0	Setting disabled (set to the initial value)
7	DIR	0/1	Selection of data transfer sequence in simplified SPI and UART modes 0: Input or output data with MSB first 1: Input or output data with LSB first
9:8	PTC[1:0]	00b	Since this bit is dedicated to UART mode, it is fixed in the simplified SPI mode.
10	EOC	0	Since this bit is dedicated to UART receive modes, it is fixed in the simplified SPI mode.
11	—	0	Setting disabled (set to the initial value)
13:12	DCP[1:0]	00b to 11b	Selection of data and clock phase in simplified SPI mode. Selection of the data and clock phase (For details about the setting, see section 23.3. Register Descriptions.)
15:14	TRXE[1:0]	10b	Setting TRXE[1:0] = 10b is fixed in the simplified SPI slave transmission mode

(c) Serial data register mn (SDRmn)

Table 23.44 Example of serial data register mn (SDRmn) contents for slave transmission of simplified SPI

Bit	Symbol	Set value	Function
7:0	DAT[7:0]	0x00 to 0xFF	Transmit data Transmit data setting
8	DAT[8]	0	0 Fixed
15:9	STCLK[6:0]	0x00	Baud rate setting (do not used in the any slave mode)

(d) Serial output register m (SOm)

Set only the bit of the target channel.

Table 23.45 Example of serial output register m (SOm) contents for slave transmission of simplified SPI

Bit	Symbol	Set value	Function
n	SO[n]	0/1	Serial data output of channel n 0: Serial data output value is 0 1: Serial data output value is 1
n+8	CKO[n]	x	Bit that cannot be used in this mode (set to the initial value when not used in any mode)

(e) Serial output enable register m (SOEm)

Set only the bit of the target channel to 1.

Table 23.46 Example of serial output enable register m (SOEm) contents for slave transmission of simplified SPI

Bit	Symbol	Set value	Function
n	SOE[n]	1	Serial output enable or stop of channel n 1: Enable output by serial communication operation

(f) Serial channel start register m (SSm)

Set only the bit of the target channel to 1.

Table 23.47 Example of serial channel start register m (SSm) contents for slave transmission of simplified SPI

Bit	Symbol	Set value	Function
n	SS[n]	1	Operation start trigger of channel n 1: Set the SEm.SE[n] bit to 1 to place the channel in communication waiting state

Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: SPI number (p = 00, 01, 10, 11, 20, 21), mn = 00 to 03, 10 to 11

Note: ×: Bits not used with serial array units (depending on the settings of other peripheral functions)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Table 23.48 shows the procedure for initial setting of slave transmission.

Table 23.48 Initial setting procedure for slave transmission

Step		Process	Detail
Procedure for initial setting of slave transmission	<1>	Starting initial setting	—
	<2>	Setting the SPSm register	Set the operation clock.
	<3>	Setting the SMRmn register	Set an operation mode.
	<4>	Setting the SCRmn register	Set a communication format.
	<5>	Setting the SDRmn register	Set the SDRmn.STCLK[6:0] bits to 0x00 for baud rate setting.
	<6>	Setting the SOm register	Set the initial output level of the serial data (SOm.SO[n]).
	<7>	Changing setting of the SOEm register	Set the SOEm.SOE[n] bit to 1 and enable data output of the target channel.
	<8>	Setting port	Enable data output of the target channel.
	<9>	Writing to the SSm register	Set the SSm.SS[n] bit of the target channel to 1 and the SEm.SE[n] bit to 1 to enable operation.
	<10>	Completing initial setting	Initial setting is completed. Set transmit data to the SDRmn.DAT[7:0] bits and wait for a clock from the master

Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 11

Table 23.49 shows the procedure for stopping slave transmission.

Table 23.49 Procedure for stopping slave transmission

Step		Process	Detail
Procedure for stopping slave transmission	<1>	Starting setting to stop	—
	<2>	Wait until the SSRmn.TSF flag is cleared (optional)	If there is any data being transferred, wait for their completion. If there is a requirement to stop, do not wait.
	<3>	Writing the STm register	Write 1 to the STm.ST[n] bit of the target channel and set the SEm.SE[n] bit to 0 to stop operation.
	<4>	Changing setting of the SOEm register	Set the SOEm.SOE[n] bit to 0 and stop the output of the target channel.
	<5>	Changing setting of the SOm register (optional)	The levels of the serial data (SOm.SO[n]) on the target channel can be changed if required.
	<6>	Stop setting is completed	After the stop setting is completed, go to the next processing.

Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 11

Table 23.50 shows the procedure for resuming slave transmission.

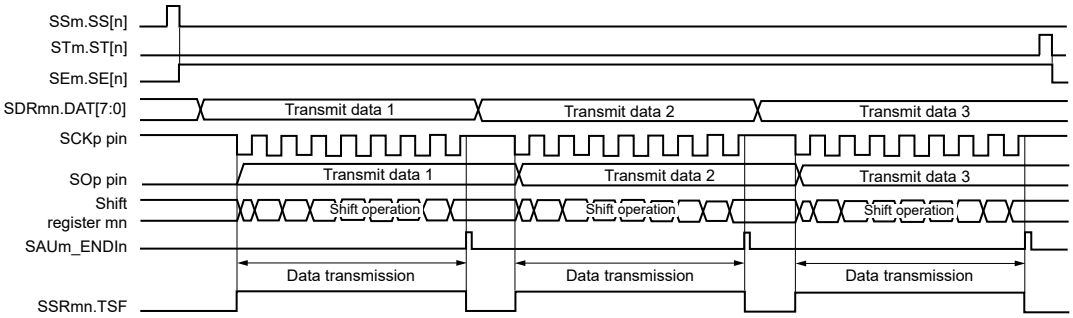
Table 23.50 Procedure for resuming slave transmission

Step		Process	Detail
Procedure for resuming slave transmission	<1>	Starting setting for resumption	—
	<2>	Wait until completing master preparations	Wait until the communication target (master) stops or communication operation completed.
	<3>	Port manipulation	Disable data output and clock output of the target channel.
	<4>	Changing setting of the SPSm register (optional)	Reset the register to change the operation clock setting.
	<5>	Changing setting of the SDRmn register (optional)	Reset the register to change the transfer baud rate setting (setting the transfer clock by dividing the operation clock (f_{MCK})).
	<6>	Changing setting of the SMRmn register (optional)	Reset the register to change serial mode register mn (SMRmn) setting.
	<7>	Changing setting of the SCRmn register (optional)	Reset the register to change serial communication operation setting register mn (SCRmn) setting.
	<8>	Clearing error flag (optional)	If the SSRmn.OVF flag remains set, clear this using serial flag clear trigger register mn (SIRmn).
	<9>	Changing setting of the SOEm register (optional)	Set the SOEm.SOE[n] bit to 0 to stop output from the target channel.
	<10>	Changing setting of the SOm register	Set the initial output level of the serial data (SOm.SO[n]).
	<11>	Changing setting of the SOEm register	Set the SOEm.SOE[n] bit to 1 and enable output from the target channel.
	<12>	Port manipulation	Enable data output of the target channel.
	<13>	Writing to the SSm register	Set the SSm.SS[n] bit of the target channel to 1 and set the SEm.SE[n] bit to 1 to enable operation.
	<14>	Starting communication	Sets transmit data to the SDRmn.DAT[7:0] bits and wait for a clock from the master.
	<15>	Completing resumption setting	—

Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 11

(3) Processing flow (in single transmission mode)

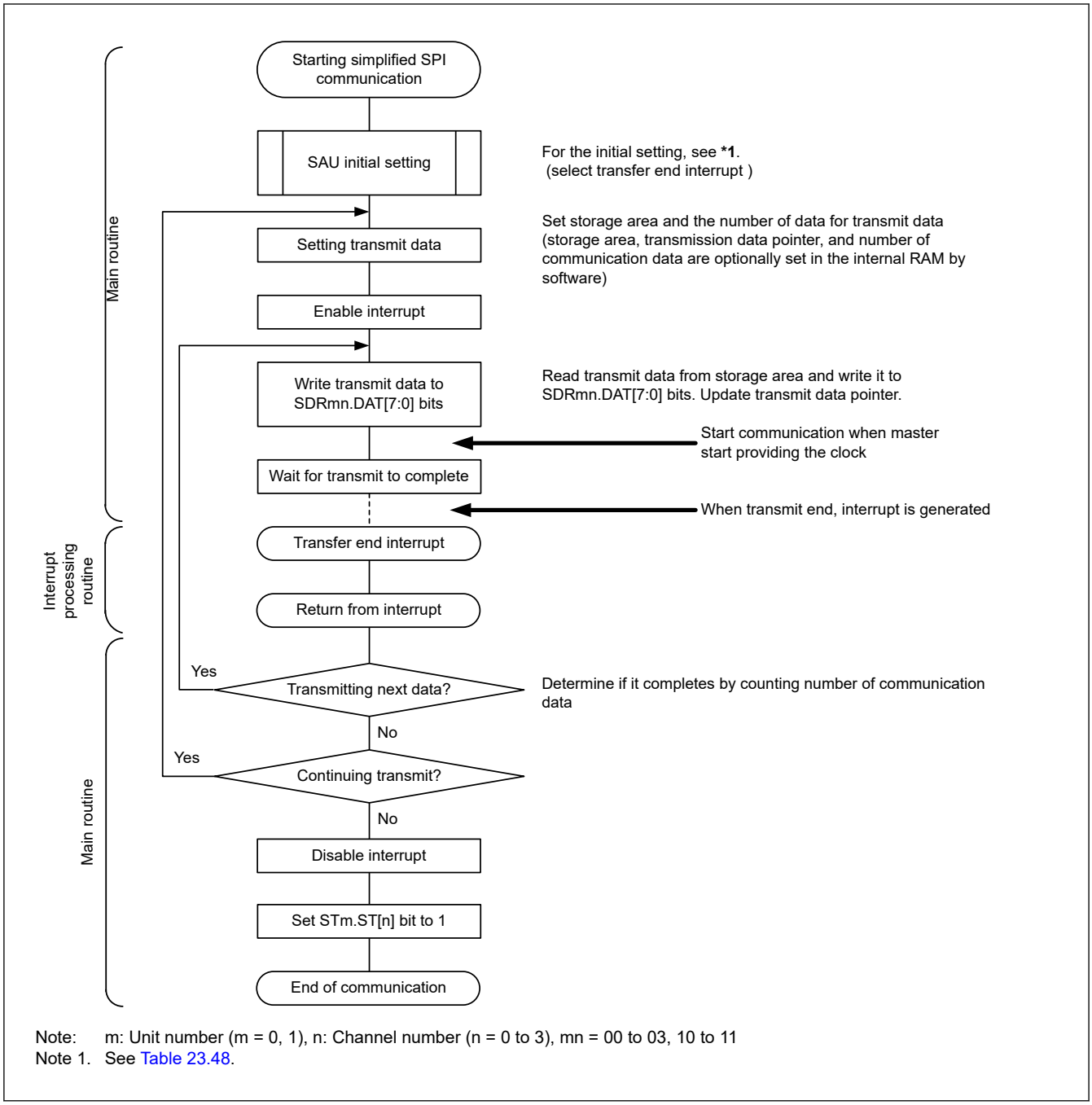
Figure 23.17 shows the timing of slave transmission (in single transmission mode) (type 1: SCRmn.DCP[1:0] = 00b).



Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: SPI number (p = 00, 01, 10, 11, 20, 21), mn = 00 to 03, 10 to 11

Figure 23.17 Timing of slave transmission (in single transmission mode) (type 1: SCRmn.DCP[1:0] = 00b)

Figure 23.18 shows the flowchart of slave transmission (in single transmission mode).

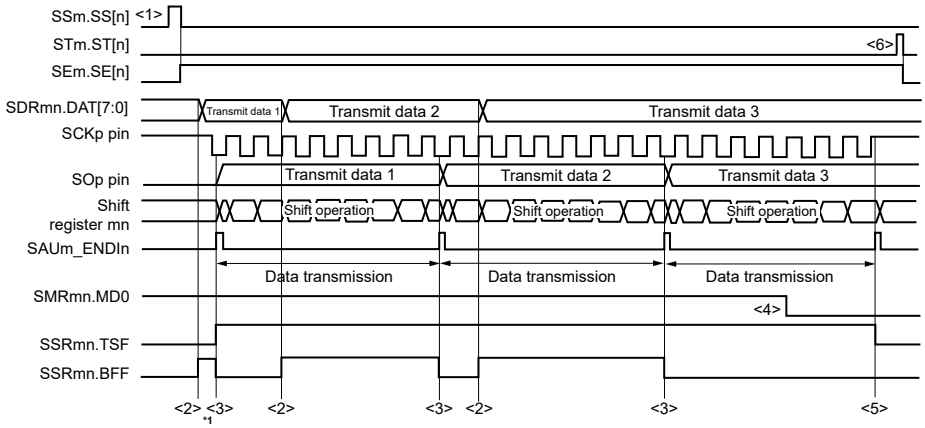


Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 11
Note 1. See Table 23.48.

Figure 23.18 Flowchart of slave transmission (in single transmission mode)

(4) Processing flow (in continuous transmission mode)

Figure 23.19 shows the timing of slave transmission (in continuous transmission mode) (type 1: SCRmn.DCP[1:0] = 00b).



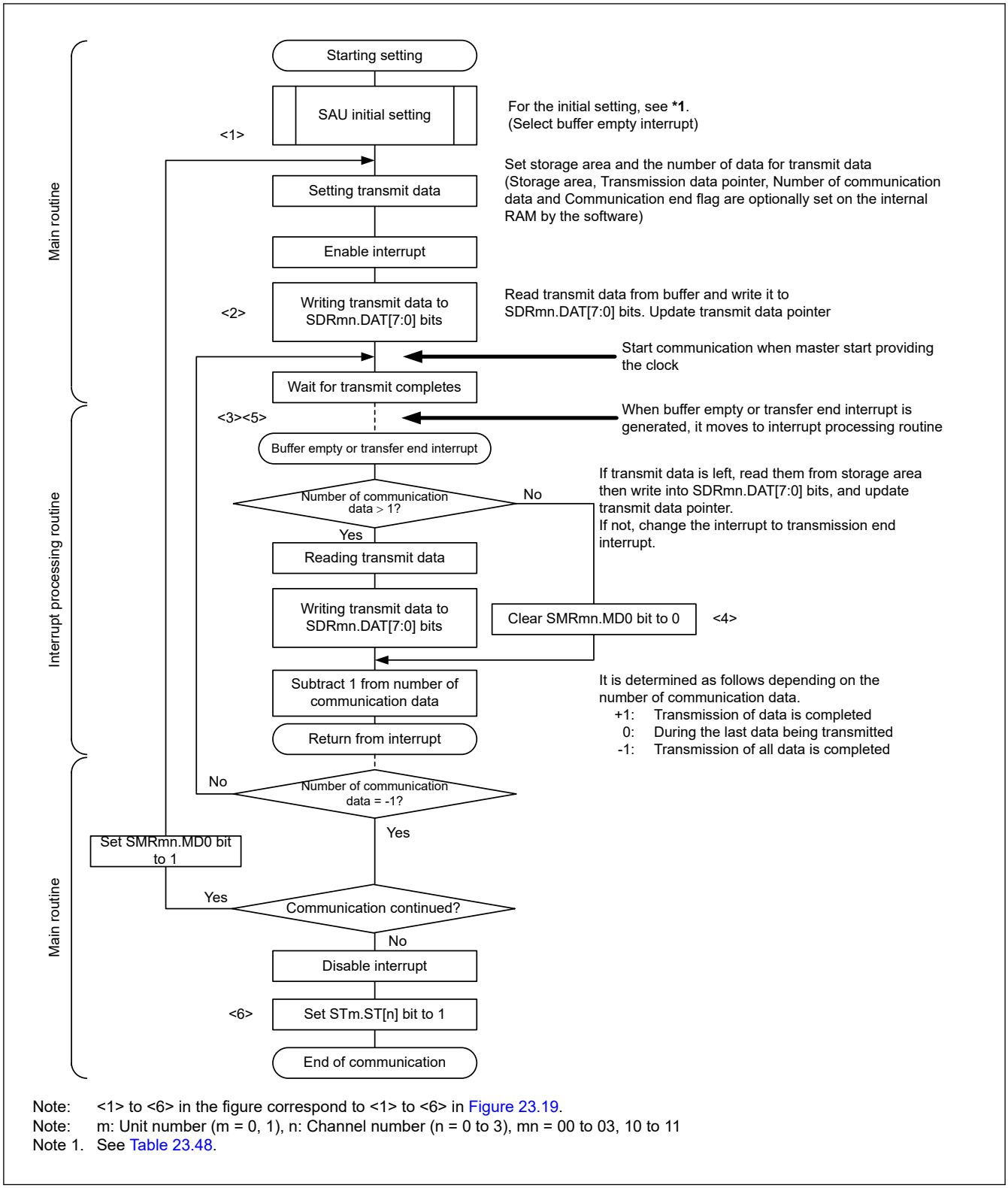
Note: The MD0 bit of serial mode register mn (SMRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started.

Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: SPI number (p = 00, 01, 10, 11, 20, 21), mn = 00 to 03, 10 to 11

Note 1. If transmit data is written to the SDRmn register while the BFF flag of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

Figure 23.19 Timing of slave transmission (in continuous transmission mode) (type 1: SCRmn.DCP[1:0] = 00b)

Figure 23.20 shows the flowchart of slave transmission (in continuous transmission mode).



Note: <1> to <6> in the figure correspond to <1> to <6> in Figure 23.19.
Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 11
Note 1. See Table 23.48.

Figure 23.20 Flowchart of slave transmission (in continuous transmission mode)

23.5.5 Slave Reception

Slave reception is when a microcontroller receives data from another device in the state of a transfer clock being input from another device.

Table 23.51 shows the specification of slave reception of simplified SPI.

Table 23.51 Specification of slave reception of simplified SPI

Simplified SPI	SPI00	SPI01	SPI10	SPI11	SPI20	SPI21
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1
Pins used	SCK00, SI00	SCK01, SI01	SCK10, SI10	SCK11, SI11	SCK20, SI20	SCK21, SI21
Interrupt	SAU0_ENDI0	SAU0_ENDI1	SAU0_ENDI2	SAU0_ENDI3	SAU1_ENDI0	SAU1_ENDI1
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)					
Error detection flag	Overrun error detection flag (SSRmn.OVF) only					
Transfer data length	7 or 8 bits					
Transfer rate	Max. $f_{MCK}/6$ [Hz]*1 *2					
Data phase	Selectable by the DCP[1] bit of the SCRmn register • DCP[1] = 0: Data input starts from the start of the operation of the serial clock. • DCP[1] = 1: Data input starts half a clock cycle before the start of the serial clock operation.					
Clock phase	Selectable by the DCP[0] bit of the SCRmn register • DCP[0] = 0: Non-reverse • DCP[0] = 1: Reverse					
Data direction	MSB or LSB first					

Note: f_{MCK}: Operation clock frequency of target channel
f_{SCK}: Serial clock frequency
Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 11
Note 1. Because the external serial clock input to the SCK00, SCK01, SCK10, SCK11, SCK20, and SCK21 pins is sampled internally and used, the fastest transfer rate is $f_{MCK}/6$ [Hz].
Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics specified in the electrical characteristics. For details, see section 37, Electrical Characteristics .

(1) Register setting

Table 23.52 to Table 23.57 show examples of the register contents for slave reception of simplified SPI.

(a) Serial mode register mn (SMRmn)

Table 23.52 Example of serial mode register mn (SMRmn) contents for slave reception of simplified SPI

Bit	Symbol	Set value	Function
0	MD0	0	Interrupt source of channel n 0: Transfer end interrupt
2:1	MD1[1:0]	00b	Setting of operation mode of channel n 0 0: Simplified SPI mode
5:3	—	100b	Setting disabled (set to the initial value)
6	SIS0	0	Setting is fixed in the simplified SPI mode
7	—	0	Setting disabled (set to the initial value)
8	STS	0	Selection of start trigger source 0: Only software trigger is valid (selected for simplified SPI, UART transmission, and simplified I^2C).
13:9	—	00000b	Setting disabled (set to the initial value)
14	CCS	1	Selection of transfer clock (f_{TCLK}) of channel n 1: Clock input f_{SCK} from the SCKp pin (slave transfer in simplified SPI mode)
15	CKS	0/1	Operation clock (f_{MCK}) of channel n 0: Prescaler output clock CKm0 set by the SPSm register 1: Prescaler output clock CKm1 set by the SPSm register

(b) Serial communication operation setting register mn (SCRmn)

Table 23.53 Example of serial communication operation setting register mn (SCRmn) contents for slave reception of simplified SPI

Bit	Symbol	Set value	Function
1:0	DLS [1:0]	10b or 11b	Setting of data length 1 0: 7-bit data length 1 1: 8-bit data length
3:2	—	01b	Setting disabled (set to the initial value)
5:4	SLC[1:0]	00b	Since this bit is dedicated to UART mode, it is fixed in the simplified SPI mode.
6	—	0	Setting disabled (set to the initial value)
7	DIR	0/1	Selection of data transfer sequence in simplified SPI and UART modes 0: Inputs or outputs data with MSB first. 1: Inputs or outputs data with LSB first.
9:8	PTC[1:0]	00b	Since this bit is dedicated to UART mode, it is fixed in the simplified SPI mode.
10	EOC	0	Since this bit is dedicated to UART receive modes, it is fixed in the simplified SPI mode.
11	—	0	Setting disabled (set to the initial value)
13:12	DCP[1:0]	00b to 11b	Selection of data and clock phase in simplified SPI mode For details about the setting, see section 23.3. Register Descriptions.
15:14	TRXE[1:0]	01b	Setting TRXE[1:0] = 01b is fixed in the simplified SPI slave reception mode

(c) Serial data register mn (SDRmn)

Read-only.

Table 23.54 Example of serial data register mn (SDRmn) contents for slave reception of simplified SPI

Bit	Symbol	Set value	Function
7:0	DAT[7:0]	0xFF	Receive data
8	DAT[8]	0	0 Fixed
15:9	STCLK[6:0]	0x00	Baud rate setting (Do not used in any slave mode)

(d) Serial output register m (SOm)

This register is not used in this mode.

Table 23.55 Example of serial output register m (SOm) contents for slave reception of simplified SPI

Bit	Symbol	Set value	Function
n	SO[n]	x	Bit that cannot be used in this mode (set to the initial value when not used in any mode)
n+8	CKO[n]	x	Bit that cannot be used in this mode (set to the initial value when not used in any mode)

(e) Serial output enable register m (SOEm)

This register is not used in this mode.

Table 23.56 Example of serial output enable register m (SOEm) contents for slave reception of simplified SPI

Bit	Symbol	Set value	Function
n	SOE[n]	x	Bit that cannot be used in this mode (set to the initial value when not used in any mode)

(f) Serial channel start register m (SSm)

Set only the bit of the target channel to 1.

Table 23.57 Example of serial channel start register m (SSm) contents for slave reception of simplified SPI

Bit	Symbol	Set value	Function
n	SS[n]	1	Operation start trigger of channel n 1: Set the SEm.SE[n] bit to 1 to place the channel in communication waiting state

Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: SPI number (p = 00, 01, 10, 11, 20, 21), mn = 00 to 03, 10 to 11

Note: ×: Bits not used with serial array units (depending on the settings of other peripheral functions)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Table 23.58 shows the procedure for initial setting of slave reception.

Table 23.58 Initial setting procedure for slave reception

Step		Process	Detail
Procedure for initial setting of slave reception	<1>	Starting initial setting	—
	<2>	Setting the SPSm register	Set the operation clock.
	<3>	Setting the SMRmn register	Set an operation mode.
	<4>	Setting the SCRmn register	Set a communication format.
	<5>	Setting the SDRmn register	Set the SDRmn.STCLK[6:0] bits to 0x00 for baud rate setting.
	<6>	Setting port	Enable data input and clock input of the target channel.
	<7>	Writing to the SSm register	Set the SSm.SS[n] bit of the target channel to 1 and set the SEm.SE[n] bit to 1 to enable operation. Wait for a clock from the master.
	<8>	Completing initial setting	—

Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 11

Table 23.59 shows the procedure for stopping slave reception.

Table 23.59 Procedure for stopping slave reception

Step		Process	Detail
Procedure for stopping slave reception	<1>	Start setting to stop	—
	<2>	Wait until the SSRmn.TSF flag is cleared (optional)	If there is any data being transferred, wait for their completion. If there is a requirement to stop, do not wait.
	<3>	Write the STm register	Write 1 to the STm.ST[n] bit of the target channel and set SEm.SE[n] = 0 to stop operation.
	<4>	Change setting of the SOEm register	Set the SOEm.SOE[n] bit to 0 and stop the output of the target channel.
	<5>	Stop setting is completed	After the stop setting is completed, go to the next processing.

Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 11

Table 23.60 shows the procedure for resuming slave reception.

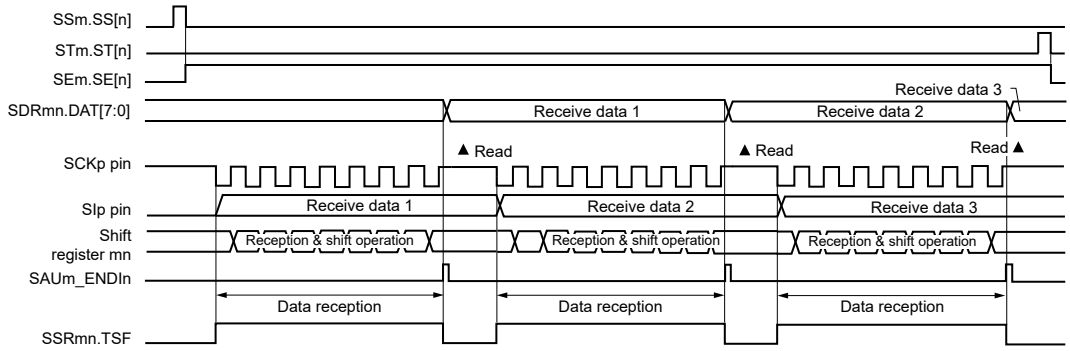
Table 23.60 Procedure for resuming slave reception

Step		Process	Detail
Procedure for resuming slave reception	<1>	Start setting for resumption	—
	<2>	Wait until completing master preparations	Wait until the communication target (master) stops or communication operation completed.
	<3>	Port manipulation	Disable data input and clock input of the target channel.
	<4>	Change setting of the SPSm register (optional)	Reset the register to change the operation clock setting.
	<5>	Change setting of the SMRmn register (optional)	Reset the register to change serial mode register mn (SMRmn) setting.
	<6>	Change setting of the SCRmn register (optional)	Reset the register to change serial communication operation setting register mn (SCRmn) setting.
	<7>	Clearing error flag (optional)	If the SSRmn.OVF flag remains set, clear this using serial flag clear trigger register mn (SIRmn).
	<8>	Port manipulation	Enable clock input of the target channel.
	<9>	Writing to the SSm register	Set the SSm.SS[n] bit of the target channel to 1 and set SEm.SE[n] = 1 to enable operation. Wait for a clock from the master.
	<10>	Completing resumption setting	—

Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 11

(3) Processing flow (in single reception mode)

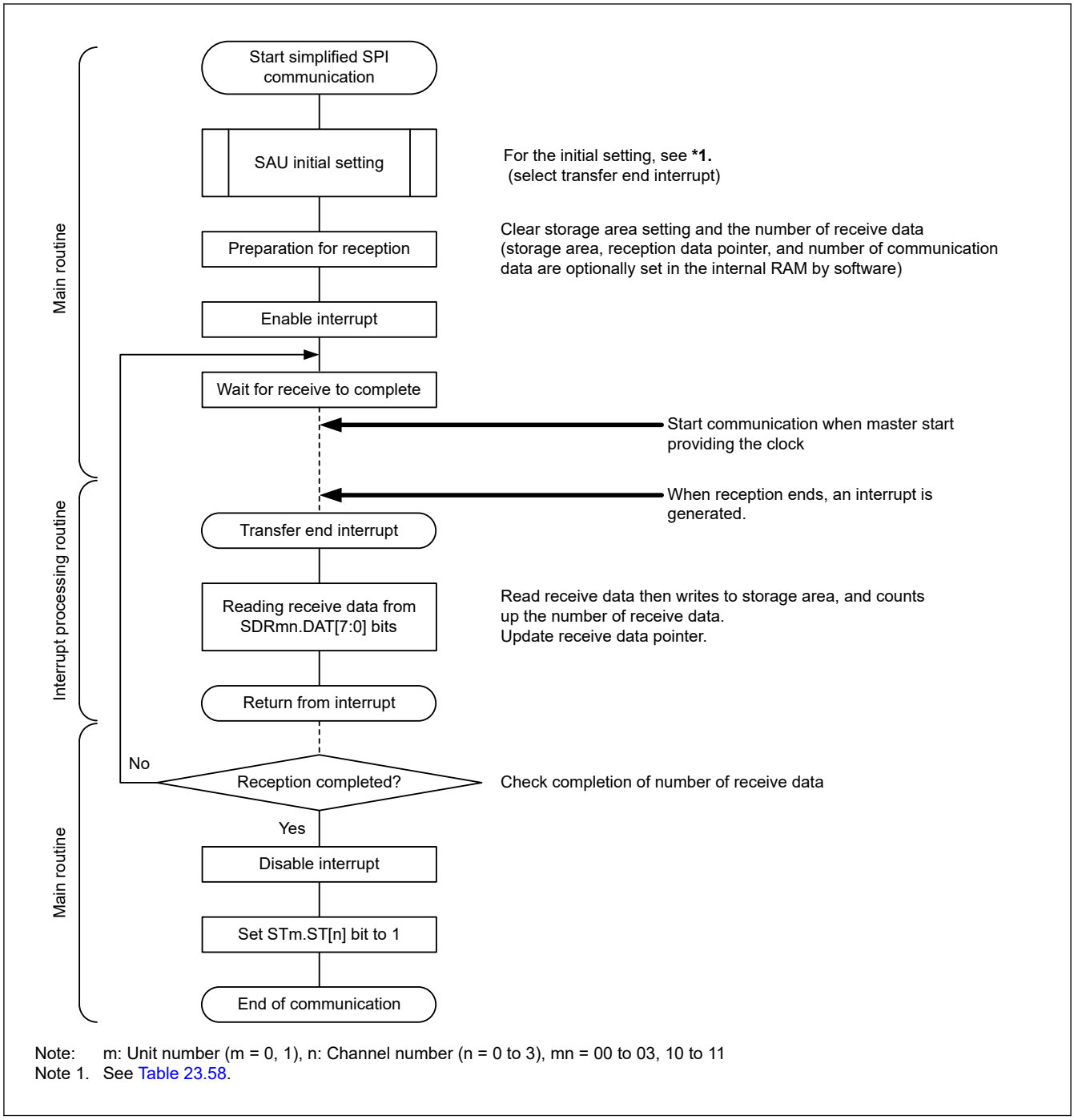
Figure 23.21 shows the timing of slave reception (in single reception mode) (type 1: SCRmn.DCP[1:0] = 00b).



Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: SPI number (p = 00, 01, 10, 11, 20, 21), mn = 00 to 03, 10 to 11

Figure 23.21 Timing of slave reception (in single reception mode) (type 1: SCRmn.DCP[1:0] = 00b)

Figure 23.22 shows the flowchart of slave reception (in single reception mode).



Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 11
Note 1. See Table 23.58.

Figure 23.22 Flowchart of slave reception (in single reception mode)

23.5.6 Slave Transmission and Reception

Slave transmission and reception is when a microcontroller transmits and receives data to and from another device in the state of a transfer clock being input from another device.

Table 23.61 shows the specification of slave transmission and reception of simplified SPI.

Table 23.61 Specification of slave transmission and reception of simplified SPI (1 of 2)

Simplified SPI	SPI00	SPI01	SPI10	SPI11	SPI20	SPI21
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1

Table 23.61 Specification of slave transmission and reception of simplified SPI (2 of 2)

Simplified SPI	SPI00	SPI01	SPI10	SPI11	SPI20	SPI21
Pins used	SCK00, SI00, SO00	SCK01, SI01, SO01	SCK10, SI10, SO10	SCK11, SI11, SO11	SCK20, SI20, SO20	SCK21, SI21, SO21
Interrupt	SAU0_ENDI0	SAU0_ENDI1	SAU0_ENDI2	SAU0_ENDI3	SAU1_ENDI0	SAU1_ENDI1
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.					
Error detection flag	Overrun error detection flag (SSRmn.OVF) only					
Transfer data length	7 or 8 bits					
Transfer rate	Max. f_{MCK}/6 [Hz]*1 *2					
Data phase	Selectable by the DCP[1] bit of the SCRmn register • DCP[1] = 0: Data I/O starts at the start of the operation of the serial clock. • DCP[1] = 1: Data I/O starts half a clock cycle before the start of the serial clock operation.					
Clock phase	Selectable by the DCP[0] bit of the SCRmn register • DCP[0] = 0: Non-reverse • DCP[0] = 1: Reverse					
Data direction	MSB or LSB first					

Note: f_{MCK}: Operation clock frequency of target channel
f_{SCK}: Serial clock frequency
Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 11
Note 1. Because the external serial clock input to the SCK00, SCK01, SCK10, SCK11, SCK20, and SCK21 pins is sampled internally and used, the fastest transfer rate is f_{MCK}/6 [Hz].
Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics specified in the electrical characteristics. For details, see section 37, Electrical Characteristics .

(1) Register setting

Table 23.62 to Table 23.67 show examples of the register contents for slave transmission and reception of simplified SPI.

(a) Serial mode register mn (SMRmn)

Table 23.62 Example of serial mode register mn (SMRmn) contents for slave transmission and reception of simplified SPI

Bit	Symbol	Set value	Function
0	MD0	0/1	Interrupt source of channel n 0: Transfer end interrupt 1: Buffer empty interrupt
2:1	MD1[1:0]	00b	Setting of operation mode of channel n 0 0: Simplified SPI mode
5:3	—	100b	Setting disabled (set to the initial value)
6	SIS0	0	Setting is fixed in the simplified SPI mode
7	—	0	Setting disabled (set to the initial value)
8	STS	0	Selection of start trigger source 0: Only software trigger is valid (selected for simplified SPI, UART transmission, and simplified I^2C).
13:9	—	00000b	Setting disabled (set to the initial value)
14	CCS	1	Selection of transfer clock (f_{TCLK}) of channel n 1: Clock input f_{SCK} from the SCKp pin (slave transfer in simplified SPI mode)
15	CKS	0/1	Operation clock (f_{MCK}) of channel n 0: Prescaler output clock CKm0 set by the SPSm register 1: Prescaler output clock CKm1 set by the SPSm register

(b) Serial communication operation setting register mn (SCRmn)

Table 23.63 Example of serial communication operation setting register mn (SCRmn) contents for slave transmission and reception of simplified SPI

Bit	Symbol	Set value	Function
1:0	DLS[1:0]	10b or 11b	Setting of data length 1 0: 7-bit data length 1 1: 8-bit data length
3:2	—	01b	Setting disabled (set to the initial value)
5:4	SLC[1:0]	00b	Since this bit is dedicated to UART mode, it is fixed in the simplified SPI mode.
6	—	0	Setting disabled (set to the initial value)
7	DIR	0/1	Selection of data transfer sequence in simplified SPI and UART modes 0: Inputs or outputs data with MSB first 1: Inputs or outputs data with LSB first
9:8	PTC[1:0]	00b	Since this bit is dedicated to UART mode, it is fixed in the simplified SPI mode.
10	EOC	0	Since this bit is dedicated to UART receive modes, it is fixed in the simplified SPI mode.
11	—	0	Setting disabled (set to the initial value)
13:12	DCP[1:0]	00b to 11b	Selection of data and clock phase in simplified SPI mode. For details about the setting, see section 23.3. Register Descriptions.
15:14	TRXE[1:0]	11b	Setting TRXE[1:0] = 11b is fixed in the simplified SPI master transmission and reception mode

(c) Serial data register mn (SDRmn)

Table 23.64 Example of serial data register mn (SDRmn) contents for slave transmission and reception of simplified SPI

Bit	Symbol	Set value	Function
7:0	DAT[7:0]	0xFF	Transmit data or Receive data (Transmit data setting and receive data read)
8	DAT[8]	0	0 Fixed
15:9	STCLK[6:0]	0x00	Baud rate setting (do not used in any slave mode)

(d) Serial output register m (SOm)

Set only the bit of the target channel.

Table 23.65 Example of serial output register m (SOm) contents for slave transmission and reception of simplified SPI

Bit	Symbol	Set value	Function
n	SO[n]	0/1	Serial data output of channel 0: Serial data output value is 0 1: Serial data output value is 1
n + 8	CKO[n]	x	Bit that cannot be used in this mode (set to the initial value when not used in any mode)

(e) Serial output enable register m (SOEm)

Set only the bit of the target channel to 1.

Table 23.66 Example of serial output enable register m (SOEm) contents for slave transmission and reception of simplified SPI

Bit	Symbol	Set value	Function
n	SOE[n]	1	Serial output enable or stop of channel n 1: Enable output by serial communication operation.

(f) Serial channel start register m (SSm)

Set only the bit of the target channel to 1.

Table 23.67 Example of serial channel start register m (SSm) contents for slave transmission and reception of simplified SPI

Bit	Symbol	Set value	Function
n	SS[n]	1	Operation start trigger of channel n 1: Set the SEm.SE[n] bit to 1 to place the channel in the communications waiting state.

Note: Be sure to set transmit data to the SIOp register before the clock from the master is started.

Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: SPI number (p = 00, 01, 10, 11, 20, 21), mn = 00 to 03, 10 to 11

Note: ×: Bits not used with serial array units (depending on the settings of other peripheral functions)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Table 23.68 shows the procedure for initial setting of slave transmission and reception.

Table 23.68 Initial setting procedure for slave transmission and reception

Step		Process	Detail
Procedure for initial setting of slave transmission and reception	<1>	Starting initial setting	—
	<2>	Setting the SPSm register	Set the operation clock.
	<3>	Setting the SMRmn register	Set an operation mode.
	<4>	Setting the SCRmn register	Set a communication format.
	<5>	Setting the SDRmn register	Set the SDRmn.STCLK[6:0] bits to 0x00 for baud rate setting.
	<6>	Setting the SOm register	Set the initial output level of the serial data (SOm.SO[n]).
	<7>	Changing setting of the SOEm register	Set the SOEm.SOE[n] bit to 1 and enable data output of the target channel.
	<8>	Setting port	Enable data output of the target channel
	<9>	Writing to the SSm register	Set the SSm.SS[n] bit of the target channel to 1 and set SEm.SE[n] = 1 to enable operation.
	<10>	Completing initial setting	Initial setting is completed. Set transmit data to the SDRmn.DAT[7:0] bits and wait for a clock from the master

Note: Be sure to set transmit data to the SIOp register before the clock from the master is started.
Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 11

Table 23.69 shows the procedure for stopping slave transmission and reception.

Table 23.69 Procedure for stopping slave transmission and reception

Step		Process	Detail
Procedure for stopping slave transmission and reception	<1>	Starting setting to stop	—
	<2>	Wait until the SSRmn.TSF flag is cleared (optional)	If there is any data being transferred, wait for their completion. If there is a requirement to stop, do not wait.
	<3>	Writing the STm register	Write 1 to the STm.ST[n] bit of the target channel and set SEm.SE[n] = 0 to stop operation.
	<4>	Changing setting of the SOEm register	Set the SOEm.SOE[n] bit to 0 and stop the output of the target channel.
	<5>	Changing setting of the SOm register (optional)	The levels of the serial data (SOm.SO[n]) on the target channel can be changed if required.
	<6>	Stop setting is completed	After the stop setting is completed, go to the next processing.

Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 11

Table 23.70 shows the procedure for resuming slave transmission and reception.

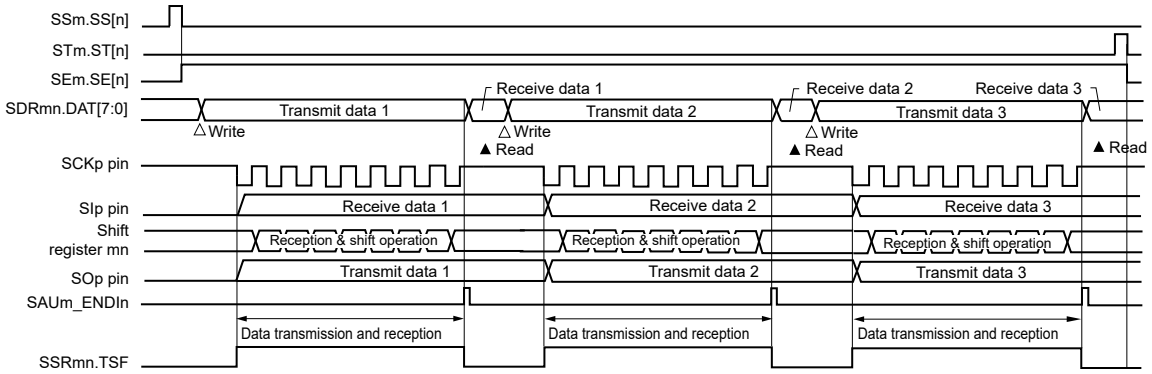
Table 23.70 Procedure for resuming slave transmission and reception

Step		Process	Detail
Procedure for resuming slave transmission and reception	<1>	Starting setting for resumption	—
	<2>	Wait until completing master preparations	Wait until the communication target (master) stops or communication operation completed.
	<3>	Port manipulation	Disable data output of the target channel.
	<4>	Changing setting of the SPSm register (optional)	Reset the register to change the operation clock setting.
	<5>	Changing setting of the SMRmn register (optional)	Reset the register to change serial mode register mn (SMRmn) setting.
	<6>	Changing setting of the SCRmn register (optional)	Reset the register to change serial communication operation setting register mn (SCRmn) setting.
	<7>	Clearing error flag (optional)	If the SSRmn.OVF flag remains set, clear this using serial flag clear trigger register mn (SIRmn).
	<8>	Changing setting of the SOEm register (optional)	Set the SOEm.SOE[n] bit to 0 to stop output from the target channel.
	<9>	Changing setting of the SOm register (optional)	Set the initial output level of the serial data (SOm.SO[n]).
	<10>	Changing setting of the SOEm register (optional)	Set the SOEm.SOE[n] bit to 1 and enable output from the target channel.
	<11>	Port manipulation	Enable data output of the target channel.
	<12>	Writing to the SSm register	Set the SSm.SS[n] bit of the target channel to 1 and set SEm.SE[n] = 1 to enable operation.
	<13>	Starting communication	Set transmit data to the SDRmn.DAT[7:0] bits and wait for a clock from the master
	<14>	Completing resumption setting	—

Note: Be sure to set transmit data to the SIOp register before the clock from the master is started.
Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 11

(3) Processing flow (in single transmission and reception mode)

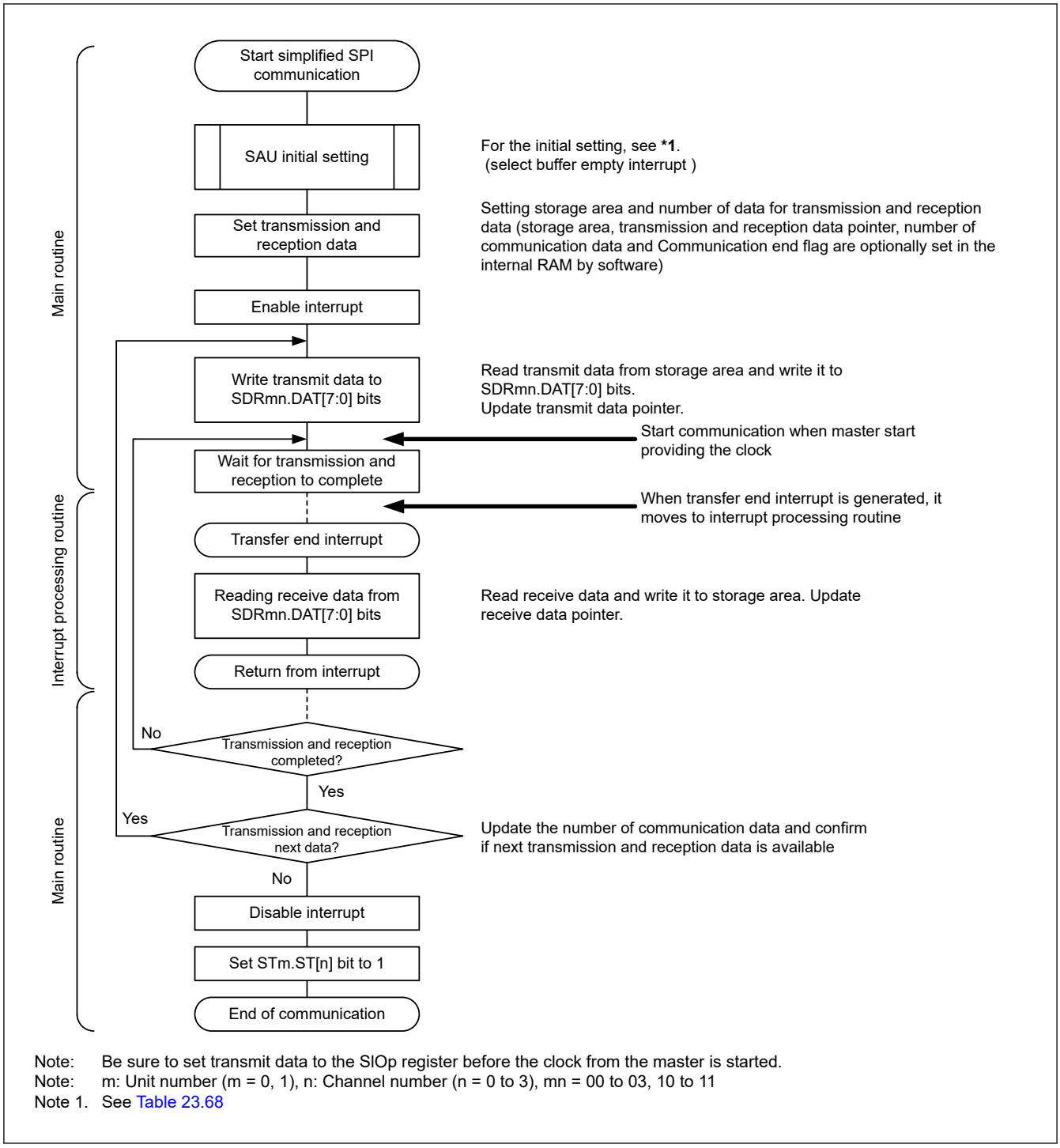
Figure 23.23 shows the timing of slave transmission and reception (in single transmission and reception mode) (type 1: SCRmn.DCP[1:0] = 00b).



Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: SPI number (p = 00, 01, 10, 11, 20, 21), mn = 00 to 03, 10 to 11

Figure 23.23 Timing of slave transmission and reception (in single transmission and reception mode)

Figure 23.24 shows the flowchart of slave transmission and reception (in single transmission and reception mode).

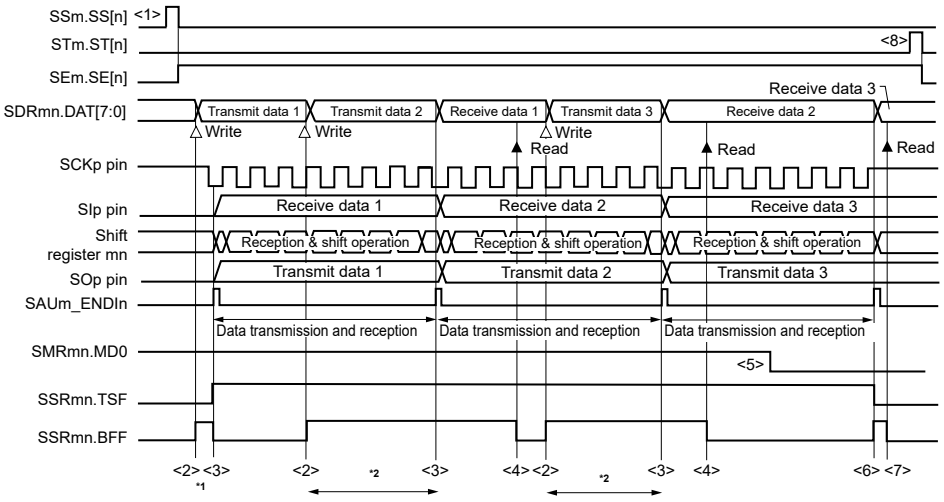


Note: Be sure to set transmit data to the SIOp register before the clock from the master is started.
Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 11
Note 1. See Table 23.68

Figure 23.24 Flowchart of slave transmission and reception (in single transmission and reception mode)

(4) Processing flow (in continuous transmission and reception mode)

Figure 23.25 shows the timing of slave transmission and reception (in continuous transmission and reception mode) (type 1: SCRmn.DCP[1:0] = 00b).



Note: The MD0 bit of serial mode register mn (SMRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last transmit data.

Note: <1> to <8> in the figure correspond to <1> to <8> in Figure 23.26.

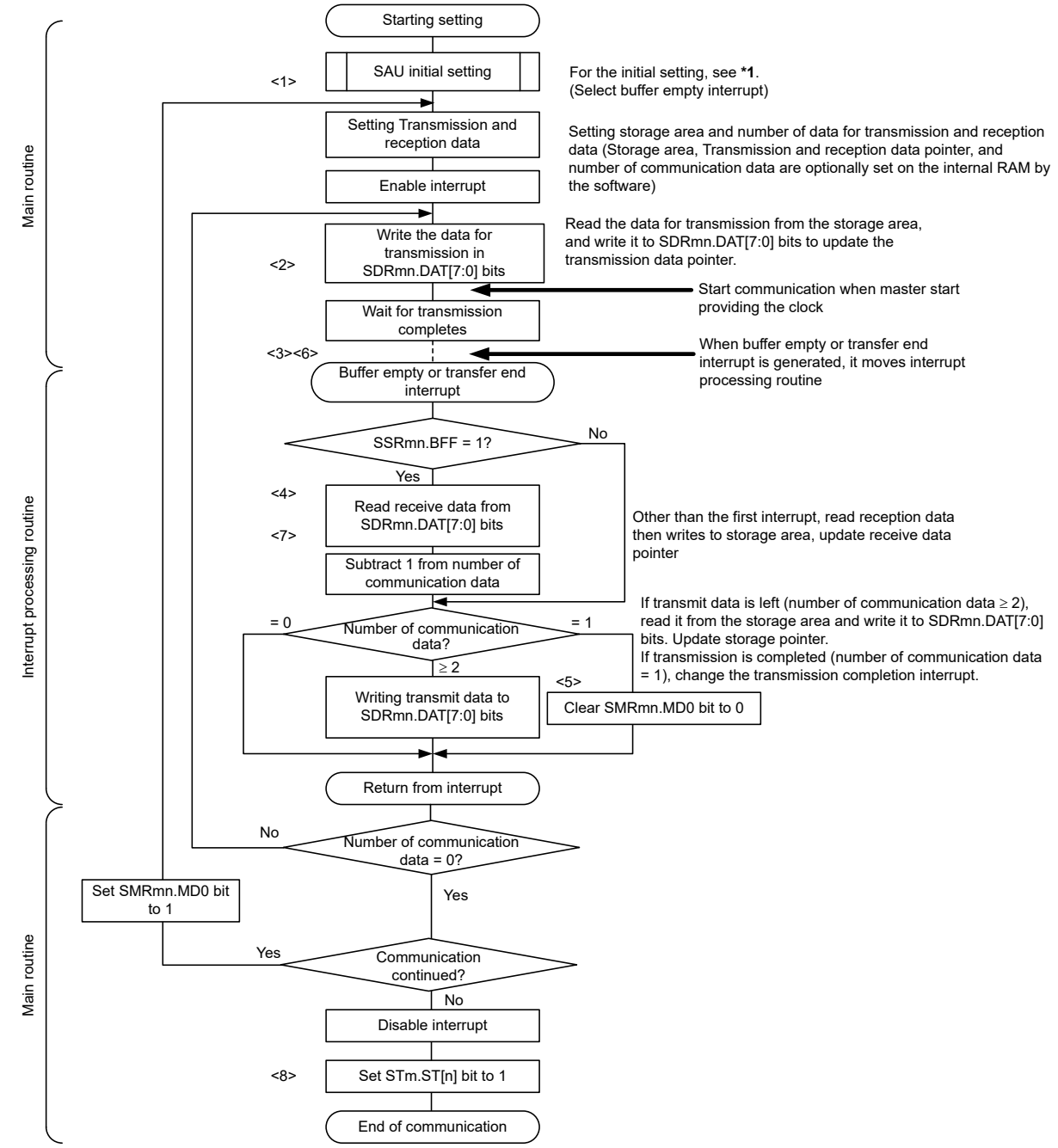
Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: SPI number (p = 00, 01, 10, 11, 20, 21) mn = 00 to 03, 10 to 11

Note 1. If transmit data is written to the SDRmn register while the BFF flag of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

Note 2. The transmit data can be read by reading the SDRmn register during this period. At this time, the transfer operation is not affected.

Figure 23.25 Timing of slave transmission and reception (in continuous transmission and reception mode) (type 1: SCRmn.DCP[1:0] = 00b)

Figure 23.26 shows the flowchart of slave transmission and reception (in continuous transmission and reception mode).



Note: Be sure to set transmit data to the SIOp register before the clock from the master is started.
Note: <1> to <8> in the figure correspond to <1> to <8> in Figure 23.25
Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 11
Note 1. See Table 23.68.

Figure 23.26 Flowchart of slave transmission and reception (in continuous transmission and reception mode)

23.5.7 Snooze Mode Function

The Snooze mode makes the simplified SPI perform reception operations on SCKp pin input detection while in the Software Standby mode. Normally the simplified SPI stops communication in the Software Standby mode. However, using the Snooze mode enables the simplified SPI to perform reception operations without CPU operation on detection of the SCKp pin input. Only SPI00 and SPI20 channels can be set to the Snooze mode.

When using the simplified SPI in Snooze mode, make the following setting before switching to the Software Standby mode (see Figure 23.28 and Figure 23.30.)

- When using the Snooze mode function, set the SWC bit of serial standby control register m (SSCm) to 1 just before switching to the Software Standby mode. After the initial setting has been completed, set the SS[0] bit of serial channel start register m (SSm) to 1.
- The CPU shifts to the Snooze mode on detecting the valid edge of the SCKp signal following a transition to the Software Standby mode.
  An SPIp starts reception on detecting input of the serial clock on the SCKp pin.

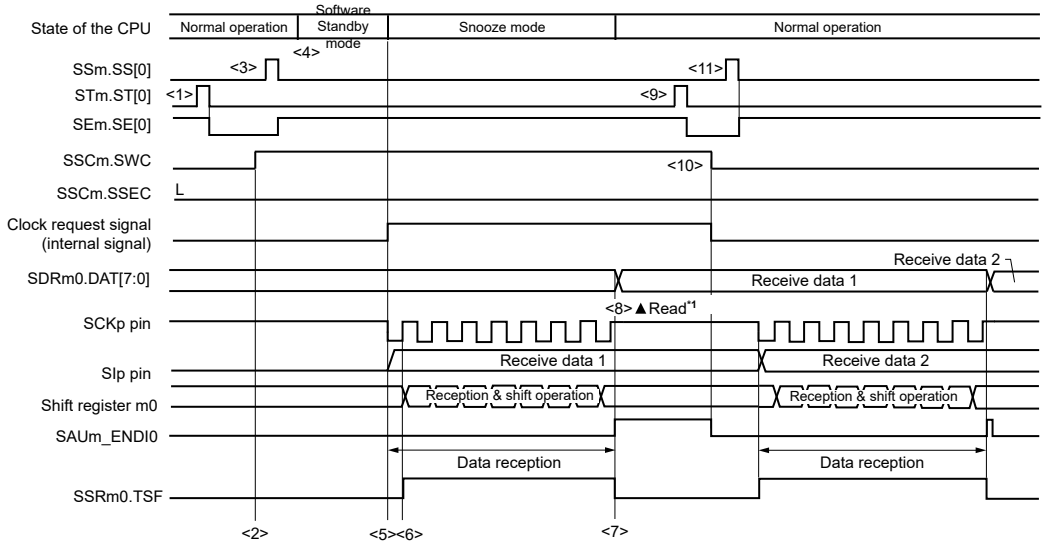
Note: The Snooze mode can only be specified when the high-speed on-chip oscillator clock (HOCO) or middle-speed on-chip oscillator clock (MOCO) is selected for PCLKB.

Note: The maximum transfer rate when using SPIp in the Snooze mode is 0.5 Mbps.

Note: m: Unit number (m = 0, 1), p: SPI number (p = 00, 20)

(1) Snooze mode operation (on startup)

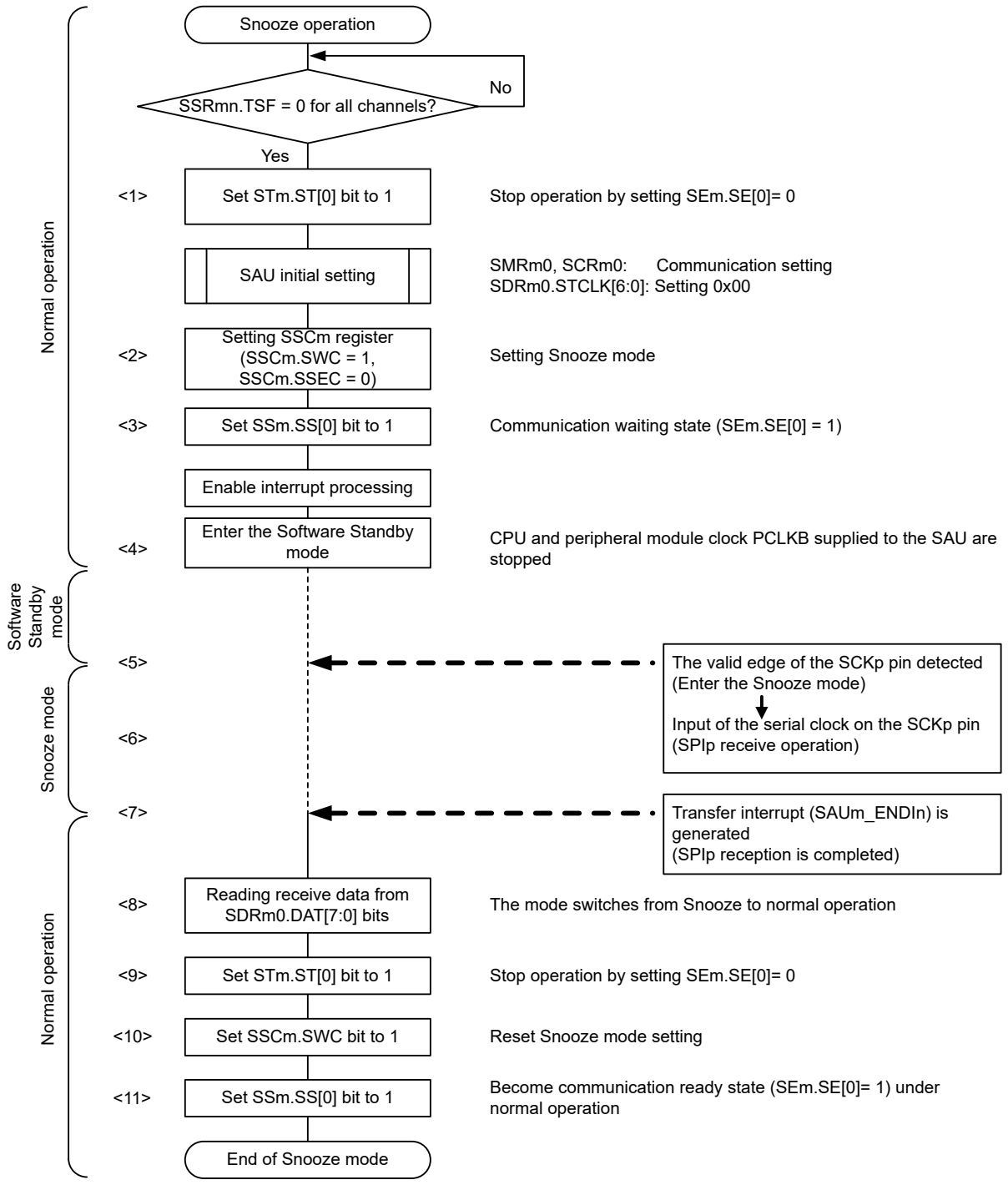
Figure 23.27 shows the timing of Snooze mode operation (on startup) (Type 1: SCRmn.DCP[1:0] = 00b).



Note: m: Unit number (m = 0, 1), p: SPI number (p = 00, 20)
Note: Before switching to the Snooze mode or after reception operation in the Snooze mode finishes, set the STm.ST[0] bit to 1 (the SEm.SE[0] bit is cleared and the operation stops). After the receive operation completes, clear the SSCm.SWC bit to 0 (Snooze mode release).
Note: When SSCm.SWC = 1, the SSRm0.BFF and OVF flags do not change.
Note: <1> to <11> in the figure correspond to <1> to <11> in Figure 23.28.
Note 1. Only read received data while SSCm.SWC = 1 and before the next valid edge of the SCKp pin input is detected.

Figure 23.27 Timing of Snooze mode operation (on startup) (type 1: SCRmn.DCP[1:0] = 00b)

Figure 23.28 shows the flowchart of Snooze mode operation (on startup).

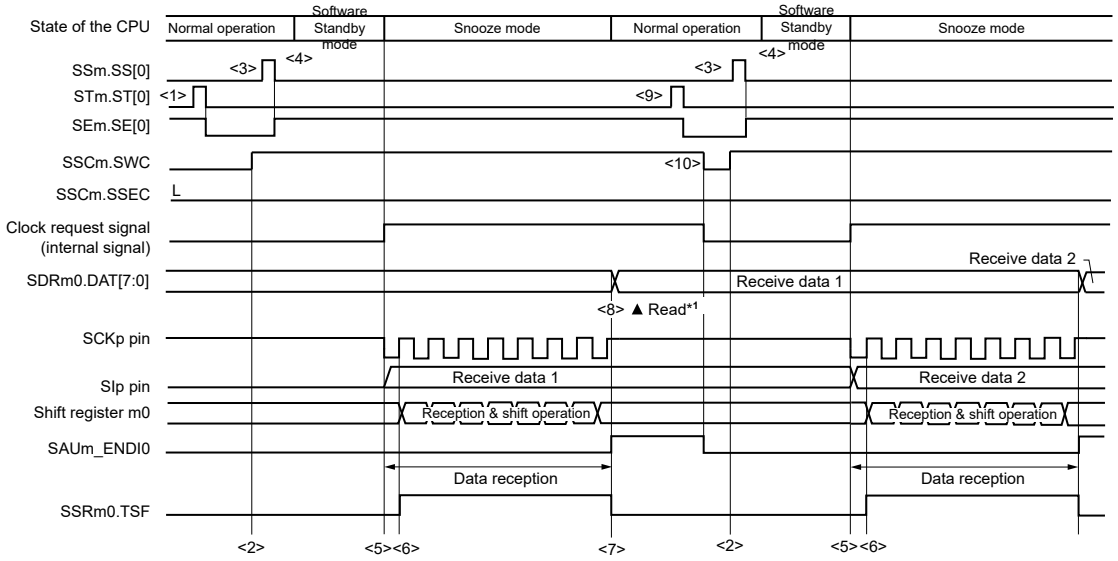


Note: <1> to <11> in the figure correspond to <1> to <11> in Figure 23.27.
Note: m: Unit number (m = 0, 1), p: SPI number (p = 00, 20)

Figure 23.28 Flowchart of Snooze mode operation (on startup)

(2) Snooze mode operation (continuous startup)

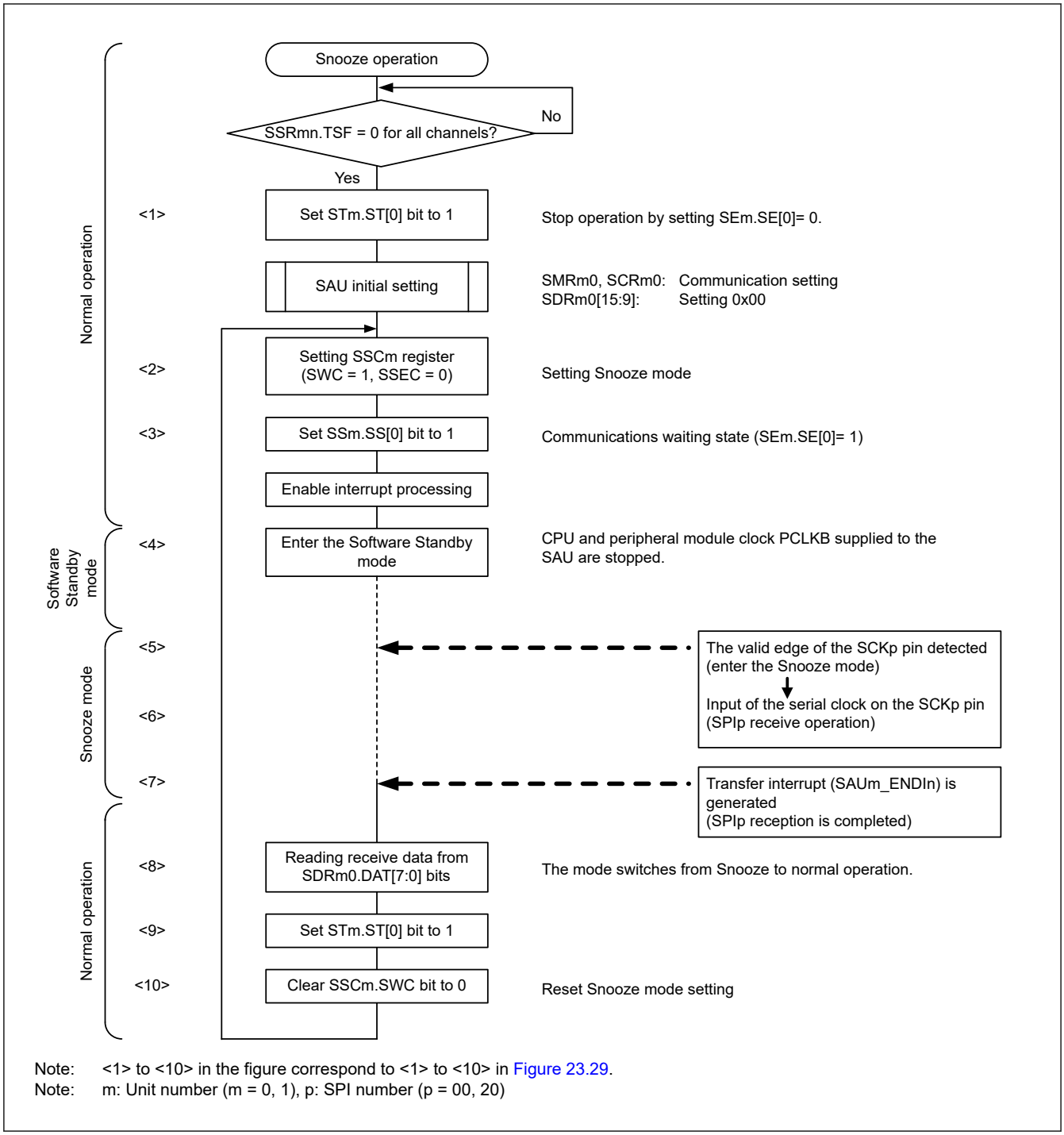
Figure 23.29 shows the timing of Snooze mode operation (continuous startup) (type 1: SCRmn.DCP[1:0] = 00b).



Note: m: Unit number (m = 0, 1), p: SPI number (p = 00, 20)
Note: Before switching to the Snooze mode or after reception operation in the Snooze mode finishes, set the STm.ST[0] bit to 1 (the SEm.SE[0] bit is cleared and the operation stops). After the receive operation completes, clear the SSCm.SWC bit to 0 (Snooze mode release).
Note: When SSCm.SWC = 1, the SSRm0.BFF and OVF flags do not change.
Note: <1> to <10> in the figure correspond to <1> to <10> in Figure 23.30.
Note 1. Only read received data while SSCm.SWC = 1 and before the next valid edge of the SCKp pin input is detected.

Figure 23.29 Timing of Snooze mode operation (continuous startup) (type 1: SCRmn.DCP[1:0] = 00b)

Figure 23.30 shows the flowchart of Snooze mode operation (continuous startup).



Note: <1> to <10> in the figure correspond to <1> to <10> in Figure 23.29.
Note: m: Unit number (m = 0, 1), p: SPI number (p = 00, 20)

Figure 23.30 Flowchart of Snooze mode operation (continuous startup)

23.5.8 Calculating Transfer Clock Frequency

The transfer clock frequency for simplified SPI communication can be calculated by the following expressions.

1. Master
   (Transfer clock frequency) = {Operation clock (f_{MCK}) frequency of target channel} ÷ (SDRmn.STCLK[6:0] + 1) ÷ 2 [Hz]
2. Slave
   (Transfer clock frequency) = {Frequency of serial clock (SCK) supplied by master}*1 [Hz]

Note 1. The permissible maximum transfer clock frequency is f_{MCK}/6.

The operation clock (f_{MCK}) is determined by serial clock select register m (SPSm) and CKS bit of serial mode register mn (SMRmn) as shown in Table 23.71.

Table 23.71 Selection of operation clock for simplified SPI, UART and simplified I^2C

SMRmn register	SPSm register		Operation clock (f_{MCK}) *1		
CKS	PRS1[3:0]	PRS0[3:0]	PCLKB/2^n	PCLKB = 32 MHz	PCLKB = 48 MHz
0	0xX	0x0	PCLKB	32 MHz	Setting prohibited
		0x1	PCLKB/2	16 MHz	24 MHz
		0x2	PCLKB/2^2	8 MHz	12 MHz
		0x3	PCLKB/2^3	4 MHz	6 MHz
		0x4	PCLKB/2^4	2 MHz	3 MHz
		0x5	PCLKB/2^5	1 MHz	1.5 MHz
		0x6	PCLKB/2^6	500 kHz	750 kHz
		0x7	PCLKB/2^7	250 kHz	375 kHz
		0x8	PCLKB/2^8	125 kHz	188 kHz
		0x9	PCLKB/2^9	62.5 kHz	93.8 kHz
		0xA	PCLKB/2^{10}	31.25 kHz	46.9 kHz
		0xB	PCLKB/2^{11}	15.63 kHz	23.4 kHz
		0xC	PCLKB/2^{12}	7.81 kHz	11.7 kHz
		0xD	PCLKB/2^{13}	3.91 kHz	5.86 kHz
		0xE	PCLKB/2^{14}	1.95 kHz	2.93 kHz
		0xF	PCLKB/2^{15}	977 Hz	1.46 kHz
1	0x0	0xX	PCLKB	32 MHz	Setting prohibited
	0x1		PCLKB/2	16 MHz	24 MHz
	0x2		PCLKB/2^2	8 MHz	12 MHz
	0x3		PCLKB/2^3	4 MHz	6 MHz
	0x4		PCLKB/2^4	2 MHz	3 MHz
	0x5		PCLKB/2^5	1 MHz	1.5 MHz
	0x6		PCLKB/2^6	500 kHz	750 kHz
	0x7		PCLKB/2^7	250 kHz	375 kHz
	0x8		PCLKB/2^8	125 kHz	188 kHz
	0x9		PCLKB/2^9	62.5 kHz	93.8 kHz
	0xA		PCLKB/2^{10}	31.25 kHz	46.9 kHz
	0xB		PCLKB/2^{11}	15.63 kHz	23.4 kHz
	0xC		PCLKB/2^{12}	7.81 kHz	11.7 kHz
	0xD		PCLKB/2^{13}	3.91 kHz	5.86 kHz
	0xE		PCLKB/2^{14}	1.95 kHz	2.93 kHz
	0xF		PCLKB/2^{15}	977 kHz	1.46 kHz
Other than above*2				Setting prohibited	Setting prohibited

Note: X: Don't care
Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 11
Note 1. When changing the clock selected for PCLKB, do so after having stopped (serial channel stop register m (STm) = 0x000F) the operation of the serial array unit (SAU).
Note 2. In the Simplified I^2C mode, setting the value greater than 0xB is prohibited.

23.5.9 Procedure for Processing Errors that Occurred During Simplified SPI Communication

The procedure for processing errors that occurred during simplified SPI communication is described in Table 23.72.

Table 23.72 Processing procedure in case of overrun error

Step	Software Manipulation		State of the Hardware	Remark
<1>	Reads serial data register mn (SDRmn).	→	The BFF flag of the SSRmn register is set to 0 and channel n is enabled to receive data.	This is to prevent an overrun error if the next reception is completed during error processing.
<2>	Reads serial status register mn (SSRmn).		—	The error type is identified and the read value is used to clear the error flag.
<3>	Writes 1 to serial flag clear trigger register mn (SIRmn).	→	The error flag is cleared.	The error only during reading can be cleared, by writing the value read from the SSRmn register to the SIRmn register without modification.

Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 11

23.6 Operation of UART Communication

This is a start/stop synchronization communication function using two lines: serial data transmission (TxD) and serial data reception (RxD) lines. By using these two communication lines, each data frame, which consist of a start bit, data, parity bit, and stop bit, is transferred asynchronously (using the internal baud rate) between the microcontroller and the other communication party. Full-duplex asynchronous UART communication can be performed by using a channel dedicated to transmission (even-numbered channel) and a channel dedicated to reception (odd-numbered channel). The LIN-bus can be implemented by using UART2, timer array unit (channel 7), and an external interrupt (IRQ2).

[Data transmission and reception]

- Data length of 7, 8, or 9 bits*1
- MSB or LSB first selectable
- Level setting of transmit and receive data (selecting whether to reverse the level)
- Parity bit appending and parity check functions
- Stop bit appending, stop bit check function

[Interrupt function]

- Transfer end interrupt or buffer empty interrupt
- Error interrupt in case of framing error, parity error, or overrun error

[Error detection flag]

- Framing error, parity error, or overrun error

In addition, UART reception of following channels supports the Snooze mode. In the Snooze mode, data can be received without CPU processing upon detecting RxD input in the Software Standby mode. The Snooze mode is only available in UART0 and UART2, which support the reception baud rate adjustment function.

The LIN-bus is accepted in UART2 (channels 0 and 1 of unit 1).

[LIN-bus functions]

- Wakeup signal detection
- Break field (BF) detection
- Sync field measurement, baud rate calculation

} Using the external interrupt (IRQ2) and timer array unit (channel 7)

Note 1. Only the following UARTs support the 9-bit data length:

- 16-, 24-, and 32-pin products: UART0
- 48-pin products: UART0 and UART2

When the middle-speed on-chip oscillator clock (MOCO) or low-speed on-chip oscillator clock (LOCO) is selected for PCLKB, use the MOCO User Trimming Control Register (MOCOUTCR) and the LOCO User Trimming Control Register (LOCOUTCR).

- UART0 uses channels 0 and 1 of SAU0
- UART1 uses channels 2 and 3 of SAU0
- UART2 uses channels 0 and 1 of SAU1.

See Table 23.1 to Table 23.3.

Select a single function for each channel. Only the selected function is possible. If UART0 is selected for channels 0 and 1 of unit 0, for example, the SPI00 and SPI01 functions cannot be used. At this time, however, channel 2 or 3 of the same unit can be used for a function other than UART0, such as SPI10, UART1, and IIC10.

Note: When using a serial array unit for UART, both the transmitter side (even-numbered channel) and the receiver side (odd-numbered channel) can only be used for UART.

UART performs the following four types of communication operations.

- UART transmission (See section 23.6.1. UART Transmission.)
- UART reception (See section 23.6.2. UART Reception.)
- LIN transmission (UART2 only) (See section 23.7.1. LIN Transmission .)
- LIN reception (UART2 only) (See section 23.7.2. LIN Reception .)

23.6.1 UART Transmission

UART transmission is an operation to transmit data from a microcontroller to another device asynchronously (start-stop synchronization).

Of the two channels used for UART, the even channel is used for UART transmission.

Table 23.73 shows the specification of UART transmission.

Table 23.73 Specification of UART transmission

UART	UART0	UART1	UART2
Target channel	Channel 0 of SAU0	Channel 2 of SAU0	Channel 0 of SAU1
Pins used	TxD0	TxD1	TxD2
Interrupt	SAU0_ENDI0	SAU0_ENDI2	SAU1_ENDI0
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.		
Error detection flag	None		
Transfer data length	7, 8, or 9 bits*1		
Transfer rate*2	Max. f_{MCK}/6 [bps] (SDRmn.STCLK[6:0] = 2 or more), Min. PCLKB/ (2 × 2^{15} × 128) [bps]		
Data phase	Non-reverse output (default: high level) Reverse output (default: low level)		
Parity bit	The following selectable • No parity bit • Appending 0 parity • Appending even parity • Appending odd parity		
Stop bit	The following selectable: • Appending 1 bit • Appending 2 bit		
Data direction	MSB or LSB first		

Note: f_{MCK}: Operation clock frequency of target channel
Note: m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 10
Note 1. Only the following UARTs support the 9-bit data length:

- 16-, 24-, and 32-pin products: UART0

- 48-pin products: UART0 and UART2

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics specified in the electrical characteristics. For details, see section 37, Electrical Characteristics .

(1) Register setting

Table 23.74 to Table 23.80 show examples of the register contents for UART transmission.

(a) Serial mode register mn (SMRmn)

Table 23.74 Example of serial mode register mn (SMRmn) contents for UART transmission

Bit	Symbol	Set value	Function
0	MD0	0/1	Interrupt source of channel n 0: Transfer end interrupt 1: Buffer empty interrupt
2:1	MD1[1:0]	01b	Setting of operation mode of channel n 0 1: UART mode
13:3	—	000_0000_0100 b	Setting disabled (set to the initial value)
14	CCS	0	Selection of transfer clock (f_{TCLK}) of channel n 0: Divided operation clock f_{MCK} specified by the CKS bit
15	CKS	0/1	Operation clock (f_{MCK}) of channel n 0: Prescaler output clock CKm0 set by the SPSm register 1: Prescaler output clock CKm1 set by the SPSm register

(b) Serial communication operation setting register mn (SCRmn)

Table 23.75 Example of serial communication operation setting register mn (SCRmn) contents for UART transmission (1 of 2)

Bit	Symbol	Set value	Function
1:0	DLS[1:0]	01b to 11b	Setting of data length 0 1: 9-bit data length 1 0: 7-bit data length 1 1: 8-bit data length
3:2	—	01b	Setting disabled (set to the initial value)
5:4	SLC[1:0]	01b or 10b	Setting of stop bit 0 1: Appending 1 bit 1 0: Appending 2 bit
6	—	0	Setting disabled (set to the initial value)
7	DIR	0/1	Selection of data transfer sequence in simplified SPI and UART modes 0: Inputs or outputs data with MSB first. 1: Inputs or outputs data with LSB first.
9:8	PTC[1:0]	00b to 11b	Setting of parity bit 0 0: No parity 0 1: Appending 0 parity 1 0: Appending Even parity 1 1: Appending Odd parity
10	EOC	0	Since this bit is dedicated to UART receive modes, it is fixed in the UART transmission mode.
11	—	0	Setting disabled (set to the initial value)

Table 23.75 Example of serial communication operation setting register mn (SCRmn) contents for UART transmission (2 of 2)

Bit	Symbol	Set value	Function
13:12	DCP[1:0]	00b	Since this bit is dedicated to other modes, it is fixed in the UART mode
15:14	TRXE[1:0]	10b	Setting TRXE[1:0] = 10b is fixed in the UART transmission mode

(c) Serial data register mn (SDRmn)

Table 23.76 Example of serial data register mn (SDRmn) contents for UART transmission

Bit	Symbol	Set value	Function
6:0	DAT[6:0]	0x00 to 0x7F	Setting transmit data [6:0]
7	DAT[7]	0/1	Setting transmit data [7] (8-bit and 9-bit data length)
		0	0 Fixed (7-bit data length)
8	DAT[8]*1	0/1	Setting transmit data [8] (9-bit data length)
		0	0 Fixed (7-bit and 8-bit data length)
15:9	STCLK[6:0]	0x02 to 0x7F	Baud rate setting (Operation clock (f_{MCK}) division setting)

Note 1. When UART performs 9-bit communication, the SDRm0.DAT[8:0] bits are used as the transmission data specification area.

(d) Serial output level register m (SOLm)

Set only the bit of the target channel.

Table 23.77 Example of serial output level register m (SOLm) contents for UART transmission

Bit	Symbol	Set value	Function
n	SOLn	0/1	Selects inversion of the level of the transmit data of channel 0 in UART mode 0: Non-reverse (normal) transmission 1: Reverse transmission

(e) Serial output register m (SOm)

Set only the bit of the target channel.

Table 23.78 Example of serial output register m (SOm) contents for UART transmission

Bit	Symbol	Set value	Function
n	SO[n]*1	0/1	Serial data output of channel n 0: Serial data output value is 0 1: Serial data output value is 1
n+8	CKO[n]	x	Bit that cannot be used in this mode (set to the initial value when not used in any mode)

Note 1. Before transmission is started, be sure to set to 1 when the SOLm.SOLn bit of the target channel is set to 0, and set to 0 when the SOLm.SOLn bit of the target channel is set to 1. The value varies depending on the communication data during communication operation.

(f) Serial output enable register m (SOEm)

Set only the bit of the target channel to 1.

Table 23.79 Example of serial output enable register m (SOEm) contents for UART transmission

Bit	Symbol	Set value	Function
n	SOE[n]	1	Serial output enable or stop of channel n 1: Enable output by serial communication operation.

(g) Serial channel start register m (SSm)

Set only the bit of the target channel to 1.

Table 23.80 Example of serial channel start register m (SSm) contents for UART transmission

Bit	Symbol	Set value	Function
n	SS[n]	1	Operation start trigger of channel n 1: Set the SEm.SE[n] bit to 1 to place the channel in communication waiting state

Note: m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), q: UART number (q = mx2 + n/2), mn = 00, 02, 10

Note: ×: Bits not used with serial array units (depending on the settings of other peripheral functions)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Table 23.81 shows the procedure for initial setting for UART transmission.

Table 23.81 Initial setting procedure for UART transmission

Step		Process	Detail
Procedure for initial setting of UART transmission	<1>	Starting initial setting	—
	<2>	Setting the SPSm register	Set the operation clock.
	<3>	Setting the SMRmn register	Set an operation mode.
	<4>	Setting the SCRmn register	Set a communication format.
	<5>	Setting the SDRmn register	Set a transfer baud rate (setting the transfer clock by dividing the operation clock (f_{MCK})).
	<6>	Changing setting of the SOLm register	Set an output data level.
	<7>	Setting the SOm register	Set the initial output level of the serial data (SOm.SO[n]).
	<8>	Changing setting of the SOEm register	Set the SOEm.SOE[n] bit to 1 and enable data output of the target channel.
	<9>	Setting port	Enable data output of the target channel.
	<10>	Writing to the SSm register	Set the SSm.SS[n] bit of the target channel to 1 and the SEm.SE[n] bit to 1 to enable operation.
	<11>	Completing initial setting	Initial setting is completed. Set transmit data to the SDRmn.DAT[7:0] bits (8 bits) or the SDRmn.DAT[8:0] bits (9 bits) and start communication.

Note: m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 10

Table 23.82 shows the procedure for stopping UART transmission.

Table 23.82 Procedure for stopping UART transmission

Step		Process	Detail
Procedure for stopping UART transmission	<1>	Starting setting to stop	—
	<2>	Wait until the SSRmn.TSF flag is cleared (optional)	If there is any data being transferred, wait for their completion. If there is a requirement to stop, do not wait.
	<3>	Writing the STm register	Write 1 to the STm.ST[n] bit of the target channel and set SEm.SE[n] = 0 to stop operation.
	<4>	Setting the SOEm register	Set the SOEm.SOE[n] bit to 0 and stop the output of the target channel.
	<5>	Changing setting of the SOm register (optional)	The levels of the serial data (SOm.SO[n]) on the target channel can be changed if required.
	<6>	Stop setting is completed	After the stop setting is completed, go to the next processing.

Note: m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 10

Table 23.83 shows the procedure for resuming UART transmission.

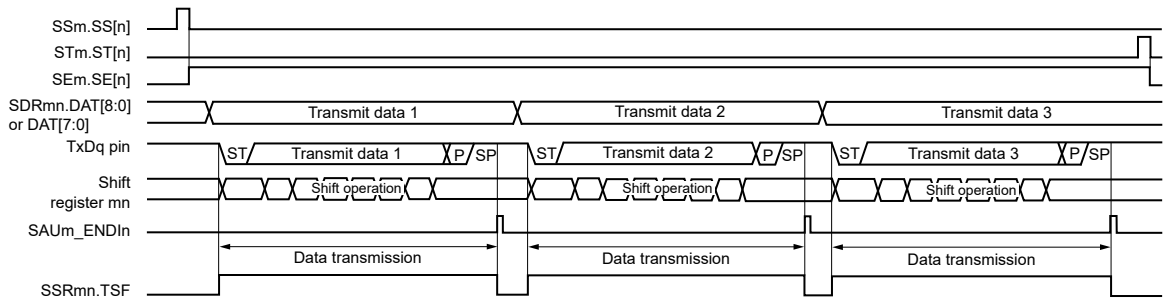
Table 23.83 Procedure for resuming UART transmission

Step		Process	Detail
Procedure for resuming UART transmission	<1>	Starting setting for resumption	—
	<2>	Wait until Communication target is ready	Wait until the communication target stops or communication operation completed.
	<3>	Port manipulation	Disable data output of the target channel.
	<4>	Changing setting of the SPSm register (optional)	Reset the register to change the operation clock setting.
	<5>	Changing setting of the SDRmn register (optional)	Reset the register to change the transfer baud rate setting (setting the transfer clock by dividing the operation clock (f_{MCK})).
	<6>	Changing setting of the SMRmn register (optional)	Reset the register to change serial mode register mn (SMRmn) setting.
	<7>	Changing setting of the SCRmn register (optional)	Reset the register to change the serial communication operation setting register mn (SCRmn) setting.
	<8>	Changing setting of the SOLm register (Selective)	Reset the register to change serial output level register m (SOLm) setting.
	<9>	Changing setting of the SOEm register (optional)	Clear the SOEm.SOE[n] bit to 0 and stop output.
	<10>	Changing setting of the SOm register (optional)	Set the initial output level of the serial data (SOm.SO[n]).
	<11>	Changing setting of the SOEm register	Set the SOEm.SOE[n] bit to 1 and enable output.
	<12>	Port manipulation	Enable data output of the target channel.
	<13>	Writing to the SSm register	Set the SSm.SS[n] bit of the target channel to 1 and set the SEm.SE[n] bit to 1 to enable operation.
	<14>	Completing resumption setting	Setting is completed. Set transmit data to the SDRmn.DAT[7:0] bits (8 bits) or the SDRmn.DAT[8:0] bits (9 bits) and start communication.

Note: m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 10

(3) Processing flow (in single transmission mode)

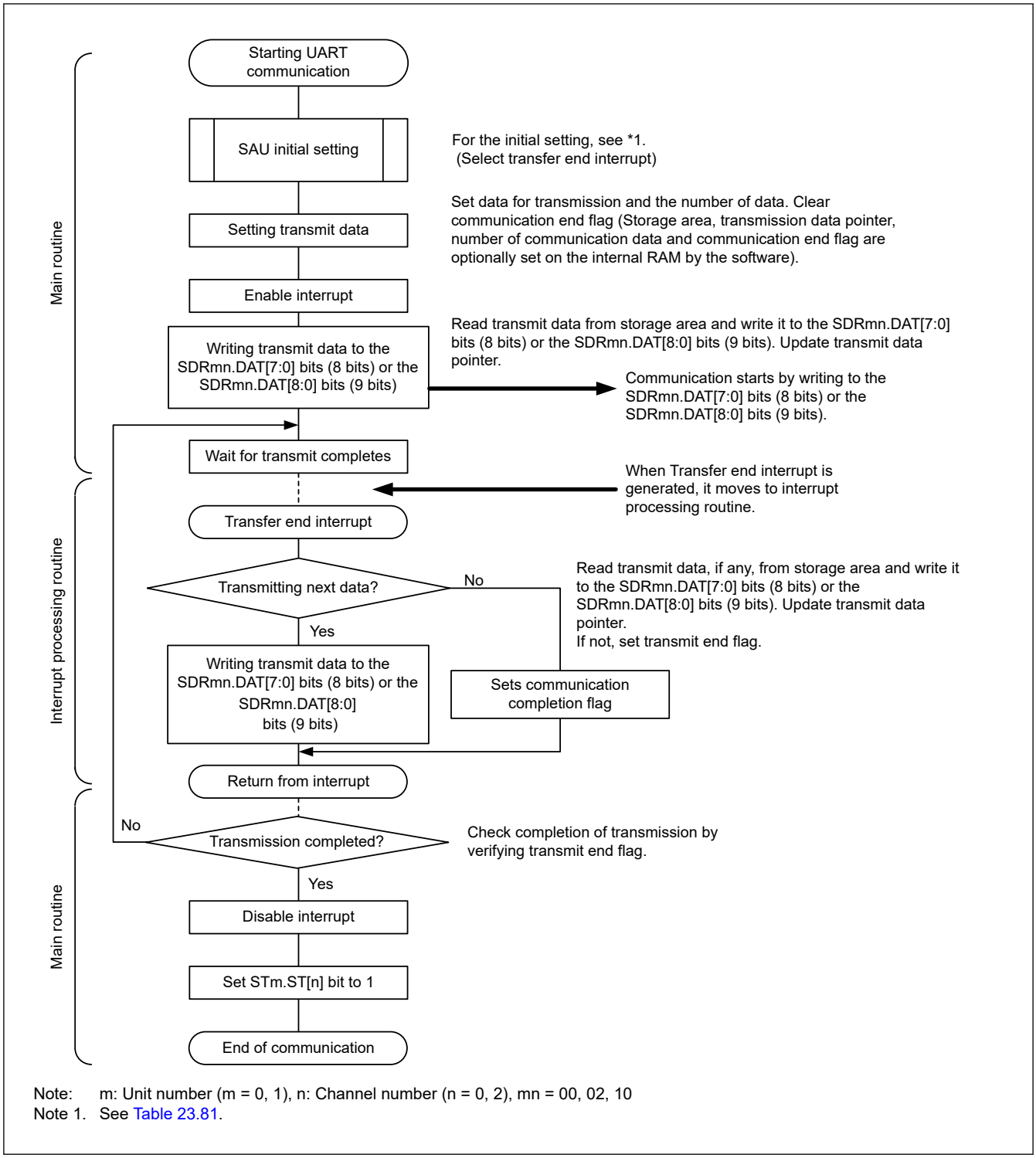
Figure 23.31 shows the timing of UART transmission (in single transmission mode).



Note: m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), q: UART number (q = mx2 + n/2), mn = 00, 02, 10

Figure 23.31 Timing of UART transmission (in single transmission mode)

Figure 23.32 shows the flowchart of UART transmission (in single transmission mode).

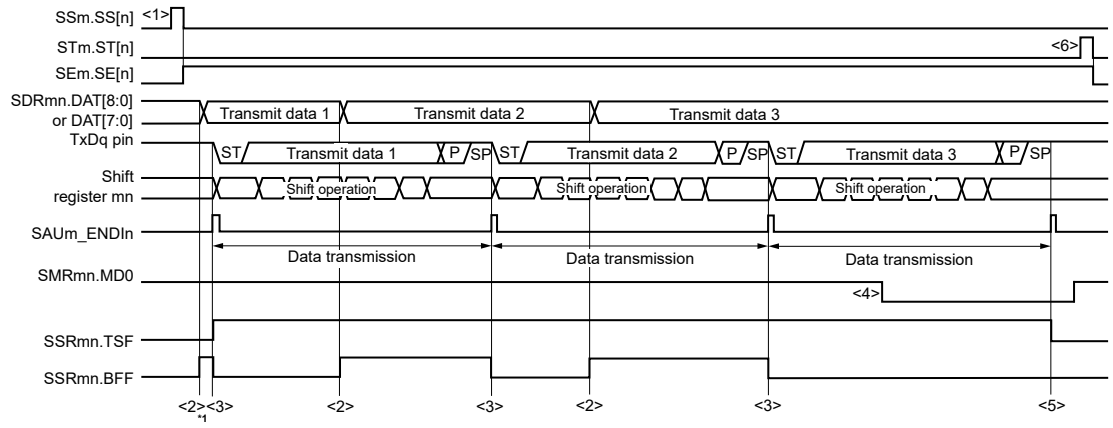


Note: m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 10
Note 1. See Table 23.81.

Figure 23.32 Flowchart of UART transmission (in single transmission mode)

(4) Processing flow (in continuous transmission mode)

Figure 23.33 shows the timing of UART transmission (in continuous transmission mode).



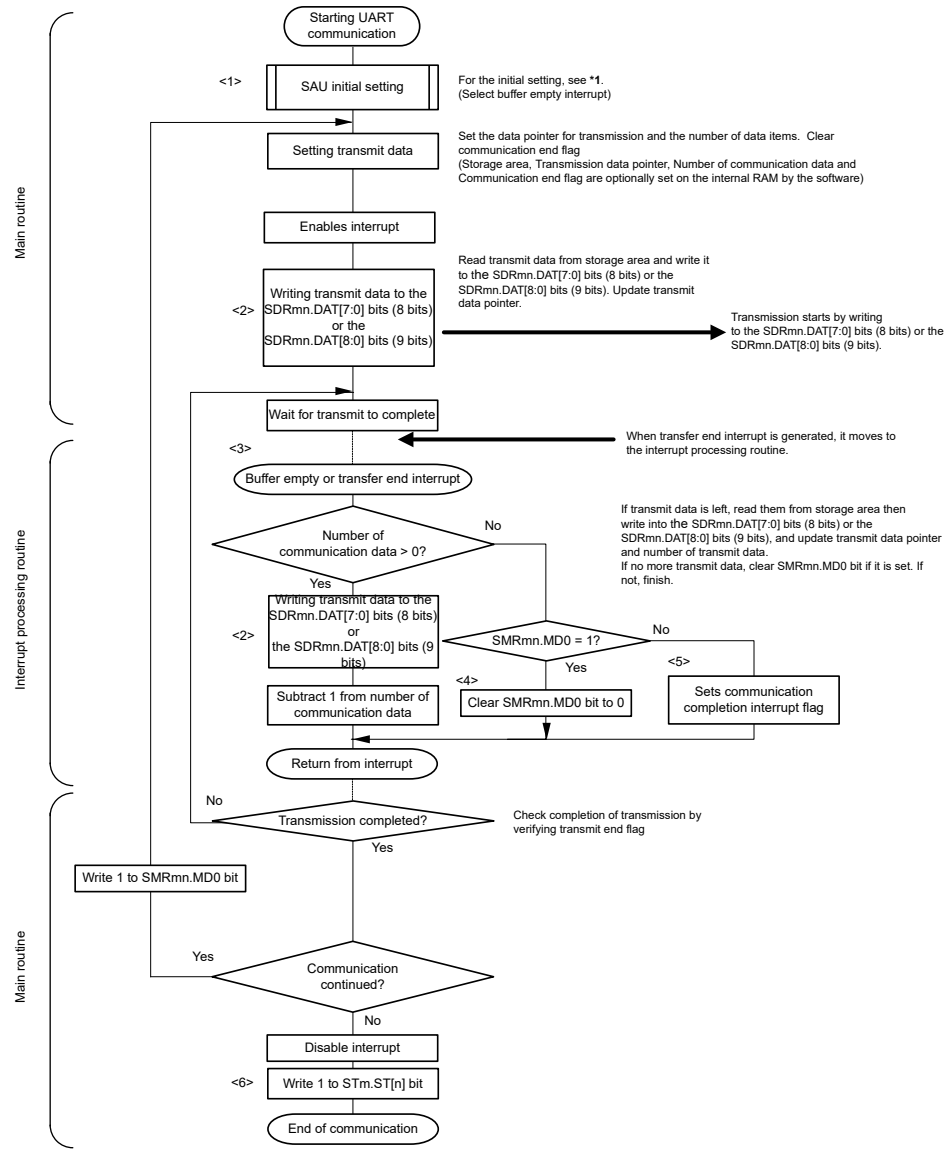
Note: The MD0 bit of serial mode register mn (SMRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last transmit data.

Note: m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), q: UART number (q = mx2 + n/2), mn = 00, 02, 10

Note 1. If transmit data is written to the SDRmn register while the BFF flag of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

Figure 23.33 Timing of UART transmission (in continuous transmission mode)

Figure 23.34 shows the flowchart of UART transmission (in continuous transmission mode).



Note: <1> to <6> in the figure correspond to <1> to <6> in Figure 23.33.
Note: m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 10
Note 1. See Table 23.81.

Figure 23.34 Flowchart of UART transmission (in continuous transmission mode)

23.6.2 UART Reception

UART reception is an operation wherein a microcontroller asynchronously receives data from another device (start-stop synchronization).

For UART reception, the odd-number channel of the two channels used for UART is used. The SMRmn register of both the odd- and even-numbered channels must be set.

Table 23.84 shows the specification of UART reception.

Table 23.84 Specification of UART reception (1 of 2)

UART	UART0	UART1	UART2
Target channel	Channel 1 of SAU0	Channel 3 of SAU0	Channel 1 of SAU1
Pins used	RxD0	RxD1	RxD2

Table 23.84 Specification of UART reception (2 of 2)

UART	UART0	UART1	UART2
Interrupt	SAU0_ENDI1	SAU0_ENDI3	SAU1_ENDI1
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)		
Error interrupt	SAU0_INTSRE0	SAU0_INTSRE1	SAU1_INTSRE2
Error detection flag	• Framing error detection flag (SSRmn.FEF) • Parity error detection flag (SSRmn.PEF) • Overrun error detection flag (SSRmn.OVF)		
Transfer data length	7, 8, or 9 bits*1		
Transfer rate*2	Max. f_{MCK}/6 [bps] (SDRmn.STCLK[6:0] = 2 or more), Min. PCLKB/ (2 × 2^{15} × 128) [bps]		
Data phase	Non-reverse output (default: high level) Reverse output (default: low level)		
Parity bit	The following are selectable: • No parity bit (no parity check) • No parity judgment (0 parity) • Even parity check • Odd parity check		
Stop bit	Appending 1 bit		
Data direction	MSB or LSB first		

Note: f_{MCK}: Operation clock frequency of target channel
f_{SCK}: Serial clock frequency
Note: m: Unit number (m = 0, 1), n: Channel number (n = 1, 3), mn = 01, 03, 11
Note 1. Only the following UARTs support the 9-bit data length:
- 16-, 24-, and 32-pin products: UART0
- 48-pin products: UART0 and UART2

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics specified in the electrical characteristics. For details, see section 37, Electrical Characteristics .

(1) Register setting

Table 23.85 to Table 23.91 show examples of the register contents for UART reception.

(a) Serial mode register mn (SMRmn)

Table 23.85 Example of serial mode register mn (SMRmn) contents for UART reception (1 of 2)

Bit	Symbol	Set value	Function
0	MD0	0	Interrupt source of channel n 0: Transfer end interrupt
2:1	MD1[1:0]	01b	Setting of operation mode of channel n 0 1: UART mode
5:3	—	100b	Setting disabled (set to the initial value)
6	SIS0	0/1	Controls inversion of level of receive data of channel n in UART mode 0: Normal reception 1: Reverse reception
7	—	0	Setting disabled (set to the initial value)
8	STS	1	Selection of start trigger source 1: Valid edge of the RxDq pin
13:9	—	0_0000b	Setting disabled (set to the initial value)
14	CCS	0	Selection of transfer clock (f_{TCLK}) of channel n 0: Divided operation clock f_{MCK} specified by the CKS bit

Table 23.85 Example of serial mode register mn (SMRmn) contents for UART reception (2 of 2)

Bit	Symbol	Set value	Function
15	CKS	0/1	Operation clock (f_{MCK}) of channel n 0: Prescaler output clock CKm0 set by the SPSm register 1: Prescaler output clock CKm1 set by the SPSm register

(b) Serial mode register mr (SMRmr)

Table 23.86 Example of serial mode register mr (SMRmr) contents for UART reception

Bit	Symbol	Set value	Function
0	MD0	0	Interrupt source of channel r 0: Transfer end interrupt
2:1	MD1[1:0]	01b	Setting of operation mode of channel r 0 1: UART mode
13:3	—	000_0000_0100b	Setting disabled (set to the initial value)
14	CCS	0	Selection of transfer clock (f_{TCLK}) of channel r 0: Divided operation clock f_{MCK} specified by the CKS bit
15	CKS	0/1	Operation clock (f_{MCK}) of channel r (same setting value as SMRmn.CKS bit) 0: Prescaler output clock CKm0 set by the SPSm register 1: Prescaler output clock CKm1 set by the SPSm register

(c) Serial communication operation setting register mn (SCRmn)

Table 23.87 Example of serial communication operation setting register mn (SCRmn) contents for UART reception (1 of 2)

Bit	Symbol	Set value	Function
1:0	DLS[1:0]	01b to 11b	Setting of data length 0 1: 9-bit data length 1 0: 7-bit data length 1 1: 8-bit data length
3:2	—	01b	Setting disabled (set to the initial value)
5:4	SLC[1:0]	01b	Setting of stop bit 0 1: Appending 1 bit
6	—	0	Setting disabled (set to the initial value)
7	DIR	0/1	Selection of data transfer sequence in simplified SPI and UART modes 0: Inputs or outputs data with MSB first. 1: Inputs or outputs data with LSB first.
9:8	PTC[1:0]	00b to 11b	Setting of parity bit 0 0: No parity 0 1: Appending 0 parity 1 0: Appending Even parity 1 1: Appending Odd parity

Table 23.87 Example of serial communication operation setting register mn (SCRmn) contents for UART reception (2 of 2)

Bit	Symbol	Set value	Function
10	EOC	0/1	Mask control of error interrupt signal SAUm_INTSREq 0: Disables generation of error interrupt SAUm_INTSREq (SAUm_ENDIn is generated). 1: Enables generation of error interrupt SAUm_INTSREq (SAUm_ENDIn is not generated if an error occurs).
11	—	0	Setting disabled (set to the initial value)
13:12	DCP[1:0]	00b	Since this bit is dedicated to other modes, it is fixed in the UART mode.
15:14	TRXE[1:0]	01b	Setting TRXE[1:0] = 01b is fixed in the UART reception mode

(d) Serial data register mn (SDRmn)

Table 23.88 Example of serial data register mn (SDRmn) contents for UART reception

Bit	Symbol	Set value	Function
6:0	DAT[6:0]	0x00 to 0x7F	Receive data[6:0]
7	DAT[7]	0/1	Receive data[7] (8-bit and 9-bit data length)
		0	0 Fixed (7-bit data length)
8	DAT[8]*1	0/1	Receive data[8] (9-bit data length)
		0	0 Fixed (7-bit and 8-bit data length)
15:9	STCLK[6:0]	0x02 to 0x7F	Baud rate setting (Operation clock (f_{MCK}) division setting)

Note 1. When UART performs 9-bit communication, the DAT[8:0] bits of the SDRm1 register are used as the reception data specification area.

(e) Serial output register m (SOm)

This register is not used in this mode.

Table 23.89 Example of serial output register m (SOm) contents for UART reception

Bit	Symbol	Set value	Function
n	SO[n]	x	Bit that cannot be used in this mode (set to the initial value when not used in any mode)
n+8	CKO[n]	x	Bit that cannot be used in this mode (set to the initial value when not used in any mode)

(f) Serial output enable register m (SOEm)

This register is not used in this mode.

Table 23.90 Example of serial output enable register m (SOEm) contents for UART reception

Bit	Symbol	Set value	Function
n	SOE[n]	x	Bit that cannot be used in this mode (set to the initial value when not used in any mode)

(g) Serial channel start register m (SSm)

Set only the bit of the target channel to 1.

Table 23.91 Example of serial channel start register m (SSm) contents for UART reception

Bit	Symbol	Set value	Function
n	SS[n]	1	Operation start trigger of channel n 1: Set the SEm.SE[n] bit to 1 to place the channel in the communications waiting state.

Note: For the UART reception, be sure to set the SMRmr register of channel r to UART transmission mode that is to be paired with channel n.

Note: m: Unit number (m = 0, 1), n: Channel number (n = 1, 3), mn = 01, 03, 11
r: Channel number (r = n – 1)

Note: ×: Bits not used with serial array units (depending on the settings of other peripheral functions)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Table 23.92 shows the procedure for initial setting for UART Reception.

Table 23.92 Initial setting procedure for UART reception

Step		Process	Detail
Procedure for initial setting of UART reception	<1>	Starting initial setting	—
	<2>	Setting the SPSm register	Set the operation clock.
	<3>	Setting the SMRmn and SMRmr registers	Set an operation mode.
	<4>	Setting the SCRmn register	Set a communication format.
	<5>	Setting the SDRmn register	Set a transfer baud rate (setting the transfer clock by dividing the operation clock (f_{MCK})).
	<6>	Setting port	Enable data input of the target channel.
	<7>	Writing to the SSm register	Set the SSm.SS[n] bit of the target channel to 1 and set the SEm.SE[n] bit to 1 to enable operation. Wait for start bit detection.
	<8>	Completing initial setting	—

Note: Set the TRXE[0] bit of SCRmn register to 1, and then be sure to set the SSm.SS[n] bit to 1 after at least 4 f_{MCK} clock cycles have elapsed.
Note: m: Unit number (m = 0, 1), n: Channel number (n = 1, 3), mn = 01, 03, 11

Table 23.93 shows the procedure for stopping UART reception.

Table 23.93 Procedure for stopping UART reception

Step		Process	Detail
Procedure for stopping UART transmission	<1>	Starting setting to stop	—
	<2>	Wait until the SSRmn.TSF flag is cleared (optional)	If there is any data being transferred, wait for its completion. If there is a requirement to stop, do not wait.
	<3>	Writing the STm register	Write 1 to the STm.ST[n] bit of the target channel and set SEm.SE[n] = 0 to stop operation.
	<4>	Stop setting is completed	After the stop setting is completed, go to the next processing.

Note: m: Unit number (m = 0, 1), n: Channel number (n = 1, 3), mn = 01, 03, 11

Table 23.94 shows the procedure for resuming UART Reception.

Table 23.94 Procedure for resuming UART reception

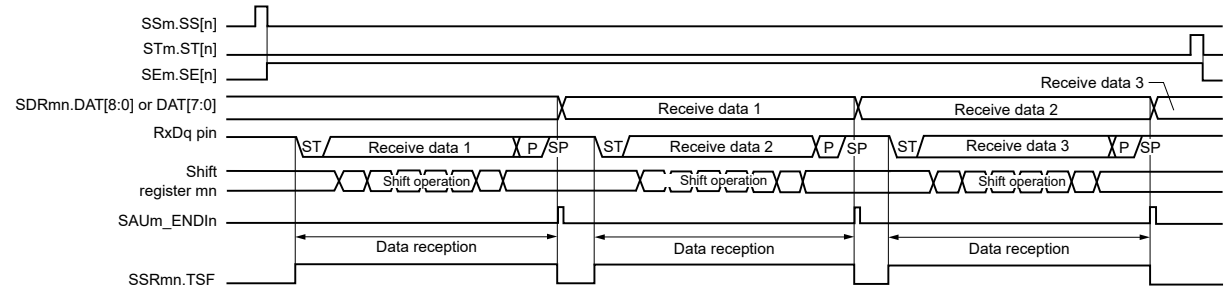
Step		Process	Detail
Procedure for resuming UART reception	<1>	Starting setting for resumption	—
	<2>	Wait until the communication target is ready	Wait until the communication target stops or communication operation completed.
	<3>	Changing setting of the SPSm register (optional)	Reset the register to change the operation clock setting.
	<4>	Changing setting of the SDRmn register (optional)	Reset the register to change the transfer baud rate setting (setting the transfer clock by dividing the operation clock (f_{MCK})).
	<5>	Changing setting of the SMRmn and SMRmr registers (optional)	Reset the registers to change serial mode registers mn, mr (SMRmn, SMRmr) setting.
	<6>	Changing setting of the SCRmn register (optional)	Reset the register to change the serial communication operation setting register mn (SCRmn) setting.
	<7>	Clearing error flag	If the SSRmn.FEF, PEF, and OVF flags remain set, clear them using serial flag clear trigger register mn (SIRmn).
	<8>	Setting port	Enable data input of the target channel.
	<9>	Writing to the SSm register	Set the SSm.SS[n] bit of the target channel to 1 and set the SEm.SE[n] bit to 1 to enable operation. Wait for start bit detection.
	<10>	Completing resumption setting	—

Note: Set the TRXE[0] bit of SCRmn register to 1, and then be sure to set the SSm.SS[n] bit to 1 after at least 4 f_{MCK} clocks have elapsed.

Note: m: Unit number (m = 0, 1), n: Channel number (n = 1, 3), r: Channel number (r = n – 1), mn = 01, 03, 11

(3) Processing flow

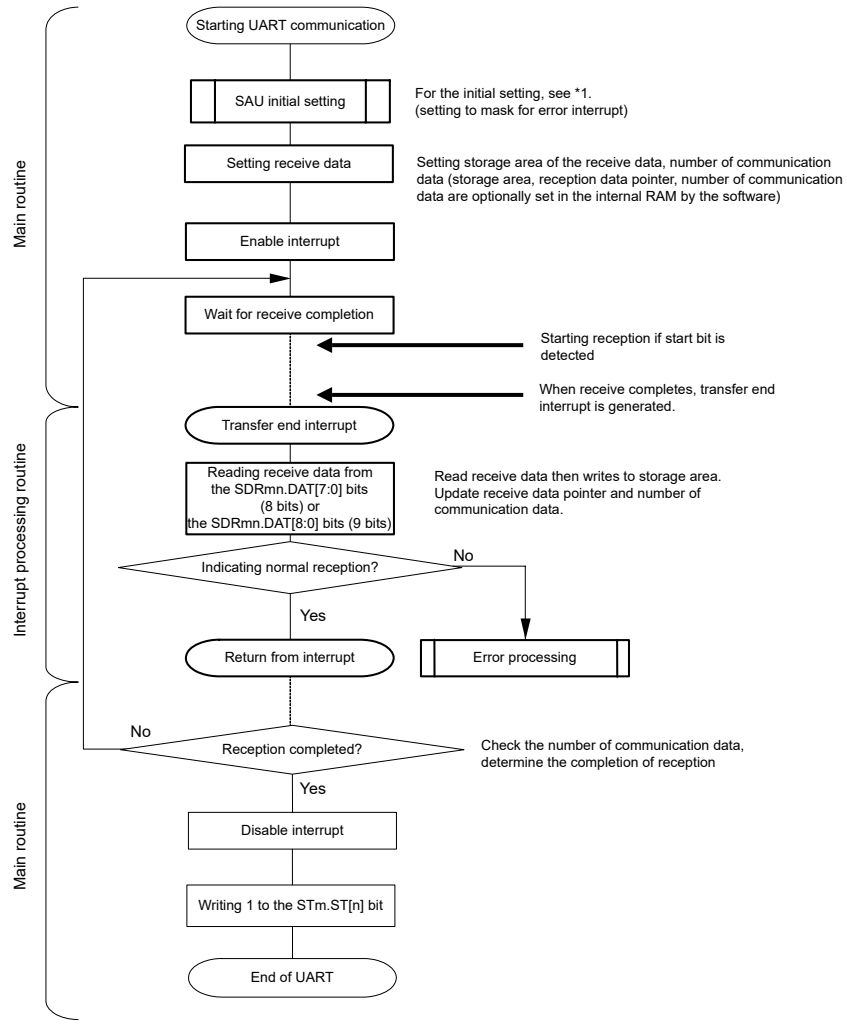
Figure 23.35 shows the timing of UART reception.



Note: m: Unit number (m = 0, 1), n: Channel number (n = 1, 3), mn = 01, 03, 11, q: UART number (q = mx2 + n/2)

Figure 23.35 Timing of UART reception

Figure 23.36 shows the flowchart of UART reception.



Note: m: Unit number (m = 0, 1), n: Channel number (n = 1, 3), mn = 01, 03, 11
Note 1. See Table 23.92.

Figure 23.36 Flowchart of UART reception

23.6.3 Snooze Mode Function

The Snooze mode makes the UART perform reception operations on RxDq pin input detection while in the Software Standby mode. Normally the UART stops communication in the Software Standby mode. However, using the Snooze mode enables the UART to perform reception operations without CPU operation.

Only UART0 and UART2 channels can be set to Snooze mode.

When using UARTq in the Snooze mode, make the following settings before entering the Software Standby mode. (See Figure 23.39 and Figure 23.41.)

- In the Snooze mode, the baud rate setting for UART reception needs to be changed to a value different from that in normal operation. Set the SPSm register and the SDRmn.STCLK[6:0] bit with reference to Table 23.95.
- Set the SCRmn.EOC and SSCm.SSEC bits. This is for enabling or stopping generation of an error interrupt (SAU0_INTSRE0) when a communication error occurs.
- When using the Snooze mode function, set the SWC bit of serial standby control register m (SSCm) to 1 just before switching to the Software Standby mode. After the initial setting has been completed, set the SS[1] bit of serial channel start register m (SSm) to 1.
- A UARTq starts reception in Snooze mode on detecting input of the start bit on the RxDq pin following a transition of the CPU to the Software Standby mode.

Note: The Snooze mode can only be used when the high-speed on-chip oscillator clock (HOCO) or middle-speed on-chip oscillator clock (MOCO) is selected for PCLKB.

When the middle-speed on-chip oscillator clock (MOCO) is selected, use the MOCO User Trimming Control Register (MOCOUTCR) to correct the accuracy of the oscillation frequency.

Note: The maximum transfer rate in the Snooze mode is 9600 bps.

Note: When SSCm.SWC = 1, UARTq can be used only when the reception operation is started in the Software Standby mode.

When used simultaneously with another Snooze mode function or interrupt, if the reception operation is started in a state other than the Software Standby mode, such as those given below, data may not be received correctly and a framing error or parity error may be generated.

- When after the SSCm.SWC bit has been set to 1, the reception operation is started before the Software Standby mode is entered
- When the reception operation is started while another function is in the Snooze mode
- When after returning from the Software Standby mode to normal operation due to an interrupt or other cause, the reception operation is started before the SSCm.SWC bit is returned to 0

Note: If a parity error, framing error, or overrun error occurs while the SSCm.SSEC bit is set to 1, the SSRmn.PEF, FEF, or OVF flag is not set and an error interrupt (SAUm_INTSREq) is not generated. Therefore, when the setting of SSCm.SSEC = 1 is made, clear the SSRmn.PEF, FEF, and OVF flags before setting the SSCm.SWC bit to 1 and read the value in DAT[7:0] bits of the SDRm1 register.

Note: The CPU shifts from the Software Standby mode to the Snooze mode on detecting the valid edge of the RxDq signal.

Note, however, that transfer through the UART channel may not start and the CPU may remain in the Snooze mode if an input pulse on the RxDq pin is too short to be detected as a start bit. In such cases, data may not be received correctly, and this may lead to a framing error or parity error in the next UART transfer.

Note: m: Unit number (m = 0, 1), n: Channel number (n = 1, 3), q: UART number (q = 0, 2), mn = 01, 11

Table 23.95 shows the baud rate setting for UART reception in Snooze mode.

Table 23.95 Baud rate setting for UART reception in Snooze mode

Baud rate	High-speed on-chip oscillator (HOCO)	Operating clock (f_{MCK})	SDRmn.STCLK[6:0]	Maximum permissible value	Minimum permissible value
4800 bps	32 MHz ± 1%*1	PCLKB/2^5	106	1.45%	−1.67%
	24 MHz ± 1%*1	PCLKB/2^5	79	1.77%	−1.37%
9600 bps	32 MHz ± 1%*1	PCLKB/2^4	106	1.45%	−1.67%
	24 MHz ± 1%*1	PCLKB/2^4	79	1.77%	−1.37%

Note 1. When the accuracy of the clock frequency of the HOCO is ±1.5% or ±2.0%, the permissible range becomes smaller as shown below.

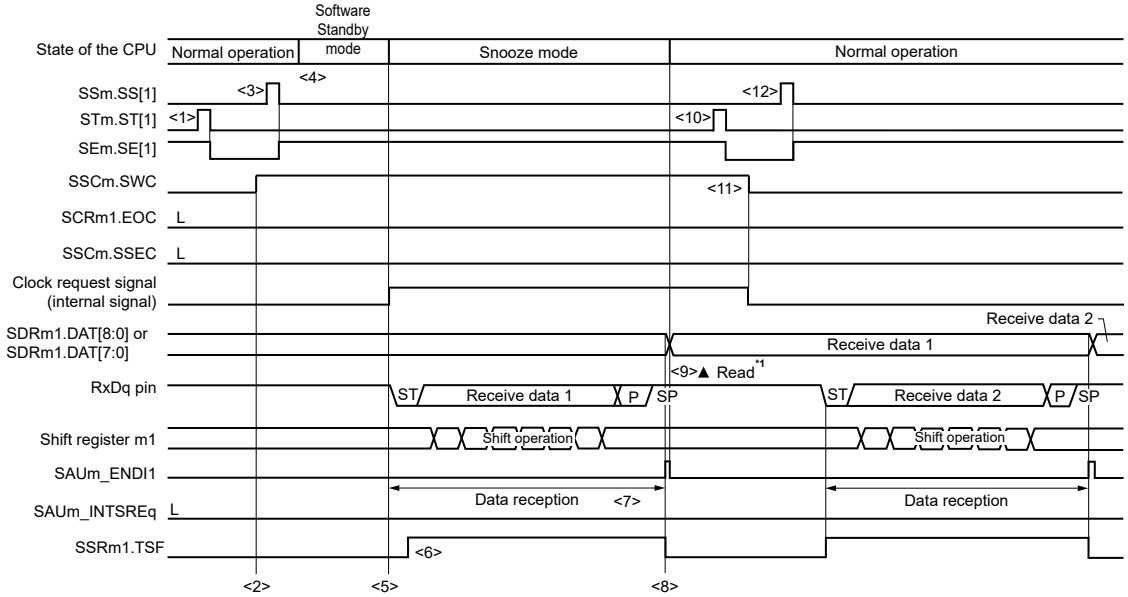
- In the case of HOCO ±1.5%, perform (Maximum permissible value – 0.5%) and (Minimum permissible value + 0.5%) to the values in the above table.
- In the case of HOCO ±2.0%, perform (Maximum permissible value – 1.0%) and (Minimum permissible value + 1.0%) to the values in the above table.

Note: The maximum permissible value and minimum permissible value are permissible values for the baud rate in UART reception. The baud rate on the transmitting side should be set to fall inside this range.

(1) Snooze mode operation (SCRm1.EOC = 0, SSCm.SSEC = 0/1)

Because of the setting of SCRm1.EOC = 0, even though a communication error occurs, an error interrupt (SAUm_INTSREq) is not generated, regardless of the setting of the SSCm.SSEC bit. However, a transfer end interrupt (SAUm_ENDI1) is generated.

Figure 23.37 shows the timing of Snooze mode operation (SCRm1.EOC = 0, SSCm.SSEC = 0/1).



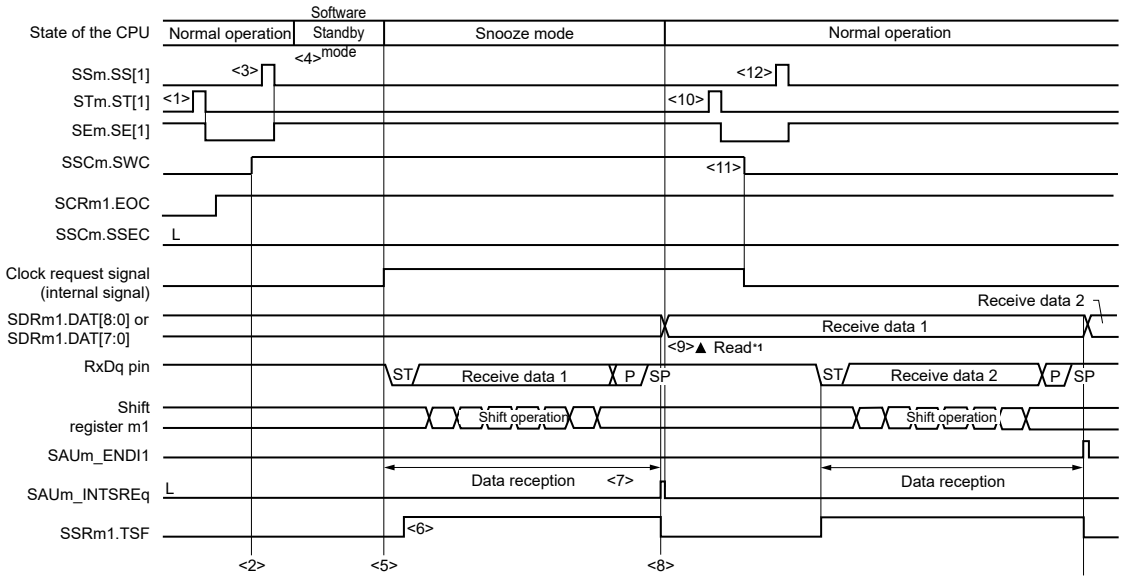
Note: m: Unit number (m = 0, 1), q: UART number (q = 0, 2)
Note: Before switching to the Snooze mode or after reception operation in the Snooze mode finishes, set the STm.ST[1] bit to 1 (the SEm.SE[1] bit is cleared and the operation stops).
After the receive operation completes, also clear the SSCm.SWC bit to 0 (Snooze mode release).
Note: <1> to <12> in the figure correspond to <1> to <12> in Figure 23.39.
Note 1. Read the received data when SSCm.SWC = 1.

Figure 23.37 Timing of Snooze mode operation (SCRm1.EOC = 0, SSCm.SSEC = 0/1)

(2) Snooze mode operation (SCRm1.EOC = 1, SSCm.SSEC = 0: Error interrupt (SAUm_INTSREq) generation is enabled)

Because SCRm1.EOC = 1 and SSCm.SSEC = 0, an error interrupt (SAUm_INTSREq) is generated when a communication error occurs.

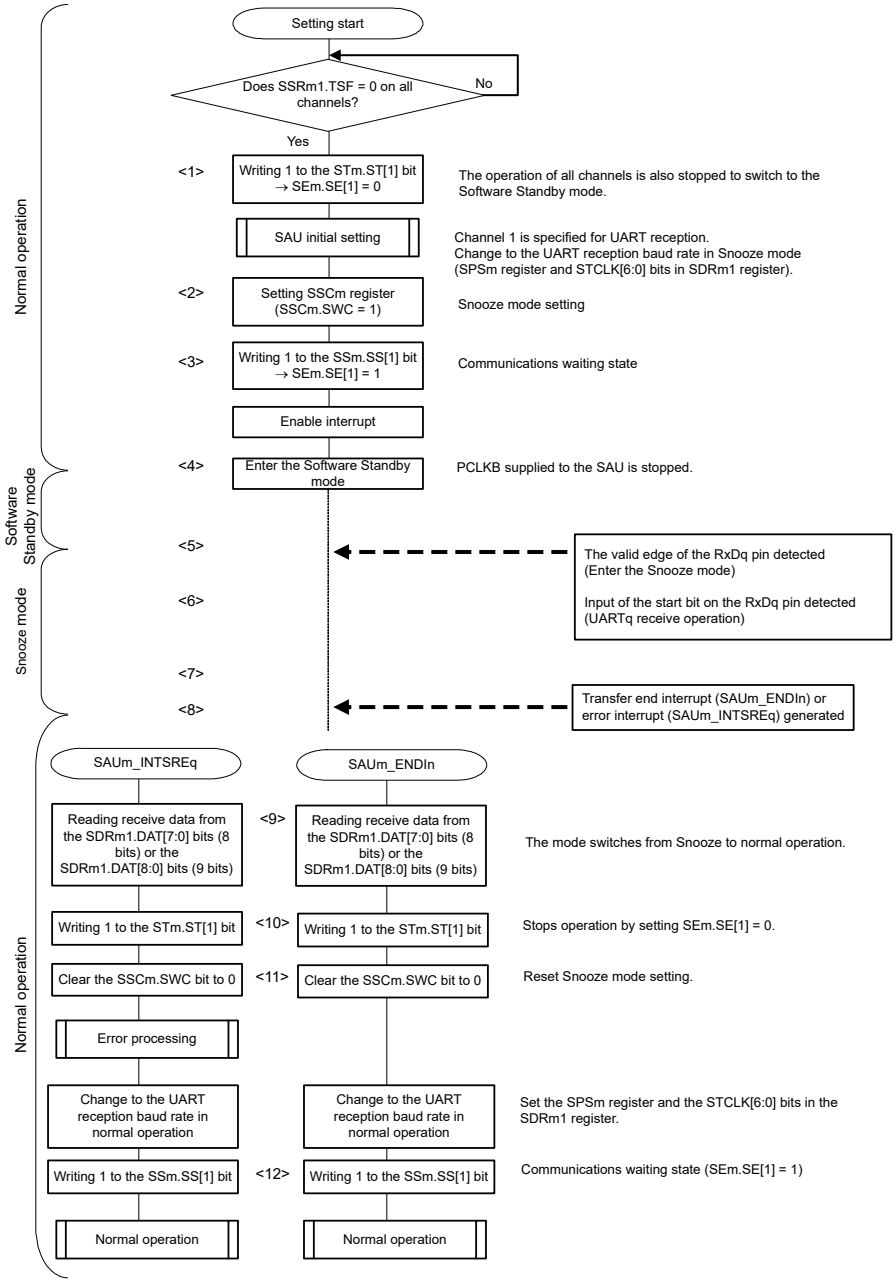
Figure 23.38 shows the timing of Snooze mode operation (SCRm1.EOC = 1, SSCm.SSEC = 0).



Note: m: Unit number (m = 0, 1), q: UART number (q = 0, 2)
Note: Before switching to the Snooze mode or after reception operation in the Snooze mode finishes, set the STm.ST[1] bit to 1 (the SEm.SE[1] bit is cleared and the operation stops).
After the receive operation completes, also clear the SSCm.SWC bit to 0 (Snooze mode release).
Note: <1> to <12> in the figure correspond to <1> to <12> in Figure 23.39.
Note 1. Read the received data when SSCm.SWC = 1.

Figure 23.38 Timing of Snooze mode operation (SCRm1.EOC = 1, SSCm.SSEC = 0)

Figure 23.39 shows the flowchart of Snooze mode operation (SCRm1.EOC = 0, SSCm.SSEC = 0/1 or SCRm1.EOC = 1, SSCm.SSEC = 0).



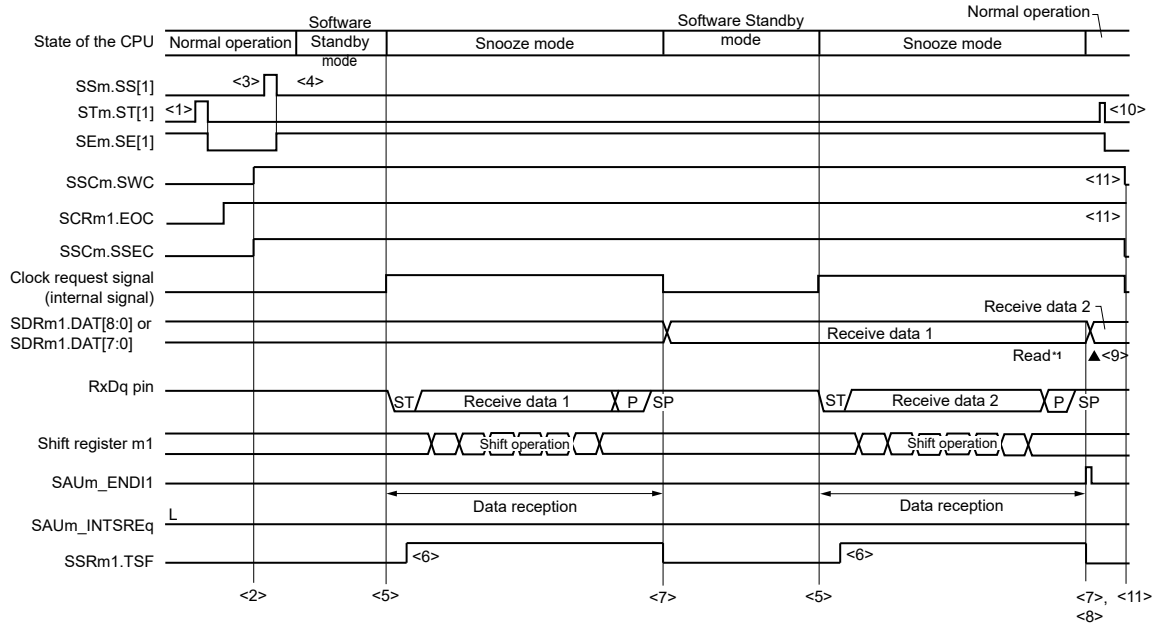
Note: m: Unit number (m = 0, 1)
Note: <1> to <12> in the figure correspond to <1> to <12> in Figure 23.37 and Figure 23.38.

Figure 23.39 Flowchart of Snooze mode operation (SCRm1.EOC = 0, SSCm.SSEC = 0/1 or SCRm1.EOC = 1, SSCm.SSEC = 0)

(3) Snooze mode operation (SCRm1.EOC = 1, SSCm.SSEC = 1: Error interrupt (SAUm_INTSREq) generation is stopped)

Because SCRm1.EOC = 1 and SSCm.SSEC = 1, an error interrupt (SAUm_INTSREq) is not generated when a communication error occurs.

Figure 23.40 shows the timing of Snooze mode operation (SCRm1.EOC = 1, SSCm.SSEC = 1).



Note: m: Unit number (m = 0, 1), q: UART number (q = 0, 2)

Note: Before switching to the Snooze mode or after reception operation in the Snooze mode finishes, set the STm.ST[1] bit to 1 (the SEm.SE[1] bit is cleared and the operation stops).
After the receive operation completes, also clear the SSCm.SWC bit to 0 (Snooze mode release).

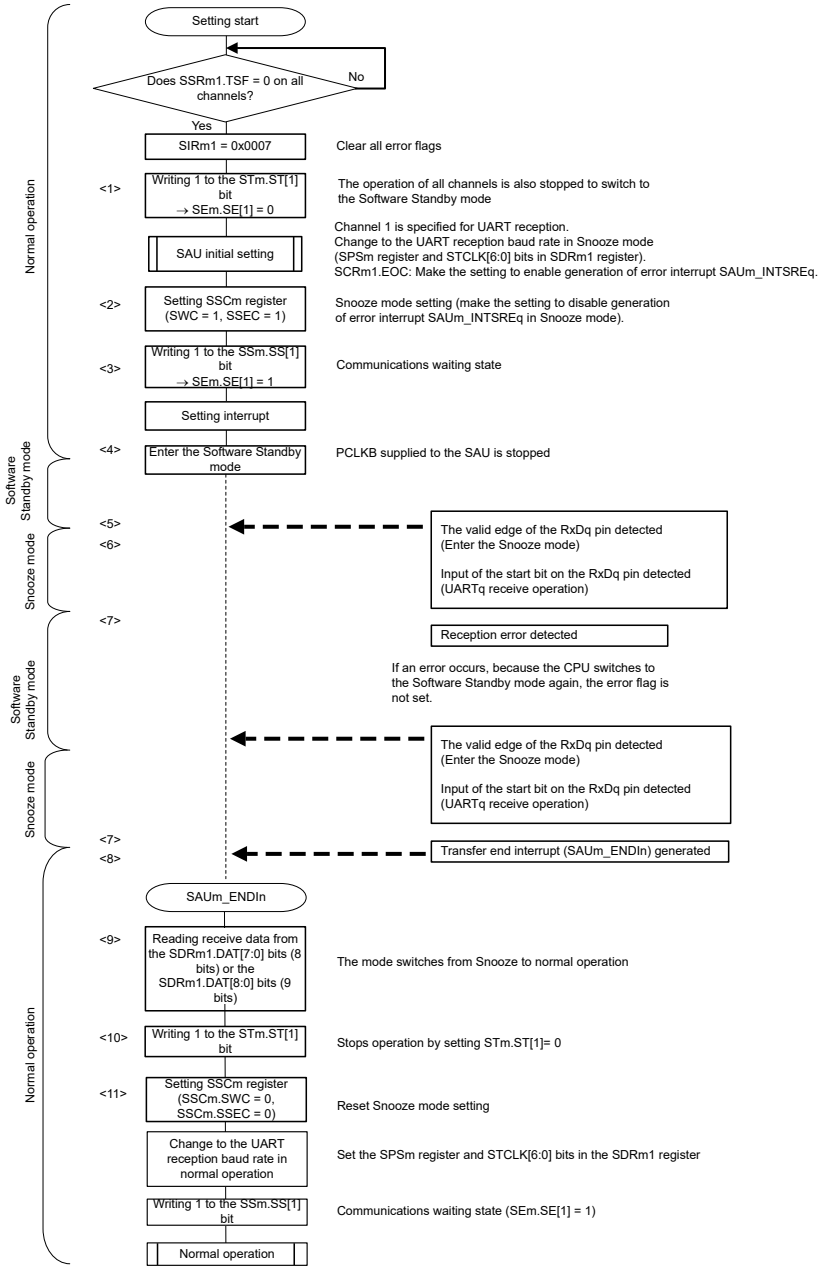
Note: If a parity error, framing error, or overrun error occurs while the SSCm.SSEC bit is set to 1, the SSRm1.PEF, FEF, or OVF flag is not set and an error interrupt (SAUm_INTSREq) is not generated. Therefore, when the setting of SSCm.SSEC = 1 is made, clear the SSRm1.PEF, FEF, and OVF flags before setting the SSCm.SWC bit to 1 and read the value in SDRm1.DAT[7:0] (8 bits) or SDRm1.DAT[8:0] (9 bits).

Note: <1> to <11> in the figure correspond to <1> to <11> in Figure 23.41.

Note 1. Read the received data when SSCm.SWC = 1.

Figure 23.40 Timing of Snooze mode operation (SCRm1.EOC = 1, SSCm.SSEC = 1)

Figure 23.41 shows the flowchart of Snooze mode operation (SCRm1.EOC = 1, SSCm.SSEC = 1).



Note: m: Unit number (m = 0, 1), q: UART number (q = 0, 2)

Note: If a parity error, framing error, or overrun error occurs while the SSCm.SSEC bit is set to 1, the SSRm1.PEF, FEF, or OVF flag is not set and an error interrupt (SAUm_INTSREq) is not generated. Therefore, when the setting of SSCm.SSEC = 1 is made, clear the SSRm1.PEF, FEF, and OVF flags before setting the SSCm.SWC bit to 1 and read the value in SDRm1.DAT[7:0] (8 bits) or SDRm1.DAT[8:0] (9 bits).

Note: <1> to <11> in the figure correspond to <1> to <11> in Figure 23.40.

Figure 23.41 Flowchart of Snooze mode operation (SCRm1.EOC = 1, SSCm.SSEC = 1)

23.6.4 Calculating Baud Rate

(1) Baud rate calculation expression

The baud rate for UART communication can be calculated by the following expressions.

(Baud rate) = {Operation clock (f_{MCK}) frequency of target channel} ÷ (SDRmn.STCLK[6:0] + 1) ÷ 2 [bps]

Note: Setting SDRmn.STCLK[6:0] = (0x00, 0x01) is prohibited.

Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 11

The operation clock (f_{MCK}) is determined by serial clock select register m (SPSm) and CKS bit of serial mode register mn (SMRmn). See Table 23.71.

(2) Baud rate error during transmission

The baud rate error of UART communication during transmission can be calculated by the following expression. Make sure that the baud rate at the transmission side is within the permissible baud rate range at the reception side.

(Baud rate error) = (Calculated baud rate value) ÷ (Target baud rate) × 100 - 100 [%]

Table 23.96 shows an example of setting a UART baud rate at PCLKB = 32 MHz.

Table 23.96 Example of setting UART baud rate at PCLKB = 32 MHz

UART baud rate (target baud rate)	PCLKB = 32 MHz			
	Operation clock (f_{MCK})	SDRmn.STCLK[6:0]	Calculated baud rate	Error from target baud rate
300 bps	PCLKB/2^9	103	300.48 bps	+0.16%
600 bps	PCLKB/2^8	103	600.96 bps	+0.16%
1200 bps	PCLKB/2^7	103	1201.92 bps	+0.16%
2400 bps	PCLKB/2^6	103	2403.85 bps	+0.16%
4800 bps	PCLKB/2^5	103	4807.69 bps	+0.16%
9600 bps	PCLKB/2^4	103	9615.38 bps	+0.16%
19200 bps	PCLKB/2^3	103	19230.8 bps	+0.16%
31250 bps	PCLKB/2^3	63	31250.0 bps	±0.0%
38400 bps	PCLKB/2^2	103	38461.5 bps	+0.16%
76800 bps	PCLKB/2	103	76923.1 bps	+0.16%
153600 bps	PCLKB	103	153846 bps	+0.16%
312500 bps	PCLKB	50	313725.5 bps	+0.39%

Note: m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 10

(3) Permissible baud rate range for reception

The permissible baud rate range for reception during UART communication can be calculated by the following expression. Make sure that the baud rate at the transmission side is within the permissible baud rate range at the reception side.

$$\text{(Maximum receivable baud rate)} = \frac{2 \times k \times \text{Nfr}}{2 \times k \times \text{Nfr} - k + 2} \times \text{Brate}$$

$$\text{(Minimum receivable baud rate)} = \frac{2 \times k \times (\text{Nfr} - 1)}{2 \times k \times \text{Nfr} - k - 2} \times \text{Brate}$$

- Brate: Calculated baud rate value at the reception side (See (1) Baud rate calculation expression.)
- k: SDRmn.STCLK[6:0] + 1
- Nfr: 1 data frame length [bits] = (Start bit) + (Data length) + (Parity bit) + (Stop bit)

Note: m: Unit number (m = 0, 1), n: Channel number (n = 1, 3), mn = 01, 03, 11

Figure 23.42 shows the permissible baud rate range for reception (1 Data Frame Length = 11 Bits).

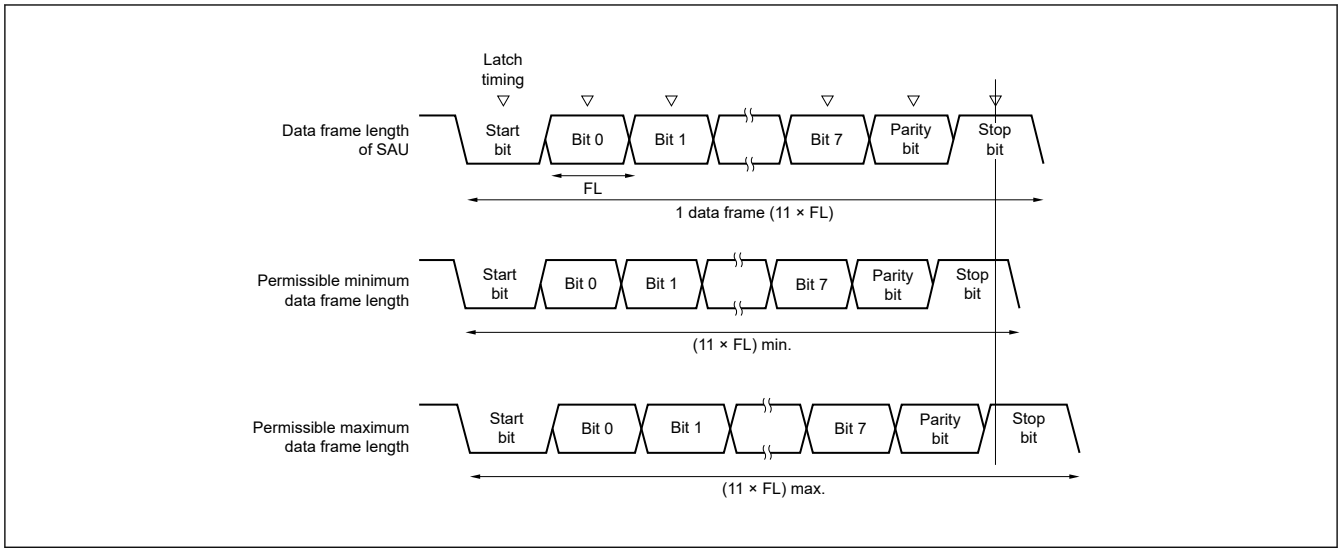


Figure 23.42 Permissible baud rate range for reception (1 data frame length = 11 bits)

As shown in Figure 23.42, the timing of latching receive data is determined by the division ratio set by the STCLK[6:0] bits of serial data register mn (SDRmn) after the start bit is detected. If the last data (stop bit) is received before this latch timing, the data can be correctly received.

23.6.5 Procedure for Processing Errors that Occurred During UART Communication

The procedure for processing errors that occurred during UART communication is described in Table 23.97 and Table 23.98.

Table 23.97 Processing procedure for parity error or overrun error

Step	Software manipulation		State of the hardware	Note
<1>	Reads serial data register mn (SDRmn).	→	The BFF flag of the SSRmn register is set to 0 and channel n is enabled to receive data	This is to prevent an overrun error if the next reception is completed during error processing.
<2>	Reads serial status register mn (SSRmn).		—	The error type is identified and the read value is used to clear the error flag.
<3>	Writes 1 to serial flag clear trigger register mn (SIRmn).	→	The error flag is cleared.	Only the error generated during reading can be cleared, by writing the value read from the SSRmn register to the SIRmn register without modification.

Table 23.98 Processing procedure for framing error (1 of 2)

Step	Software manipulation		State of the hardware	Note
<1>	Reads serial data register mn (SDRmn).	→	The BFF flag of the SSRmn register is set to 0 and channel n is enabled to receive data	This is to prevent an overrun error if the next reception is completed during error processing.
<2>	Reads serial status register mn (SSRmn).		—	The error type is identified and the read value is used to clear the error flag.
<3>	Writes serial flag clear trigger register mn (SIRmn).	→	The error flag is cleared.	Only the error generated during reading can be cleared, by writing the value read from the SSRmn register to the SIRmn register without modification.
<4>	Sets the ST[n] bit of serial channel stop register m (STm) to 1.	→	The SE[n] bit of serial channel enable status register m (SEm) is set to 0 and channel n stops operation.	—

Table 23.98 Processing procedure for framing error (2 of 2)

Step	Software manipulation		State of the hardware	Note
<5>	Synchronization with other party of communication		—	Synchronization with the other party of communication is re-established and communication is resumed because it is considered that a framing error has occurred because the start bit has been shifted.
<6>	Sets the SS[n] bit of serial channel start register m (SSm) to 1.	→	The SE[n] bit of serial channel enable status register m (SEm) is set to 1 and channel n is enabled to operate.	—

Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 11

23.7 LIN Communication Operation

23.7.1 LIN Transmission

Of UART transmission, UART2 supports LIN communication.

For LIN transmission, channel 0 of unit 1 is used.

Table 23.99 shows the specification of LIN transmission.

Table 23.99 Specification of LIN transmission

UART	UART0	UART1	UART2
Support of LIN communication	Not supported	Not supported	Supported
Target channel	—	—	Channel 0 of SAU1
Pins used	—	—	TxD2
Interrupt	—	—	SAU1_ENDI0
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.		
Error detection flag	None		
Transfer data length	8 bits		
Transfer rate*1	Max. f_{MCK}/6 [bps] (SDR10.STCLK[6:0] = 2 or more), Min. PCLKB/ (2 × 2^{15} × 128) [bps]		
Data phase	Non-reverse output (default: high level) Reverse output (default: low level)		
Parity bit	No parity bit		
Stop bit	Appending 1 bit		
Data direction	LSB first		

Note: f_{MCK}: Operation clock frequency of target channel

Note 1. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics specified in the electrical characteristics. For details, see section 37, Electrical Characteristics . In general, 2.4, 9.6, or 19.2 kbps is often used in LIN communication.

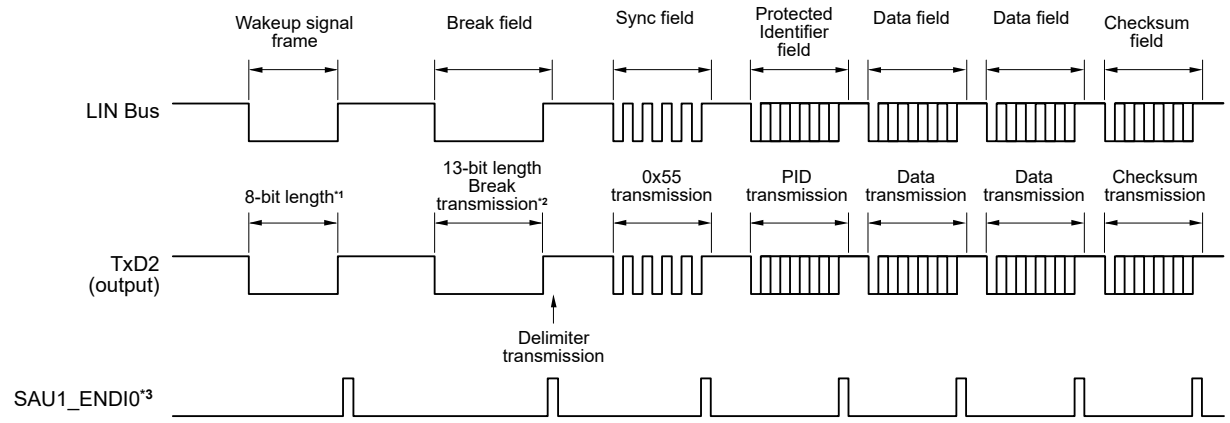
LIN stands for Local Interconnect Network and is a low-speed (1 to 20 kbps) serial communication protocol designed to reduce the cost of an automobile network.

Communication of LIN is single-master communication and up to 15 slaves can be connected to one master. The slaves are used to control switches, actuators, and sensors, which are connected to the master via LIN. Usually, the master is connected to a network such as CAN (Controller Area Network).

A LIN bus is a single-wire bus to which nodes are connected via transceiver conforming to ISO9141.

According to the protocol of LIN, the master transmits a frame by attaching baud rate information to it. A slave receives this frame and corrects a baud rate error from the master. If the baud rate error of a slave is within ±15%, communication can be established.

Figure 23.43 outlines a transmission operation of LIN.



Note: The interval between fields is controlled by software.

Note 1. Set the baud rate in accordance with the wakeup signal regulations and transmit data of 0x80.

Note 2. A break field is defined to have a width of 13 bits and output a low level. Where the baud rate for main transfer is N [bps], therefore, the baud rate of the break field is calculated as follows.

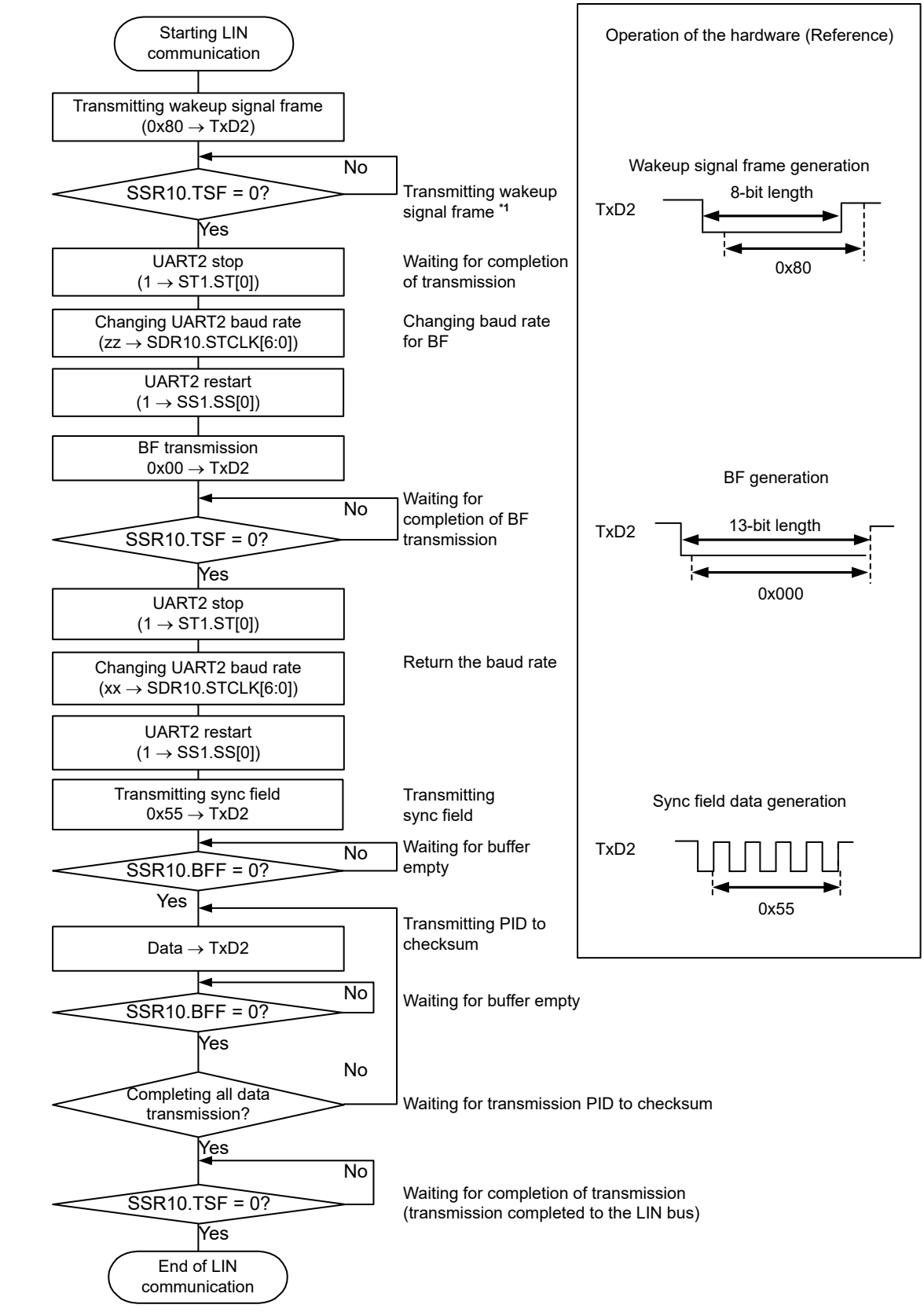
$$(\text{Baud rate of break field}) = \frac{9}{13} \times N$$

By transmitting data of 0x00 at this baud rate, a break field is generated.

Note 3. SAU1_ENDI0 is output on completion of transmission. SAU1_ENDI0 is also output at BF transmission.

Figure 23.43 Transmission operation of LIN

Figure 23.44 shows the flowchart for LIN transmission.



Note: This flow assumes that the initial setting of the UART is completed and transmission is enabled.
Note 1. This is only required if the LIN bus is started from Sleep mode.

Figure 23.44 Flowchart for LIN transmission

23.7.2 LIN Reception

Of UART reception, UART2 supports LIN communication.

For LIN reception, channel 1 of unit 1 is used.

Table 23.100 shows the specification of LIN reception.

Table 23.100 Specification of LIN reception

UART	UART0	UART1	UART2
Support of LIN communication	Not supported	Not supported	Supported
Target channel	—	—	Channel 1 of SAU1
Pins used	—	—	RxD2
Interrupt	—	—	SAU1_ENDI1
	Transfer end interrupt only (setting the buffer empty interrupt is prohibited)		
Error interrupt	—	—	SAU1_INTSRE2
Error detection flag	• Framing Error detection flag (SSR11.FEF) • Overrun Error detection flag (SSR11.OVF)		
Transfer data length	8 bits		
Transfer rate*1	Max. f_{MCK}/6 [bps] (SDR11.STCLK[6:0] = 2 or more), Min. PCLKB/ (2 × 2^{15} × 128) [bps]		
Data phase	Non-reverse output (default: high level) Reverse output (default: low level)		
Parity bit	No parity bit (the parity bit is not checked)		
Stop bit	Check the first bit		
Data direction	LSB first		

Note: f_{MCK}: Operation clock frequency of target channel

Note 1. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics specified in the electrical characteristics. For details, see section 37, Electrical Characteristics .

Figure 23.45 shows a reception operation of LIN.

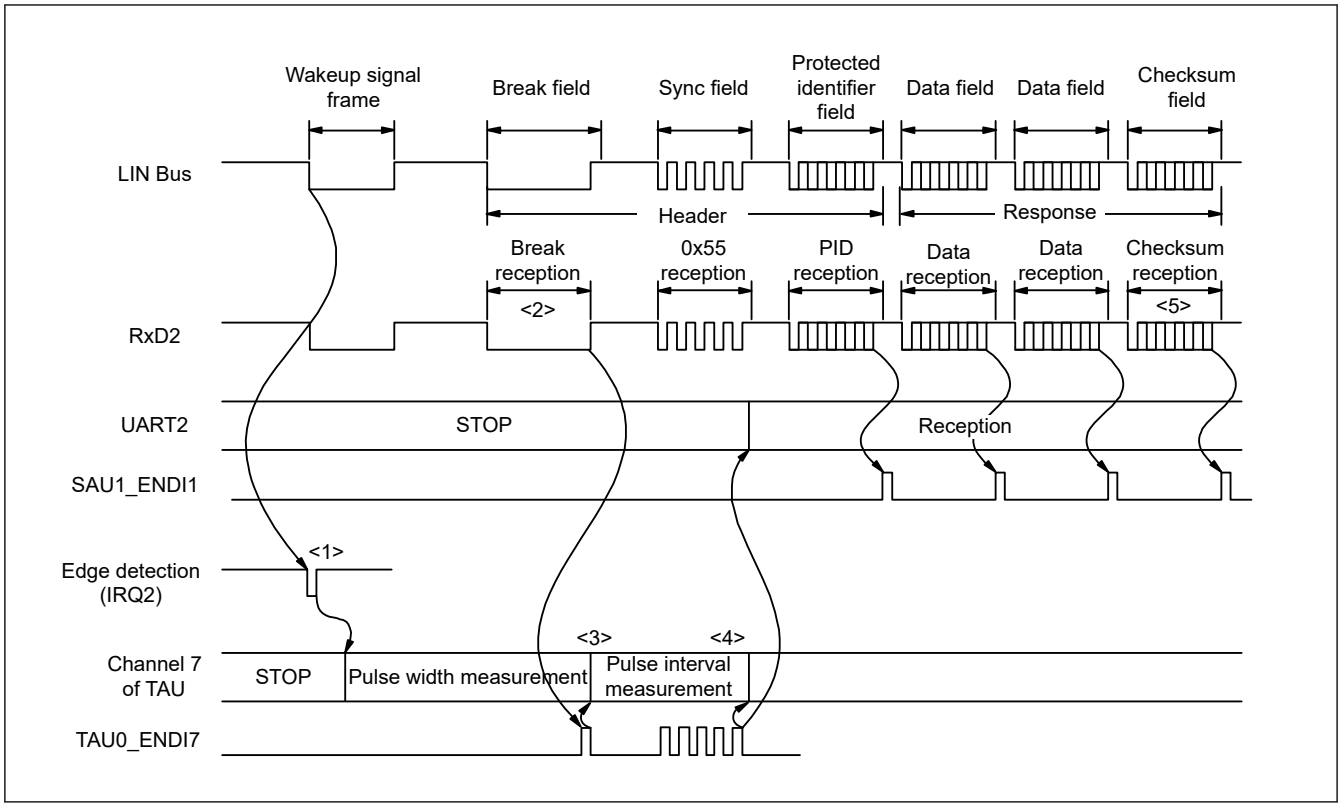


Figure 23.45 Reception operation of LIN

The flow of reception processing is as follows.

<1> The wakeup signal is detected by using an edge on the external interrupt pin (IRQ2). When the wakeup signal is detected, set channel 7 of TAU to the pulse width measurement function to measure the low-level width of the BF signal. Then wait for BF signal reception.

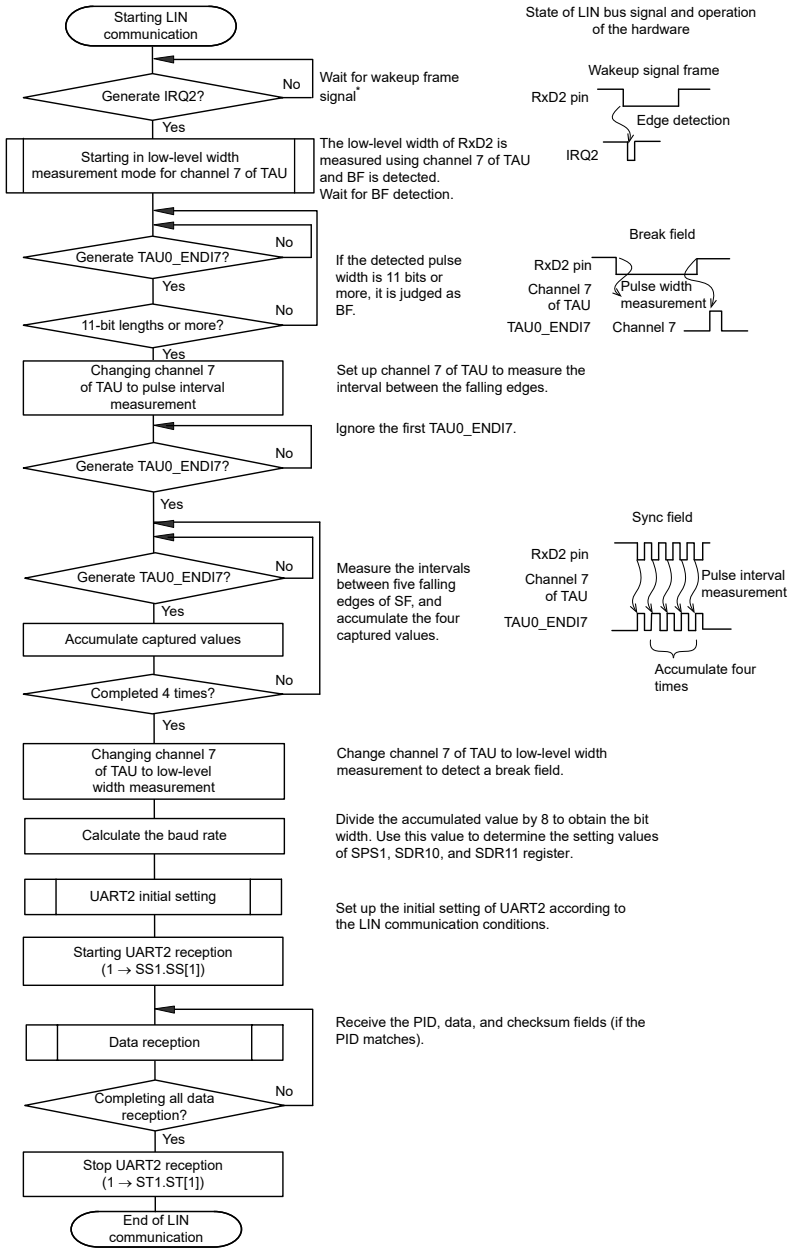
<2> Channel 7 of TAU starts measuring the low-level width on detection of the falling edge of the BF signal, and then captures the data on detection of the rising edge of the BF signal. The captured data is used to determine whether it is the BF signal.

<3> When the BF signal has been received normally, change channel 7 of TAU to pulse interval measurement and measure the interval between the falling edges of the RxD2 signal in the Sync field four times. (See section 18.7.4. Operation for Input Pulse Interval Measurement).

<4> Calculate a baud rate error from the bit interval of sync field (SF). Stop UART2 once and adjust (reset) the baud rate.

<5> The checksum field should be distinguished by software. In addition, processing to initialize UART2 after the checksum field is received and to wait for reception of BF should also be performed by software.

Figure 23.46 shows the flowchart of LIN reception.



Note: This is only required if the LIN bus is in Sleep mode.

Figure 23.46 Flowchart of LIN reception

The wakeup signal transmitted from the master of LIN is received by detecting an edge of an external interrupt (IRQ2). The length of the sync field transmitted from the master can be measured by using the external event capture operation of the timer array unit to calculate a baud-rate error.

By using the port input switching control (the ISC.ISC1 bit), the signal input to the reception port (RxD2) sent to the timer array unit without additional external connections.

To use RxD2 as IRQ2 input pin, set the corresponding P102PFS.ISEL bit to 1. For details, see section 16, I/O Ports.

The peripheral functions used for the LIN communication operation are as follows.

<Peripheral functions used>

- External interrupt (IRQ2): wakeup signal detection
  Usage: To detect an edge of the wakeup signal and the start of communication
- Channel 7 of timer array unit: baud rate error detection, break field detection.

Usage: To detect the length of the sync field (SF) and divide it by the number of bits in order to detect a baud rate error. (The interval of the edge input to RxD2 is measured in the capture mode.)
To measure the low-level width to detect the break field (BF).

- Channels 0 and 1 (UART2) of serial array unit 1 (SAU1)

23.8 Operation of Simplified I²C Communication

This is a clocked communication function to communicate with two or more devices by using two lines: serial clock (SCL) and serial data (SDA). This simplified I²C is designed for single communication with a device such as EEPROM, flash memory, or A/D converter, and therefore, it functions only as a master.

Operate the control registers by software to set the start and stop conditions while observing the specifications of the I²C bus line.

[Data transmission and reception]

- Master transmission, master reception (only master function with a single master)
- ACK output function*1 and ACK detection function
- Data length of 8 bits
  (when an address is transmitted, the address is specified by the upper 7 bits, and the least significant bit is used for R/W control.)
- Generation of start condition and stop condition for software

[Interrupt function]

- Transfer end interrupt

[Error detection flag]

- Overrun error
- ACK error

[Functions not supported by simplified I²C]

- Slave transmission, slave reception
- Multi-master function (arbitration loss detection function)
- Clock stretch detection

Note 1. When receiving the last data, ACK is not output if 0 is written to the SOEm.SOE[n] bit and serial communication data output is stopped. See (2) Processing flow for details.

Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 11

The channel supporting simplified I²C is channels 0 to 3 of SAU0 and channel 0 and 1 of SAU1. section 23, Serial Array Unit (SAU) to section 23, Serial Array Unit (SAU) show the channels supporting simplified I²C for each product.

Simplified I²C performs the following four types of communication operations:

- Address field transmission (see section 23.8.1. Address Field Transmission)
- Data transmission (see section 23.8.2. Data Transmission)
- Data reception (see section 23.8.3. Data Reception)
- Stop condition generation (see section 23.8.4. Stop Condition Generation.)

23.8.1 Address Field Transmission

Address field transmission is a transmission operation that first executes in I²C communication to identify the target for transfer (slave). After a start condition is generated, an address (7 bits) and a transfer direction (1 bit) are transmitted in one frame.

Table 23.101 shows the specification of address field transmission of Simplified I²C.

Table 23.101 Specification of address field transmission of simplified I²C

Simplified I²C	IIC00	IIC01	IIC10	IIC11	IIC20	IIC21
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1
Pins used	SCL00, SDA00*1	SCL01, SDA01*1	SCL10, SDA10*1	SCL11, SDA11*1	SCL20, SDA20*1	SCL21, SDA21*1
Interrupt	SAU0_ENDI0	SAU0_ENDI1	SAU0_ENDI2	SAU0_ENDI3	SAU1_ENDI0	SAU1_ENDI1
	Transfer end interrupt only (setting the buffer empty interrupt is prohibited)					
Error detection flag	ACK error detection flag (SSRmn.PEF)					
Transfer data length	8 bits (transmitted with specifying the higher 7 bits as address and the least significant bit as R/W control)					
Transfer rate*2	Max.f_{MCK}/4 [Hz] (SDRmn.STCLK[6:0] = 1 or more). However, the following condition must be satisfied in each mode of I²C: ● Max. 1 MHz (fast mode plus) ● Max. 400 kHz (fast mode) ● Max. 100 kHz (standard mode)					
Data level	Non-reverse output (default: high level)					
Parity bit	No parity bit					
Stop bit	Appending 1 bit (for ACK transmission and reception timing)					
Data direction	MSB first					

Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 11, gh: Port number (g = 0 to 4, h = 00 to 15)
Note: f_{MCK}: Operation clock frequency of target channel
Note 1. To perform communication using simplified I²C, set the NMOS open drain output mode with the Port gh Pin Function Select Register (PghPFS). For details, see section 16, I/O Ports.
Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics specified in the electrical characteristics. For details, see section 37, Electrical Characteristics .

(1) Register setting

Table 23.102 to Table 23.107 show examples of the register contents for address field transmission of simplified I²C.

(a) Serial mode register mn (SMRmn)

Table 23.102 Example of serial mode register mn (SMRmn) contents for address field transmission of simplified I²C (1 of 2)

Bit	Symbol	Set value	Function
0	MD0	0	Interrupt source of channel n 0: Transfer end interrupt
2:1	MD1[1:0]	10b	Setting of operation mode of channel n 1 0: Simplified I²C mode
5:3	—	100b	Setting disabled (set to the initial value)
6	SIS0	0	Setting is fixed in the simplified I²C mode
7	—	0	Setting disabled (set to the initial value)
8	STS	0	Selection of start trigger source 0: Only software trigger is valid (selected for simplified SPI, UART transmission, and simplified I²C).
13:9	—	00000b	Setting disabled (set to the initial value)
14	CCS	0	Selection of transfer clock (f_{TCLK}) of channel n 0: Divided operation clock f_{MCK} specified by the CKS bit

Table 23.102 Example of serial mode register mn (SMRmn) contents for address field transmission of simplified I^2C (2 of 2)

Bit	Symbol	Set value	Function
15	CKS	0/1	Operation clock (f_{MCK}) of channel n 0: Prescaler output clock CKm0 set by the SPSm register 1: Prescaler output clock CKm1 set by the SPSm register

(b) Serial communication operation setting register mn (SCRmn)

Table 23.103 Example of serial communication operation setting register mn (SCRmn) contents for address field transmission of simplified I^2C

Bit	Symbol	Set value	Function
1:0	DLS[1:0]	11b	Setting of data length 1 1: 8-bit data length
3:2	—	01b	Setting disabled (set to the initial value)
5:4	SLC[1:0]	01b	Setting of stop bit 0 1: Appending 1 bit (ACK)
6	—	0	Setting disabled (set to the initial value)
7	DIR	0	This bit is fixed in simplified I^2C mode because it is for simplified SPI and UART modes.
9:8	PTC[1:0]	00b	This bit is fixed in simplified I^2C mode because it is for UART mode.
10	EOC	0	This bit is fixed in simplified I^2C mode because it is for UART receive mode.
11	—	0	Setting disabled (set to the initial value)
13:12	DCP[1:0]	00b	This bit is fixed in simplified I^2C mode because it is for simplified SPI mode.
15:14	TRXE[1:0]	10b	Setting TRXE[1:0] = 10b is fixed in the simplified I^2C address field transmission

(c) Serial data register mn (SDRmn)

Table 23.104 Example of serial data register mn (SDRmn) contents for address field transmission of simplified I^2C

Bit	Symbol	Set value	Function
7:0	DAT[7:0]	0x00 to 0xFF	Slave address + R/W (Transmit data setting)
8	DAT[8]	0	0 Fixed
15:9	STCLK[6:0]	0x01 to 0x7F	Baud rate setting (Operation clock (f_{MCK}) division setting)

(d) Serial output register m (SOm)

Start condition is generated by manipulating the SOm.SO[n] bit.

Table 23.105 Example of serial output register m (SOm) contents for address field transmission of simplified I^2C

Bit	Symbol	Set value	Function
n	SO[n]	0/1	Serial data output of channel n 0: Serial data output value is 0 1: Serial data output value is 1
n+8	CKO[n]	0/1	Communication starts when a bit is 1 if the clock phase is non-reversed (SCRmn.DCP[0] = 0). If the clock phase is reversed (SCRmn.DCP[0] = 1), communication starts when a bit is 0

(e) Serial output enable register m (SOEm)

SOEm.SOE[n] = 0 until the start condition is generated, and SOEm.SOE[n] = 1 after generation.

Table 23.106 Example of serial output enable register m (SOEm) contents for address field transmission of simplified I^2C

Bit	Symbol	Set value	Function
n	SOE[n]	0/1	Serial output enable or stop of channel n 0: Stop output by serial communication operation 1: Enable output by serial communication operation

(f) Serial channel start register m (SSm)

Set only the bit of the target channel to 1. SSm.SS[n] = 0 until the start condition is generated, and SSm.SS[n] = 1 after generation.

Table 23.107 Example of serial channel start register m (SSm) contents for address field transmission of simplified I^2C

Bit	Symbol	Set value	Function
n	SS[n]	0/1	Operation start trigger of channel n 0: No trigger operation 1: Set the SEm.SE[n] bit to 1 to place the channel in the communications waiting state

Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 11

Note: 0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Table 23.108 shows the procedure for initial setting of simplified I^2C address field transmission.

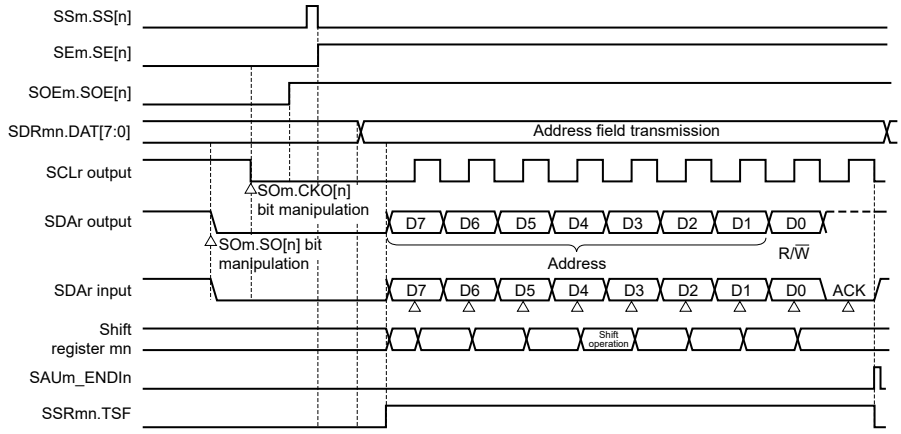
Table 23.108 Initial setting procedure for simplified I^2C address field transmission

Step		Process	Detail
Procedure for initial setting of I^2C address field transmission	<1>	Starting initial setting	—
	<2>	Setting the SPSm register	Set the operation clock.
	<3>	Setting the SMRmn register	Set an operation mode.
	<4>	Setting the SCRmn register	Set a communication format.
	<5>	Setting the SDRmn register	Set a transfer baud rate (setting the transfer clock by dividing the operation clock (f_{MCK})).
	<6>	Setting the SOm register	Set the initial output level (1) of the serial data (SOm.SO[n]) and serial clock (SOm.CKO[n]).
	<7>	Setting port	Enable data output, clock output, and NMOS open-drain output of the target channel.
	<8>	Completing initial setting	—

Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 11

(3) Processing flow

Figure 23.47 shows the timing of address field transmission.



Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), r: IIC number (r = 00, 01, 10, 11, 20, 21), mn = 00 to 03, 10 to 11

Figure 23.47 Timing of address field transmission

Table 23.109 shows the procedure for simplified I^2C address field transmission.

Table 23.109 Procedure for simplified I^2C address field transmission

Step		Process	Detail
Procedure for simplified I^2C address field transmission	<1>	Transmitting address field	—
	<2>	Default setting	For the initial setting, see Table 23.108.
	<3>	Writing 0 to the SOm.SO[n] bit	Set the SOm.SO[n] bit to 0 Start condition generate Secure a hold time of SCL signal
	<4>	Wait	
	<5>	Writing 0 to the SOm.CKO[n] bit	Drive the SCL signal low and prepare for communications.
	<6>	Writing 1 to the SOEm.SOE[n] bit	Enable serial output
	<7>	Writing 1 to the SSm.SS[n] bit	Enable serial communications.
	<8>	Writing address and R/W data to SDRmn.DAT[7:0] bits	Transmitting address field
	<9>	Wait until transfer end interrupt generated.	Wait for address field transmission to complete. Approve the interrupt request.
	<10>	Check if ACK responded. If yes, go to step <11>. If no, go to communication error processing	ACK response from the slave is confirmed in the SSRmn.PEF flag. If ACK (SSRmn.PEF = 0), go to the next processing, if NACK (SSRmn.PEF = 1), go to error processing.
	<11>	Address field transmission completed	—
	<12>	Go to data transmission flow and data reception flow	—

Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 11

23.8.2 Data Transmission

Data transmission is an operation to transmit data to the target for transfer (slave) after transmission of an address field. After all data are transmitted to the slave, a stop condition is generated and the bus is released.

Table 23.110 shows the specification of data transmission of simplified I^2C.

Table 23.110 Specification of data transmission of simplified I²C

Simplified I²C	IIC00	IIC01	IIC10	IIC11	IIC20	IIC21
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1
Pins used	SCL00, SDA00*1	SCL01, SDA01*1	SCL10, SDA10*1	SCL11, SDA11*1	SCL20, SDA20*1	SCL21, SDA21*1
Interrupt	SAU0_ENDI0	SAU0_ENDI1	SAU0_ENDI2	SAU0_ENDI3	SAU1_ENDI0	SAU1_ENDI1
	Transfer end interrupt only (setting the buffer empty interrupt is prohibited.)					
Error detection flag	ACK error flag (SSRmn.PEF)					
Transfer data length	8 bits					
Transfer rate*2	Max.f_{MCK}/4 [Hz] (SDRmn.STCLK[6:0] = 1 or more). However, the following condition must be satisfied in each mode of I²C: • Max. 1 MHz (fast mode plus) • Max. 400 kHz (fast mode) • Max. 100 kHz (standard mode)					
Data level	Non-reverse output (default: high level)					
Parity bit	No parity bit					
Stop bit	Appending 1 bit (for ACK reception timing)					
Data direction	MSB first					

Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 11, gh: Port number (g = 0 to 4, h = 00 to 15)
Note: f_{MCK}: Operation clock frequency of target channel
Note 1. To perform communication using simplified I²C, set the NMOS open drain output mode with the Port gh Pin Function Select Register (PghPFS). For details, see section 16, I/O Ports.
Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics specified in the electrical characteristics. For details, see section 37, Electrical Characteristics.

(1) Register setting

Table 23.111 to Table 23.116 show examples of the register contents for data transmission of simplified I²C.

(a) Serial mode register mn (SMRmn)

Do not manipulate this register during data transmission and reception.

Table 23.111 Example of serial mode register mn (SMRmn) contents for data transmission of simplified I²C (1 of 2)

Bit	Symbol	Set value	Function
0	MD0	0	Interrupt source of channel n 0: Transfer end interrupt
2:1	MD1[1:0]	10b	Setting of operation mode of channel n 1 0: Simplified I²C mode
5:3	—	100b	Setting disabled (set to the initial value)
6	SIS0	0	Setting is fixed in the simplified I²C mode
7	—	0	Setting disabled (set to the initial value)
8	STS	0	Selection of start trigger source 0: Only software trigger is valid (selected for simplified SPI, UART transmission, and simplified I²C)
13:9	—	00000b	Setting disabled (set to the initial value)
14	CCS	0	Selection of transfer clock (f_{TCLK}) of channel n 0: Divided operation clock f_{MCK} specified by the CKS bit

Table 23.111 Example of serial mode register mn (SMRmn) contents for data transmission of simplified I^2C (2 of 2)

Bit	Symbol	Set value	Function
15	CKS	0/1	Operation clock (f_{MCK}) of channel n 0: Prescaler output clock CKm0 set by the SPSm register 1: Prescaler output clock CKm1 set by the SPSm register

(b) Serial communication operation setting register mn (SCRmn)

Do not manipulate the bits of this register, except the TRXE[1:0] bits, during data transmission and reception.

Table 23.112 Example of serial communication operation setting register mn (SCRmn) contents for data transmission of simplified I^2C

Bit	Symbol	Set value	Function
1:0	DLS[1:0]	11b	Setting of data length 1 1: 8-bit data length
3:2	—	01b	Setting disabled (set to the initial value)
5:4	SLC[1:0]	01b	Setting of stop bit 0 1: Appending 1 bit (ACK)
6	—	0	Setting disabled (set to the initial value)
7	DIR	0	This bit is fixed in simplified I^2C mode because it is for simplified SPI and UART modes.
9:8	PTC[1:0]	00b	This bit is fixed in simplified I^2C mode because it is for UART mode.
10	EOC	0	This bit is fixed in simplified I^2C mode because it is for UART receive mode.
11	—	0	Setting disabled (set to the initial value)
13:12	DCP[1:0]	00b	This bit is fixed in simplified I^2C mode because it is for simplified SPI mode.
15:14	TRXE[1:0]	10b	Setting TRXE[1:0] = 10b is fixed in the simplified I^2C data transmission

(c) Serial data register mn (SDRmn)

During data transmission and reception, valid only lower 8-bits.

Table 23.113 Example of serial data register mn (SDRmn) contents for data transmission of simplified I^2C

Bit	Symbol	Set value	Function
7:0	DAT[7:0]	0x00 to 0xFF	Transmit data (Transmit data setting)
8	DAT[8]	0	0 Fixed
15:9	STCLK[6:0]	0x01 to 0x7F	Baud rate setting Because the setting is completed by address field transmission, set the same value as before.

(d) Serial output register m (SOm)

Do not manipulate this register during data transmission and reception.

Table 23.114 Example of serial output register m (SOm) contents for data transmission of simplified I^2C

Bit	Symbol	Set value	Function
n	SO[n]	0/1	The value varies depending on the communication data during communication operation.
n+8	CKO[n]	0/1	The value varies depending on the communication data during communication operation.

(e) Serial output enable register m (SOEm)

Do not manipulate this register during data transmission and reception.

Table 23.115 Example of serial output enable register m (SOEm) contents for data transmission of simplified I^2C

Bit	Symbol	Set value	Function
n	SOE[n]	1	Serial output enable or stop of channel n 1: Enables output by serial communication operation.

(f) Serial channel start register m (SSm)

Do not manipulate this register during data transmission and reception.

Table 23.116 Example of serial channel start register m (SSm) contents for data transmission of simplified I^2C

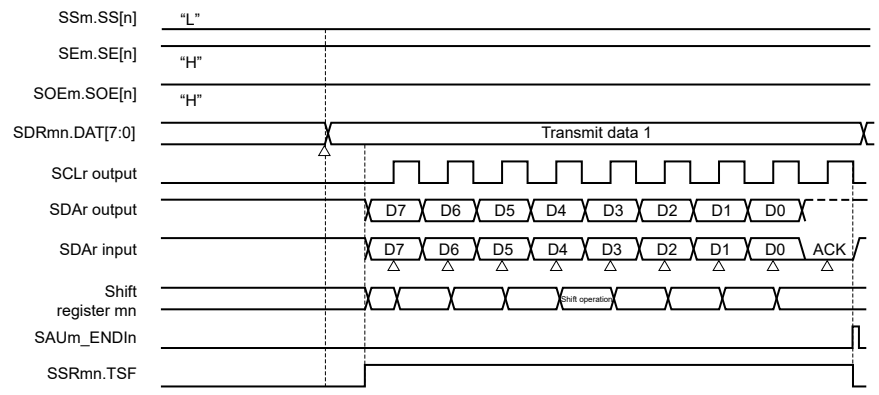
Bit	Symbol	Set value	Function
n	SS[n]	0/1	Operation start trigger of channel n 0: No trigger operation 1: Set the SEm.SE[n] bit to 1 to place the channel in the communications waiting state.

Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 11.

Note: 0/1: Set to 0 or 1 depending on the usage of the user.

(2) Processing flow

Figure 23.48 shows the timing of data transmission.



Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), r: IIC number (r = 00, 01, 10, 11, 20, 21), mn = 00 to 03, 10 to 11

Figure 23.48 Timing of data transmission

Table 23.117 shows the procedure for simplified I^2C data transmission.

Table 23.117 Procedure for simplified I²C data transmission

Step		Process	Detail
Procedure for simplified I²C data transmission	<1>	Address field transmission completed	—
	<2>	Starting data transmission	—
	<3>	Writing data to SDRmn.DAT[7:0] bits	Transmission start by writing
	<4>	Wait until transfer end interrupt generated.	Wait for transmission to complete. Approve the interrupt request.
	<5>	Check if ACK is responded. If yes, go to step <6>. If no, go to Communication error processing.	ACK acknowledgment from the slave. If ACK (SSRmn.PEF = 0), go to the next process. If NACK (SSRmn.PEF = 1), go to error handling.
	<6>	If data transfer completed, go to step <7>. If no, go to step <3>.	—
	<7>	Data transmission completed	—
	<8>	Stop condition generation	—

Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 11

23.8.3 Data Reception

Data reception is an operation to receive data from the target for transfer (slave) after transmission of an address field. After all data are received from the slave, a stop condition is generated and the bus is released.

Table 23.118 shows the specification of data reception of simplified I²C.

Table 23.118 Specification of data reception of simplified I²C

Simplified I²C	IIC00	IIC01	IIC10	IIC11	IIC20	IIC21
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1
Pins used	SCL00, SDA00*1	SCL01, SDA01*1	SCL10, SDA10*1	SCL11, SDA11*1	SCL20, SDA20*1	SCL21, SDA21*1
Interrupt	SAU0_ENDI0	SAU0_ENDI1	SAU0_ENDI2	SAU0_ENDI3	SAU1_ENDI0	SAU1_ENDI1
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)					
Error detection flag	Overrun error detection flag (SSRmn.OVF) only					
Transfer data length	8 bits					
Transfer rate*2	Max.f_{MCK}/4 [Hz] (SDRmn.STCLK[6:0] = 1 or more). However, the following conditions must be satisfied in each mode of I²C: • Max. 1 MHz (fast mode plus) • Max. 400 kHz (fast mode) • Max. 100 kHz (standard mode)					
Data level	Non-reverse output (default: high level)					
Parity bit	No parity bit					
Stop bit	Appending 1 bit (ACK transmission)					
Data direction	MSB-first					

Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 11, gh: Port number (g = 0 to 4, h = 00 to 15)
Note: f_{MCK}: Operation clock frequency of target channel
Note 1. To perform communication using simplified I²C, set the NMOS open drain output mode with the Port gh Pin Function Select Register (PghPFS). For details, see section 16, I/O Ports.
Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics specified in the electrical characteristics. For details, see section 37, Electrical Characteristics.

(1) Register setting

Table 23.119 to Table 23.124 show examples of the register contents for data reception of simplified I²C.

(a) Serial mode register mn (SMRmn)

Do not manipulate this register during data transmission and reception.

Table 23.119 Example of serial mode register mn (SMRmn) contents for data reception of simplified I^2C

Bit	Symbol	Set value	Function
0	MD0	0	Interrupt source of channel n 0: Transfer end interrupt
2:1	MD1[1:0]	10b	Setting of operation mode of channel n 1 0: Simplified I^2C mode
5:3	—	100b	Setting disabled (set to the initial value)
6	SIS0	0	Setting is fixed in the simplified I^2C mode
7	—	0	Setting disabled (set to the initial value)
8	STS	0	Selection of start trigger source 0: Only software trigger is valid (selected for simplified SPI, UART transmission, and simplified I^2C).
13:9	—	00000b	Setting disabled (set to the initial value)
14	CCS	0	Selection of transfer clock (f_{TCLK}) of channel n 0: Divided operation clock f_{MCK} specified by the CKS bit
15	CKS	0/1	Operation clock (f_{MCK}) of channel n 0: Prescaler output clock CKm0 set by the SPSm register 1: Prescaler output clock CKm1 set by the SPSm register

(b) Serial communication operation setting register mn (SCRmn)

Do not manipulate the bits of this register, except the TRXE[1:0] bits, during data transmission and reception.

Table 23.120 Example of serial communication operation setting register mn (SCRmn) contents for data reception of simplified I^2C

Bit	Symbol	Set value	Function
1:0	DLS[1:0]	11b	Setting of data length 1 1: 8-bit data length
3:2	—	01b	Setting disabled (set to the initial value)
5:4	SLC[1:0]	01b	Setting of stop bit 0 1: Appending 1 bit (ACK)
6	—	0	Setting disabled (set to the initial value)
7	DIR	0	This bit is fixed in simplified I^2C mode because it is for simplified SPI and UART modes.
9:8	PTC[1:0]	00b	This bit is fixed in simplified I^2C mode because it is for UART mode.
10	EOC	0	This bit is fixed in simplified I^2C mode because it is for UART receive mode.
11	—	0	Setting disabled (set to the initial value)
13:12	DCP[1:0]	00b	This bit is fixed in simplified I^2C mode because it is for simplified SPI mode.
15:14	TRXE[1:0]	01b	Setting TRXE[1:0] = 01b is fixed in the simplified I^2C data reception

(c) Serial data register mn (SDRmn)

Table 23.121 Example of serial data register mn (SDRmn) contents for data reception of simplified I²C

Bit	Symbol	Set value	Function
7:0	DAT[7:0]	0xFF	Receive data (Dummy transmit data setting 0xFF)
8	DAT[8]	0	0 Fixed
15:9	STCLK[6:0]	0x01 to 0x7F	Baud rate setting Because the setting is completed by address field transmission, set the same value as before.

(d) Serial output register m (SOm)

Do not manipulate this register during data transmission and reception.

Table 23.122 Example of serial output register m (SOm) contents for data reception of simplified I²C

Bit	Symbol	Set value	Function
n	SO[n]	0/1	The value varies depending on the communication data during communication operation.
n+8	CKO[n]	0/1	The value varies depending on the communication data during communication operation.

(e) Serial output enable register m (SOEm)

Do not manipulate this register during data transmission, and reception.

Table 23.123 Example of serial output enable register m (SOEm) contents for data reception of simplified I²C

Bit	Symbol	Set value	Function
n	SOE[n]	0/1	Serial output enable or stop of channel n 0: Stop output by serial communication operation 1: Enable output by serial communication operation

(f) Serial channel start register m (SSm)

Do not manipulate this register during data transmission and reception.

Table 23.124 Example of serial channel start register m (SSm) contents for data reception of simplified I²C

Bit	Symbol	Set value	Function
n	SS[n]	0/1	Operation start trigger of channel n 0: No trigger operation 1: Set the SEm.SE[n] bit to 1 to place the channel in the communications waiting state.

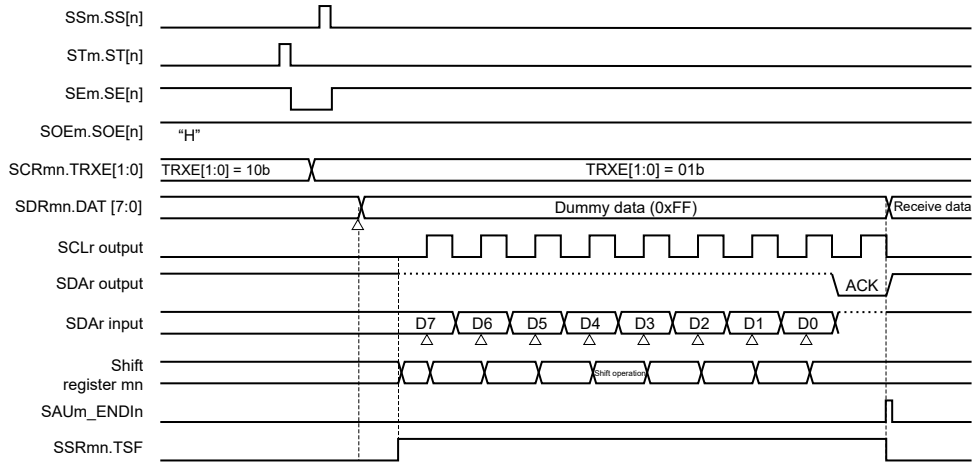
Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 11

Note: 0/1: Set to 0 or 1 depending on the usage of the user

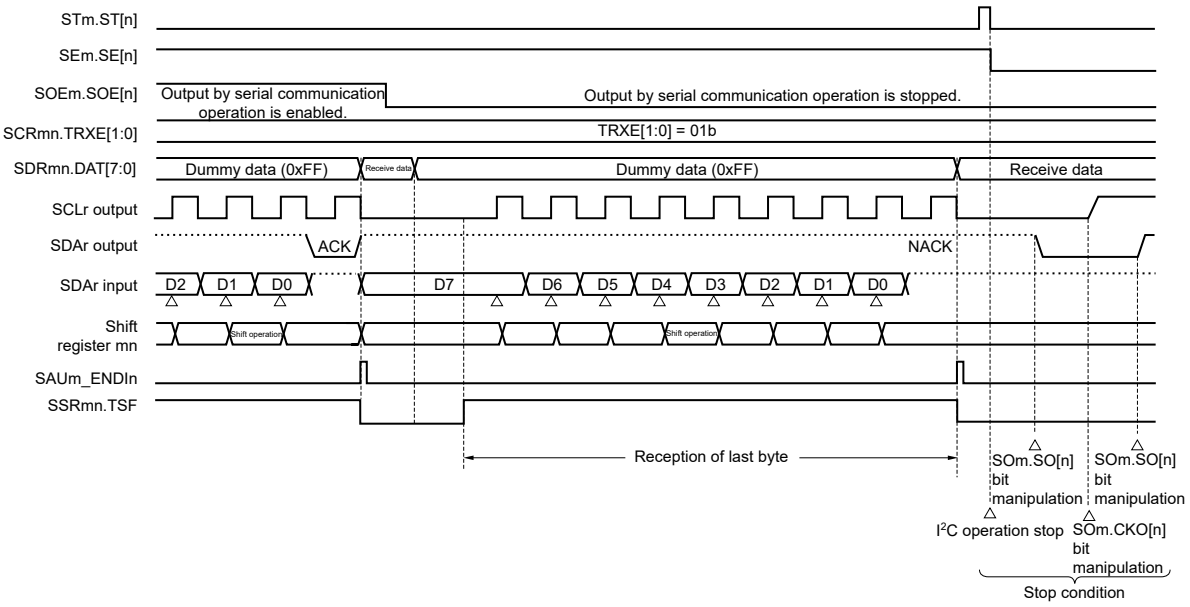
(2) Processing flow

Figure 23.49 shows the timing of data reception.

(a) When starting data reception



(b) When receiving last data



Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), r: IIC number (r = 00, 01, 10, 11, 20, 21), mn = 00 to 03, 10 to 11

Figure 23.49 Timing of data reception

Table 23.125 shows the procedure for data reception.

Table 23.125 Procedure for data reception

Step		Process	Detail
Procedure for data reception	<1>	Address field transmission completed	—
	<2>	Data reception	—
	<3>	Writing 1 to the STm.ST[n] bit	Stop operation for rewriting SCRmn register.
	<4>	Writing 01b to the SCRmn.TRXE[1:0] bits	Set the operation of the channel to receive-only mode.
	<5>	Writing 1 to the SSm.SS[n] bit	Operation restart
	<6>	Check if the last byte is received. If yes, go to step <7>. If No, go to step <8>.	Disable output so that it is not the ACK response to the last received data.
	<7>	Writing 0 to the SOEm.SOE[n] bit	
	<8>	Writing dummy data (0xFF) to the SDRmn.DAT[7:0] bits	Starting reception operation
	<9>	Check if transfer end interrupt generated If yes, go to step <10>. If No, go to step <9>.	Wait for reception to complete. Approve the interrupt request.
	<10>	Reading the SDRmn.DAT[7:0] bits	Reading receive data, perform processing (data is stored in the RAM, for example).
	<11>	Check if data transfer completed. If yes, go to step <12>. If No, go to step <6>.	—
	<12>	Data reception completed	—
	<13>	Stop condition generation	—

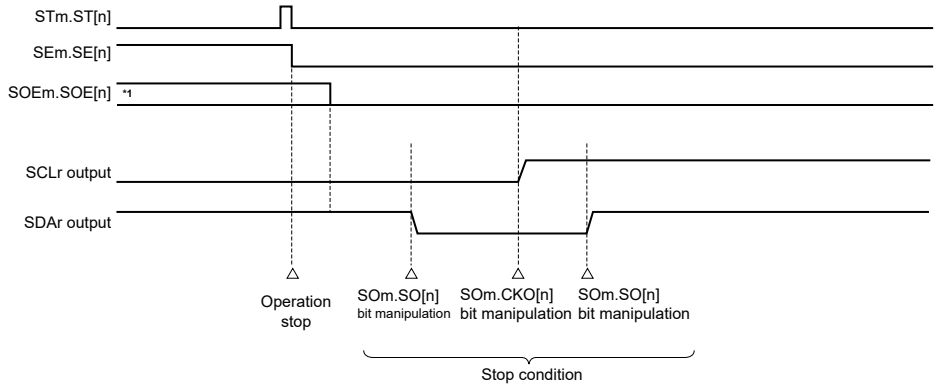
Note: ACK is not output when the last data is received (NACK). Communication is then completed by setting 1 in the ST[n] bit of serial channel stop register m (STm) to stop operation and generating a stop condition.
Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 11

23.8.4 Stop Condition Generation

After all data are transmitted to or received from the target slave, a stop condition is generated and the bus is released.

(1) Processing flow

Figure 23.50 shows the timing of stop condition generation.



Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), r: IIC number (r = 00, 01, 10, 11, 20, 21), mn = 00 to 03, 10 to 11
Note 1. During a receive operation, the SOE[n] bit of serial output enable register m (SOEm) is cleared to 0 before receiving the last data.

Figure 23.50 Timing of stop condition generation

Table 23.126 shows the procedure for stop condition generation.

Table 23.126 Procedure for stop condition generation

Step		Process	Detail
Procedure for stop condition generation	<1>	Completion of data transmission and data reception	—
	<2>	Starting generation of stop condition	—
	<3>	Writing 1 to the STm.ST[n] bit (the SEm.SE[n] bit is cleared to 0)	Stop operation (SOm.CKO[n] can be manipulated).
	<4>	Writing 0 to the SOEm.SOE[n] bit	Disable output (SOm.SO[n] can be manipulated).
	<5>	Writing 0 to the SOm.SO[n] bit	—
	<6>	Writing 1 to the SOm.CKO[n] bit	Timing to satisfy the low width standard of SCL for the I^2C bus.
	<7>	Wait	Secure a wait time so that the specifications of I^2C on the slave side are satisfied.
	<8>	Writing 1 to the SOm.SO[n] bit	—
	<9>	End of I^2C communication	—

Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 11

23.8.5 Calculating Transfer Rate

The transfer rate for simplified I^2C communication can be calculated by the following expressions.

(Transfer rate) = {Operation clock (f_{MCK}) frequency of target channel} ÷ (SDRmn.STCLK[6:0] + 1) ÷ 2

Note: SDRmn.STCLK[6:0] must not be set to 0x00. Set SDRmn.STCLK[6:0] to 0x01 or greater.

The duty ratio of the SCL signal output by the simplified I^2C is 50%. The I^2C bus specifications define that the low-level width of the SCL signal is longer than the high-level width. If 400 kbps (fast mode) or 1 Mbps (fast mode plus) is specified, therefore, the low-level width of the SCL output signal becomes shorter than the value specified in the I^2C bus specifications. Make sure that the SDRmn.STCLK[6:0] bits value satisfies the I^2C bus specifications.

Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 11

The operation clock (f_{MCK}) is determined by serial clock select register m (SPSm) and CKS bit of serial mode register mn (SMRmn). See Table 23.71.

Table 23.127 shows an example of setting an I^2C transfer rate where f_{MCK} = PCLKB = 32 MHz.

Table 23.127 Example of setting I^2C transfer rate where f_{MCK} = PCLKB = 32 MHz

I^2C transfer mode (desired transfer rate)	PCLKB = 32 MHz			
	Operation clock (f_{MCK})	SDRmn.STCLK[6:0]	Calculated transfer rate	Error from desired transfer rate
100 kHz	PCLKB/2	79	100 kHz	0.0%
400 kHz	PCLKB	41	380 kHz	5.0%*1
1 MHz	PCLKB	18	0.84 MHz	16.0%*1

Note 1. The error cannot be set to about 0% because the duty ratio of the SCL signal is 50%.

23.8.6 Procedure for Processing Errors that Occurred during Simplified I^2C Communication

The procedure for processing errors that occurred during simplified I^2C communication is described in Table 23.128 and Table 23.129.

Table 23.128 Processing procedure for overrun error

Step	Software manipulation		State of the hardware	Remark
<1>	Read serial data register mn (SDRmn).	→	The BFF flag of the SSRmn register is set to 0 and channel n is enabled to receive data	This is to prevent an overrun error if the next reception is completed during error processing.
<2>	Read serial status register mn (SSRmn).		—	The error type is identified and the read value is used to clear the error flag.
<3>	Write 1 to serial flag clear trigger register mn (SIRmn).	→	The error flag is cleared.	Only the error during reading can be cleared, by writing the value read from the SSRmn register to the SIRmn register without modification.

Table 23.129 Processing procedure for ACK error in simplified I^2C mode

Step	Software manipulation		State of the hardware	Remark
<1>	Read serial status register mn (SSRmn).		—	The error type is identified and the read value is used to clear the error flag.
<2>	Write serial flag clear trigger register mn (SIRmn).	→	The error flag is cleared.	Only the error during reading can be cleared, by writing the value read from the SSRmn register to the SIRmn register without modification.
<3>	Set the ST[n] bit of serial channel stop register m (STm) to 1.	→	The SE[n] bit of serial channel enable status register m (SEm) is set to 0 and channel n stops operation.	The slave is not ready for reception because ACK is not returned. Therefore, a stop condition is created, the bus is released, and communication is started again from the start condition. Or, a restart condition is generated and transmission can be redone from address transmission.
<4>	Create a stop condition.		—	
<5>	Create a start condition.		—	
<6>	Set the SS[n] bit of serial channel start register m (SSm) to 1.	→	The SE[n] bit of serial channel enable status register m (SEm) is set to 1 and channel n is enabled to operate.	—

Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 11

24. I2C Bus Interface (IICA)

24.1 Overview

The I2C bus interface has the following three modes:

- Operation stop mode
- I2C bus mode (multi-master supported)
- Wakeup mode

Table 24.1 shows specifications of the I2C bus interface.

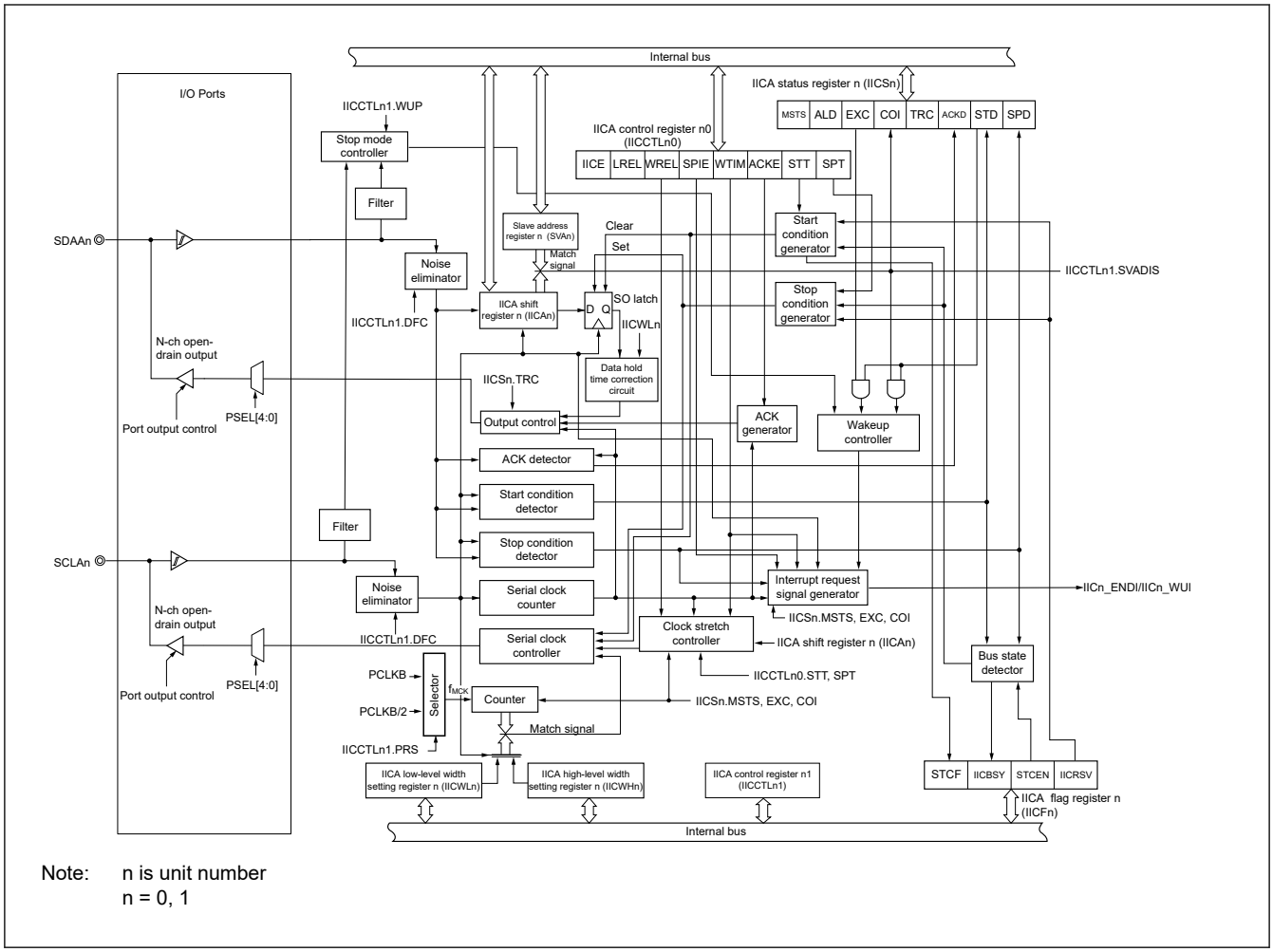
Table 24.1 I2C specifications (1 of 2)

Parameter	Specifications
Communications format	• I2C-bus format • Master or slave mode selectable • Automatic securing of the setup times, hold times, and bus-free times for the transfer rate
Transfer rate	• Standard-mode, up to 100 kbps • Fast-mode supported, up to 400 kbps • Fast-mode Plus supported, up to 1 Mbps
SCL clock	For master operation, the duty cycle of the SCLAn clock is selectable
Issuing and detecting conditions	• Start, restart, and stop conditions are automatically generated • Start conditions (including restart conditions) and stop conditions are detectable
Slave address	7- and 10-bit address formats supported, including simultaneous use
Acknowledgment	• The reception side returns ACK each time it has received 8-bit data • The transmission side usually receives ACK after transmitting 8-bit data • How ACK is generated when data is received depends on the setting of the timing of clock stretching as follows: – 8th cycle clock stretching is selected: By setting the IICCTLn0.ACKE bit to 1 before releasing from the clock stretch state, ACK is generated at the falling edge of the 8th clock cycle of the SCLAn pin – 9th cycle clock stretching is selected: ACK is generated if the IICCTLn0.ACKE bit is set to 1 in advance
Wait function (clock stretching)	During reception, the following wait periods are available by holding the SCLAn clock low: • Waiting between the 8th and 9th clock cycles • Waiting between the 9th clock cycle and the 1st clock cycle of the next transfer
Arbitration	• When several master devices simultaneously generate a start condition, communication among the master devices is performed as the number of clocks are adjusted until the data differs • When one of the master devices loses in arbitration, the SCLAn and SDAAn lines are both set to high impedance, which releases the bus • The arbitration loss is detected by checking IICSn.ALD = 1 by software at the timing of the next interrupt request
Noise cancellation	Digital noise filters for both the SCLAn and SDAAn signals
Interrupt sources	• the local address is received • an address is received while the all address match function is enabled • an extension code is received • a stop condition is detected

Table 24.1 I²C specifications (2 of 2)

Parameter	Specifications
Module-stop function	Module-stop state can be set
IIC operating modes	• Master transmit • Master receive • Slave transmit • Slave receive
Event link function (output)	• the local address is received • an address is received while the all address match function is enabled • an extension code is received • a stop condition is detected
Wakeup function	CPU can return from Software Standby mode and Snooze mode using a wakeup event

Figure 24.1 shows a block diagram of the I²C bus interface.



Note: n is unit number
n = 0, 1

Figure 24.1 Block diagram of I²C bus interface

(1) Operation stop mode

This mode is used when serial transfers are not performed. The operating power can be reduced in this mode.

(2) I²C bus mode (multi-master supported)

This mode is used for 8-bit data transfers with several devices using two lines: a serial clock (SCLAn) line and a serial data bus (SDAAn) line.

This mode complies with the I^2C bus format and the master device can send start conditions, addresses, transfer directions, acknowledges (ACK), data, and stop conditions to the slave devices, through the serial data bus. The slave device automatically detects these states and data by hardware. This function can simplify the part of application program that controls the I^2C bus.

Since the SCLAn and SDAAn pins are used for open drain outputs, I^2C bus interface requires pull-up resistors for the serial clock line and the serial data bus line.

(3) Wakeup mode

The Software Standby mode can be released by generating an interrupt request signal (IICn_ENDI/IICn_WUI) when an extension code from the master device or the local address has been received while in Software Standby mode. This can be set by using the WUP bit of IICA control register n1 (IICCTLn1).

The all address match function is enabled by setting the SVADIS bit of the ICCTLn1 register to 1, allowing any received address is to be determined as a matched address.

(4) SO latch

The SO latch is used to retain the output level of SDAAn pin.

(5) Wakeup controller

This circuit generates an interrupt request signal (IICn_ENDI/IICn_WUI) when the received address matches the address value set to the slave address register n (SVAn), when any address is received while the all address match function is enabled, or when an extension code is received.

(6) Serial clock counter

This counter counts the serial clock cycles that are output or input during transmit/receive operations and is used to verify that 8-bit data was transmitted or received.

(7) Interrupt request signal generator

This circuit controls the generation of interrupt request signals (IICn_ENDI/IICn_WUI). An I^2C interrupt request is generated by the following two triggers:

- Falling edge of the 8th or 9th clock of the serial clock (set by the IICCTLn0.WTIM bit)
- Interrupt request generated when a stop condition is detected (set by the IICCTLn0.SPIE bit)

(8) Serial clock controller

In master mode, this circuit generates the serial clock, which is output using the SCLAn pin.

(9) Clock stretch controller

This circuit controls the timing of clock stretching.

(10) ACK generator, stop condition detector, start condition detector, and ACK detector

These circuits generate or detect each state.

(11) Data hold time correction circuit

This circuit corrects the hold time for data after the falling edge of the serial clock is detected.

(12) Start condition generator

This circuit generates a start condition when the IICCTLn0.STT bit is set to 1. However, while communication reservations are disabled (IICFn.IICRSV = 1) and the bus is busy (IICFn.IICBSY = 1), start condition requests are ignored and the IICFn.STCF bit is set to 1.

(13) Stop condition generator

This circuit generates a stop condition when the IICCTLn0.SPT bit is set to 1.

(14) Bus state detector

This circuit detects whether or not the bus is released by detecting start conditions and stop conditions. However, as the bus state cannot be detected immediately after the IICA operation is enabled, the initial state is set by the IICFn.STCEN bit.

Figure 24.2 shows an example of the serial bus configuration using the I^2C bus.

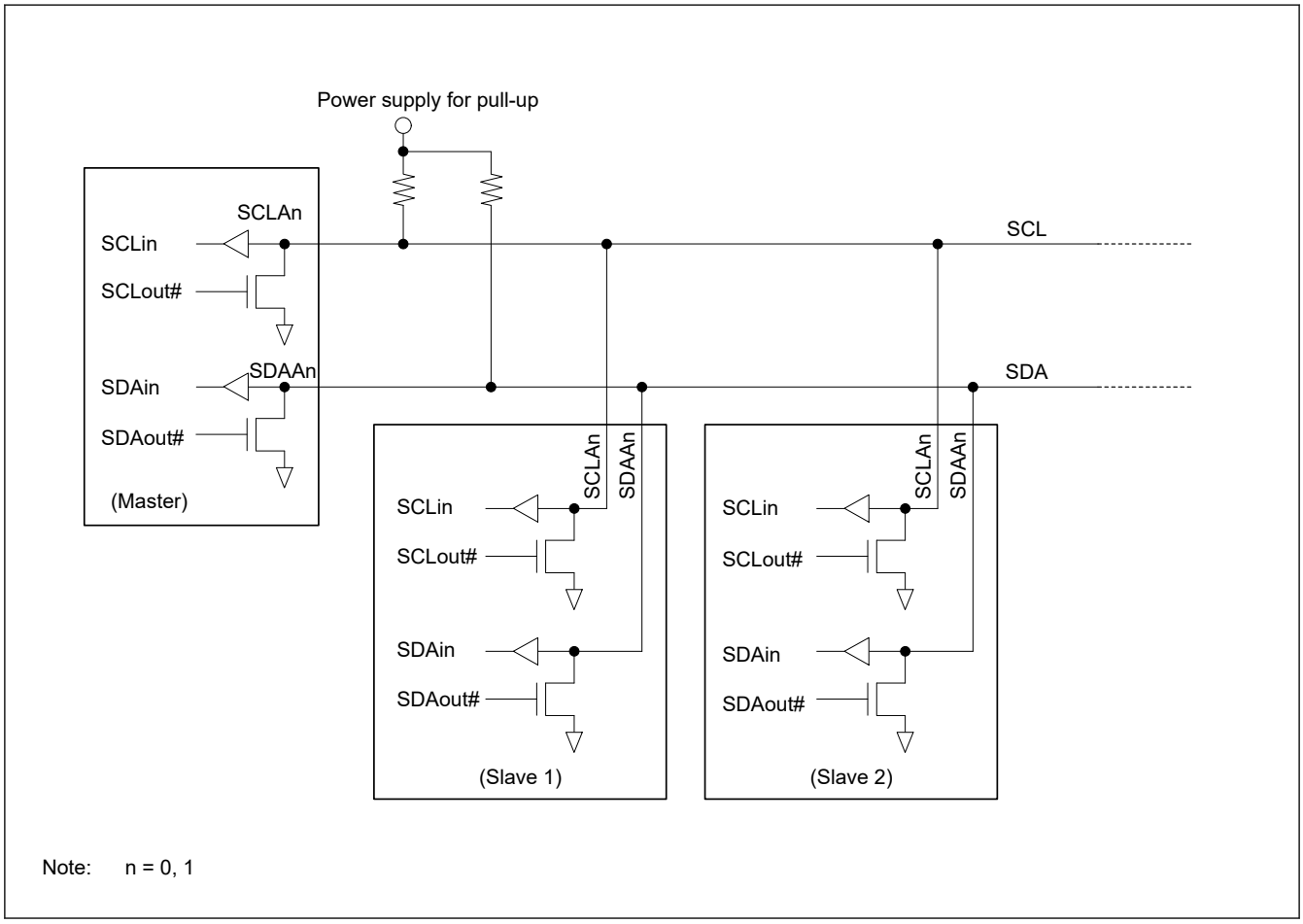


Figure 24.2 Example of the serial bus configuration using the I^2C bus

24.2 Register Descriptions

24.2.1 IICAn : IICA Shift Register n (n = 0, 1)

Base address: IICA = 0x4009_3000

Offset address: 0x00 + 0x100 × n

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

Bit	Symbol	Function	R/W
7:0	n/a	8-bit transmit and receive data for IICA of unit n	R/W

The IICAn register is used to convert 8-bit serial data to 8-bit parallel data and vice versa in synchronization with the serial clock. The IICAn register can be used for both transmission and reception.

The actual transmit and receive operations can be controlled by writing to and reading from the IICAn register. Release I^2C bus interface from the clock stretch state and start data transfer by writing data to the IICAn register during the clock stretch period.

Do not write data to the IICAn register during data transfer.

Write to or read from the IICAn register only during the clock stretch period. Accessing the IICAn register in a communication state other than during the clock stretch period is prohibited. When the device serves as the master, however, the IICAn register can be written only once after the communication trigger bit (IICCTLn0.STT) is set to 1.

When communication is reserved, write data to the IICAn register after the interrupt triggered by a stop condition is detected.

24.2.2 SVAn : Slave Address Register n (n = 0, 1)

Base address: IICA = 0x4009_3000

Offset address: 0x84 + 0x100 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	A[6:0]							—
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
7:1	A[6:0]	7-bit local address when in slave mode of unit n	R/W

This register holds 7 bits (A[6:0]) of the local address when in slave mode.

However, rewriting to this register is prohibited while IICSn.STD = 1 (while the start condition is detected).

24.2.3 IICCTLn0 : IICA Control Register n0 (n = 0, 1)

Base address: IICA = 0x4009_3000

Offset address: 0x80 + 0x100 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	IICE	LREL	WREL	SPIE	WTIM	ACKE	STT	SPT
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SPT*1	Stop condition trigger 0: Stop condition is not generated 1: Stop condition is generated (termination of master device transfer)	R/W
1	STT*2 *3	Start condition trigger 0: Do not generate a start condition 1: When bus is released (in standby state, when IICFn.IICBSY = 0): If this bit is set to 1, a start condition is generated (startup as the master). When a 3rd-party is communicating: ● When communication reservation function is enabled (IICFn.IICRSV = 0): This bit functions as the start condition reservation flag. When set to 1, automatically generates a start condition after the bus is released. ● When communication reservation function is disabled (IICFn.IICRSV = 1): Even if this bit is set to 1, the STT bit is cleared and the STT clear flag (IICFn.STCF) is set to 1. No start condition is generated. In the clock stretch state (for a master device): Generates a restart condition after release from the clock stretch state.	R/W

Bit	Symbol	Function	R/W
2	ACKE*4 *5	Acknowledgment control 0: Disable acknowledgment 1: Enable acknowledgment. During the 9th clock period, the SDAAn line is set to low level.	R/W
3	WTIM*4	Control of clock stretching and interrupt request generation 0: An interrupt request is generated on the falling edge of the 8th clock cycle. Master mode: After the output of eight clock pulses, the clock output is set to the low level and clock stretching is set. Slave mode: After the input of eight clock pulses, the clock is set to the low level and clock stretching is set for the master device. 1: An interrupt request is generated on the falling edge of the 9th clock cycle. Master mode: After the output of nine clock pulses, the clock output is set to the low level and clock stretching is set. Slave mode: After the input of 9 clock pulses, the clock is set to the low level and clock stretching is set for the master device.	R/W
4	SPIE*6	Enable and disable generation of interrupt request when stop condition is detected 0: Disable 1: Enable	R/W
5	WREL*6 *7	Release from the clock stretch state 0: The interface is not released from the clock stretch state. 1: The interface is released from the clock stretch state. After release from the clock stretch state, this bit is automatically cleared to 0.	R/W
6	LREL*6 *7	Exit from communications 0: Normal operation 1: IICA exits from the current communications and sets standby mode. This setting is automatically cleared to 0 after being executed. Its uses include cases in which a locally irrelevant extension code has been received. The SCLAn and SDAAn lines are set to high impedance. The following flags of IICA control register n0 (IICCTLn0) and the IICA status register n (IICSn) are cleared to 0. ● IICCTLn0.STT, SPT ● IICSn.MSTS, EXC, COI, TRC, ACKD, STD	R/W
7	IICE	I^2C operation enable 0: Stop operation. Reset the IICA status register n (IICSn)*8. Stop internal operation. 1: Enable operation.	R/W

Note 1. The SPT bit is always read as 0.
Note 2. The signal of this bit is invalid while the IICE bit is 0.
Note 3. The STT bit is always read as 0.
Note 4. The signal of this bit is invalid while the IICE bit is 0. Set this bit during that period.
Note 5. The set value is invalid during address transfer and if the code is not an extension code, and the all address match function is disabled.
When the device serves as a slave and the addresses match, an acknowledgment is generated regardless of the set value.
Note 6. The setting of this bit has no effect while the setting of the IICE bit is 0.
Note 7. Reading the LREL and WREL bits always returns 0.
Note 8. The IICA status register n (IICSn), the STCF and IICBSY bits of the IICA flag register n (IICFn), and the CLD and DAD bits of IICA control register n1 (IICCTLn1) are reset.

This register is used to enable or disable the I^2C operations, set the timing of clock stretching, and set other I^2C operations.

Note that the SPIE, WTIM, and ACKE bits must be set while the setting of the IICE bit is 0 or this module is in the clock stretch state. These bits can be set at the same time as setting the IICE bit 1.

When TRC bit of the IICA status register n (IICSn) is set to 1 (transmission state), WREL bit of IICA control register n0 (IICCTLn0) is set to 1 during the 9th clock and the interface is released from the clock stretch state, after which the IICSn.TRC bit is cleared (reception state) and the SDAAn line is set to the high impedance state. Release the interface from the clock stretch state while the IICSn.TRC bit is 1 (transmission state) by writing to the IICA shift register n.

If the I^2C operation is enabled (IICE = 1) when the SCLAn line is high level, the SDAAn line is low level, and the digital filter is turned on (IICCTLn1.DFC = 1), a start condition is inadvertently detected immediately. In this case, set 1 to the LREL bit immediately after enabling the I^2C operation (IICE = 1).

SPT bit (Stop condition trigger)

Cautions concerning set timing

- For master reception: Cannot be set to 1 during transfer.

Can be set to 1 only during the clock stretch period when the ACKE bit has been cleared to 0 and slave has been notified of final reception.

- For master transmission: A stop condition cannot be generated normally during the acknowledge period. Therefore, set it during the clock stretch period that follows output of the ninth clock.
- Cannot be set to 1 at the same time as start condition trigger (STT).
- The SPT bit can be set to 1 only when in master mode.
- When the WTIM bit has been cleared to 0, if the SPT bit is set to 1 during the clock stretch period that follows output of eight clock pulses, note that a stop condition will be generated during the high-level period of the ninth clock after release from the clock stretch state. The WTIM bit should be changed from 0 to 1 during the clock stretch period following the output of eight clock pulses, and the SPT bit should be set to 1 during the clock stretch period that follows the output of the ninth clock.
- Once SPT is set to 1, setting it to 1 again before the clear condition is met is not allowed.

Condition for clearing (SPT = 0)

- Cleared by loss in arbitration
- Automatically cleared after stop condition is detected
- Cleared by LREL = 1 (exit from communications)
- When IICE = 0 (operation stop)
- Reset

Condition for setting (SPT = 1)

- Set by instruction

Note: SPT bit becomes 0 when it is read after data setting.

STT bit (Start condition trigger)

Cautions concerning set timing

- For master reception: Cannot be set to 1 during transfer.
  Can be set to 1 only during the clock stretch period when the ACKE bit has been cleared to 0 and slave has been notified of final reception.
- For master transmission: A start condition cannot be generated normally during the acknowledge period.
  Set to 1 during the clock stretch period that follows output of the ninth clock.
- Cannot be set to 1 at the same time as stop condition trigger (SPT).
- Once STT is set to 1, setting it to 1 again before the clear condition is met is not allowed.

Condition for clearing (STT = 0)

- Cleared by setting the STT bit to 1 while communication reservation is prohibited.
- Cleared by loss in arbitration
- Cleared after start condition is generated by master device
- Cleared by LREL = 1 (exit from communications)
- When IICE = 0 (operation stop)
- Reset

Condition for setting (STT = 1)

- Set by instruction

ACKE bit (Acknowledgment control)

Condition for clearing (ACKE = 0)

- Cleared by instruction

- Reset

Condition for setting (ACKE = 1)

- Set by instruction

WTIM bit (Control of clock stretching and interrupt request generation)

An interrupt is generated on the falling edge of the ninth clock cycle during address transfer independently of the setting of this bit. The setting of this bit is valid when the address transfer is completed. In master mode, clock stretching is inserted at the falling edge of the ninth clock cycle during address transfer. For a slave device that has received a local address, clock stretching is inserted at the falling edge of the ninth clock cycle after an acknowledge (ACK) is issued. However, if the slave device has received an extension code, clock stretching is inserted at the falling edge of the eighth clock cycle. When an address is received while the all address match function is enabled, clock stretching is inserted at the falling edge of the eighth clock cycle.

Condition for clearing (WTIM = 0)

- Cleared by instruction
- Reset

Condition for setting (WTIM = 1)

- Set by instruction

SPIE bit (Enable and disable generation of interrupt request when stop condition is detected)

If the WUP bit of IICA control register n1 (IICCTLn1) is 1, no stop condition interrupt will be generated even if SPIE = 1.

Condition for clearing (SPIE = 0)

- Cleared by instruction
- Reset

Condition for setting (SPIE = 1)

- Set by instruction

WREL bit (Release from the clock stretch state)

When the WREL bit is set (for release from the clock stretch state) during the clock stretch period at the ninth clock pulse in the transmission state (IICSn.TRC = 1), the SDAAn line goes into the high impedance state (IICSn.TRC = 0).

Condition for clearing (WREL = 0)

- Automatically cleared after execution
- Reset

Condition for setting (WREL = 1)

- Set by instruction

LREL bit (Exit from communications)

The standby mode following exit from communications remains in effect until the following communications entry conditions are met.

- After a stop condition is detected, restart is in master mode.
- An address match, extension code reception, or address reception with the all address match function enabled occurs after the start condition.

Condition for clearing (LREL = 0)

- Automatically cleared after execution
- Reset

Condition for setting (LREL = 1)

- Set by instruction

IICE bit (I^2C operation enable)

Be sure to set this bit to 1 while the SCLAn and SDAAn lines are at high level.

Condition for clearing (IICE = 0)

- Cleared by instruction
- Reset

Condition for setting (IICE = 1)

- Set by instruction

24.2.4 IICSn : IICA Status Register n (n = 0, 1)

Base address: IICA = 0x4009_3000

Offset address: 0x01 + 0x100 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	MSTS	ALD	EXC	COI	TRC	ACKD	STD	SPD
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SPD	Detection of stop condition 0: Stop condition was not detected. 1: Stop condition was detected. Communication of the master device is terminated and the bus is released.	R
1	STD	Detection of start condition 0: Start condition was not detected. 1: Start condition was detected. This indicates that the address transfer period is in effect.	R
2	ACKD	Detection of acknowledge (ACK) 0: Acknowledge was not detected. 1: Acknowledge was detected.	R
3	TRC	Detection of transmit and receive status 0: Receive status (other than transmit status). The SDAAn line is set for high impedance. 1: Transmit status. The value in the SOn latch is enabled for output to the SDAAn line (valid starting at the falling edge of the first byte's ninth clock).	R
4	COI	Detection of matching addresses 0: Addresses do not match. 1: Addresses match. Or, the all address match function is enabled.	R
5	EXC	Detection of extension code reception 0: Extension code was not received. 1: Extension code was received. Or, the all address match function is enabled.	R
6	ALD	Detection of arbitration loss 0: This status means either that there was no arbitration, or that the arbitration result was a win. 1: This status indicates the arbitration result was a loss. The MSTS bit is cleared.	R
7	MSTS	Master status check flag 0: Slave device status or communication standby status 1: Master device communication status	R

This register indicates the state of the I^2C.

The IICSn register can only be read while the setting of IICCTLn0.STT is 1 or this module is in the clock stretch state.

Reading the IICSn register while the address match wakeup function is enabled (IICCTLn1.WUP = 1) in Software Standby mode is prohibited. When the IICCTLn1.WUP bit is changed from 1 to 0 (wakeup operation is stopped), regardless of the

IICn_ENDI/IICn_WUI interrupt request signal, the change in status is not reflected until the next start condition or stop condition is detected. To use the wakeup function, enable (IICCTLn0.SPIE = 1) the interrupt generated by detecting a stop condition and read the IICSn register after the interrupt has been detected.

SPD bit (Detection of stop condition)

Condition for clearing (SPD = 0)

- At the rising edge of the address transfer byte's first clock following setting of this bit and detection of a start condition
- When the IICCTLn1.WUP bit changes from 1 to 0
- When the IICCTLn0.IICE bit changes from 1 to 0 (operation stop)
- Reset

Condition for setting (SPD = 1)

- When a stop condition is detected

STD bit (Detection of start condition)

Condition for clearing (STD = 0)

- When a stop condition is detected
- At the rising edge of the next byte's first clock following address transfer
- Cleared by IICCTLn0.LREL = 1 (exit from communications)
- When the IICCTLn0.IICE bit changes from 1 to 0 (operation stop)
- Reset

Condition for setting (STD = 1)

- When a start condition is detected

ACKD bit (Detection of acknowledge (ACK))

Condition for clearing (ACKD = 0)

- When a stop condition is detected
- At the rising edge of the first clock of the next byte
- Cleared by IICCTLn0.LREL = 1 (exit from communications)
- When the IICCTLn0.IICE bit changes from 1 to 0 (operation stop)
- Reset

Condition for setting (ACKD = 1)

- After the SDAAn line is set to low level at the rising edge of the 9th clock of the SCLAn line

TRC bit (Detection of transmit and receive status)

Condition for clearing (TRC = 0)

<Both master and slave>

- When a stop condition is detected
- Cleared by IICCTLn0.LREL = 1 (exit from communications)
- When the IICCTLn0.IICE bit changes from 1 to 0 (operation stop)
- Cleared by IICCTLn0.WREL = 1 (release from the clock stretch state)[*1]
- When the ALD bit changes from 0 to 1 (arbitration loss)
- Reset
- When not used for communication (MSTS, EXC, COI = 0)

<Master>

- When 1 is output to the LSB of the first byte (transfer direction specification bit)

<Slave>

- When a start condition is detected
- When 0 is input to the LSB of the first byte (transfer direction specification bit)

Note 1. When the TRC bit is set to 1 (transmission state), the WREL bit of IICA control register n0 (IICCTLn0) is set to 1 during the 9th clock, the interface is released from the clock stretch state, after which the TRC bit is cleared (reception state) and the SDAAn line is set to the high impedance state. Release the interface from the clock stretch state while the TRCn bit is 1 (transmission state) by writing to the IICA shift register n.

Condition for setting (TRC = 1)

<Master>

- When a start condition is generated
- When 0 (master transmission) is output to the LSB (transfer direction specification bit) of the first byte (during address transfer)

<Slave>

- When 1 (slave transmission) is input to the LSB (transfer direction specification bit) of the first byte from the master (during address transfer)

COI bit (Detection of matching addresses)

Condition for clearing (COI = 0)

- When a start condition is detected
- When a stop condition is detected
- Cleared by IICCTLn0.LREL = 1 (exit from communications)
- When the IICCTLn0.IICE bit changes from 1 to 0 (operation stop)
- Reset

Condition for setting (COI = 1)

- When the received address matches the local address (slave address register n (SVAn)) (set at the rising edge of the eighth clock).
- When an address is received while the all address match function is enabled (IICCTLn1.SVADIS = 1) (set at the rising edge of the eighth clock).

EXC bit (Detection of extension code reception)

Condition for clearing (EXC = 0)

- When a start condition is detected
- When a stop condition is detected
- Cleared by IICCTLn0.LREL = 1 (exit from communications)
- When the IICCTLn0.IICE bit changes from 1 to 0 (operation stop)
- Reset

Condition for setting (EXC = 1)

- When the higher four bits of the received address data is either 0000b or 1111b (set at the rising edge of the eighth clock).
- When an address is received while the all address match function is enabled (IICCTLn1.SVADIS = 1) (set at the rising edge of the eighth clock).

ALD bit (Detection of arbitration loss)

Condition for clearing (ALD = 0)

- Automatically cleared after the IICSn register is read
- When the IICCTLn0.IICE bit changes from 1 to 0 (operation stop)
- Reset

Condition for setting (ALD = 1)

- When the arbitration result is a "loss".

MSTS flag (Master status check flag)

Condition for clearing (MSTS = 0)

- When a stop condition is detected
- When ALD = 1 (arbitration loss)
- Cleared by IICCTLn0.LREL = 1 (exit from communications)
- When the IICCTLn0.IICE bit changes from 1 to 0 (operation stop)
- Reset

Condition for setting (MSTS = 1)

- When a start condition is generated

24.2.5 IICFn : IICA Flag Register n (n = 0, 1)

Base address: IICA = 0x4009_3000

Offset address: 0x02 + 0x100 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	STCF	IICBSY	—	—	—	—	STCEN	IICRSV
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	IICRSV	Communication reservation function disable bit 0: Enable communication reservation 1: Disable communication reservation	R/W
1	STCEN	Initial start enable trigger 0: After operation is enabled (IICCTLn0.IICE = 1), enable generation of a start condition upon detection of a stop condition. 1: After operation is enabled (IICCTLn0.IICE = 1), enable generation of a start condition without detecting a stop condition.	R/W
5:2	—	These bits are read as 0. The write value should be 0.	R/W
6	IICBSY	I^2C bus status flag 0: Bus release status (communication initial status when STCEN = 1) 1: Bus communication status (communication initial status when STCEN = 0)	R
7	STCF	IICCTLn0.STT clear flag 0: Generate start condition 1: Start condition generation unsuccessful: clear the IICCTLn0.STT flag	R

This register sets the operation mode of I^2C and indicates the state of the I^2C bus.

The IICRSV bit can be used to enable or disable the communication reservation.

The STCEN bit can be used to set the initial value of the IICBSY bit.

The IICRSV and STCEN bits can be written only when the I^2C operation is disabled (IICCTLn0.IICE=0). The IICFn register is read-only while the I^2C operation is enabled.

The bus release status (IICBSY = 0) is recognized regardless of the actual bus status when STCEN = 1. When generating the first start condition (IICCTLn0.STT = 1), it is necessary to verify that no third-party communications are in progress in order to prevent such communications from being destroyed.

IICRSV bit (Communication reservation function disable bit)

Write to the IICRSV bit only when the operation is stopped (IICCTLn0.IICE = 0).

Condition for clearing (IICRSV = 0)

- Cleared by instruction
- Reset

Condition for setting (IICRSV = 1)

- Set by instruction

STCEN bit (Initial start enable trigger)

Write to the STCEN bit only when the operation is stopped (IICCTLn0.IICE = 0).

Condition for clearing (STCEN = 0)

- Cleared by instruction
- When a start condition is detected
- Reset

Condition for setting (STCEN = 1)

- Set by instruction

IICBSY flag (I^2C bus status flag)

Condition for clearing (IICBSY = 0)

- When a stop condition is detected
- When IICCTLn0.IICE = 0 (operation stop)
- Reset

Condition for setting (IICBSY = 1)

- When a start condition is detected
- Setting of the IICCTLn0.IICE bit when STCEN = 0

STCF flag (IICCTLn0.STT clear flag)

Condition for clearing (STCF = 0)

- Cleared by IICCTLn0.STT = 1
- When IICCTLn0.IICE = 0 (operation stop)
- Reset

Condition for setting (STCF = 1)

- Generating start condition unsuccessful and the IICCTLn0.STT bit cleared to 0 when communication reservation is disabled (IICRSV = 1).

24.2.6 IICCTLn1 : IICA Control Register n1 (n = 0, 1)

Base address: IICA = 0x4009_3000

Offset address: 0x81 + 0x100 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	WUP	SVADIS	CLD	DAD	SMC	DFC	—	PRS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PRS	IICA operation clock (f_{MCK}) 0: Selects PCLKB (1 MHz ≤ PCLKB ≤ 20 MHz) 1: Selects PCLKB/2 (20 MHz < PCLKB ≤ 32 MHz)	R/W
1	—	This bit is read as 0. The write value should be 0.	R/W
2	DFC	Digital filter operation control 0: Digital filter off 1: Digital filter on	R/W
3	SMC	Operation mode switching 0: Operates in standard mode (fastest transfer rate: 100 kbps) 1: Operates in fast mode (fastest transfer rate: 400 kbps) or fast mode plus (fastest transfer rate: 1 Mbps)	R/W
4	DAD	Detection of SDAAn pin level (valid only when IICCTLn0.IICE = 1) 0: The SDAAn pin was detected at low level 1: The SDAAn pin was detected at high level	R
5	CLD	Detection of SCLAn pin level (valid only when IICCTLn0.IICE = 1) 0: The SCLAn pin was detected at low level 1: The SCLAn pin was detected at high level	R
6	SVADIS	Address match disabling flag 0: Disables the all address match function 1: Enables the all address match function	R/W
7	WUP	Control of address match wakeup 0: Stops operation of address match wakeup function in Software Standby mode 1: Enables operation of address match wakeup function in Software Standby mode	R/W

This register is used to set the operation mode of I^2C and detect the states of the SCLAn and SDAAn pins.

Set the IICCTLn1 register, except the WUP bit, while I^2C operation is disabled (IICCTLn0.IICE = 0).

The fastest operation frequency of the IICA operation clock (f_{MCK}) is 20 MHz (max.).

Set PRS bit to 1 only when the PCLKB exceeds 20 MHz.

Setting more than 32 MHz to PCLKB is prohibited when using I^2C bus interface.

Note the minimum PCLKB operation frequency when setting the transfer clock.

The minimum PCLKB operation frequency for I^2C bus interface is determined according to the mode.

Fast mode: PCLKB = 3.5 MHz (min.)

Fast mode plus: PCLKB = 10 MHz (min.)

Normal mode: PCLKB = 1 MHz (min.)

PRS bit (IICA operation clock (f_{MCK}))

The PRS bit is used to set IICA operation clock (f_{MCK}).

DFC bit (Digital filter operation control)

Use the digital filter only in fast mode and fast mode plus.

The digital filter is used for noise elimination.

The transfer clock does not vary, regardless of the DFC bit being set to 1 or cleared to 0.

SMC bit (Operation mode switching)

The SMC bit is used for operation mode switching.

DAD bit (Detection of SDAAn pin level (valid only when IICCTLn0.IICE = 1))

Condition for clearing (DAD = 0)

- When the SDAAn pin is at low level
- When IICCTLn0.IICE = 0 (operation stop)
- Reset

Condition for setting (DAD = 1)

- When the SDAAn pin is at high level

CLD bit (Detection of SCLAn pin level (valid only when IICCTLn0.IICE = 1))

Condition for clearing (CLD = 0)

- When the SCLAn pin is at low level
- When IICCTLn0.IICE = 0 (operation stop)
- Reset

Condition for setting (CLD = 1)

- When the SCLAn pin is at high level

SVADIS bit (Address match disabling flag)

When SVADIS = 1, IICA considers any address as address match, and performs the same operation as that when an extension code is received.

Therefore, the IICSn.COI bit is set to 1, and the IICSn.EXC bit is set to 1.

For details about extension code reception, see section 24.4.11. Extension Code.

WUP flag (Control of address match wakeup)

To shift to Software Standby mode when WUP = 1, execute the WFI instruction at least 3 cycles of f_{MCK} after setting 1 to the WUP bit (see Table 24.5).

Clear the WUP bit to 0 after the address has matched, an address has been received while the all address match function is enabled, or an extension code has been received. The subsequent communication can be entered by the clearing the WUP bit to 0. The interface must be released from the clock stretch state and transmit data must be written after the WUP bit has been cleared to 0.

The interrupt timing when the address has matched, when an address has been received while the all address match function is enabled, or when an extension code has been received, while WUP = 1, is identical to the interrupt timing when WUP = 0. (A delay of the difference of sampling by the clock will occur.) Furthermore, when WUP = 1, a stop condition interrupt is not generated even if the IICCTLn0.SPIE bit is set to 1.

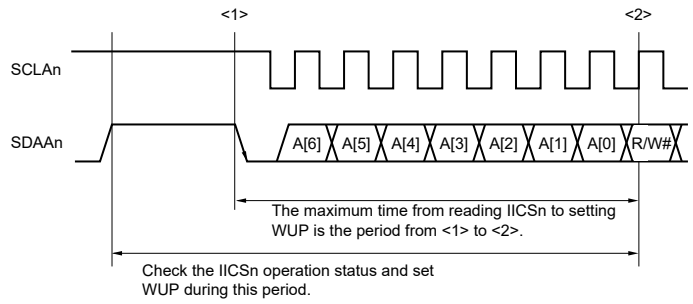
Condition for clearing (WUP = 0)

- Cleared by instruction (after address match, address reception with the all address match function enabled, or extension code reception)

Condition for setting (WUP = 1)

- Set by instruction (when the IISCn.MSTS, EXC, COI bits are 0, and the IICSn.STD bit also 0 (communication not entered))

The status of the IICA status register n (IICSn) must be checked and the WUP bit must be set during the period shown in Figure 24.3.



Note: n = 0, 1

Figure 24.3 WUP bit setting period

24.2.7 IICWLn : IICA Low-level Width Setting Register n (n = 0, 1)

Base address: IICA = 0x4009_3000

Offset address: 0x82 + 0x100 × n

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

Bit	Symbol	Function	R/W
7:0	n/a	SCLAn pin low-width configuration data of unit n	R/W

This register is used to set the low-level width (t_{LOW}) of the SCLAn pin signal that is output by I^2C bus interface and to control the SDAAn pin signal.

Set the IICWLn register while the I^2C operation is disabled (IICCTLn0.IICE = 0).

For details about setting the IICWLn register, see section 24.3.2. Setting Transfer Clock Using IICWLn and IICWHn Registers. The data hold time is one-quarter of the time set by the IICWLn register.

24.2.8 IICWHn : IICA High-level Width Setting Register n (n = 0, 1)

Base address: IICA = 0x4009_3000

Offset address: 0x83 + 0x100 × n

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

Bit	Symbol	Function	R/W
7:0	n/a	SCLAn pin high-width configuration data of unit n	R/W

This register is used to set the high-level width of the SCLAn pin signal that is output by I^2C bus interface and to control the SDAAn pin signal.

Set the IICWHn register while the I^2C operation is disabled (IICCTLn0.IICE = 0).

For setting procedures of the transfer clock on master side and of the IICWLn and IICWHn registers on slave side, see (1) Setting transfer clock on master side and (2) Setting IICWLn and IICWHn registers on slave side, respectively.

24.2.9 Registers to Control the Port Function Multiplexed with the I²C I/O Pins

For information on how to set up the I/O ports, see section 16, I/O Ports .

Set the IICCTLn0.IICE bit to 1 before setting the output mode because the SCLAn and the SDAAn pins output a low level (fixed) when the IICCTLn0.IICE bit is 0.

24.3 I²C Bus Mode Functions

24.3.1 Pin Configuration

The serial clock pin (SCLAn) and the serial data bus pin (SDAAn) are configured as follows.

1. SCLAn: This pin is used for serial clock input and output.
   This pin is an N-ch open-drain output for both master and slave devices. Input is Schmitt input.
2. SDAAn: This pin is used for serial data input and output.
   This pin is an N-ch open-drain output for both master and slave devices. Input is Schmitt input.

Since outputs from the serial clock line and the serial data bus line are N-ch open-drain outputs, an external pull-up resistor is required. Figure 24.2 shows a serial bus configuration example using the I²C bus.

24.3.2 Setting Transfer Clock Using IICWLn and IICWHn Registers

(1) Setting transfer clock on master side

$$\text{Transfer clock} = \frac{f_{MCK}}{\text{IICWLn} + \text{IICWHn} + f_{MCK}(t_R + t_F)}$$

At this time, the optimal setting values of the IICWLn and IICWHn registers are as follows. (The fractional parts of all setting values are rounded up.)

- When the fast mode
  $\text{IICWLn} = \frac{0.52}{\text{Transfer clock}} \times f_{MCK}$
  $\text{IICWHn} = \left(\frac{0.48}{\text{Transfer clock}} - t_R - t_F\right) \times f_{MCK}$
- When the normal mode
  $\text{IICWLn} = \frac{0.47}{\text{Transfer clock}} \times f_{MCK}$
  $\text{IICWHn} = \left(\frac{0.53}{\text{Transfer clock}} - t_R - t_F\right) \times f_{MCK}$
- When the fast mode plus
  $\text{IICWLn} = \frac{0.50}{\text{Transfer clock}} \times f_{MCK}$
  $\text{IICWHn} = \left(\frac{0.50}{\text{Transfer clock}} - t_R - t_F\right) \times f_{MCK}$

(2) Setting IICWLn and IICWHn registers on slave side

The fractional parts of all setting values are rounded up.

- When the fast mode
  $\text{IICWLn} = 1.3\ \mu s \times f_{MCK}$
  $\text{IICWHn} = (1.2\ \mu s - t_R - t_F) \times f_{MCK}$
- When the normal mode
  $\text{IICWLn} = 4.7\ \mu s \times f_{MCK}$
  $\text{IICWHn} = (5.3\ \mu s - t_R - t_F) \times f_{MCK}$
- When the fast mode plus
  $\text{IICWLn} = 0.50\ \mu s \times f_{MCK}$
  $\text{IICWHn} = (0.50\ \mu s - t_R - t_F) \times f_{MCK}$

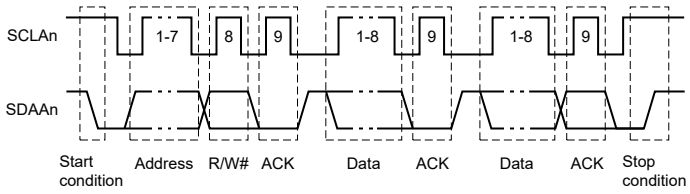
Note: Calculate the rise time (t_R) and fall time (t_F) of the SDAAn and SCLAn signals separately, because they differ depending on the pull-up resistance and wire load.

Note: IICWLn: IICA low-level width setting register n
IICWHn: IICA high-level width setting register n
t_F: SDAAn and SCLAn signal falling times
t_R: SDAAn and SCLAn signal rising times
f_{MCK}: IICA operation clock frequency

Note: n = 0, 1

24.4 I2C Bus Definitions and Control Methods

The following section describes the I2C serial data bus communication format and the signals used by the I2C bus. Figure 24.4 shows the transfer timing for the "start condition", "address", "data", and "stop condition" output using the I2C serial data bus.



Note: n = 0, 1

Figure 24.4 I2C serial data bus transfer timing

The master device generates the start condition, slave address, and stop condition.

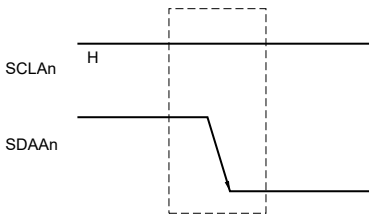
The acknowledge (ACK) can be generated by either the master or slave device (normally, it is output by the device that receives 8-bit data).

The serial clock (SCLAn) is continuously output by the master device. However, for the slave device, the period over which the SCLAn pin is at the low level can be extended and clock stretching can be inserted.

24.4.1 Start Conditions

When the SCLAn pin is at high level, changing the SDAAn pin from the high level to the low level generates a start condition.

A start condition is a signal that the master device transmits to the slave device when starting a serial transfer. When the device is used as a slave, start conditions can be detected. Figure 24.5 shows the start conditions.



Note: n = 0, 1

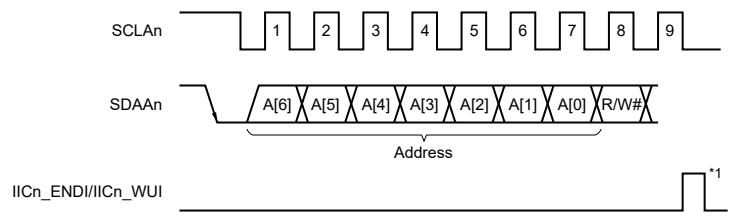
Figure 24.5 Start conditions

A start condition is output when STT bit of IICA control register n0 (IICCTLn0) is set to 1 after a stop condition has been detected (IICSn.SPD = 1). When a start condition is detected, STD bit of the IICSn register is set to 1.

24.4.2 Address

The address is defined by the 7 bits of data that follow the start condition.

An address is a 7-bit data segment that is output in order to select one of the slave devices that are connected to the master device via the bus lines. Therefore, each slave device connected via the bus lines must have a unique address. The slave devices include hardware that detects the start condition and checks whether or not the 7-bit address data matches the data values stored in the slave address register n (SVAn). If the address data matches the SVAn register values, the slave device is selected and communicates with the master device until the master device generates a start condition or stop condition. Figure 24.6 shows the address.



Note: n = 0, 1
Note 1. IICn_ENDI/IICn_WUI is not issued if data other than a local address or extension code is received while the all address match function is disabled during slave device operation.

Figure 24.6 Addresses

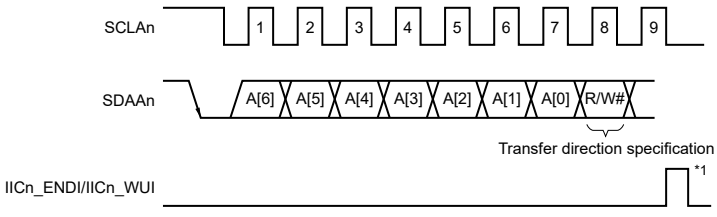
Addresses are output when a total of 8 bits consisting of the slave address and the transfer direction described in section 24.4.3. Transfer Direction Specification are written to the IICA shift register n (IICAn). The received addresses are written to the IICAn register.

The slave address is assigned to the higher 7 bits of the IICAn register.

24.4.3 Transfer Direction Specification

In addition to the 7-bit address data, the master device sends 1 bit that specifies the transfer direction.

When this transfer direction specification bit has a value of 0, it indicates that the master device is transmitting data to a slave device. When the transfer direction specification bit has a value of 1, it indicates that the master device is receiving data from a slave device. Figure 24.7 shows the transfer direction specification.



Note: n = 0, 1
Note 1. IICn_ENDI/IICn_WUI is not issued if data other than a local address or extension code is received while the all address match function is disabled during slave device operation.

Figure 24.7 Transfer direction specification

24.4.4 Acknowledge (ACK)

ACK is used to check the status of serial data at the transmission and reception sides. The reception side returns ACK each time it has received 8-bit data.

The transmission side usually receives ACK after transmitting 8-bit data. When ACK is returned from the reception side, it is assumed that reception has been correctly performed and processing is continued. Whether ACK has been detected can be checked by using ACKD bit of the IICA status register n (IICSn).

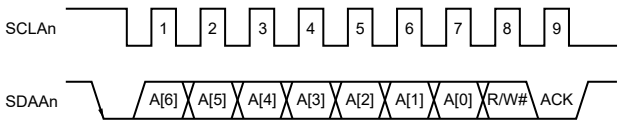
When the master receives the last data item, it does not return ACK and instead generates a stop condition. If a slave does not return ACK after receiving data, the master outputs a stop condition or restart condition and stops transmission. If ACK is not returned, the possible causes are as follows.

1. Reception was not performed normally.
2. The final data item was received.
3. The reception side specified by the address does not exist.

To generate ACK, the reception side makes the SDAAn line low at the ninth clock (indicating normal reception). Automatic generation of ACK is enabled by setting ACKE bit of IICA control register n0 (IICCTLn0) to 1. TRC bit of the IICSn register is set to the value of the eighth bit that follows 7-bit address information. Usually, set the IICCTLn0.ACKE bit to 1 for reception (IICCTLn0.TRC = 0).

If a slave can receive no more data during reception (IICCTLn0.TRC = 0) or does not require the next data item, then the slave must inform the master, by clearing the IICCTLn0.ACKE bit to 0, that it will not receive any more data.

When the master does not require the next data item during reception (IICCTLn0.TRC = 0), it must clear the IICCTLn0.ACKE bit to 0 so that ACK is not generated. In this way, the master informs a slave at the transmission side that it does not require any more data (transmission will be stopped). Figure 24.8 shows the ACK.



Note: n = 0, 1

Figure 24.8 ACK

When the local address is received, ACK is automatically generated, regardless of the value of the IICCTLn0.ACKE bit. When an address other than that of the local address is received, ACK is not generated (NACK).

When an extension code is received, or when an address is received while the all address match function is enabled, ACK is generated if the IICCTLn0.ACKE bit is set to 1 in advance.

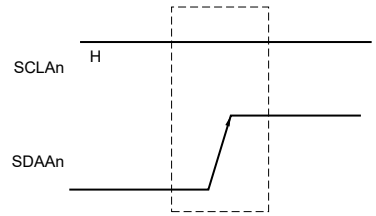
How ACK is generated when data is received depends on the setting of the timing of clock stretching as follows.

- When 8th cycle clock stretching is selected (IICCTLn0.WTIM=0):
  By setting the IICCTLn0.ACKE bit to 1 before release from the clock stretch state, ACK is generated at the falling edge of the eighth clock cycle of the SCLAn pin.
- When 9th cycle clock stretching is selected (IICCTLn0.WTIM=1):
  ACK is generated if the IICCTLn0.ACKE bit is set to 1 in advance.

24.4.5 Stop Condition

When the SCLAn pin is at high level, changing the SDAAn pin from low level to high level generates a stop condition. A stop condition is a signal that the master device generates to the slave device when serial transfer has been completed. When the device is used as a slave, stop conditions can be detected.

Figure 24.9 shows the stop conditions.



Note: n = 0, 1

Figure 24.9 Stop condition

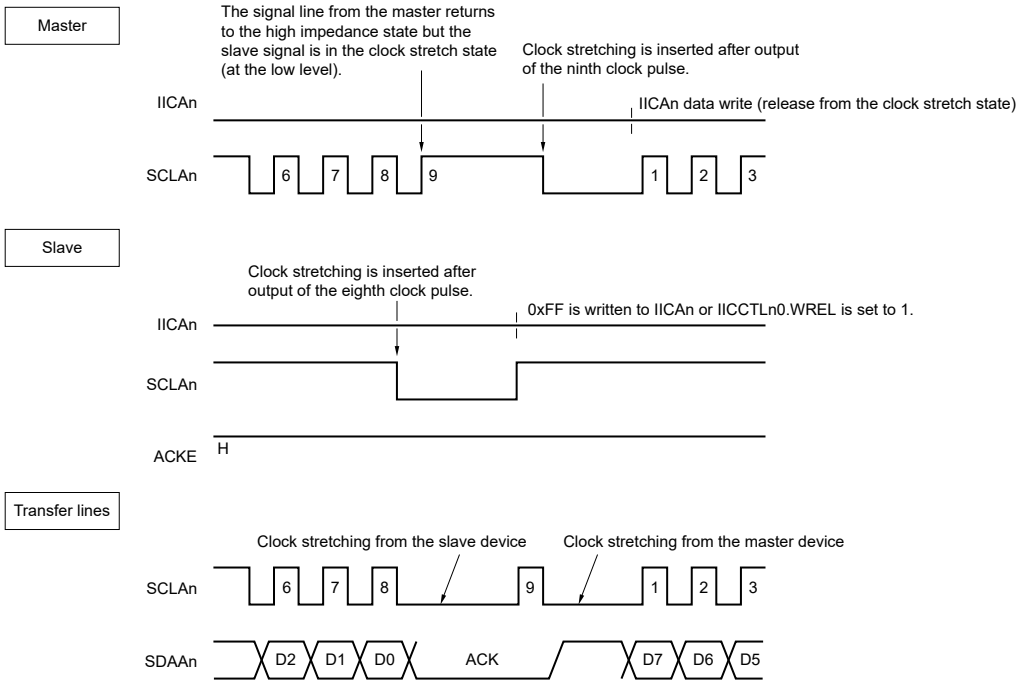
A stop condition is generated when SPT bit of IICA control register n0 (IICCTLn0) is set to 1. When the stop condition is detected, SPD bit of the IICA status register n (IICSn) is set to 1 and IICn_ENDI/IICn_WUI is generated when SPIE bit of the IICCTLn0 register is set to 1.

24.4.6 Clock Stretching

Clock stretching is used to notify the other party in communications that a device (master or slave) is preparing to transmit or receive data (for example, the interface is in the clock stretch state).

Setting the SCLAn pin to the low level indicates the clock stretch state to the other party. When clock stretching is released for both the master and slave devices, the next data transfer can start. Figure 24.10 and Figure 24.11 show the clock stretching.

(1) When clock stretching is set for the 9th and 8th clock cycles for the master and slave devices, respectively (master: transmission, slave: reception, and IICCTLn0.ACKE = 1)



Note: n = 0, 1

Figure 24.10 Clock stretching (1/2)

(2) When clock stretching is set for the 9th clock cycle for both the master and slave devices (master: transmission, slave: reception, and IICCTLn0.ACKE = 1)

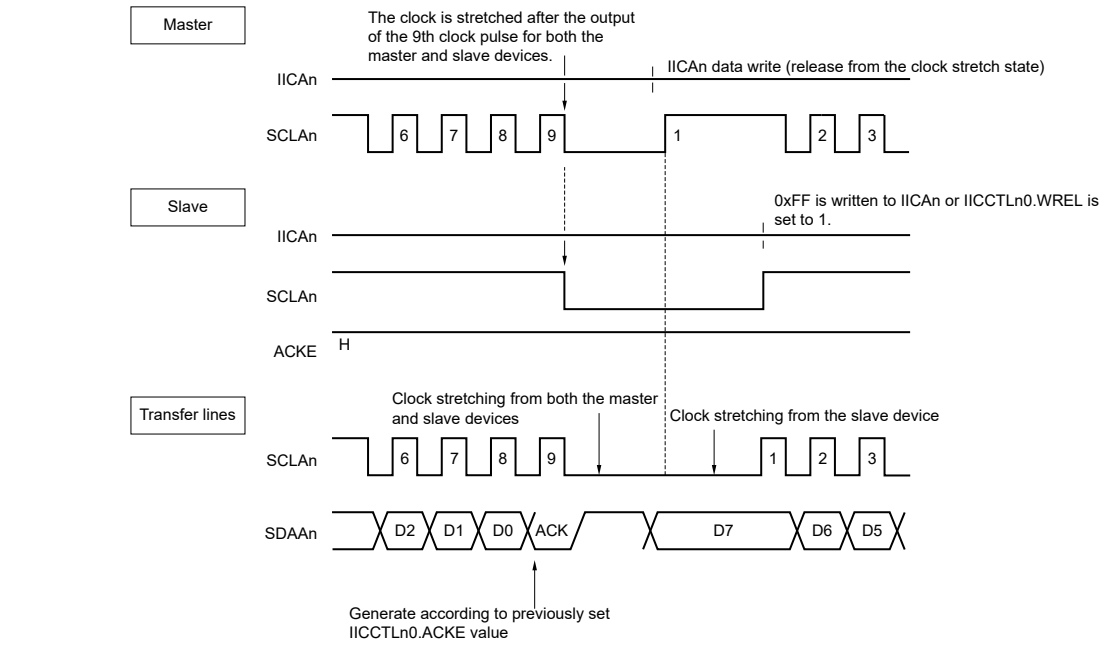


Figure 24.11 Clock stretching (2/2)

Clock stretching is automatically generated depending on the setting of WTIM bit of IICA control register n0 (IICCTLn0).

Normally, the receiving side releases the clock stretch state when WREL bit of the IICCTLn0 register is set to 1 or when 0xFF is written to the IICA shift register n (IICAn), and the transmitting side releases the clock stretch state when data is written to the IICAn register.

The master device can also release the clock stretch state via either of the following methods.

- By setting STT bit of the IICCTLn0 register to 1
- By setting SPT bit of the IICCTLn0 register to 1

24.4.7 Release from Clock Stretching

The I^2C interface usually releases the clock stretch state by the following processing.

- Writing data to the IICA shift register n (IICAn)
- Setting WREL bit of IICA control register n0 (IICCTLn0) (release from the clock stretch state)
- Setting STT bit of the IICCTLn0 register (generating start condition)*1
- Setting SPT bit of the IICCTLn0 register (generating stop condition)*1

Note 1. Master only

Executing the above processing for release from clock stretching leads to IICA releasing the clock stretch state after which communications are resumed.

To release the clock stretch state and transmit data (including addresses), write the data to the IICAn register.

To receive data after release from the clock stretch state, or to complete data transmission, set WREL bit of the IICCTLn0 register to 1.

To generate a restart condition after release from the clock stretch state, set STT bit of the IICCTLn0 register to 1.

To generate a stop condition after release from the clock stretch state, set SPT bit of the IICCTLn0 register to 1.

Execute the processing for release only once for each period in the clock stretch state.

If, for example, data is written to the IICAn register after release from the clock stretch state by setting the IICCTLn0.WREL bit to 1, an incorrect value may be output to SDAAn line because the timing for changing the SDAAn line conflicts with the timing for writing the IICAn register.

In addition to the above, communications are stopped if the IICCTLn0.IICE bit is cleared to 0 when communications have been aborted, so that the clock stretch state can be released.

If the I^2C bus has deadlocked due to noise, the device can exit from communications by setting LREL bit of the IICCTLn0 register to 1, so that the clock stretch state can be released.

If the processing for release from clock stretching is executed when IICCTLn1.WUP = 1, the clock stretch state is not released.

24.4.8 Timing of Generation of the Interrupt Request Signal (IICn_ENDI/IICn_WUI) and Control of Clock Stretching

The setting of WTIM bit of IICA control register n0 (IICCTLn0) determines the timing by which IICn_ENDI/IICn_WUI is generated and controls clock stretching, as shown in Table 24.2.

The numbers in the table indicate the pulses of the serial clock signal. Interrupt requests and control of clock stretching are both synchronized with the falling edge of these clock pulses.

Table 24.2 IICn_ENDI/IICn_WUI generation timing and control of clock stretching

IICCTLn0.WTIM	During slave device operation			During master device operation		
	Address	Data reception	Data transmission	Address	Data reception	Data transmission
0	9*1 *2	8*2	8*2	9	8	8
1	9*1 *2	9*2	9*2	9	9	9

Note 1. The IICn_ENDI/IICn_WUI signal of the slave device and clock stretching occur at the falling edge of the 9th clock cycle only when there is a match with the address set to the slave address register n (SVAn).
At this point, ACK is generated regardless of the value set to the IICCTLn0.ACKE bit. For a slave device that has received an extension code, or has received an address while the all address match function is enabled, IICn_ENDI/IICn_WUI occurs at the falling edge of the eighth clock.
However, if the address does not match after restart, IICn_ENDI/IICn_WUI is generated at the falling edge of the 9th clock cycle, but clock stretching does not occur.

Note 2. If the received address does not match the contents of the slave address register n (SVAn), the all address match function is disabled, and extension code is not received, neither IICn_ENDI/IICn_WUI nor clock stretching occurs.

1. During address transmission/reception
   - Slave device operation: The timing of the interrupt and clock stretching depends on the conditions described in Note 1 and Note 2 of Table 24.2, regardless of the setting of the IICCTLn0.WTIM bit.
   - Master device operation: The interrupt and clock stretching occur at the falling edge of the ninth clock cycle, regardless of the setting of the IICCTLn0.WTIM bit.
2. During data reception
   - All operation: The timing of the interrupt and clock stretching depends on the setting of the IICCTLn0.WTIM bit.
3. During data transmission
   - All operation: The timing of the interrupt and clock stretching depends on the setting of the IICCTLn0.WTIM bit.
4. Release from clock stretching
   The four types of processing for release from clock stretching are as follows.
   - Writing data to the IICA shift register n (IICAn)
   - Setting WREL bit of IICA control register n0 (IICCTLn0) (release from the clock stretch state)
   - Setting STT bit of the IICCTLn0 register (generating start condition)*1
   - Setting SPT bit of the IICCTLn0 register (generating stop condition)*1

   Note 1. Master only

When 8th cycle clock stretching has been selected (IICCTLn0.WTIM = 0), the presence or absence of ACK generation must be determined before release from the clock stretch state.

5. Detection of stop condition
IICn_ENDI/IICn_WUI is generated when a stop condition is detected (only when IICCTLn0.SPIE = 1).

24.4.9 Address Match Detection Method

In I^2C bus mode, the master device can select a particular slave device by transmitting the corresponding slave address.

Address match can be detected automatically by hardware. An interrupt request signal (IICn_ENDI/IICn_WUI) occurs only when the address set to the slave address register n (SVAn) matches the slave address sent by the master device, when an address is received while the all address match function is enabled (IICCTLn1.SVADIS = 1), or when an extension code has been received.

24.4.10 Error Detection

In I^2C bus mode, the status of the serial data bus (SDAAn) during data transmission is captured by the IICA shift register n (IICAn) of the transmitting device, so the IICA data prior to transmission can be compared with the transmitted IICA data to enable detection of transmission errors. A transmission error is judged as having occurred when the compared data values do not match.

24.4.11 Extension Code

1. When the higher 4 bits of the receive address are either 0000b or 1111b, the extension code reception flag (IICSn.EXC) is set to 1 for extension code reception and an interrupt request signal (IICn_ENDI/IICn_WUI) is issued at the falling edge of the eighth clock.
When an address is received while the all address match function is enabled, it is also determined that an extension code has been received.
The local address stored in the slave address register n (SVAn) is not affected.
2. The settings below are specified if 11110xx0b is transferred from the master by using a 10-bit address transfer while the SVAn register is set to 11110xx0b or if an address is received while the all address match function is enabled. Note that IICn_ENDI/IICn_WUI occurs at the falling edge of the eighth clock.
   - Higher four bits of data match or the all address match function is enabled: IICSn.EXC = 1
   - Seven bits of data match or the all address match function is enabled: IICSn.COI = 1
3. Since the processing after the interrupt request occurs differs according to the data that follows the extension code, such processing is performed by software.
If the extension code is received or an address is received with the all address match function enabled during operation as a slave, then the slave device is participating in communication even if its address does not match.
For example, after the extension code is received, if you do not wish to operate the target device as a slave device, set LREL bit of IICA control register n0 (IICCTLn0) to 1 to set the standby mode for the next communication operation.

Table 24.3 shows the bit definitions for the major extension codes.

Table 24.3 Bit definitions of major extension codes

Slave address	R/W# bit	Description
0 0 0 0 0 0 0	0	General call address
1 1 1 1 0 x x	0	10-bit slave address specification (during address authentication)
1 1 1 1 0 x x	1	10-bit slave address specification (after address match, when read command is issued)

Note: See the I^2C bus specifications issued by NXP Semiconductors for details of extension codes other than those described above.

24.4.12 Arbitration

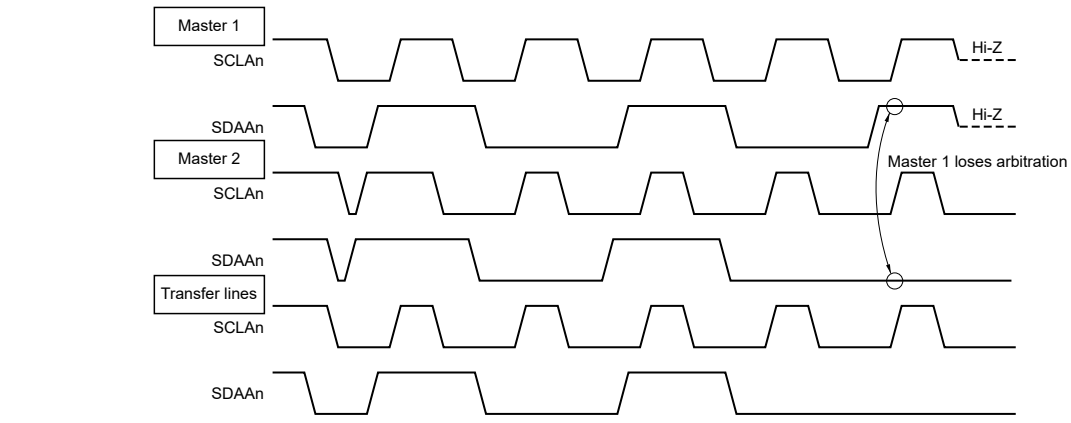
When several master devices simultaneously generate a start condition (when the IICCTLn0.STT bit is set to 1 before the IICSn.STD bit is set to 1), communication among the master devices is performed as the number of clocks are adjusted until the data differs. This kind of operation is called arbitration.

When one of the master devices loses in arbitration, an arbitration loss flag (IICSn.ALD) is set to 1 through the timing by which the arbitration loss occurred, and the SCLAn and SDAAn lines are both set to high impedance, which releases the bus.

The arbitration loss is detected by checking IICSn.ALD = 1 with software at the timing of the next interrupt request (the 8th or 9th clock cycle, when a stop condition is detected, for instance).

For details of interrupt request timing, see section 24.4.8. Timing of Generation of the Interrupt Request Signal (IICn_ENDI/IICn_WUI) and Control of Clock Stretching.

Figure 24.12 shows the arbitration timing example.



Note: n = 0, 1

Figure 24.12 Arbitration timing example

Table 24.4 shows the status during arbitration and when interrupt requests are generated.

Table 24.4 Status during arbitration and interrupt request generation timing

Status during arbitration	Interrupt request generation timing
During address transmission	At falling edge of 8th or 9th clock following byte transfer*1
Read/write data after address transmission	
During extension code transmission	
Read/write data after extension code transmission	
During data transmission	
During ACK transfer period after data transmission	
When restart condition is detected during data transfer	
When stop condition is detected during data transfer	When stop condition is generated (when IICCTLn0.SPIE = 1)*2
When data is at low level while attempting to generate a restart condition	At falling edge of eighth or ninth clock following byte transfer*1
When stop condition is detected while attempting to generate a restart condition	When stop condition is generated (when IICCTLn0.SPIE = 1)*2
When data is at low level while attempting to generate a stop condition	At falling edge of 8th or 9th clock following byte transfer*1
When SCLAn is at low level while attempting to generate a restart condition	

Note 1. When IICCTLn0.WTIM = 1, an interrupt request occurs at the falling edge of the 9th clock. When IICCTLn0.WTIM = 0, the slave address of the extension code is received, and an address is received while the all address match function is enabled, an interrupt request occurs at the falling edge of the 8th clock.

Note 2. When there is a chance that arbitration will occur, set IICCTLn0.SPIE = 1 for master device operation.

24.4.13 Wakeup Function

The I^2C bus slave function is a function that generates an interrupt request signal (IICn_ENDI/IICn_WUI) when the local address is received, an address is received while the all address match function is enabled, or an extension code is received.

This function makes processing more efficient by preventing unnecessary IICn_ENDI/IICn_WUI signal from occurring when addresses do not match while the all address match function is disabled.

When a start condition is detected, wakeup standby mode is set. Even a master that has generated a start condition enters the wakeup standby state while transmitting an address because the master may become a slave due to an arbitration loss.

To use the wakeup function in the Software Standby mode, set the IICCTLn1.WUP bit to 1. Addresses can be received regardless of the operation clock. An interrupt request signal (IICn_ENDI/IICn_WUI) is also generated when the local address is received, an address is received while the all address match function is enabled, or an extension code is received. Operation returns to normal operation by using an instruction to clear (0) the IICCTLn1.WUP bit after this interrupt has been generated.

Table 24.5 shows the step for setting IICCTLn1.WUP = 1 and Table 24.6 shows the step for setting IICCTLn1.WUP = 0 upon an address match (or when the all address match function is enabled).

Table 24.5 Step when setting IICCTLn1.WUP = 1

Step		Process	Detail
Setting IICCTLn1.WUP = 1	<1>	Start operation	—
	<2>	Status check	Wait until IICSn (IICA status register n) is in the following state: ● MSTS bit = 0 ● STD bit = 0 ● EXC bit = 0 ● COI bit = 0
	<3>	Enable operation of address match wakeup function	Set IICCTLn1.WUP bit.
	<4>	Wait	Waits for three cycles of f_{MCK}
	<5>	WFI instruction execution	—

Table 24.6 Flow when setting IICCTLn1.WUP = 0 on address match (or when the all address match function is enabled) (including extension code reception)

Step		Process	Detail
Setting IICCTLn1.WUP = 0 on address match (or when the all address match function is enabled) (including extension code reception)	<1>	Start operation	Software Standby mode state
	<2>	Interrupt check	Wait until IICn_ENDI/IICn_WUI = 1
	<3>	Disable operation of address match wakeup function	Clear IICCTLn1.WUP bit.
	<4>	Wait	Waits for five cycles of f_{MCK}
	<5>	Reading IICSn	Executes processing corresponding to the operation to be executed after checking the operation state of I^2C bus interface.
	<6>	Executes next processing	—

Use the following flows to perform the processing to release the Software Standby mode other than by an interrupt request signal (IICn_ENDI/IICn_WUI) generated from I^2C bus interface.

- When operating next IIC communication as master: Flow shown in Table 24.7.
- When operating next IIC communication as slave:
  When released by IICn_ENDI/IICn_WUI interrupt: Same as the flow in Table 24.6.
  When released by other than IICn_ENDI/IICn_WUI interrupt: Wait for IICn_ENDI/IICn_WUI interrupt with IICCTLn1.WUP left set to 1.

Table 24.7 When operating as master device after releasing Software Standby mode other than by IICn_ENDI/ IICn_WUI

Step		Process	Detail
When operating as master device after releasing Software Standby mode other than by IICn_ENDI/ IICn_WUI	<1>	Start operation	—
	<2>	Enable generation of interrupt request	Set IICCTLn0.SPIE bit.
	<3>	Enable operation of address match wakeup function	Set IICCTLn1.WUP bit.
	<4>	Wait	Waits for 3 cycles of f_{MCK}
	<5>	WFI instruction execution	Software standby state
	<6>	Releasing Software Standby mode	Releases Software Standby mode by an interrupt other than IICn_ENDI/IICn_WUI
	<7>	Disable operation of address match wakeup function	Clear IICCTLn1.WUP bit.
	<8>	Interrupt check	Wait until IICn_ENDI/IICn_WUI = 1
	<9>	Reading IICSn	Executes processing corresponding to the operation to be executed after checking the operation state of I^2C bus interface.
	<10>	Executes next processing	—

24.4.14 Communication Reservation

(1) When communication reservation function is enabled (IICFn.IICRSV = 0)

To start master device communications when not currently using a bus, a communication reservation can be made to enable transmission of a start condition when the bus is released. There are two modes under which the bus is not used.

- When arbitration results in neither master nor slave operation
- While the all address match function is disabled, when an extension code is received and slave operation is disabled (ACK is not returned and the bus was released by setting LREL bit of IICA control register n0 (IICCTLn0) to 1 and exiting from communication)

If STT bit of the IICCTLn0 register is set to 1 while the bus is not used, a start condition is automatically generated and wait state is entered after the bus is released (after a stop condition is detected).

If an address is written to the IICA shift register n (IICAn) after SPIE bit of the IICCTLn0 register was set to 1, and it was detected by generation of an interrupt request signal (IICn_ENDI/IICn_WUI) that the bus was released (detection of the stop condition), then the device automatically starts communication as the master. Data written to the IICAn register before the stop condition is detected is invalid.

When the IICCTLn0.STT bit has been set to 1, the operation mode (as start condition or as communication reservation) is determined according to the bus status.

- If the bus has been released …… a start condition is generated
- If the bus has not been released (standby mode) … communication reservation

Check whether the communication reservation operates or not using the IICSn.MSTS bit after the IICCTLn0.STT bit is set to 1 and the wait time elapses.

Use software to secure the wait time calculated by the following expression.

Wait time from setting IICCTLn0.STT = 1 to checking the IICSn.MSTS flag:

(IICWLn setting value + IICWHn setting value + 4) / f_{MCK} + t_F × 2

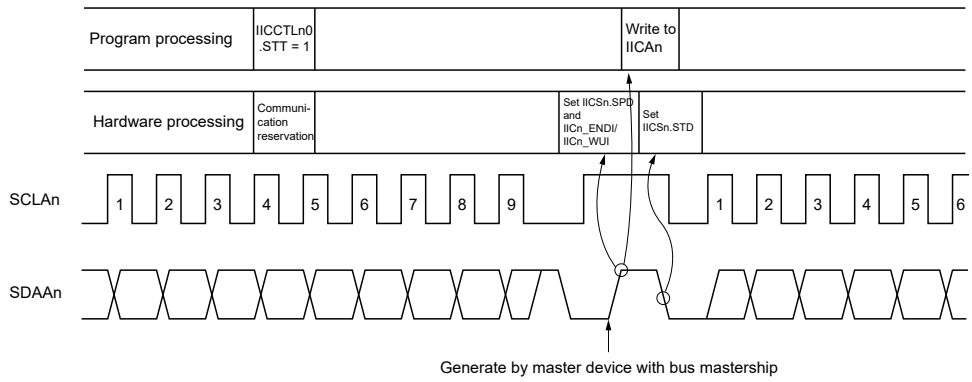
Note: IICWLn: IICA low-level width setting register n

IICWHn: IICA high-level width setting register n

t_F: SDAAn and SCLAn signal falling times

f_{MCK}: IICA operation clock frequency

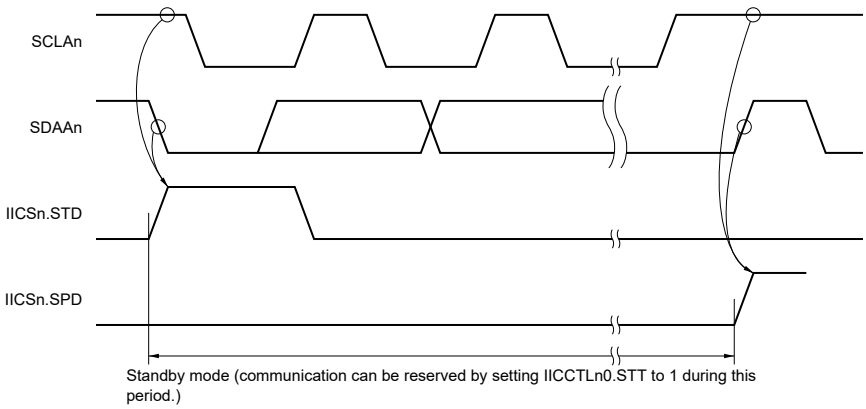
Figure 24.13 shows the communication reservation timing.



Note: n = 0, 1

Figure 24.13 Communication reservation timing

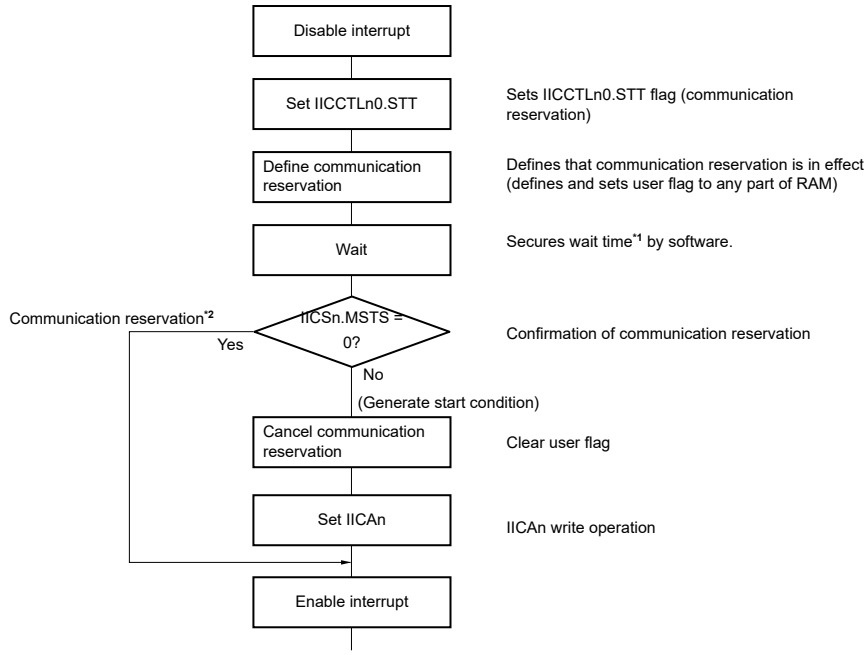
Communication reservations are accepted with the timing shown in Figure 24.14. After STD bit of the IICA status register n (IICSn) is set to 1, a communication reservation can be made by setting STT bit of IICA control register n0 (IICCTLn0) to 1 before a stop condition is detected.



Note: n = 0, 1

Figure 24.14 Timing for accepting communication reservations

Figure 24.15 shows the communication reservation protocol.



Note: n = 0, 1
Note 1. The wait time is calculated as follows.
(IICWLn setting value + IICWHn setting value + 4) / f_{MCK} + t_F × 2
t_F: SDAAn and SCLAn signal falling times
f_{MCK}: IICA operation clock frequency
Note 2. The communication reservation operation executes a write to the IICA shift register n (IICAn) when a stop condition interrupt request occurs.

Figure 24.15 Communication reservation protocol

(2) When communication reservation function is disabled (IICFn.IICRSV = 1)

When STT bit of IICA control register n0 (IICCTLn0) is set to 1 when the bus is not used in a communication during bus communication, this request is rejected and a start condition is not generated. The following two statuses are included in the status where bus is not used.

- When arbitration results in neither master nor slave operation
- While the all address match function is disabled, when an extension code is received and slave operation is disabled (ACK is not returned and the bus was released by setting LREL bit of the IICCTLn0 register to 1 and exiting from communication)

To confirm whether the start condition was generated or request was rejected, check IICFn.STCF bit. It takes up to 5 cycles of f_{MCK} until the IICFn.STCF bit is set to 1 after setting IICCTLn0.STT = 1. Therefore, secure the time by software.

24.4.15 Usage Notes

1. When IICFn.STCEN = 0
   Immediately after I^2C operation is enabled (IICCTLn0.IICE = 1), the bus communication status (IICFn.IICBSY = 1) is recognized regardless of the actual bus status. When changing from a mode in which no stop condition has been detected to a master device communication mode, first generate a stop condition to release the bus, then perform master device communication.
   When using multiple masters, it is not possible to perform master device communication when the bus has not been released (when a stop condition has not been detected).
   Use the following sequence for generating a stop condition.
   <1> Set IICA control register n1 (IICCTLn1).
   <2> Set IICE bit of IICA control register n0 (IICCTLn0) to 1.
   <3> Set SPT bit of the IICCTLn0 register to 1.

2. When IICFn.STCEN = 1
   Immediately after I^2C operation is enabled (IICCTLn0.IICE = 1), the bus released status (IICFn.IICBSY = 0) is recognized regardless of the actual bus status. To generate the first start condition (IICCTLn0.STT = 1), it is necessary to confirm that the bus has been released, so as to not disturb other communications.
3. If other I^2C communications are already in progress
   If I^2C operation is enabled and the device participates in communication already in progress when the SDAAn pin is low and the SCLAn pin is high, the IICA recognizes that the SDAAn pin has gone low (detects a start condition). If the value on the bus at this time can be recognized as an extension code or the all address match function is enabled, ACK is returned, but this interferes with other I^2C communications. To avoid this, start the IICA in the following sequence.
   <1> Clear SPIE bit of the IICCTLn0 register to 0 to disable generation of an interrupt request signal (IICn_ENDI/IICn_WUI) when the stop condition is detected.
   <2> Set IICE bit of the IICCTLn0 register to 1 to enable the operation of the IICA.
   <3> Wait for detection of the start condition.
   <4> Set LREL bit of the IICCTLn0 register to 1 before ACK is returned (4 to 72 cycles of f_{MCK} after setting the IICCTLn0.IICE bit to 1), to forcibly disable detection.
4. Setting the IICCTLn0.STT and IICCTLn0.SPT bits again after they are set and before they are cleared to 0 is prohibited.
5. When transmission is reserved, set the IICCTLn0.SPIE bit to 1 so that an interrupt request is generated when the stop condition is detected. Transfer is started when communication data is written to the IICA shift register n (IICAn) after the interrupt request is generated. Unless the interrupt is generated when the stop condition is detected, the device stops in the wait state because the interrupt request is not generated when communication is started. However, it is not necessary to set the IICCTLn0.SPIE bit to 1 when the IICSn.MSTS bit is detected by software.

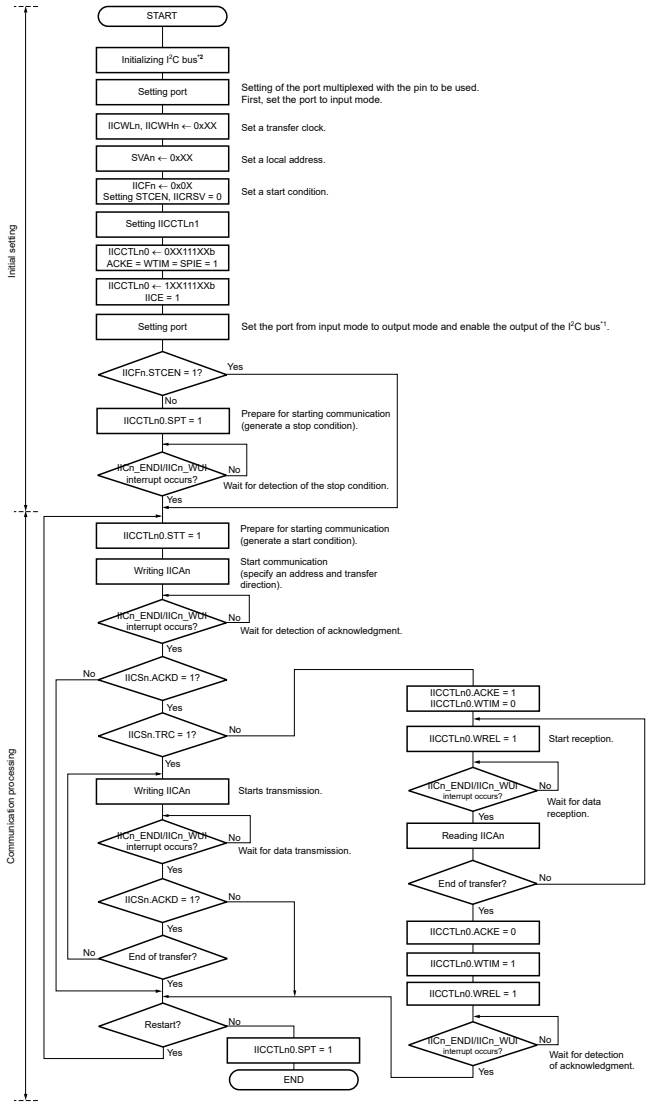
24.4.16 Communication Operations

The following shows three operation procedures with the flowchart.

1. Master operation in single-master system
   The flowchart when using this product as the master in a single master system is shown in Figure 24.16.
   This flowchart is broadly divided into the initial settings and communication processing. Execute the initial settings at startup. If communication with the slave is required, prepare the communication and then execute communication processing.
2. Master operation in multi-master system
   In the I^2C bus multi-master system, whether the bus is released or used cannot be judged by the I^2C bus specifications when a device takes part in a communication. Here, when data and clock are at a high level for a certain period (1 frame), this product takes part in a communication with bus released state.
   This flowchart is broadly divided into the initial settings, communication waiting, and communication processing. The processing when this product loses in arbitration and is specified as the slave is omitted in Figure 24.17, and only the processing as the master is shown. Execute the initial settings at startup to take part in a communication. Then, wait for the communication request as the master or wait for the specification as the slave. The actual communication is performed in the communication processing, and it supports the transmission and reception with the slave and the arbitration with other masters.
3. Slave operation
   An example of when this product is used as the I^2C bus slave is shown in Figure 24.21 and Figure 24.22.
   When used as the slave, operation is started by an interrupt. Execute the initial settings at startup, then wait for the IICn_ENDI/IICn_WUI interrupt occurrence (communication waiting). When an IICn_ENDI/IICn_WUI interrupt occurs, the communication status is judged and its result is passed as a flag over to the main processing.
   By checking the flags, necessary communication processing is performed.

(1) Master operation in single-master system

Conform to the specifications of the product that is communicating, with respect to the transmission and reception formats.



Note: n = 0, 1

Note 1. See section 24.2.9. Registers to Control the Port Function Multiplexed with the I2C I/O Pins.

Note 2. Release (SCLAn and SDAAn pins = high level) the I2C bus in conformance with the specifications of the product that is communicating. If EEPROM is outputting a low level to the SDAAn pin, for example, set the SCLAn pin in the output port mode, and output a clock pulse from the output port until the SDAAn pin is constantly at high level.

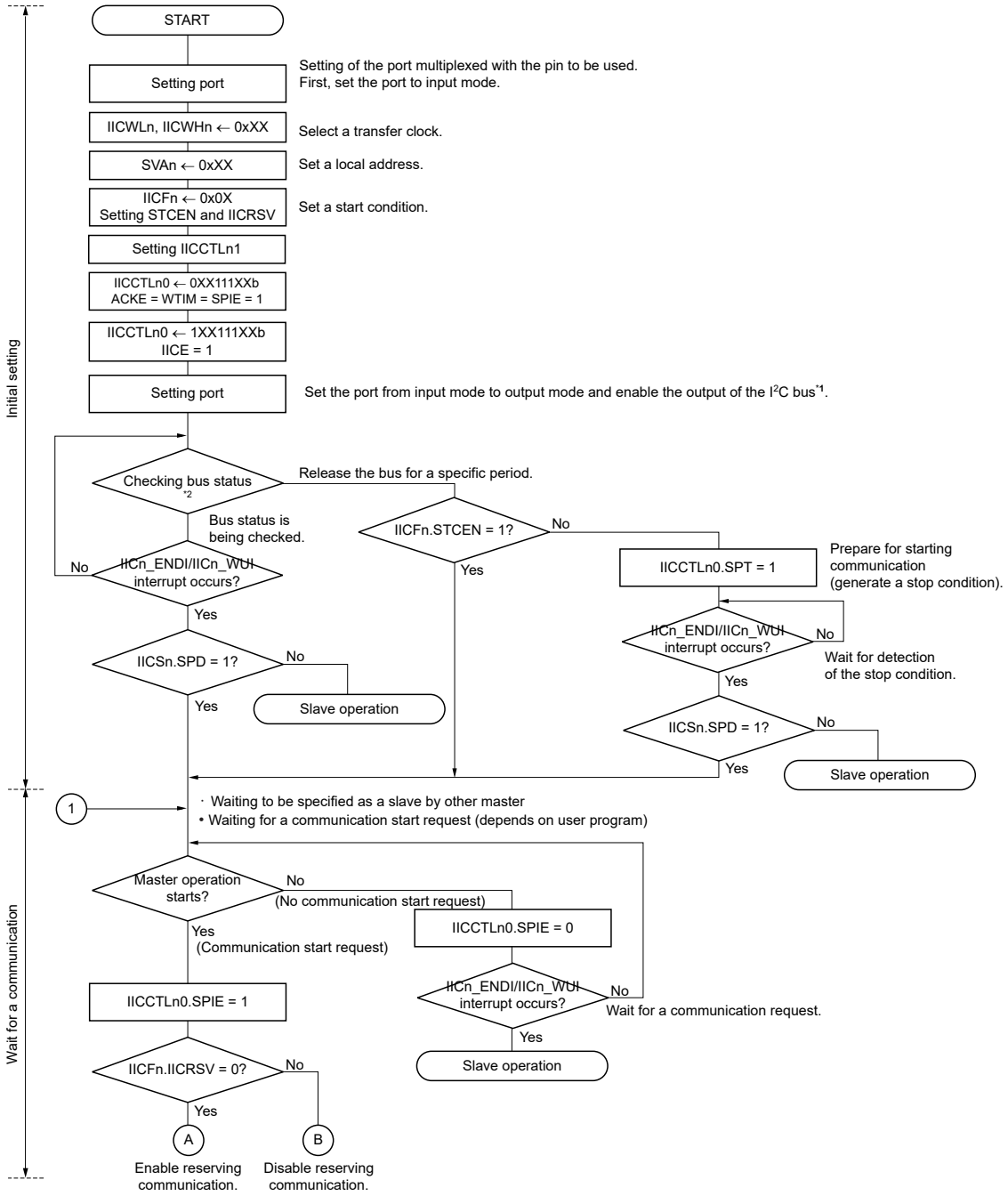
Figure 24.16 Master operation in single-master system

(2) Master operation in multi-master system

Conform to the specifications of the product that is communicating, with respect to the transmission and reception formats.

To use the device as a master in a multi-master system, read the IICSn.MSTS bit each time interrupt IICn_ENDI/IICn_WUI occurred to check the arbitration result.

To use the device as a slave in a multi-master system, check the status by using the IICA status register n (IICSn) and IICA flag register n (IICFn) each time interrupt IICn_ENDI/IICn_WUI occurred, and determine the processing to be performed next.

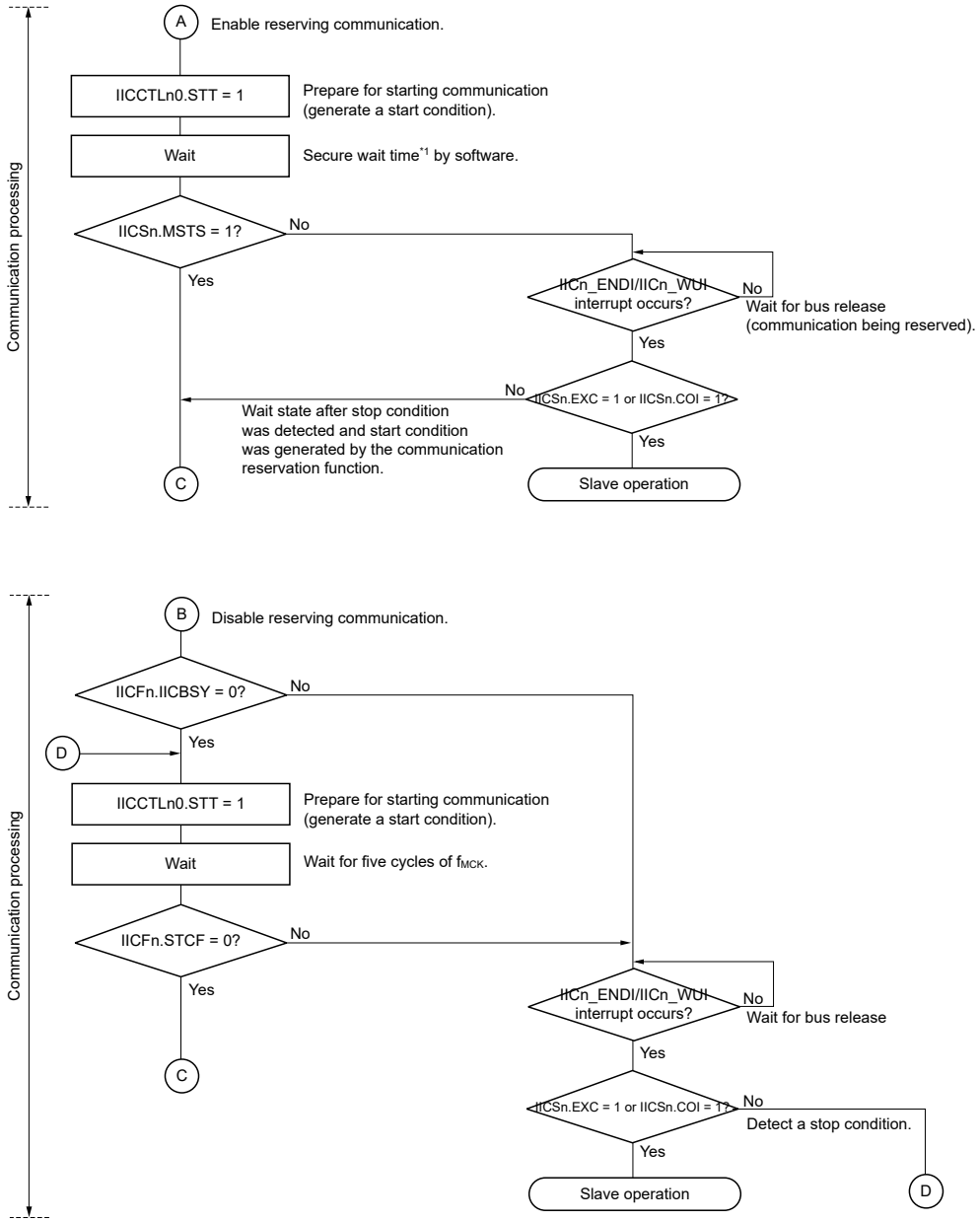


Note: n = 0, 1

Note 1. section 24.2.9. Registers to Control the Port Function Multiplexed with the I2C I/O Pins.

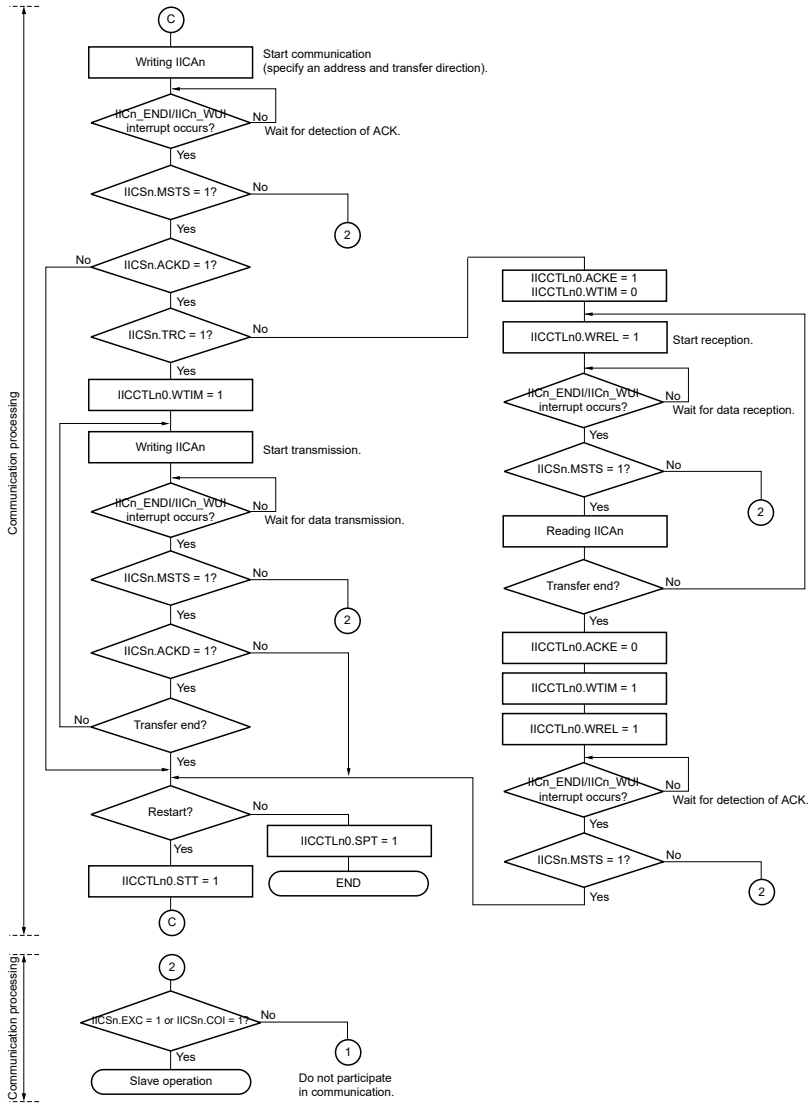
Note 2. Confirm that the bus is released (IICCTLn1.CLD = 1, IICCTLn1.DAD = 1) for a specific period (for example, for a period of one frame). If the SDAAn pin is constantly at low level, decide whether to release the I2C bus (SCLAn and SDAAn pins = high level) in conformance with the specifications of the product that is communicating.

Figure 24.17 Master operation in multi-master system (1/3)



Note: n = 0, 1
Note: t_F: SDAAn and SCLAn signal falling times
f_{MCK}: IICA operation clock frequency
Note 1. The wait time is calculated as follows.
(IICWLn setting value + IICWHn setting value + 4) / f_{MCK} + t_F × 2

Figure 24.18 Master operation in multi-master system (2/3)



Note: n = 0, 1

Figure 24.19 Master operation in multi-master system (3/3)

(3) Slave operation

The processing procedure of the slave operation is as follows.

Basically, the slave operation is event-driven. Therefore, processing by the IICn_ENDI/IICn_WUI interrupt (processing that must substantially change the operation status such as detection of a stop condition during communication) is necessary.

In the following explanation, it is assumed that the all address match function is disabled and the extension code is not supported for data communication. It is also assumed that the IICn_ENDI/IICn_WUI interrupt servicing only performs status transition processing, and that actual data communication is performed by the main processing.

Figure 24.20 shows an interface configuration with the main processor in slave operation.

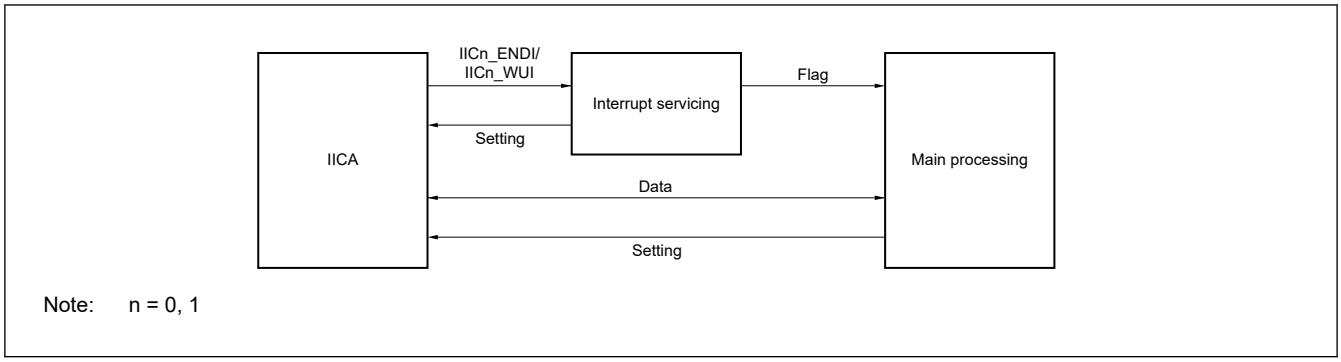


Figure 24.20 Interface configuration with the main processor in slave operation

Therefore, data communication processing is performed by preparing the following three flags and passing them to the main processing instead of IICn_ENDI/IICn_WUI.

<1> Communication mode flag

This flag indicates the following two communication statuses.

- Clear mode: Status in which data communication is not performed
- Communication mode: Status in which data communication is performed (from valid address detection to stop condition detection, no detection of ACK from master, address mismatch)

<2> Ready flag

This flag indicates that data communication is enabled. Its function is the same as the IICn_ENDI/IICn_WUI interrupt for ordinary data communication. This flag is set by interrupt servicing and cleared by the main processing. Clear this flag by interrupt servicing when communication is started. However, the ready flag is not set by interrupt servicing when the first data is transmitted. Therefore, the first data is transmitted without the flag being cleared (an address match is interpreted as a request for the next data).

<3> Communication direction flag

This flag indicates the direction of communication. Its value is the same as the TRC bit.

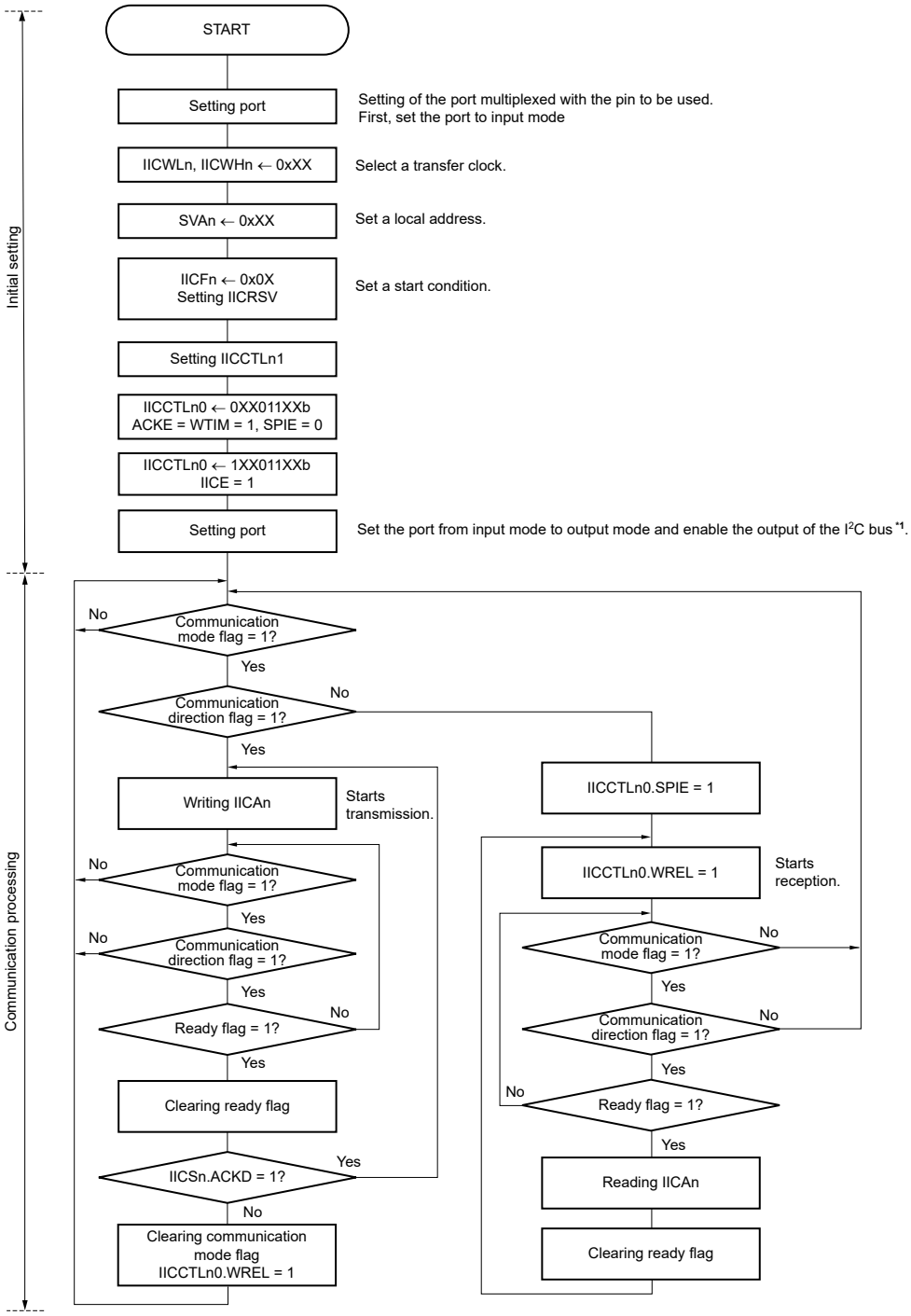
The main processing of the slave operation is explained next.

Start I^2C bus interface and wait until communication is enabled. When communication is enabled, execute communication by using the communication mode flag and ready flag (processing of the stop condition and start condition is performed by an interrupt. Here, check the status by using the flags).

The transmission operation is repeated until the master no longer returns ACK. If ACK is not returned from the master, communication is completed.

For reception, the necessary amount of data is received. When communication is completed, ACK is not returned as the next data. After that, the master generates a stop condition or restart condition. Exit from the communication status occurs in this way.

Conform to the specifications of the product that is communicating, with respect to the transmission and reception formats.



Note: n = 0, 1
Note 1. See section 24.2.9. Registers to Control the Port Function Multiplexed with the I2C I/O Pins.

Figure 24.21 Slave operation flowchart (1)

An example of the processing procedure of the slave with the IICn_ENDI/IICn_WUI interrupt is explained below (processing is performed assuming that the all address match function is disabled and no extension code is used). The IICn_ENDI/IICn_WUI interrupt checks the status, and the following operations are performed.

<1> Communication is stopped if the stop condition is issued.

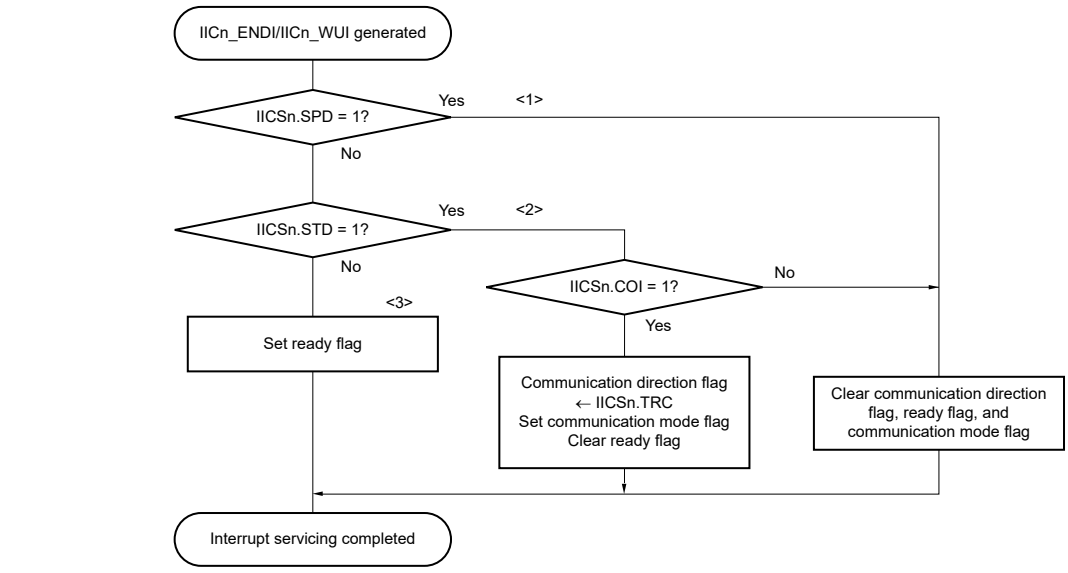
<2> If the start condition is issued, the address is checked and communication is completed if the address does not match.

If the address matches, the communication mode is set, wait is canceled, and processing returns from the interrupt (the ready flag is cleared).

<3> For data transmit/receive, only the ready flag is set. Processing returns from the interrupt with the I^2C bus remaining in the wait state.

Note: <1> to <3> above correspond to <1> to <3> in Figure 24.22.

Figure 24.22 shows the interrupt flowchart for slave operation.



Note: n = 0, 1

Figure 24.22 Slave operation flowchart (2)

24.4.17 Timing of I^2C Interrupt Request Signal (IICn_ENDI/IICn_WUI) Occurrence

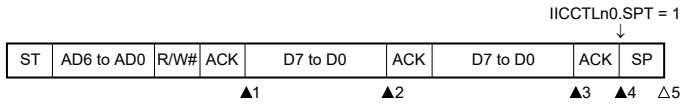
The timing of transmitting or receiving data and generation of interrupt request signal IICn_ENDI/IICn_WUI, and the value of the IICA status register n (IICSn) when the IICn_ENDI/IICn_WUI signal is generated are shown in Figure 24.23 to Figure 24.62.

Note: ST: Start condition
AD6 to AD0: Address bits
R/W#: Transfer direction specification bit
ACK: Acknowledge bit
D7 to D0: Data bits
SP: Stop condition
n = 0, 1

(1) Master device operation

(a) Start ~ Address ~ Data ~ Data ~ Stop (reception/transmission)

1. When IICCTLn0.WTIM = 0

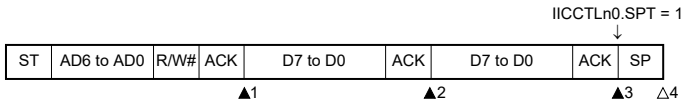


▲1: IICSn = 1000×110b
▲2: IICSn = 1000×000b
▲3: IICSn = 1000×000b (set the IICCTLn0.WTIM bit to 1)*1
▲4: IICSn = 1000××00b (set the IICCTLn0.SPT bit to 1)
△5: IICSn = 00000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
×: Don't care
Note 1. To generate a stop condition, set the IICCTLn0.WTIM bit to 1 and change the timing for generating the IICn_ENDI/IICn_WUI interrupt request signal.

Figure 24.23 Master device operation reception/transmission (IICCTLn0.WTIM = 0)

2. When IICCTLn0.WTIM = 1



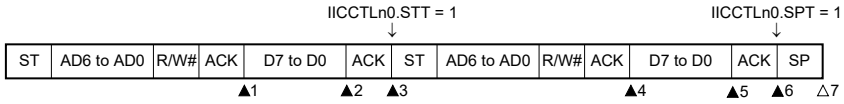
▲1: IICSn = 1000×110b
▲2: IICSn = 1000×100b
▲3: IICSn = 1000××00b (set the IICCTLn0.SPT bit to 1)
△4: IICSn = 00000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
×: Don't care

Figure 24.24 Master device operation reception/transmission (IICCTLn0.WTIM = 1)

(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop (restart)

1. When IICCTLn0.WTIM = 0

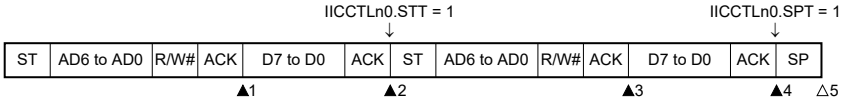


▲1: IICSn = 1000×110b
▲2: IICSn = 1000×000b (set the IICCTLn0.WTIM bit to 1)*1
▲3: IICSn = 1000××00b (clear the IICCTLn0.WTIM bit to 0*2, set the IICCTLn0.STT bit to 1)
▲4: IICSn = 1000×110b
▲5: IICSn = 1000×000b (set the IICCTLn0.WTIM bit to 1)*3
▲6: IICSn = 1000××00b (set the IICCTLn0.SPT bit to 1)
△7: IICSn = 00000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
×: Don't care
Note 1. To generate a start condition, set the IICCTLn0.WTIM bit to 1 and change the timing for generating the IICn_ENDI/IICn_WUI interrupt request signal.
Note 2. Clear the IICCTLn0.WTIM bit to 0 to restore the original setting.
Note 3. To generate a stop condition, set the IICCTLn0.WTIM bit to 1 and change the timing for generating the IICn_ENDI/IICn_WUI interrupt request signal.

Figure 24.25 Master device operation restart (IICCTLn0.WTIM = 0)

2. When IICCTLn0.WTIM = 1



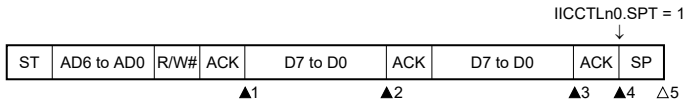
▲1: IICSn = 1000×110b
▲2: IICSn = 1000××00b (set the IICCTLn0.STT bit to 1)
▲3: IICSn = 1000×110b
▲4: IICSn = 1000××00b (set the IICCTLn0.SPT bit to 1)
△5: IICSn = 00000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
×: Don't care

Figure 24.26 Master device operation restart (IICCTLn0.WTIM = 1)

(c) Start ~ Code ~ Data ~ Data ~ Stop (extension code transmission)

1. When IICCTLn0.WTIM = 0

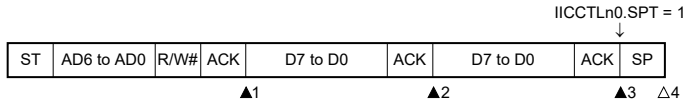


▲1: IICSn = 1010×110b
▲2: IICSn = 1010×000b
▲3: IICSn = 1010×000b (set the IICCTLn0.WTIM bit to 1)*1
▲4: IICSn = 1010××00b (set the IICCTLn0.SPT bit to 1)
△5: IICSn = 00000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
×: Don't care
Note 1. To generate a stop condition, set the IICCTLn0.WTIM bit to 1 and change the timing for generating the IICn_ENDI/IICn_WUI interrupt request signal.

Figure 24.27 Master device operation extension code transmission (IICCTLn0.WTIM = 0)

2. When IICCTLn0.WTIM = 1



▲1: IICSn = 1010×110b
▲2: IICSn = 1010×100b
▲3: IICSn = 1010××00b (set the IICCTLn0.SPT bit to 1)
△4: IICSn = 00000001b

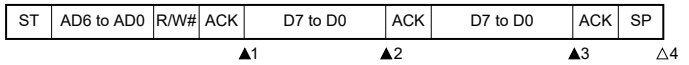
Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
×: Don't care

Figure 24.28 Master device operation extension code transmission (IICCTLn0.WTIM = 1)

(2) Slave device operation (slave address data reception)

(a) Start ~ Address ~ Data ~ Data ~ Stop

1. When IICCTLn0.WTIM = 0

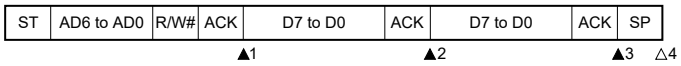


▲1: IICSn = 0001×110b
▲2: IICSn = 0001×000b
▲3: IICSn = 0001×000b
△4: IICSn = 00000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
×: Don't care

Figure 24.29 Slave device operation slave address data reception (IICCTLn0.WTIM = 0)

2. When IICCTLn0.WTIM = 1



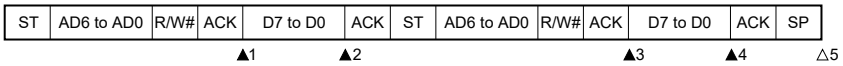
▲1: IICSn = 0001×110b
▲2: IICSn = 0001×100b
▲3: IICSn = 0001××00b
△4: IICSn = 00000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
×: Don't care

Figure 24.30 Slave device operation slave address data reception (IICCTLn0.WTIM = 1)

(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

1. When IICCTLn0.WTIM = 0 (after restart, matching SVAn, the all address match function is disabled)



▲1: IICSn = 0001×110b
▲2: IICSn = 0001×000b
▲3: IICSn = 0001×110b
▲4: IICSn = 0001×000b
△5: IICSn = 00000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
×: Don't care

Figure 24.31 Slave device operation after normal access and matching SVAn (IICCTLn0.WTIM = 0)

2. When IICCTLn0.WTIM = 1 (after restart matching SVAn, the all address match function is disabled)

▲1: IICSn = 0001×110b
▲2: IICSn = 0001××00b
▲3: IICSn = 0001×110b
▲4: IICSn = 0001××00b
△5: IICSn = 00000001b

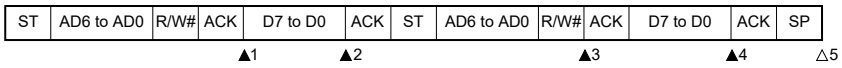
Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
×: Don't care

Figure 24.32 Slave device operation after normal access and matching SVAn (IICCTLn0.WTIM = 1)

(c) Start ~ Address ~ Data ~ Start ~ Code ~ Data ~ Stop

1. When IICCTLn0.WTIM = 0

(after restart address does not match (= extension code, the all address match function is disabled))



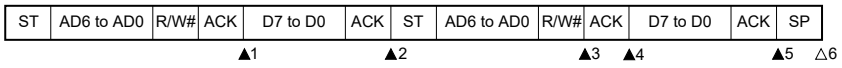
▲1: IICSn = 0001×110b
▲2: IICSn = 0001×000b
▲3: IICSn = 0010×010b
▲4: IICSn = 0010×000b
△5: IICSn = 00000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
×: Don't care

Figure 24.33 Slave device operation after normal access and matching the extension code (IICCTLn0.WTIM = 0)

2. When IICCTLn0.WTIM = 1

(after restart address does not match (= extension code, the all address match function is disabled))



▲1: IICSn = 0001×110b
▲2: IICSn = 0001××00b
▲3: IICSn = 0010×010b
▲4: IICSn = 0010×110b
▲5: IICSn = 0010××00b
△6: IICSn = 00000001b

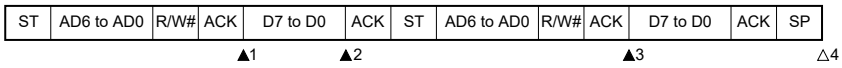
Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
×: Don't care

Figure 24.34 Slave device operation after normal access and matching the extension code (IICCTLn0.WTIM = 1)

(d) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

1. When IICCTLn0.WTIM = 0

(after restart address does not match (= not extension code, the all address match function is disabled))



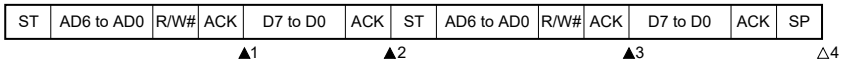
▲1: IICSn = 0001×110b
▲2: IICSn = 0001×000b
▲3: IICSn = 00000×10b
△4: IICSn = 00000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
×: Don't care

Figure 24.35 Slave device operation after normal access and mismatch (IICCTLn0.WTIM = 0)

2. When IICCTLn0.WTIM = 1

(after restart address does not match (= not extension code, the all address match function is disabled))



▲1: IICSn = 0001×110b
▲2: IICSn = 0001××00b
▲3: IICSn = 00000×10b
△4: IICSn = 00000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
×: Don't care

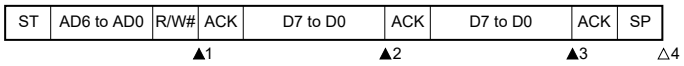
Figure 24.36 Slave device operation after normal access and mismatch (IICCTLn0.WTIM = 1)

(3) Slave device operation (when receiving extension code and the all address match function is disabled)

The device is always participating in communication when it receives an extension code.

(a) Start ~ Code ~ Data ~ Data ~ Stop

1. When IICCTLn0.WTIM = 0

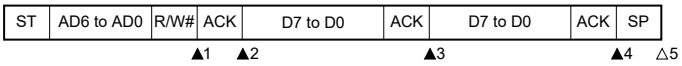


▲1: IICSn = 0010×010b
▲2: IICSn = 0010×000b
▲3: IICSn = 0010×000b
△4: IICSn = 00000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
×: Don't care

Figure 24.37 Slave device operation receiving extension code (IICCTLn0.WTIM = 0)

2. When IICCTLn0.WTIM = 1



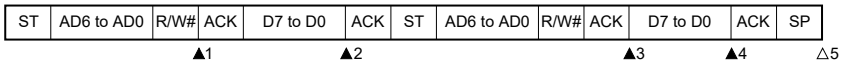
▲1: IICSn = 0010×010b
▲2: IICSn = 0010×110b
▲3: IICSn = 0010×100b
▲4: IICSn = 0010××00b
△5: IICSn = 00000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
×: Don't care

Figure 24.38 Slave device operation receiving extension code (IICCTLn0.WTIM = 1)

(b) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

1. When IICCTLn0.WTIM = 0 (after restart matching SVAn, the all address match function is disabled)

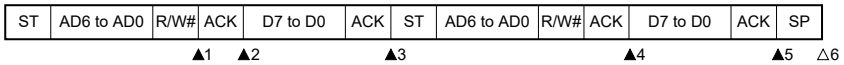


▲1: IICSn = 0010×010b
▲2: IICSn = 0010×000b
▲3: IICSn = 0001×110b
▲4: IICSn = 0001×000b
△5: IICSn = 00000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
×: Don't care

Figure 24.39 Slave device operation after code access and matching SVAn (IICCTLn0.WTIM = 0)

2. When IICCTLn0.WTIM = 1 (after restart matching SVAn, the all address match function is disabled)



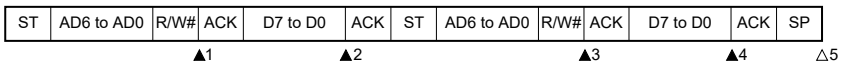
▲1: IICSn = 0010×010b
▲2: IICSn = 0010×110b
▲3: IICSn = 0010××00b
▲4: IICSn = 0001×110b
▲5: IICSn = 0001××00b
△6: IICSn = 00000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
×: Don't care

Figure 24.40 Slave device operation after code access and matching SVAn (IICCTLn0.WTIM = 1)

(c) Start ~ Code ~ Data ~ Start ~ Code ~ Data ~ Stop

1. When IICCTLn0.WTIM = 0 (after restart, extension code is received, the all address match function is disabled)



▲1: IICSn = 0010×010b
▲2: IICSn = 0010×000b
▲3: IICSn = 0010×010b
▲4: IICSn = 0010×000b
△5: IICSn = 00000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
×: Don't care

Figure 24.41 Slave device operation after code access and matching the extension code (IICCTLn0.WTIM = 0)

When IICCTLn0.WTIM = 1 (after restart, extension code is received, the all address match function is disabled)

▲1: IICSn = 0010×010b
▲2: IICSn = 0010×110b
▲3: IICSn = 0010××00b
▲4: IICSn = 0010×010b
▲5: IICSn = 0010×110b
▲6: IICSn = 0010××00b
△7: IICSn = 00000001b

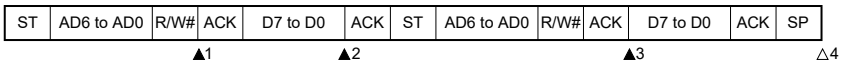
Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
×: Don't care

Figure 24.42 Slave device operation after code access and matching the extension code (IICCTLn0.WTIM = 1)

(d) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

1. When IICCTLn0.WTIM = 0

(after restart address does not match (= not extension code, the all address match function is disabled))



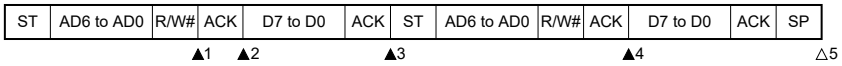
▲1: IICSn = 0010×010b
▲2: IICSn = 0010×000b
▲3: IICSn = 00000×10b
△4: IICSn = 00000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
×: Don't care

Figure 24.43 Slave device operation after code access and mismatch (IICCTLn0.WTIM = 0)

2. When IICCTLn0.WTIM = 1

(after restart address does not match (= not extension code, the all address match function is disabled))



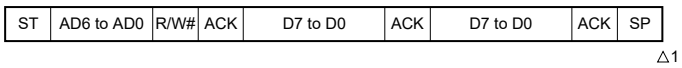
▲1: IICSn = 0010×010b
▲2: IICSn = 0010×110b
▲3: IICSn = 0010××00b
▲4: IICSn = 00000×10b
△5: IICSn = 00000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
×: Don't care

Figure 24.44 Slave device operation after code access and mismatch (IICCTLn0.WTIM = 1)

(4) Operation without communication

(a) Start ~ Code ~ Data ~ Data ~ Stop



△1: IICSn = 00000001b

Note: n = 0, 1
Note: △: Generated only when IICCTLn0.SPIE = 1

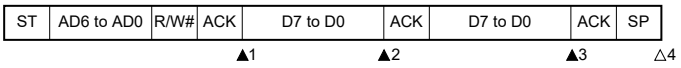
Figure 24.45 Operation without communication

(5) Arbitration loss operation (operation as slave after arbitration loss)

When the device is used as a master in a multi-master system, read the IICSn.MSTS bit each time interrupt request signal IICn_ENDI/IICn_WUI has occurred to check the arbitration result.

(a) When arbitration loss occurs during transmission of slave address data

1. When IICCTLn0.WTIM = 0

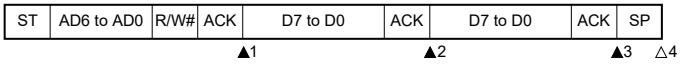


▲1: IICSn = 0101×110b
▲2: IICSn = 0001×000b
▲3: IICSn = 0001×000b
△4: IICSn = 00000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
×: Don't care

Figure 24.46 Arbitration loss when sending slave address data (IICCTLn0.WTIM = 0)

2. When IICCTLn0.WTIM = 1



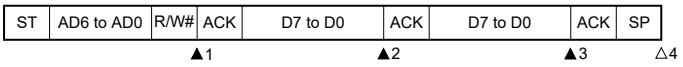
▲1: IICSn = 0101×110b
▲2: IICSn = 0001×100b
▲3: IICSn = 0001××00b
△4: IICSn = 00000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
×: Don't care

Figure 24.47 Arbitration loss when sending slave address data (IICCTLn0.WTIM = 1)

(b) When arbitration loss occurs during transmission of extension code (the all address match function is disabled)

1. When IICCTLn0.WTIM = 0

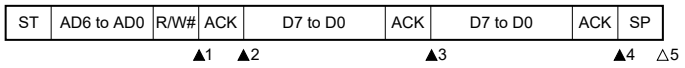


▲1: IICSn = 0110×010b
▲2: IICSn = 0010×000b
▲3: IICSn = 0010×000b
△4: IICSn = 00000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
×: Don't care

Figure 24.48 Arbitration loss when sending extension code (IICCTLn0.WTIM = 0)

2. When IICCTLn0.WTIM = 1



▲1: IICSn = 0110×010b
▲2: IICSn = 0010×110b
▲3: IICSn = 0010×100b
▲4: IICSn = 0010××00b
△5: IICSn = 00000001b

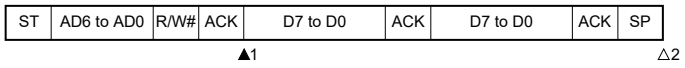
Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
×: Don't care

Figure 24.49 Arbitration loss when sending extension code (IICCTLn0.WTIM = 1)

(6) Operation when arbitration loss occurs (no communication after arbitration loss)

When the device is used as a master in a multi-master system, read the IICSn.MSTS bit each time interrupt request signal IICn_ENDI/IICn_WUI has occurred to check the arbitration result.

(a) When arbitration loss occurs during transmission of slave address data (when IICCTLn0.WTIM = 1)

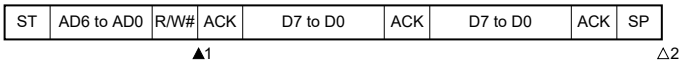


▲1: IICSn = 01000110b
△2: IICSn = 00000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1

Figure 24.50 Operation when arbitration loss occurs during slave address data transfer (IICCTLn0.WTIM = 1)

(b) When arbitration loss occurs during transmission of extension code (the all address match function is disabled)



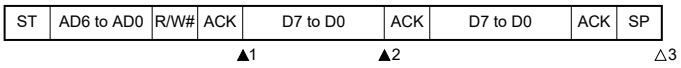
▲1: IICSn = 0110×010b
Sets IICCTLn0.LREL = 1 by software
△2: IICSn = 00000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
×: Don't care

Figure 24.51 Operation when arbitration loss occurs during extension code transfer

(c) When arbitration loss occurs during transmission of data

1. When IICCTLn0.WTIM = 0

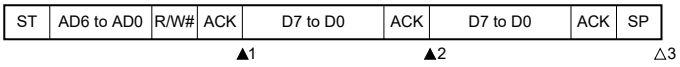


▲1: IICSn = 10001110b
▲2: IICSn = 01000000b
△3: IICSn = 00000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1

Figure 24.52 Operation when arbitration loss occurs during data transfer (IICCTLn0.WTIM = 0)

2. When IICCTLn0.WTIM = 1



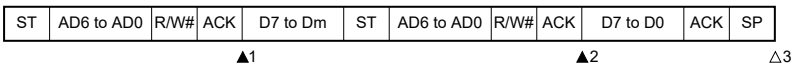
▲1: IICSn = 10001110b
▲2: IICSn = 01000100b
△3: IICSn = 00000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1

Figure 24.53 Operation when arbitration loss occurs during data transfer (IICCTLn0.WTIM = 1)

(d) When loss occurs due to restart condition during data transfer

1. Not extension code (example: unmatches with SVAn, the all address match function is disabled)

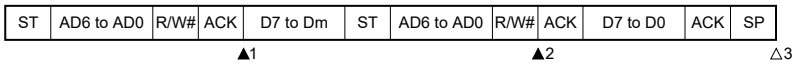


▲1: IICSn = 1000×110b
▲2: IICSn = 01000110b
△3: IICSn = 00000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
×: Don't care
m = 6 to 0

Figure 24.54 Operation when arbitration loss occurs due to restart during data transfer (not extension code)

2. Extension code (the all address match function is disabled)

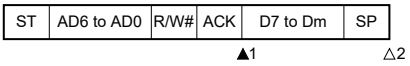


▲1: IICSn = 1000×110b
▲2: IICSn = 01100010b
Sets IICCTLn0.LREL = 1 by software
△3: IICSn = 00000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
×: Don't care
m = 6 to 0

Figure 24.55 Operation when arbitration loss occurs due to restart during data transfer (extension code)

(e) When loss occurs due to stop condition during data transfer



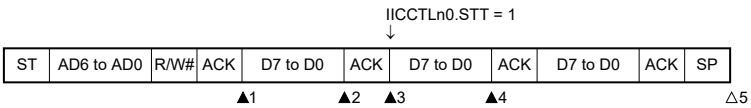
▲1: IICSn = 10000110b
△2: IICSn = 01000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
×: Don't care
m = 6 to 0

Figure 24.56 Operation when arbitration loss occurs due to stop condition during data transfer

(f) When arbitration loss occurs due to low-level data when attempting to generate a restart condition

1. When IICCTLn0.WTIM = 0

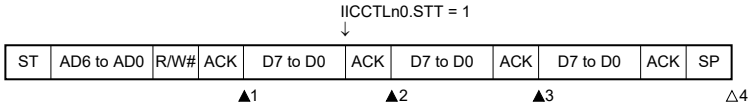


▲1: IICSn = 1000×110b
▲2: IICSn = 1000×000b (set the IICCTLn0.WTIM bit to 1)
▲3: IICSn = 1000×100b (clear the IICCTLn0.WTIM bit to 0)
▲4: IICSn = 01000000b
△5: IICSn = 00000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
×: Don't care

Figure 24.57 Operations when arbitration loss occurs due to low-level data when attempting to generate a restart condition (IICCTLn0.WTIM = 0)

2. When IICCTLn0.WTIM = 1



▲1: IICSn = 1000×110b
▲2: IICSn = 1000×100b (set the IICCTLn0.STT bit to 1)
▲3: IICSn = 01000100b
△4: IICSn = 00000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
×: Don't care

Figure 24.58 Operations when arbitration loss occurs due to low-level data when attempting to generate a restart condition (IICCTLn0.WTIM = 1)

(g) When arbitration loss occurs due to a stop condition when attempting to generate a restart condition

1. When IICCTLn0.WTIM = 0

▲1: IICSn = 1000×110b
▲2: IICSn = 1000×000b (set the IICCTLn0.WTIM bit to 1)
▲3: IICSn = 1000××00b (set the IICCTLn0.STT bit to 1)
△4: IICSn = 01000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
×: Don't care

Figure 24.59 Operations when arbitration loss occurs due to stop condition when attempting to generate a restart condition (IICCTLn0.WTIM = 0)

2. When IICCTLn0.WTIM = 1

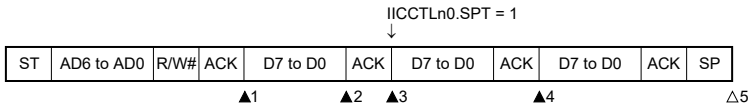
▲1: IICSn = 1000×110b
▲2: IICSn = 1000××00b (set the IICCTLn0.STT bit to 1)
△3: IICSn = 01000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
×: Don't care

Figure 24.60 Operations when arbitration loss occurs due to stop condition when attempting to generate a restart condition (IICCTLn0.WTIM = 1)

(h) When arbitration loss occurs due to low-level data when attempting to generate a stop condition

1. When IICCTLn0.WTIM = 0

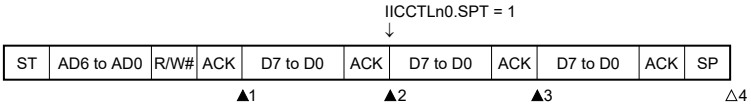


▲1: IICSn = 1000×110b
▲2: IICSn = 1000×000b (Set the IICCTLn0.WTIM bit to 1)
▲3: IICSn = 1000×100b (Clear the IICCTLn0.WTIM bit to 0)
▲4: IICSn = 01000100b
△5: IICSn = 00000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
×: Don't care

Figure 24.61 Operations when arbitration loss occurs due to low-Level data when attempting to generate a stop condition (IICCTLn0.WTIM = 0)

2. When IICCTLn0.WTIM = 1



▲1: IICSn = 1000×110b
▲2: IICSn = 1000×100b (set the IICCTLn0.SPT bit to 1)
▲3: IICSn = 01000100b
△4: IICSn = 00000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
×: Don't care

Figure 24.62 Operations when arbitration loss occurs due to low-Level data when attempting to generate a stop condition (IICCTLn0.WTIM = 1)

24.5 Timing Charts

When using the I^2C bus mode, the master device outputs an address through the serial bus to select one of several slave devices as its communication partner.

After outputting the slave address, the master device transmits the IICSn.TRC bit, which specifies the data transfer direction, and then starts serial communication with the slave device.

(1) Example of Master to Slave Communications (9th Cycle Clock Stretching Is Selected for Both the Master and Slave) and (2) Example of Slave to Master Communications (8th Cycle Clock Stretching Is Selected for the Master and 9th Cycle Clock Stretching Is Selected for the Slave) show timing charts of the data communication.

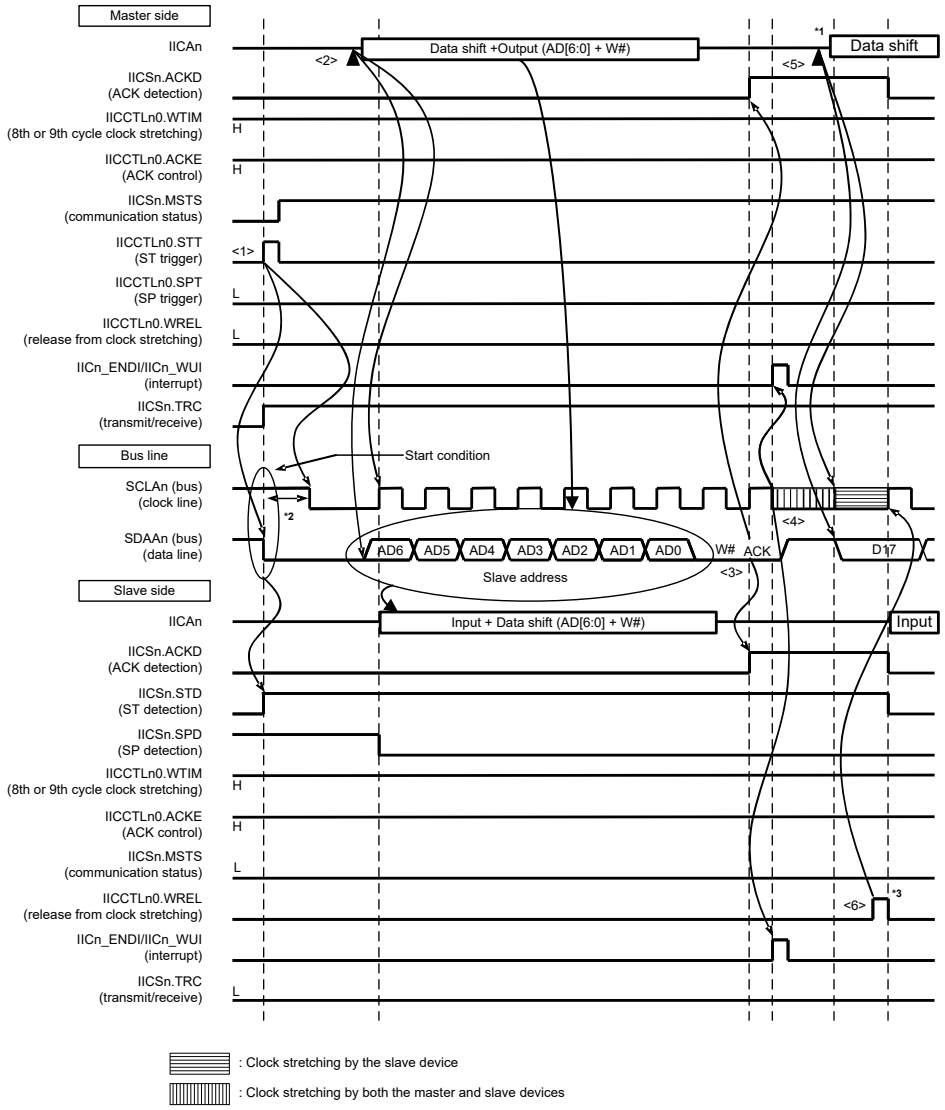
The shift operation of the IICA shift register n (IICAn) is synchronized with the falling edge of the serial clock (SCLAn). The transmit data is transferred to the SO latch and is output (MSB first) using the SDAAn pin.

Data input through the SDAAn pin is captured into IICAn at the rising edge of SCLAn.

In the timing diagrams described in this section, it is assumed that the all address match function is disabled.

(1) Example of Master to Slave Communications (9th Cycle Clock Stretching Is Selected for Both the Master and Slave)

1. Start condition → address → data



Note: n = 0, 1

Note 1. For release from the clock stretch state during transmission by a master device, write data to the IICAn register instead of setting the IICCTLn0.WREL bit.

Note 2. Make sure that the time between the fall of the SDAAn pin signal and the fall of the SCLAn pin signal is at least 4.0 µs when specifying standard mode and at least 0.6 µs when specifying fast mode.

Note 3. For release from the clock stretch state during reception by a slave device, write 0xFF to IICAn or set the IICCTLn0.WREL bit.

Figure 24.63 Example of master to slave communications (9th cycle clock stretching is selected for both the master and slave) (1/4)

The meanings of <1> to <6> in Figure 24.63 are explained below.

<1> The start condition trigger is set by the master device (IICCTLn0.STT = 1) and a start condition (SCLAn = 1 and SDAAn changes from 1 to 0) is generated once the bus data line goes low (SDAAn).

When the start condition is subsequently detected, the master device enters the master device communication status (IICSn.MSTS = 1). The master device is ready to communicate once the bus clock line goes low (SCLAn = 0) after the hold time has elapsed.

<2> The master device writes the address + W (transmission) to the IICA shift register n (IICAn) and transmits the slave address.

<3> In the slave device if the address received matches the address (SVAn value) of a slave device, that slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (IICSn.ACKD = 1) at the rising edge of the 9th clock.

<4> The master device issues an interrupt (IICn_ENDI/IICn_WUI: end of address transmission) at the falling edge of the 9th clock. The slave device with the address matching the transmitted slave address sets the clock stretch state (SCLAn = 0) and issues an interrupt (IICn_ENDI/IICn_WUI: address match).

<5> The master device writes the data to transmit to the IICAn register and releases the clock stretch state set by the master device.

<6> If the slave device releases the clock stretch state (IICCTLn0.WREL = 1), the master device starts transferring data to the slave device.

If the transmitted address does not match the address of the slave device, the slave device does not return an ACK to the master device (NACK: SDAAn = 1). The slave device also does not issue the IICn_ENDI/IICn_WUI interrupt (address match) and does not set the clock stretch state.

The master device, however, issues the IICn_ENDI/IICn_WUI interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

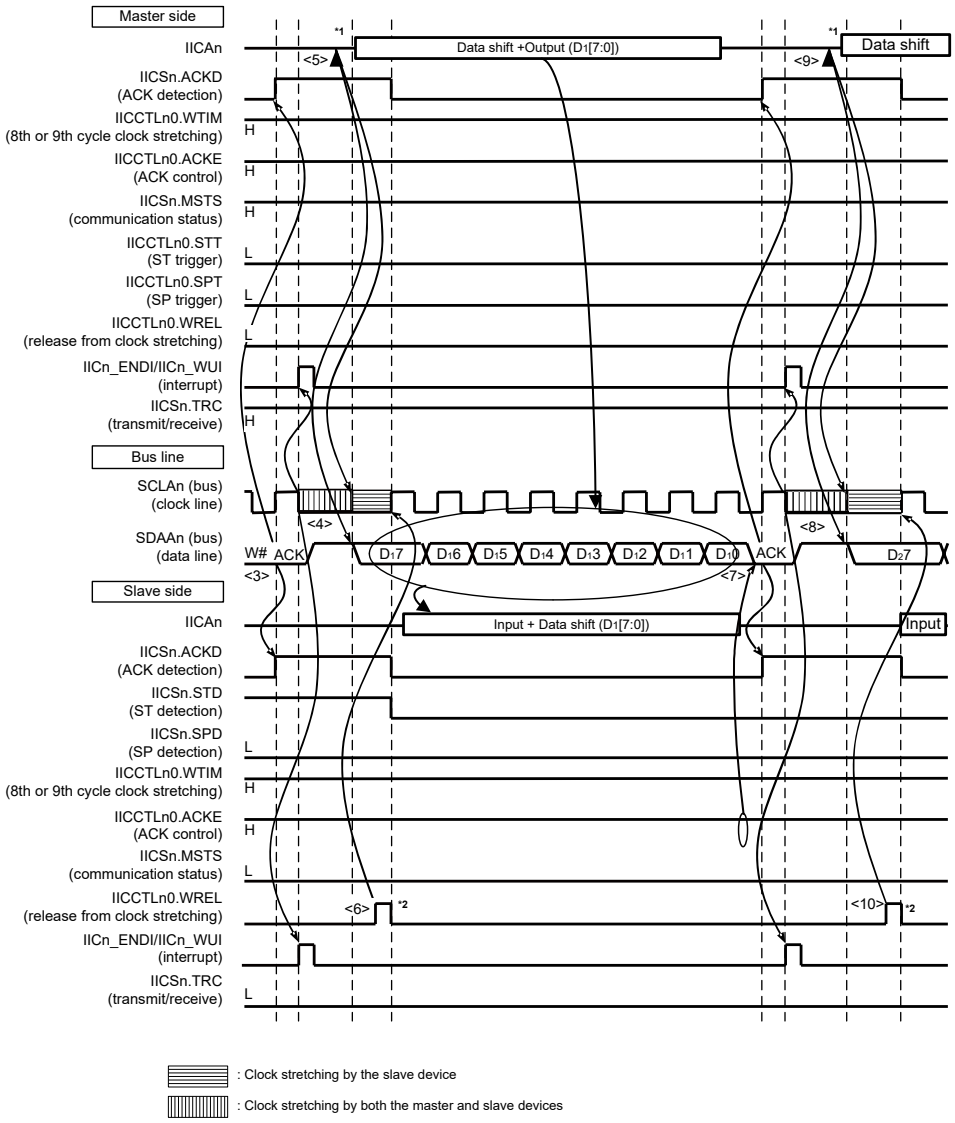
Note: <1> to <15> in (1) Example of Master to Slave Communications (9th Cycle Clock Stretching Is Selected for Both the Master and Slave) represent the entire procedure for communicating data using the I^2C bus.

Figure 24.63 shows the processing from <1> to <6>,

Figure 24.64 shows the processing from <3> to <10>, and

Figure 24.65 shows the processing from <7> to <15>.

2. Address → data → data



Note: n = 0, 1

Note 1. For release from the clock stretch state during transmission by a master device, write data to the IICAn register instead of setting the IICCTLn0.WREL bit.

Note 2. For release from the clock stretch state during reception by a slave device, write 0xFF to IICAn or set the IICCTLn0.WREL bit.

Figure 24.64 Example of master to slave communications (9th cycle clock stretching is selected for both the master and slave) (2/4)

The meanings of <3> to <10> in Figure 24.64 are explained below.

<3> In the slave device if the address received matches the address (SVAn value) of a slave device, that slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (IICSn.ACKD = 1) at the rising edge of the 9th clock.

<4> The master device issues an interrupt (IICn_ENDI/IICn_WUI: end of address transmission) at the falling edge of the 9th clock. The slave device with the address matching the transmitted slave address sets the clock stretch state (SCLAn = 0) and issues an interrupt (IICn_ENDI/IICn_WUI: address match).

<5> The master device writes the data to transmit to the IICA shift register n (IICAn) and releases the clock stretch state set by the master device.

<6> If the slave device releases the clock stretch state (IICCTLn0.WREL = 1), the master device starts transferring data to the slave device.

<7> After data transfer is completed, because of IICCTLn0.ACKE = 1, the slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (IICSn.ACKD = 1) at the rising edge of the 9th clock.

<8> The master device and slave device set the clock stretch state (SCLAn = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (IICn_ENDI/IICn_WUI: end of transfer).

<9> The master device writes the data to transmit to the IICAn register and releases the clock stretch state set by the master device.

<10>The slave device reads the received data and releases the clock stretch state (IICCTLn0.WREL = 1). The master device then starts transferring data to the slave device.

If the transmitted address does not match the address of the slave device, the slave device does not return an ACK to the master device (NACK: SDAAn = 1). The slave device also does not issue the IICn_ENDI/IICn_WUI interrupt (address match) and does not set the clock stretch state.

The master device, however, issues the IICn_ENDI/IICn_WUI interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

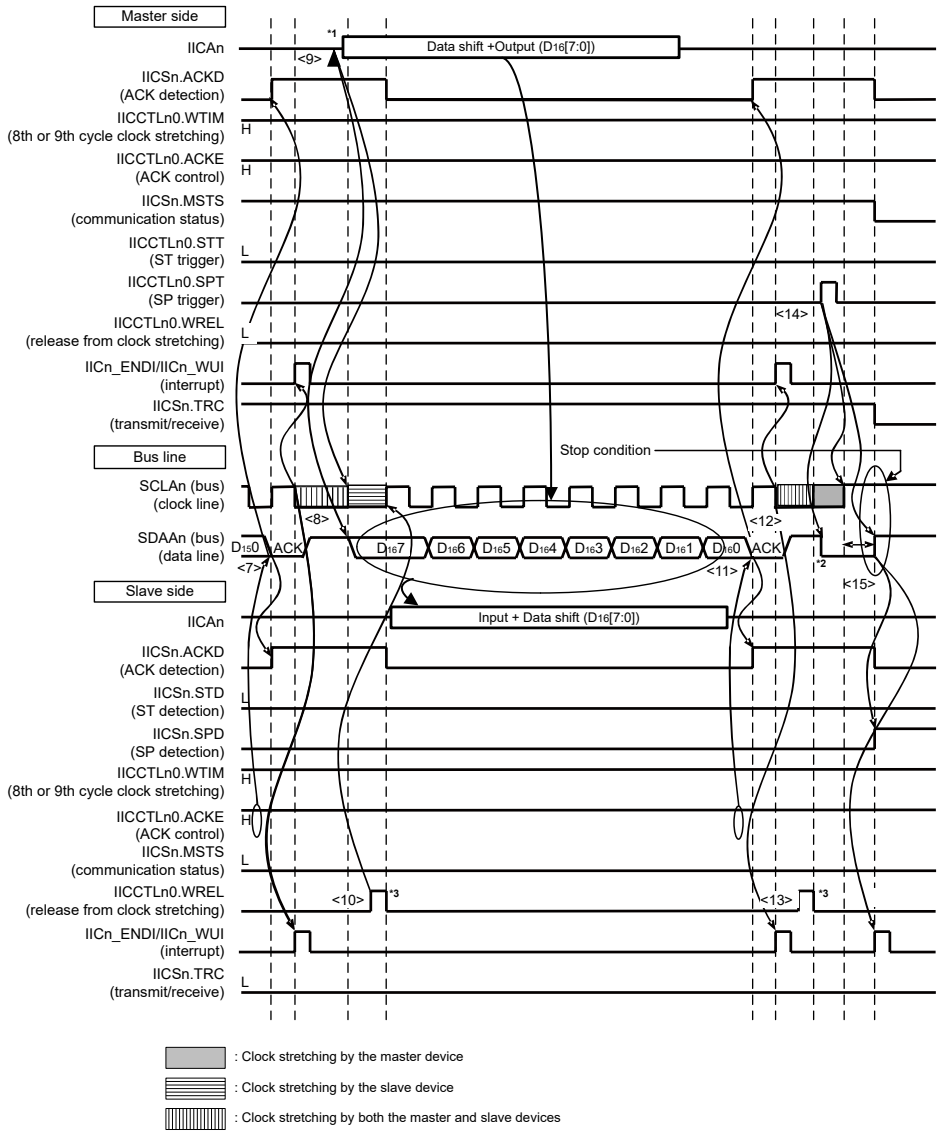
Note: <1> to <15> in (1) Example of Master to Slave Communications (9th Cycle Clock Stretching Is Selected for Both the Master and Slave) represent the entire procedure for communicating data using the I^2C bus.

Figure 24.63 shows the processing from <1> to <6>,

Figure 24.64 shows the processing from <3> to <10>, and

Figure 24.65 shows the processing from <7> to <15>.

3. Data → data → stop condition



Note: n = 0, 1

Note 1. For release from the clock stretch state during transmission by a master device, write data to the IICAn register instead of setting the IICCTLn0.WREL bit.

Note 2. Make sure that the time between the rise of the SCLAn pin signal and the generation of the stop condition after a stop condition has been issued is at least 4.0 µs when specifying standard mode and at least 0.6 µs when specifying fast mode.

Note 3. For release from the clock stretch state during reception by a slave device, write 0xFF to IICAn or set the IICCTLn0.WREL bit.

Figure 24.65 Example of master to slave communications (9th cycle clock stretching is selected for both the master and slave) (3/4)

The meanings of <7> to <15> in Figure 24.65 are explained below.

<7> After data transfer is completed, because of IICCTLn0.ACKE = 1, the slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (IICSn.ACKD = 1) at the rising edge of the 9th clock.

<8> The master device and slave device set the clock stretch state (SCLAn = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (IICn_ENDI/IICn_WUI: end of transfer).

<9> The master device writes the data to transmit to the IICA shift register n (IICAn) and releases the clock stretch state set by the master device.

<10>The slave device reads the received data and releases the clock stretch state (IICCTLn0.WREL = 1). The master device then starts transferring data to the slave device.

<11>When data transfer is complete, the slave device (IICCTLn0.ACKE = 1) sends an ACK by hardware to the master device.

The ACK is detected by the master device (IICSn.ACKD = 1) at the rising edge of the 9th clock.

<12>The master device and slave device set the clock stretch state (SCLAn = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (IICn_ENDI/IICn_WUI: end of transfer).

<13>The slave device reads the received data and releases the clock stretch state (IICCTLn0.WREL = 1).

<14>By the master device setting a stop condition trigger (IICCTLn0.SPT = 1), the bus data line is cleared (SDAAn = 0) and the bus clock line is set (SCLAn = 1). After the stop condition setup time has elapsed, by setting the bus data line (SDAAn = 1), the stop condition is then generated (SCLAn = 1 and SDAAn changes from 0 to 1).
<15>When a stop condition is generated, the slave device detects the stop condition and issues an interrupt (IICn_ENDI/IICn_WUI: stop condition).

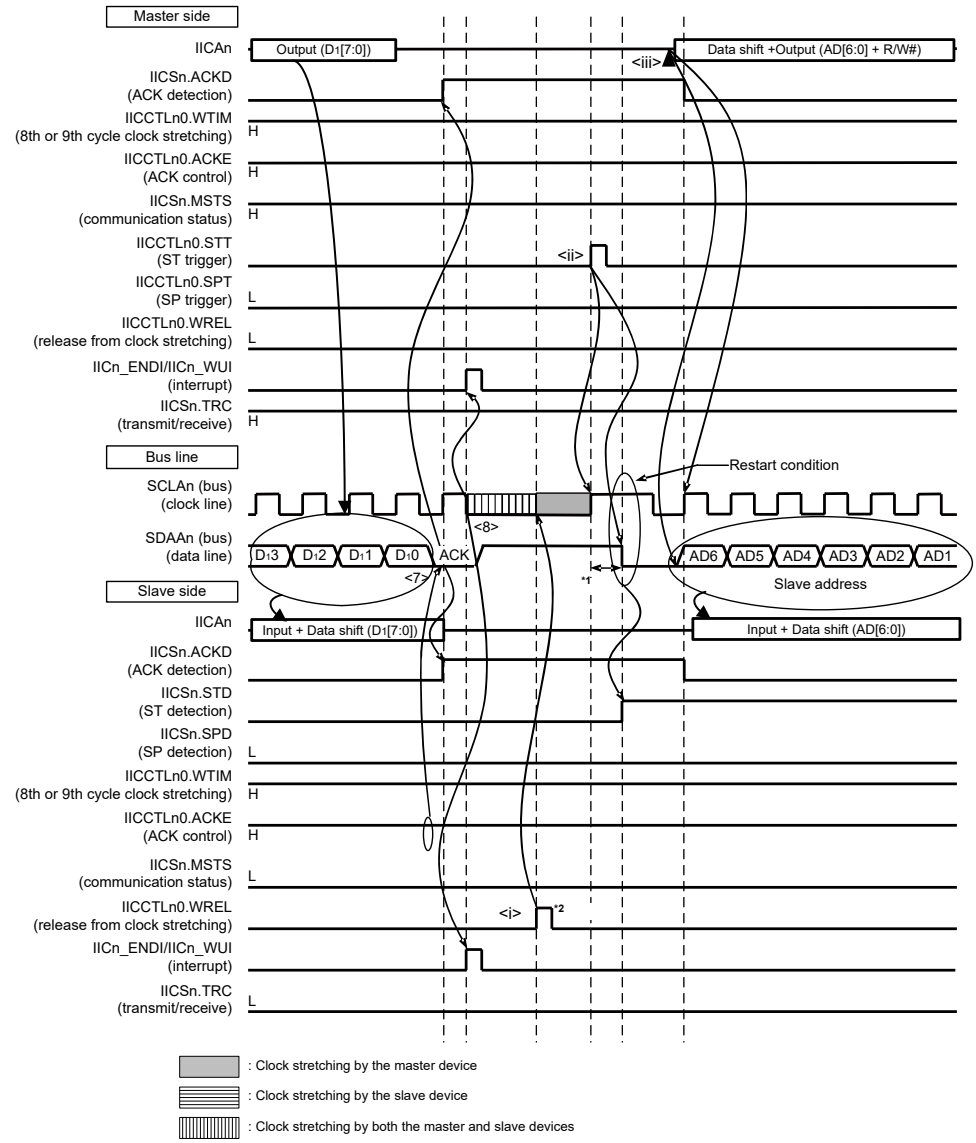
Note: <1> to <15> in (1) Example of Master to Slave Communications (9th Cycle Clock Stretching Is Selected for Both the Master and Slave) represent the entire procedure for communicating data using the I^2C bus.

Figure 24.63 shows the processing from <1> to <6>,

Figure 24.64 shows the processing from <3> to <10>, and

Figure 24.65 shows the processing from <7> to <15>.

4. Data → restart condition → address



Note: n = 0, 1

Note 1. Make sure that the time between the rise of the SCLAn pin signal and the generation of the start condition after a restart condition has been issued is at least 4.7 µs when specifying standard mode and at least 0.6 µs when specifying fast mode.

Note 2. For release from the clock stretch state during reception by a slave device, write 0xFF to IICAn or set the IICCTLn0.WREL bit.

Figure 24.66 Example of master to slave communications (9th cycle clock stretching is selected for both the master and slave) (4/4)

The following describes the operations in Figure 24.66. After the operations in steps <7> and <8>, the operations in steps <i> to <iii> are performed. These steps return the processing to step <3>, the data transmission step.

<7> After data transfer is completed, because of IICCTLn0.ACKE = 1, the slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (IICSn.ACKD = 1) at the rising edge of the 9th clock.

<8> The master device and slave device set the clock stretch state (SCLAn = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (IICn_ENDI/IICn_WUI: end of transfer).

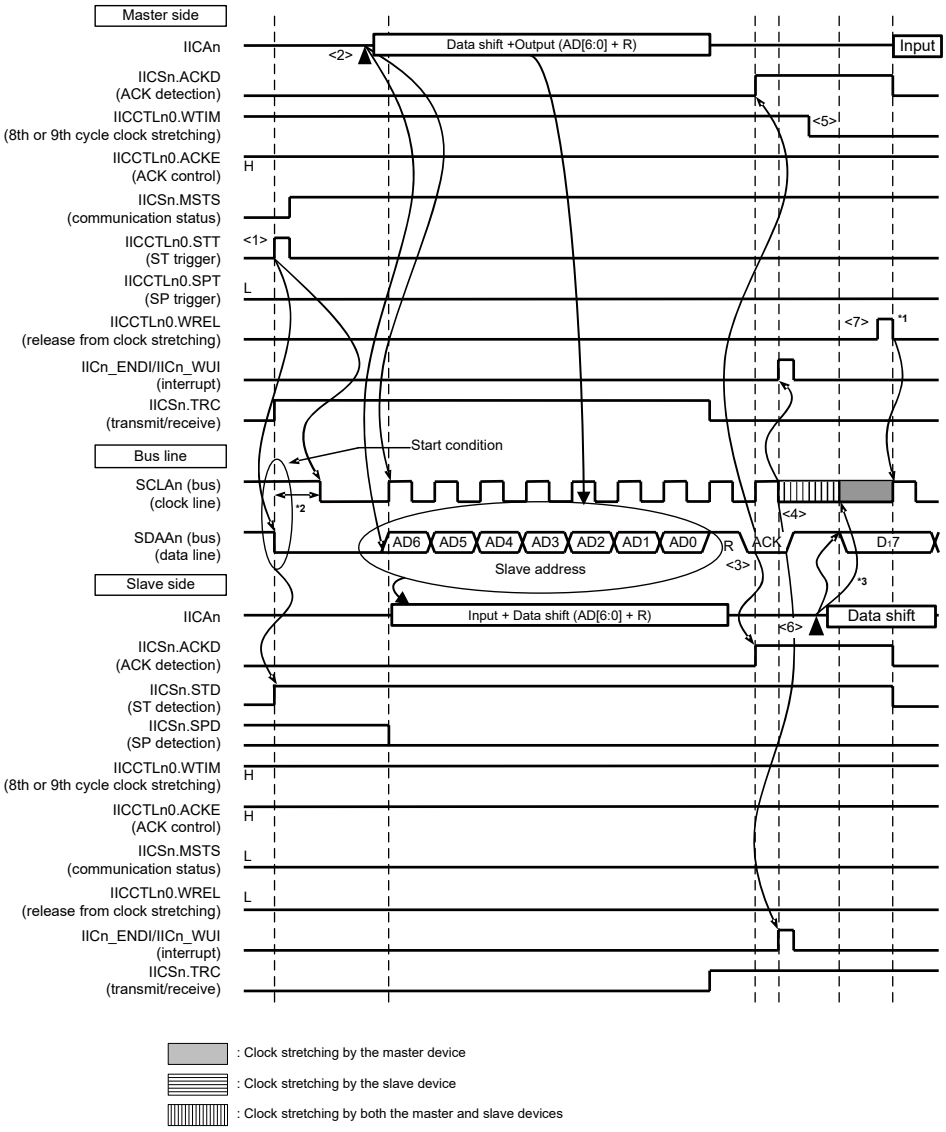
<i> The slave device reads the received data and releases the clock stretch state (IICCTLn0.WREL = 1).

<ii> The start condition trigger is set again by the master device (IICCTLn0.STT = 1) and a start condition (SCLAn = 1 and SDAAn changes from 1 to 0) is generated once the bus clock line goes high (SCLAn = 1) and the bus data line goes low (SDAAn = 0) after the restart condition setup time has elapsed. When the start condition is subsequently detected, the master device is ready to communicate once the bus clock line goes low (SCLAn = 0) after the hold time has elapsed.

<iii> The master device writing the address + R/W (transmission) to the IICA shift register n (IICAn) enables the slave address to be transmitted.

(2) Example of Slave to Master Communications (8th Cycle Clock Stretching Is Selected for the Master and 9th Cycle Clock Stretching Is Selected for the Slave)

1. Start condition → address → data



Note: n = 0, 1

Note 1. For release from the clock stretch state during reception by a master device, write 0xFF to IICAn or set the IICCTLn0.WREL bit.

Note 2. Make sure that the time between the fall of the SDAAn pin signal and the fall of the SCLAn pin signal is at least 4.0 µs when specifying standard mode and at least 0.6 µs when specifying fast mode.

Note 3. For release from the clock stretch state during transmission by a slave device, write data to the IICAn register instead of setting the IICCTLn0.WREL bit.

Figure 24.67 Example of slave to master communications (8th cycle clock stretching is selected for the master and 9th cycle clock stretching is selected for the slave) (1/3)

The meanings of <1> to <7> in Figure 24.67 are explained below.

<1> The start condition trigger is set by the master device (IICCTLn0.STT = 1) and a start condition (SCLAn = 1 and SDAAn changes from 1 to 0) is generated once the bus data line goes low (SDAAn). When the start condition is subsequently detected, the master device enters the master device communication status (IICSn.MSTS = 1). The master device is ready to communicate once the bus clock line goes low (SCLAn = 0) after the hold time has elapsed.

<2> The master device writes the address + R (reception) to the IICA shift register n (IICAn) and transmits the slave address.

<3> In the slave device if the address received matches the address (SVAn value) of a slave device, that slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (IICSn.ACKD = 1) at the rising edge of the 9th clock.

<4> The master device issues an interrupt (IICn_ENDI/IICn_WUI: end of address transmission) at the falling edge of the 9th clock. The slave device with the address matching the transmitted slave address sets the clock stretch state (SCLAn = 0) and issues an interrupt (IICn_ENDI/IICn_WUI: address match).

<5> The timing at which the master device sets the clock stretch state changes to the 8th clock (IICCTLn0.WTIM = 0).

<6> The slave device writes the data to transmit to the IICAn register and releases the clock stretch state set by the slave device.

<7> The master device releases the clock stretch state (IICCTLn0.WREL = 1) and starts transferring data from the slave device to the master device.

If the transmitted address does not match the address of the slave device, the slave device does not return an ACK to the master device (NACK: SDAAn = 1). The slave device also does not issue the IICn_ENDI/IICn_WUI interrupt (address match) and does not set the clock stretch state.

The master device, however, issues the IICn_ENDI/IICn_WUI interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

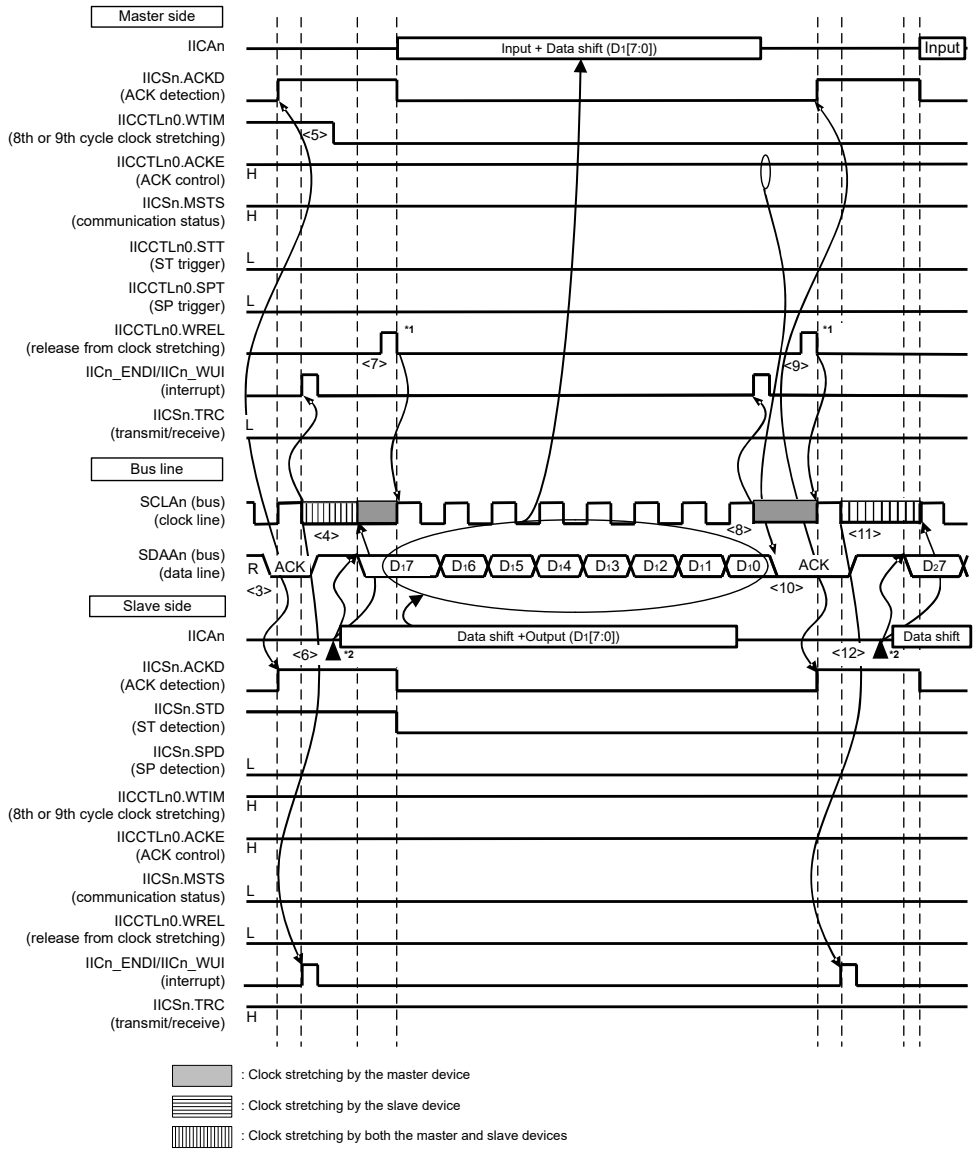
Note: <1> to <19> in (2) Example of Slave to Master Communications (8th Cycle Clock Stretching Is Selected for the Master and 9th Cycle Clock Stretching Is Selected for the Slave) represent the entire procedure for communicating data using the I^2C bus.

Figure 24.67 shows the processing from <1> to <7>,

Figure 24.68 shows the processing from <3> to <12>, and

Figure 24.69 shows the processing from <8> to <19>.

2. Address → data → data



Note: n = 0, 1

Note 1. For release from the clock stretch state during reception by a master device, write 0xFF to IICAn or set the IICCTLn0.WREL bit.

Note 2. For release from the clock stretch state during transmission by a slave device, write data to the IICAn register instead of setting the IICCTLn0.WREL bit.

Figure 24.68 Example of slave to master communications (8th cycle clock stretching is selected for the master and 9th cycle clock stretching is selected for the slave) (2/3)

The meanings of <3> to <12> in Figure 24.68 are explained below.

<3> In the slave device if the address received matches the address (SVAn value) of a slave device, that slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (IICSn.ACKD = 1) at the rising edge of the 9th clock.

<4> The master device issues an interrupt (IICn_ENDI/IICn_WUI: end of address transmission) at the falling edge of the 9th clock. The slave device with the address matching the transmitted slave address sets the clock stretch state (SCLAn = 0) and issues an interrupt (IICn_ENDI/IICn_WUI: address match).

<5> The master device changes the timing of clock stretching to the 8th clock (IICCTLn0.WTIM = 0).

<6> The slave device writes the data to transmit to the IICA shift register n (IICAn) and releases the clock stretch state set by the slave device.

<7> The master device releases the clock stretch state (IICCTLn0.WREL = 1) and starts transferring data from the slave device to the master device.

<8> The master device sets the clock stretch state (SCLAn = 0) at the falling edge of the 8th clock, and issues an interrupt (IICn_ENDI/IICn_WUI: end of transfer). Because of IICCTLn0.ACKE = 1 in the master device, the master device then sends an ACK by hardware to the slave device.

<9> The master device reads the received data and releases the clock stretch state (IICCTLn0.WREL = 1).

<10>The ACK is detected by the slave device (IICSn.ACKD = 1) at the rising edge of the 9th clock.

<11>The slave device sets the clock stretch state (SCLAn = 0) at the falling edge of the 9th clock, and the slave device issue an interrupt (IICn_ENDI/IICn_WUI: end of transfer).

<12>By the slave device writing the data to transmit to the IICAn register, the clock stretch state set by the slave device is released. The slave device then starts transferring data to the master device.

If the transmitted address does not match the address of the slave device, the slave device does not return an ACK to the master device (NACK: SDAAn = 1). The slave device also does not issue the IICn_ENDI/IICn_WUI interrupt (address match) and does not set the clock stretch state.

The master device, however, issues the IICn_ENDI/IICn_WUI interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

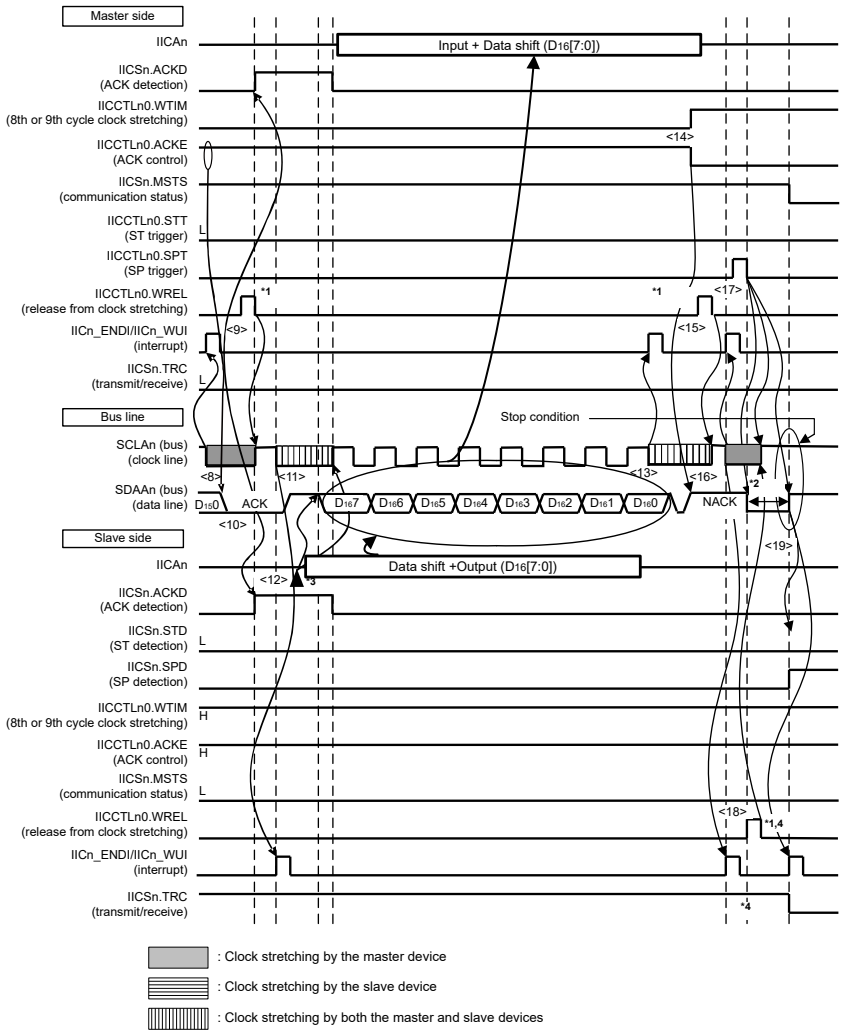
Note: <1> to <19> in (2) Example of Slave to Master Communications (8th Cycle Clock Stretching Is Selected for the Master and 9th Cycle Clock Stretching Is Selected for the Slave) represent the entire procedure for communicating data using the I^2C bus.

Figure 24.67 shows the processing from <1> to <7>,

Figure 24.68 shows the processing from <3> to <12>, and

Figure 24.69 shows the processing from <8> to <19>.

3. Data → data → stop condition



Note: n = 0, 1
Note 1. For release from the clock stretch state, write 0xFF to IICAn or set the IICCTLn0.WREL bit.
Note 2. Make sure that the time between the rise of the SCLAn pin signal and the generation of the stop condition after a stop condition has been issued is at least 4.0 µs when specifying standard mode and at least 0.6 µs when specifying fast mode.
Note 3. For release from the clock stretch state during transmission by a slave device, write data to the IICAn register instead of setting the IICCTLn0.WREL bit.
Note 4. If the clock stretch state during transmission by a slave device is released by setting the IICCTLn0.WREL bit, the IICSn.TRC bit is cleared.

Figure 24.69 Example of slave to master communications (8th cycle clock stretching is selected for the master and 9th cycle clock stretching is selected for the slave) (3/3)

The meanings of <8> to <19> in Figure 24.69 are explained below.

<8> The master device sets the clock stretch state (SCLAn = 0) at the falling edge of the 8th clock, and issues an interrupt (IICn_ENDI/IICn_WUI: end of transfer). Because of IICCTLn0.ACKE = 0 in the master device, the master device then sends an ACK by hardware to the slave device.

<9> The master device reads the received data and releases the clock stretch state (IICCTLn0.WREL = 1).

<10>The ACK is detected by the slave device (IICSn.ACKD = 1) at the rising edge of the 9th clock.

<11>The slave device sets the clock stretch state (SCLAn = 0) at the falling edge of the 9th clock, and the slave device issue an interrupt (IICn_ENDI/IICn_WUI: end of transfer).

<12>By the slave device writing the data to transmit to the IICAn register, the clock stretch state set by the slave device is released. The slave device then starts transferring data to the master device.

<13>The master device issues an interrupt (IICn_ENDI/IICn_WUI: end of transfer) at the falling edge of the 8th clock, and sets the clock stretch state (SCLAn = 0). Because ACK control (IICCTLn0.ACKE = 1) is performed, the bus data line is at the low level (SDAAn = 0) at this stage.

<14>The master device sets NACK as the response (IICCTLn0.ACKE = 0) and changes the timing at which it sets the clock stretch state to the 9th clock (IICCTLn0.WTIM = 1).

<15>If the master device releases the clock stretch state (IICCTLn0.WREL = 1), the slave device detects the NACK (ACK = 0) at the rising edge of the 9th clock.

<16>The master device and slave device set the clock stretch state (SCLAn = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (IICn_ENDI/IICn_WUI: end of transfer).

<17>When the master device issues a stop condition (IICCTLn0.SPT = 1), the bus data line is cleared (SDAAn = 0) and the master device releases the clock stretch state. The master device then waits until the bus clock line is set (SCLAn = 1).

<18>The slave device acknowledges the NACK, halts transmission, and releases the clock stretch state (IICCTLn0.WREL = 1) to end communication. Once the slave device releases the clock stretch state, the bus clock line is set (SCLAn = 1).

<19>Once the master device recognizes that the bus clock line is set (SCLAn = 1) and after the stop condition setup time has elapsed, the master device sets the bus data line (SDAAn = 1) and issues a stop condition (SCLAn = 1 and SDAAn changes from 0 to 1). The slave device detects the generated stop condition and slave device issue an interrupt (IICn_ENDI/IICn_WUI: stop condition).

Note: <1> to <19> in (2) Example of Slave to Master Communications (8th Cycle Clock Stretching Is Selected for the Master and 9th Cycle Clock Stretching Is Selected for the Slave) represent the entire procedure for communicating data using the I^2C bus.

Figure 24.67 shows the processing from <1> to <7>,

Figure 24.68 shows the processing from <3> to <12>, and

Figure 24.69 shows the processing from <8> to <19>.

25. Serial Interface UARTA (UARTA)

25.1 Overview

Table 25.1 lists specifications of the serial interface UARTA.

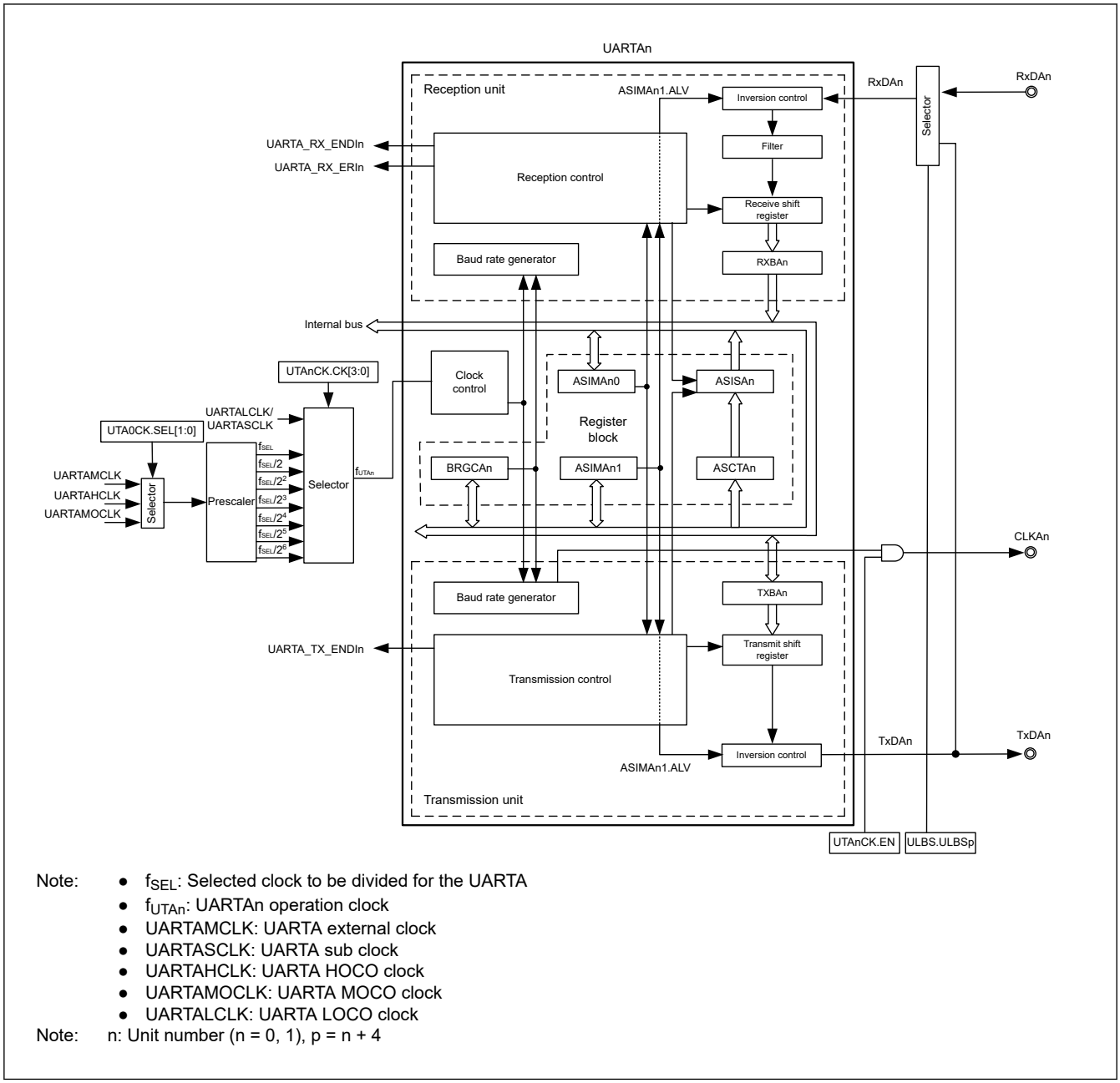
Table 25.1 UARTA specifications

Item	Specifications
Serial interface modes	• Operation stop mode • UART mode
Interfaces	• TxDAn: Transmit data output pin • RxDAn: Receive data input pin • CLKAn : Transmit clock output pin
Operation clock sources	Operating clock independent of the system/peripheral module clock selectable to UARTAMCLK, UARTALCLK/UARTASCLK, UARTAHCLK, and UARTAMOCLK
Transfer rate	Up to 153.6 kbps
Baud rate	Settable with the dedicated internal 8-bit baud rate generator
Data format	• MSB-first or LSB-first selectable • Transfer bit length selectable to 5, 7, or 8 bits
Interrupt sources	• Transfer completion interrupt • Reception transfer end • Reception error interrupt
Other functions	• Transmission and reception independent of each other (full-duplex communication) • Inversion control of communication logic level provided • Loopback mode
Module-stop function	Module-stop state can be set to reduce power consumption

Note: • UARTAMCLK: UARTA external clock
• UARTASCLK: UARTA sub clock
• UARTAHCLK: UARTA HOCO clock
• UARTAMOCLK: UARTA MOCO clock
• UARTALCLK: UARTA LOCO clock

Note: n: Unit number (n = 0, 1)

Figure 25.1 shows a block diagram of UARTAn and Table 25.2 shows the pin configuration of UARTAn.



Note: • f_{SEL}: Selected clock to be divided for the UARTA
- f_{UTAn}: UARTAn operation clock
- UARTAMCLK: UARTA external clock
- UARTASCLK: UARTA sub clock
- UARTAHCLK: UARTA HOCO clock
- UARTAMOCLK: UARTA MOCO clock
- UARTALCLK: UARTA LOCO clock

Note: n: Unit number (n = 0, 1), p = n + 4

Figure 25.1 Block diagram of UARTAn

Table 25.2 UARTAn pin configuration

Name	I/O	Function
RxDAn	Input	Serial data input signal
TxDAn	Output	Serial data output signal
CLKAn	Output	Serial clock output signal

25.2 Register Descriptions

25.2.1 TXBAn : Transmit Buffer Register n (n = 0, 1)

Base address: UARTA = 0x4009_6000

Offset address: 0x00 + 0x8 × n

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

Bit	Symbol	Function	R/W
7:0	n/a	Transmit data buffer	R/W

TXBAn is a buffer register for setting transmit data.

Transmission starts by writing data for transmission to the TXBAn register.

When a character length of 8 bits is specified:

- Data in bits [7:0] of TXBAn are transferred.

When a character length of 7 bits is specified:

- Data in bits [6:0] of TXBAn are transferred in either MSB- or LSB-first mode. Bit 7 is invalid.

When a character length of 5 bits is specified:

- Data in bits [4:0] of TXBAn are transferred in either MSB- or LSB-first mode. Bits [7:5] are invalid.

Note: When the TXBFA bit of the ASISAn register is 1, do not write data for transmission to the TXBAn register.

Note: After setting the TXEA bit of the ASIMAn0 register to 1, wait for the period of at least one cycle of the UARTAn operation clock (f_{UTAn}) before setting the first data for transmission in the TXBAn register. If data for transmission is set within one cycle of the UARTAn operation clock after the ASIMAn0.TXEA bit is set to 1, the start of transmission is delayed by one cycle of the UARTAn operation clock.

Note: Data is transferred from the TXBAn register to this register, and is then transmitted as serial data through the TxDAn pin. In the first transmission, data is transferred from the TXBAn register to this register immediately after data is written to the TXBAn register. In continuous transmission, data is transferred after transmission of one frame and just before generation of the transfer completion interrupt.

The transmit shift register cannot be manipulated directly by software.

25.2.2 RXBAn : Receive Buffer Register n (n = 0, 1)

Base address: UARTA = 0x4009_6000

Offset address: 0x01 + 0x8 × n

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

Bit	Symbol	Function	R/W
7:0	n/a	Receive data buffer	R

The RXBAn register stores the parallel data converted by the receive shift register. Every time one byte of data is received, the next receive data is transferred from the receive shift register *1 to this register.

Note 1. The receive shift register converts the serial data that is input through the RxDAn pin to parallel data.

The receive shift register cannot be manipulated directly by software.

When a character length of 8 bits is specified:

- Receive data is transferred to bits [7:0] of this register.

When a character length of 7 bits is specified:

- Receive data is transferred to bits [6:0] of this register in either MSB- or LSB-first mode. Bit 7 is always 0.

When a character length of 5 bits is specified:

- Receive data is transferred to bits [4:0] of this register in either MSB- or LSB-first mode. Bits [7:5] are always 0.

Note: If an overrun error (ASISAn.OVEA) occurs, the data received at that time is not stored in the RXBAn register.

25.2.3 ASIMAn0 : Operation Mode Setting Register n0 (n = 0, 1)

Base address: UARTA = 0x4009_6000

Offset address: 0x02 + 0x8 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	EN	TXEA	RXEA	—	—	—	ISSMA	ISRMA
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	ISRMA	Receive interrupt mode select 0: The UARTA_RX_ERIn interrupt is generated when a reception error occurs (UARTA_RX_ENDIn is not generated) 1: The UARTA_RX_ENDIn interrupt is generated when a reception error occurs (UARTA_RX_ERIn is not generated)	R/W
1	ISSMA	Transmit interrupt mode select 0: The UARTA_TX_ENDIn interrupt is generated on completion of transmission 1: The UARTA_TX_ENDIn interrupt is generated when the transmit buffer becomes empty (for continuous transmission)	R/W
4:2	—	These bits are read as 0. The write value should be 0.[*1]	R/W
5	RXEA	Reception enable 0: Disables reception (reset the reception circuit) 1: Enables reception	R/W
6	TXEA	Transmission enable 0: Disables transmission (resets the transmission circuit) 1: Enables transmission	R/W
7	EN[*2]	UART operation enable 0: Disables the UART operation clock (resets the internal circuits[*3]) 1: Enables the UART operation clock	R/W

Note 1. Be sure to clear bits [4:2] to 0.

Note 2. When EN = 0, the level being output from the TxDAn pin and the level being input from the RxDAn pin are determined according to the setting of the ASIMAn1.ALV bit as follow:
- When ASIMAn1.ALV = 0, output from the TxDAn pin is high.
- When ASIMAn1.ALV = 1, output from the TxDAn pin is low.

Note 3. The ASISAn and RXBAn registers are reset by clearing the EN bit to 0.

The ASIMAn0 register controls serial communication of the serial interface UARTAn.

To start transmission, set the EN bit to 1 and then set the TXEA bit to 1. To stop transmission, clear the TXEA bit to 0 and then clear the EN bit to 0.

To start reception, set the EN bit to 1 and then set the RXEA bit to 1. To stop reception, clear the RXEA bit to 0 and then clear the EN bit to 0.

Use the following procedure when setting the EN bit to 1 and then setting the RXEA bit to 1.

- When ASIMAn1.ALV = 0

The setting must be made while the level being input to the RxDAn pin is high. Otherwise, reception starts at that point and a framing error may occur.

- When ASIMAn1.ALV = 1
  The setting must be made while the level being input to the RxDAn pin is low. Otherwise, reception starts at that point and a framing error may occur.

The TXEA and RXEA bits are synchronized with the UARTAn operation clock (f_{UTAn}).

To enable transmission or reception again, set the TXEA or RXEA bit to 1 at least two cycles of the UARTAn operation clock after clearing the TXEA or RXEA bit to 0. If the bit is set to 1 within two cycles of the UARTAn operation clock after the clearing, the transmission or reception circuit may not be able to initialize.

After setting TXEA bit to 1, wait for at least one cycle of the UARTAn operation clock (f_{UTAn}) before setting the transmit data in the TXBAn register.

Clear the RXEA bit to 0 before modifying the ISRMA bit.

25.2.4 ASIMAn1 : Operation Mode Setting Register n1 (n = 0, 1)

Base address: UARTA = 0x4009_6000

Offset address: 0x03 + 0x8 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	PS[1:0]		CL[1:0]		SL	DIR	ALV
Value after reset:	0	0	0	1	1	0	1	0

Bit	Symbol	Function	R/W
0	ALV	Transmission and reception level setting 0: Positive logic (wait state = high level, start bit = low level, stop bit = high level) 1: Negative logic (wait state = low level, start bit = high level, stop bit = low level)	R/W
1	DIR	Transmission and reception order setting 0: MSB-first 1: LSB-first	R/W
2	SL	Transmission stop bit length setting 0: Stop bit length = 1 bit 1: Stop bit length = 2 bits	R/W
4:3	CL[1:0]	Transmission and reception character length setting 0 0: Character length of data = 5 bits 0 1: Setting prohibited 1 0: Character length of data = 7 bits 1 1: Character length of data = 8 bits	R/W
6:5	PS[1:0]	Transmission and reception parity bit setting 0 0: Transmission: No parity bit is output. Reception: Data is received without parity. 0 1: Transmission: 0 parity is output. Reception: Data is received with 0 parity.*1 1 0: Transmission: Odd parity is output. Reception: Check is made for odd parity. 1 1: Transmission: Even parity is output. Reception: Check is made for even parity.	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

Note 1. When "Data is received with 0 parity" is set, parity check is not performed. Accordingly the PEA bit of the ASISAn register is not set: no reception error interrupts are generated.

The ASIMAn1 register controls serial communication of the serial interface UARTAn.

The ASIMAn1 register must be modified while ASIMAn0.TXEA = 0 and ASIMAn0.RXEA = 0.

Clear both the ASIMAn0.TXEA and RXEA bits to 0 before modifying the ASIMAn1 register. Reception is always handled as including a stop bit. The setting of the SL bit does not affect reception.

25.2.5 BRGCAn : Baud Rate Generator Control Register n (n = 0, 1)

Base address: UARTA = 0x4009_6000

Offset address: 0x04 + 0x8 × n

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

Bit	Symbol	Function	R/W
7:0	n/a	Controls the UART baud rate (serial transfer speed) Selection of 8-bit counter output clock (f_{UTAn} / BRGCAn) 0x02: $f_{UTAn}/2$ 0x03: $f_{UTAn}/3$ ⋮ ⋮ 0xFC: $f_{UTAn}/252$ 0xFD: $f_{UTAn}/253$ 0xFE: $f_{UTAn}/254$ 0xFF: $f_{UTAn}/255$ Others: Setting prohibited	R/W

The BRGCAn register sets the frequency divisor for the 8-bit counter in the serial interface UARTAn.

Modify the BRG[7:0] bits while the ASIMAn0.TXEA and RXEA bits are 0 (in the transmission and reception stopped state). The baud rate is one half the frequency of the output clock signal from the 8-bit counter. For an example of the baud rate setting, see (c) Baud rate setting example.

25.2.6 ASISAn : Status Register n (n = 0, 1)

Base address: UARTA = 0x4009_6000

Offset address: 0x05 + 0x8 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	TXBFA	TXSFA	—	PEA	FEA	OVEA
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	OVEA	Overrun error flag 0: No error has occurred 1: An error has occurred	R
1	FEA	Framing error flag 0: No error has occurred 1: An error has occurred	R
2	PEA	Parity error flag 0: No error has occurred 1: An error has occurred	R
3	—	This bit is read as 0.	R
4	TXSFA	Transmit shift register data flag 0: Data is not being transmitted 1: Data is being transmitted	R
5	TXBFA	Transmit buffer data flag 0: No valid data exists in the TXBAn register 1: Valid data exists in the TXBAn register	R
7:6	—	These bits are read as 0.	R

The ASISAn register indicates the error status and the transmission status on completion of reception by the serial interface UARTAn. It consists of three error flag bits (PEA, FEA, and OVEA) and two transmission status flag bits (TXBFA and TXSFA).

The PEA, FEA, and OVEA bits are initialized by clearing the ASIMAn0.EN or RXEA bit to 0. These bits are also cleared by writing to the corresponding bit of the ASCTAn register. The TXBFA and TXSFA flags are initialized by clearing the ASIMAn0.EN or TXEA bit to 0.

For continuous transmission, be sure to check that the TXBFA flag is 0 after writing the first transmit data (the first byte) to the TXBAn register and then write the next transmit data (the second byte) to the TXBAn register. Otherwise, the transmit data becomes undefined. However, the TXBFA flag need not be checked when continuous transmission is performed using the buffer empty interrupt (ASIMAn0.ISSMA bit = 1).

When initializing the transmission unit (ASIMAn0.TXEA = 0) after completion of continuous transmission, be sure to check that the TXSFA flag is 0 after the transfer completion interrupt is generated, and then initialize the unit. Otherwise, the transmit data becomes undefined.

The operation of the PEA bit depends on the setting of the PS[1:0] bits of the ASIMAn1 register. For receive data, only the first one bit of the stop bits is checked regardless of the stop bit length. When an overrun error occurs, the next receive data is not written to the RXBAn register and is discarded.

OVEA flag (Overrun error flag)

[Clearing condition]

- The ASIMAn0.EN or RXEA bit is cleared to 0.
- 1 is written to the ASCTAn.OVECTA bit.

[Setting condition]

- The next reception is completed before the receive data in the RXBAn register is read.

FEA flag (Framing error flag)

[Clearing condition]

- The ASIMAn0.EN or RXEA bit is cleared to 0.
- 1 is written to the ASCTAn.FECTA bit.

[Setting condition]

- A stop bit is not detected when receiving data.

PEA flag (Parity error flag)

[Clearing condition]

- The ASIMAn0.EN or RXEA bit is cleared to 0.
- 1 is written to the ASCTAn.PECTA bit.

[Setting condition]

- The parity of the received data does not match the parity bit.

TXSFA flag (Transmit shift register data flag)

[Clearing condition]

- The ASIMAn0.EN or TXEA bit is cleared to 0.
- Data is transferred from the transmit shift register and then no subsequent data is transferred from the TXBAn register.

[Setting condition]

- Data is transferred from the TXBAn register. (Data is being transmitted.)

TXBFA flag (Transmit buffer data flag)

[Clearing condition]

- The ASIMAn0.EN or TXEA bit is cleared to 0.
- Data is transferred to the transmit shift register.

[Setting condition]

- Data is written to the TXBAn register. (Data exists in the TXBAn register.)

25.2.7 ASCTAn : Status Clear Trigger Register n (n = 0, 1)

Base address: UARTA = 0x4009_6000

Offset address: 0x06 + 0x8 × n

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

Bit	Symbol	Function	R/W
0	OVECTA*1	Overrun error flag clear trigger 0: Does not clear the ASISAn.OVEA flag (the flag is retained) 1: Clears the ASISAn.OVEA flag	R/W
1	FECTA*1	Framing error flag clear trigger 0: Does not clear the ASISAn.FEA flag (the flag is retained) 1: Clears the ASISAn.FEA flag	R/W
2	PECTA*1	Parity error flag clear trigger 0: Does not clear the ASISAn.PEA flag (the flag is retained) 1: Clears the ASISAn.PEA flag	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. When reading the ASCTAn register, 0 is returned.

The ASCTAn register sets the trigger to clear the error status on completion of reception of the serial interface UARTAn. It contains 3 bits of the error clear trigger flags (PECTA, FECTA, and OVECTA).

When the ASCTAn register is read, 0x00 is always read.

Writing 1 to the PECTA, FECTA, and OVECTA bits clears the PEA, FEA, and OVEA flags of the ASISAn register, respectively. When writing 0, the corresponding error flags are not cleared.

After writing 1 to the trigger bit, the corresponding error flag is cleared on the next rising edge of the operating clock (f_{UTAn}). Accordingly, if reading the ASISAn register immediately after writing 1 to the trigger bit, the corresponding error flag may not have been cleared yet.

25.2.8 UTA0CK : UARTA Clock Select Register 0

Base address: UARTA = 0x4009_6000

Offset address: 0x10

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

Bit	Symbol	Function	R/W
3:0	CK[3:0]	UARTA0 operation clock select (f_{UTA0}) 0x0: f_{SEL} 0x1: $f_{SEL}/2$ 0x2: $f_{SEL}/4$ 0x3: $f_{SEL}/8$ 0x4: $f_{SEL}/16$ 0x5: $f_{SEL}/32$ 0x6: $f_{SEL}/64$ 0x8: UARTALCLK/UARTASCLK Others: Setting prohibited	R/W
5:4	SEL[1:0]	f_{SEL} clock select 0 0: Stop 0 1: UARTAMCLK 1 0: UARTAHCLK 1 1: UARTAMOCLK	R/W
6	—	This bit is read as 0. The write value should be 0.	R/W
7	EN	UARTA0 clock output function enable 0: Disables CLKA0 output 1: Enables CLKA0 output	R/W

Note:
- f_{SEL}: Selected clock to be divided for the UARTA
- UARTAMCLK: UARTA external clock
- UARTASCLK: UARTA sub clock
- UARTAHCLK: UARTA HOCO clock
- UARTAMOCLK: UARTA MOCO clock
- UARTALCLK: UARTA LOCO clock

The UTA0CK register selects the operating clock of the UARTAn. The SEL[1:0] bits select the clock source, f_{SEL}, for UARTAn from UARTAMCLK, UARTAHCLK, and UARTAMOCLK. The CK[3:0] bits select the operating clock for UARTA0 from f_{SEL} to $f_{SEL}/64$, and UARTALCLK/UARTASCLK.

Setting the EN bit to 1 selects the UARTA0 clock output mode, and the operating clock for UARTA0 is output from the CLKA0 pin.

This register should be read or written when the ASIMAn0.TXEA and RXEA bits are 0 (in the transmission and reception stopped state).

25.2.9 UTA1CK : UARTA Clock Select Register 1

Base address: UARTA = 0x4009_6000

Offset address: 0x11

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

Bit	Symbol	Function	R/W
3:0	CK[3:0]	UARTA1 operation clock select (f_{UTA1}) 0x0: f_{SEL} 0x1: $f_{SEL}/2$ 0x2: $f_{SEL}/4$ 0x3: $f_{SEL}/8$ 0x4: $f_{SEL}/16$ 0x5: $f_{SEL}/32$ 0x6: $f_{SEL}/64$ 0x8: UARTALCLK/UARTASCLK 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	EN	UARTA1 clock output function enable 0: Disables CLKA1 output 1: Enables CLKA1 output	R/W

Note:
- f_{SEL}: Selected clock to be divided for the UARTA
- UARTALCLK: UARTA LOCO clock
- UARTASCLK: UARTA sub clock

The UTA1CK register selects the operating clock of UARTA1. The CK[3:0] bits select the operating clock for UARTA1 from f_{SEL} to $f_{SEL}/64$, and UARTALCLK/UARTASCLK.

Setting the EN bit to 1 selects the UARTA1 clock output mode, and the operating clock for UARTA1 is output from the CLKA1 pin.

This register should be read or written when the ASIMA10.TXEA and RXEA bits are 0 (in the transmission and reception stopped state).

25.2.10 ULBS : UART Loopback Select Register

Base address: PORGA = 0x4009_1000

Offset address: 0x0009

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	ULBS5	ULBS4	—	ULBS2	ULBS1	ULBS0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ULBS0	Selection of the UART0 loopback function 0: Inputs the states of the RxD0 pin of serial array unit UART0 to the reception shift register 1: Loops back output from the transmission shift register to the reception shift register	R/W
1	ULBS1	Selection of the UART1 loopback function 0: Inputs the states of the RxD1 pin of serial array unit UART1 to the reception shift register 1: Loops back output from the transmission shift register to the reception shift register	R/W
2	ULBS2	Selection of the UART2 loopback function 0: Inputs the states of the RxD2 pin of serial array unit UART2 to the reception shift register 1: Loops back output from the transmission shift register to the reception shift register	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	ULBS4	Selection of the UARTA0 loopback function 0: Inputs the states of the RxDA0 pin of serial interface UARTA0 to the reception shift register 1: Loops back output from the transmission shift register to the reception shift register	R/W
5	ULBS5	Selection of the UARTA1 loopback function 0: Inputs the states of the RxDA1 pin of serial interface UARTA1 to the reception shift register 1: Loops back output from the transmission shift register to the reception shift register	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W

The ULBS register is used to enable the UART loopback function. This register has bits to individually control UART channels. When the bit corresponding to each channel is set to 1, the UART loopback function is selected, and output from the transmission shift register is looped back to the reception shift register.

25.3 Operation

UARTAn operates in the following two modes:

- Operation stop mode
- UART mode

25.3.1 Operation Stop Mode

In the operation stop mode, serial communication is not performed, and thus the power consumption can be reduced. In addition, in this mode, the pins can be used as ordinary port pins. To set the operation stop mode, clear the EX, TXEA, and RXEA bits of the ASIMAn0 register to 0.

25.3.2 UART Mode

In this mode, one byte of data is transmitted and one byte is received following the start bit. This means, operation is full duplex.

A dedicated UART baud rate generator is incorporated, so that communication can be executed at a wide range of baud rates.

(1) Communication procedure

Table 25.3 shows the step of communication procedure.

Table 25.3 Step of communication procedure

Step		Process	Detail
Step of communication procedure	<1>	Baud rate setting	Set the BRGCAn register.
	<2>	Operation mode setting 1	Set the ALV, DIR, SL, CL[1:0], and PS[1:0] bits of the ASIMAn1 register.
	<3>	Operation mode setting 2	Set the ISSMA and ISRMA bits of the ASIMAn0 register.
	<4>	Enable operation	Set the EN bit of the ASIMAn0 register to 1.
	<5>	Enable communication	Set the TXEA bit of the ASIMAn0 register to 1 to enable transmission. Set the RXEA bit of the ASIMAn0 register to 1 to enable reception.
	<6>	Write transmit data	Write transmit data to the TXBAn register.
	<7>	Start of transmission	—

Note: When using the receiving function, set the port pin allocated for reception to input mode by using the port control registers. When using the transmitting function, set the port pin allocated for transmission to output mode by using the port control registers.
Note: n: Unit number (n = 0, 1)

For information on how to set up the I/O ports, see the descriptions given in section 16, I/O Ports.

(2) Format and waveform example of transmit and receive data

The following describes the communication data format of UARTAn.

Figure 25.2 shows the data format.

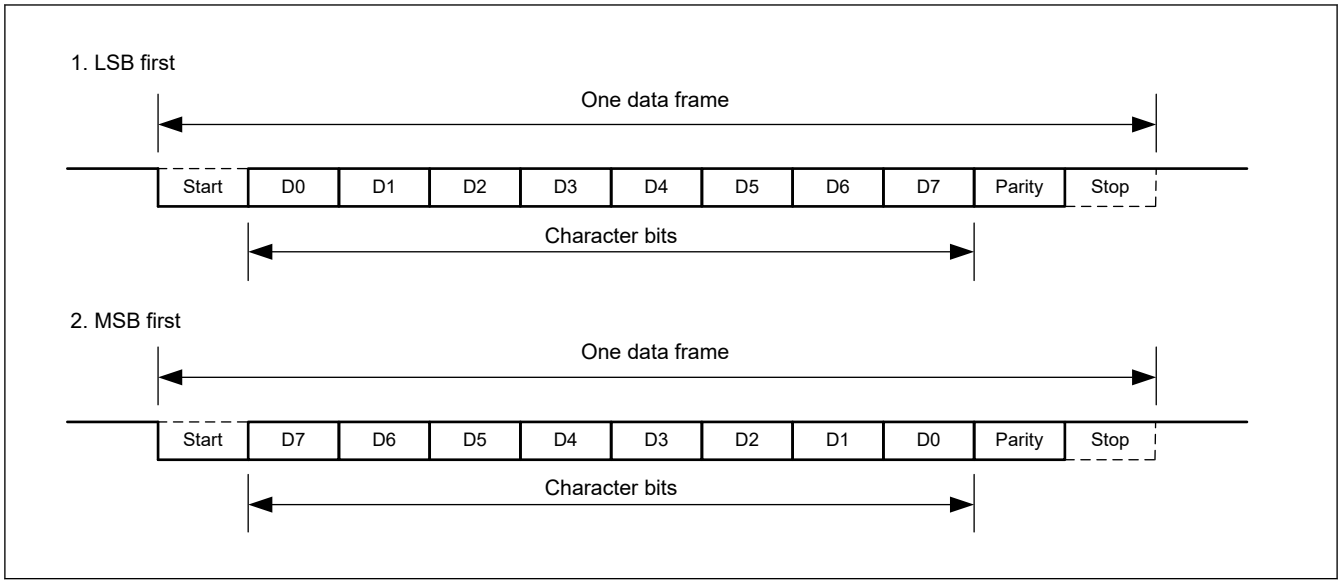


Figure 25.2 Transmit and receive data format

One data frame consists of the following bits.

- Start bit: 1 bit
- Character bits: 5, 7 or 8 bits
- Parity bit: Even parity, odd parity, 0 parity, or no parity
- Stop bit: 1 or 2 bits

The character bit length, the parity, the stop bit length, the transfer direction (LSB or MSB first), and the TxDAn pin output (direct or inverted) in one data frame are specified by the ASIMAn1 register.

Note: n: Unit number (n = 0, 1)

Figure 25.3 shows the examples of transmit and receive data waveforms.

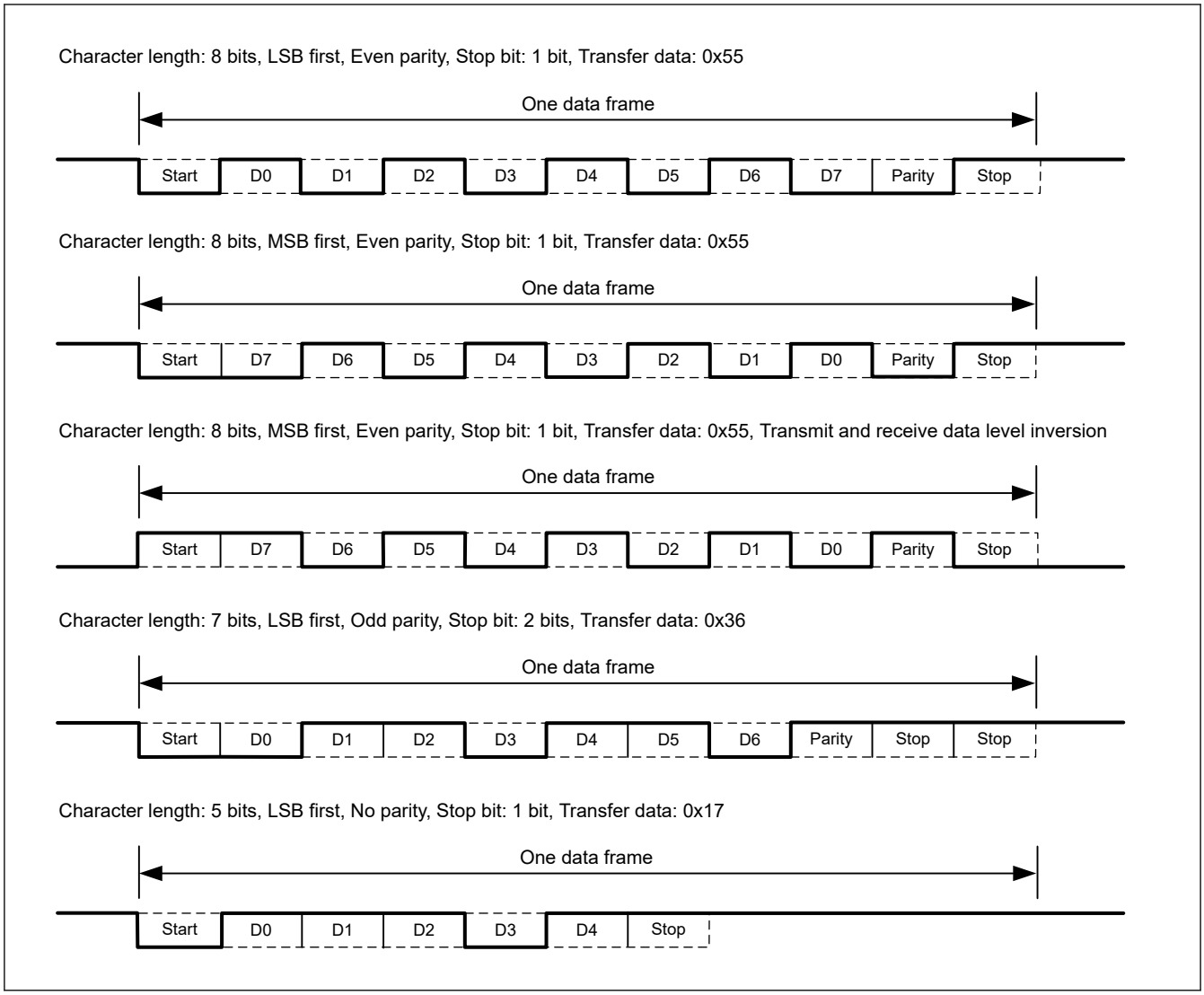


Figure 25.3 Example of transmit and receive data waveform

(3) Parity types and operation

The parity bit is used to detect a bit error in communication data. Usually, the same type of parity bit is used on both the transmitting and reception sides. With even and odd parity, a 1-bit (odd number) error can be detected. With zero and no parity, an error cannot be detected.

(a) Even parity

- In transmission
  Data for transmission, including the parity bit, are controlled so that an even number of bits have the value 1. The value of the parity bit is set as follows.
  If the data for transmission have an odd number of bits with the value 1:1
  If the data for transmission have an even number of bits with the value 1:0
- In reception
  In the data for reception, including the parity bit, the number of bits with the value 1, is counted. If it is odd, a parity error occurs.

(b) Odd parity

- In transmission
  Unlike even parity, data for transmission, including the parity bit, are controlled so that an odd number of bits have the value 1.
  If the data for transmission have an odd number of bits with the value 1:0
  If the data for transmission have an even number of bits with the value 1:1

- In reception
  In the data for reception, including the parity bit, the number of bits with the value 1, is counted. If it is even, a parity error occurs.

(c) 0 parity

The parity bit is cleared to 0 when data is transmitted, regardless of the transmit data.

The parity bit is not detected when the data is received. Therefore, a parity error does not occur regardless of whether the parity bit is 0 or 1.

(d) No parity

No parity bit is appended to the transmit data.

Reception is performed assuming that there is no parity bit. A parity error does not occur, because there is no parity bit.

(4) Normal transmission

Transmission is enabled by setting the EN bit of the operation mode setting register 0 (ASIMAn0) to 1 and then setting the TXEA bit of ASIMAn0 to 1. Transmission can be started by writing the data for transmission to the transmission buffer register (TXBAn). The start bit, parity bit, and stop bit are automatically appended to the data. When transmission is started, the data in the TXBAn register are transferred to the transmit shift register. After that, the transmit data are sequentially output from the transmit shift register to the TxDAn pin in the specified transfer direction. When transmission is completed, the parity and stop bits which are set by the ASIMAn0 register are appended and a transfer completion interrupt request signal (UARTA_TX_ENDIn) is generated.

Transmission is suspended until the next transmit data is written to the TXBAn register.

Figure 25.4 shows the timing of the transfer completion interrupt request signal (UARTA_TX_ENDIn).
UARTA_TX_ENDIn is issued at the following timing.

a. When ASIMAn0.ISSMA = 0 (UARTA_TX_ENDIn functions as a transfer completion interrupt.)
   UARTA_TX_ENDIn is issued after the output of the last stop bit.
b. ASIMAn0.ISSMA = 1 (UARTA_TX_ENDIn functions as a buffer empty interrupt.)
   UARTA_TX_ENDIn is issued when the start bit is output.

Note: n: Unit number (n = 0, 1)

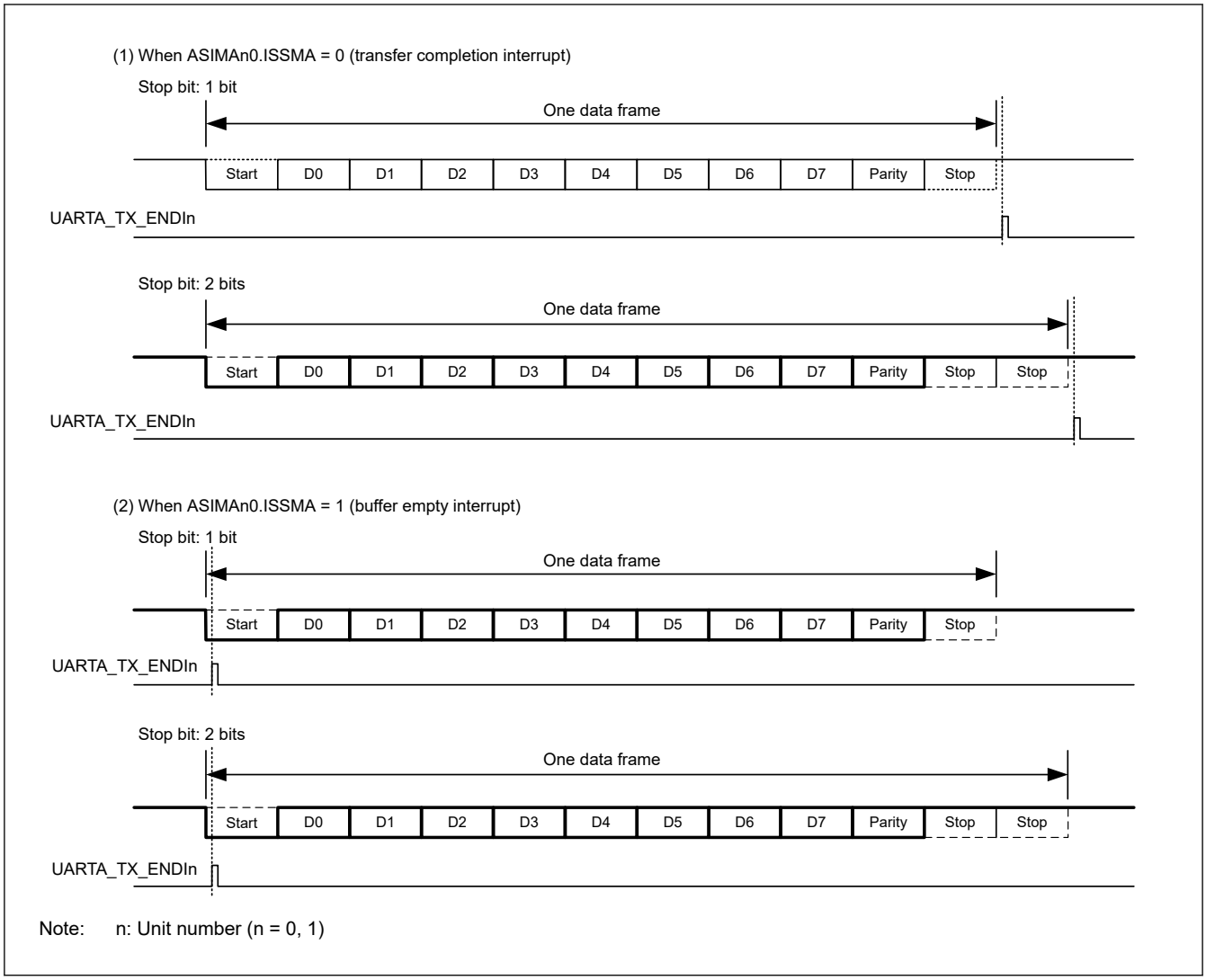


Figure 25.4 Interrupt output timing

(5) Continuous transmission

UARTAn has two separate registers for continuous transmission: the transmit buffer register (TXBAn) and the transmit shift register.

At the moment the transmit shift register starts a shift operation, the next transmit data can be written to the transmit buffer register (TXBAn). This operation enables continuous transmission, thereby improving communication rate.

Note that continuous transmission is not achieved when writing to the TXBAn register is not completed within the maximum number of clock cycles defined below from generation of the buffer empty interrupt.

Maximum number of clock cycles = Data transfer length × 2k − (2k + 3)

k: the value set with the BRGCAn register (k = 2, 3, 4, 5, 6, …, 255)

An example of calculating the maximum number of clock cycles is described below. When the BRGCAn register = 0x02 (k = 2),

start bit = 1 bit, character length = 8 bits, parity used, and stop bit = 1 bit:

The maximum number of clock cycles = Transfer length × 2k − (2k + 3) = 11 × 2 × 2 − (2 × 2 + 3) = 37

(Writing must be completed within 37 cycles of the UARTAn operating clock (f_{UTAn}).)

Continuous transmission is achieved by the following two methods.

(a) Continuous transmission by polling

Continuous transmission is achieved by polling the TXBFA and TXSFA flags of the status register (ASISAn).

When using this method, clear the ISSMA bit of the operation mode setting register 0 (ASIMAn0) to 0.

At the start of and during continuous transmission

At the start of continuous transmission, write the first byte of data to the TXBAn register, check that the transmit buffer data flag (ASISAn.TXBFA) is 0, and then write the second byte of data. In a similar way, check that the ASISAn.TXBFA flag is 0 and then write the subsequent data to the TXBAn register.

Table 25.4 shows the determination flag indicating that writing to the TXBAn register is enabled or disabled at the start of continuous transmission.

Table 25.4 Determination flag indicating that writing to the TXBAn register is enabled or disabled at the start of continuous transmission

ASISAn.TXBFA	Description
0	Writing is enabled.
1	Writing is disabled.

Note: To determine if continuous transmission is enabled or disabled, only check the ASISAn.TXBFA flag. The ASISAn.TXSFA flag must not be used for the determination in combination with this flag.
Note: n: Unit number (n = 0, 1)

Completion of continuous transmission

In continuous transmission, when data in the transmit shift register and the TXBAn register are transmitted after the required number of transmit data are written to the TXBAn register, the continuous transmission is completed. To confirm the completion, check the setting of the transmit shift register data flag (ASISAn.TXSFA).

Table 25.5 shows the confirmation flag indicating whether transmission is in progress or not.

Table 25.5 Confirmation flag indicating whether transmission is in progress or not

ASISAn.TXSFA	Description
0	Transmission is completed.
1	Transmission is in progress.

Note: When initializing the transmission unit after completion of continuous transmission, check that the ASISAn.TXSFA flag is 0 after the transfer completion interrupt is generated, and then initialize the unit.
Note: During continuous transmission, after transmission of one data frame, the subsequent transmission may be completed before execution of the UARTA_TX_ENDIn interrupt processing.
This can be detected by incorporating the program that counts the number of transmit data and by referencing the ASISAn.TXSFA flag.
Note: n: Unit number (n = 0, 1)

Table 25.6 shows a step example of continuous transmission processing by polling.

Table 25.6 Step example of continuous transmission processing by polling

Step		Process	Detail
Step example of continuous transmission processing by polling	<1>	Set registers	ASIMAn0.ISSMA = 0
	<2>	Check if the required number of transmit data are written to the TXBAn register. If yes, go to <5> If no, go to <3>	—
	<3>	Wait until the ASISAn.TXBFA flag is cleared.	Data is transferred to the transmit shift register.
	<4>	Write transmit data to the TXBAn register. Go to <2>	—
	<5>	Wait until the ASISAn.TXSFA flag is cleared.	Data is transferred from the transmit shift register and then no subsequent data is transferred from the TXBAn register.
	<6>	End of transmission processing	—

Note: n: Unit number (n = 0, 1)

(b) Continuous transmission by using an interrupt

Continuous transmission is achieved by using the interrupt (UARTA_TX_ENDIn).

An interrupt can be generated when data in the transmit buffer register (TXBAn) are transferred to the transmit shift register by setting the ISSMA bit to 1 in the operation mode setting register 0 (ASIMAn0).

With this setting, continuous transmission is enabled by writing data to the TXBAn register on occurrence of the buffer empty interrupt.

In addition, the transfer completion interrupt can be generated on completion of continuous transmission by clearing the ISSMA bit to 0 after writing the last transmit data to the TXBAn register.

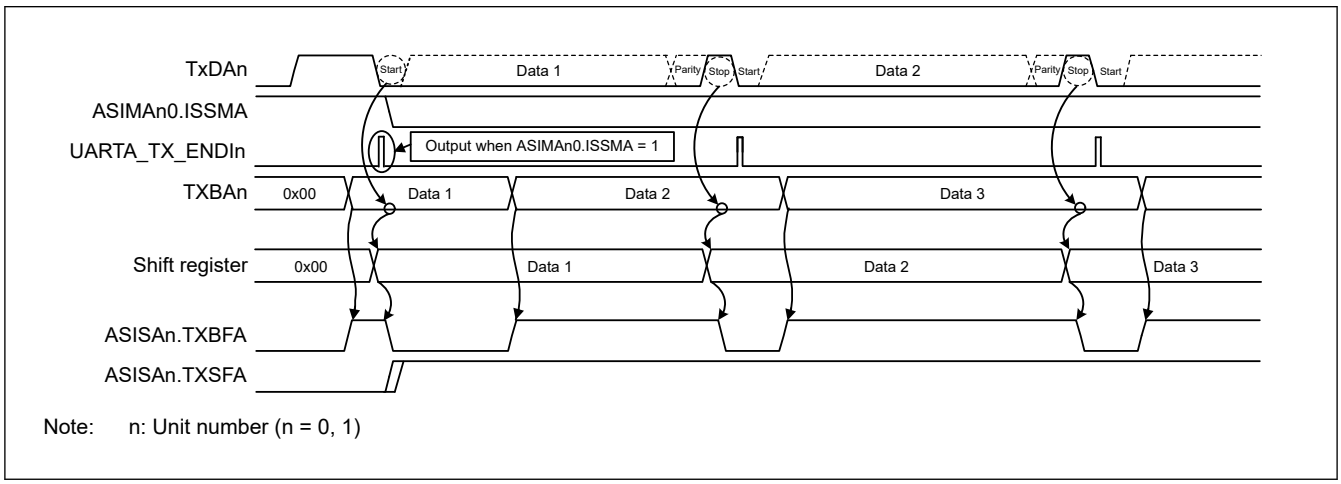
Table 25.7 shows a step example of continuous transmission using interrupt.

Table 25.7 Step example of continuous transmission using interrupt

Step		Process	Detail
Step example of continuous transmission using interrupt	<1>	Set registers	ASIMAn0.ISSMA = 1
	<2>	Write to the TXBAn register	—
	<3>	Waiting for UARTA_TX_ENDIn	Buffer empty interrupt
	<4>	Generation of UARTA_TX_ENDIn	—
	<5>	Check if the required number of transmit data are written to the TXBAn register. If yes, go to step <6>. If no, go to step <2>.	—
	<6>	Set the ASIMAn0 register	ASIMAn0.ISSMA = 0
	<7>	Waiting for UARTA_TX_ENDIn	Transfer completion interrupt
	<8>	Generation of UARTA_TX_ENDIn	—
	<9>	End of transmission processing	—

Note: n: Unit number (n = 0, 1)

Figure 25.5 and Figure 25.6 show the timing when continuous transmission is started and completed, respectively.



Note: n: Unit number (n = 0, 1)

Figure 25.5 Timing when continuous transmission is started

Note: When the ASISAn register is read, both the ASISAn.TXBFA and TXSFA flags are read as 1 within this period. Accordingly, use only the ASISAn.TXBFA flag to determine if writing is enabled or disabled.

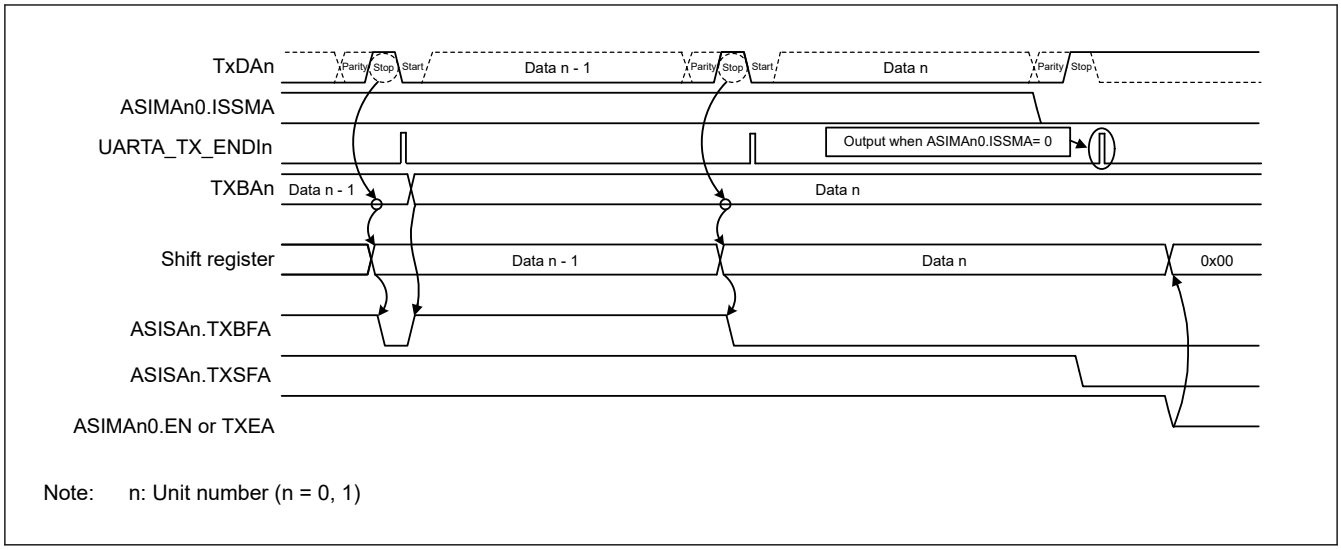


Figure 25.6 Timing when continuous transmission is completed

(6) Normal reception

When setting the EN bit of the operation mode setting register 0 (ASIMAn0) to 1 and then setting the RXEA bit of the ASIMAn0 register to 1, reception is enabled, and sampling of the input to the RxDAn pin is performed.

When the ASIMAn1.ALV bit is 0, the 8-bit counter of the baud rate generator starts counting on detection of the falling edge on the RxDAn pin. When the counter reaches the set value of the baud rate generator control register (BRGCAn), the input to the RxDAn pin is sampled again (at the point indicated with ∇ in Figure 25.7). If the RxDAn pin is low, it is regarded as a start bit.

When the ASIMAn1.ALV bit is 1, the 8-bit counter of the baud rate generator starts counting on detection of the rising edge on the RxDAn pin. When the counter reaches the set value of the baud rate generator control register (BRGCAn), the input to the RxDAn pin is sampled again (at the point indicated with ∇ in Figure 25.7). If the RxDAn pin is high, it is regarded as a start bit.

Figure 25.7 shows the timing chart of receive operation.

On detection of a start bit, receive operation is started: serial data is sequentially stored in the receive shift register at a specified baud rate. On reception of a stop bit, the transfer completion interrupt (UARTA_RX_ENDIn) is generated, and at the same time, the data in the receive shift register is written to the receive buffer register (RXBAn).

Note that when an overrun error (ASISAn.OVEA) occurs, the data received on occurrence of the error is not written to the RXBAn register.

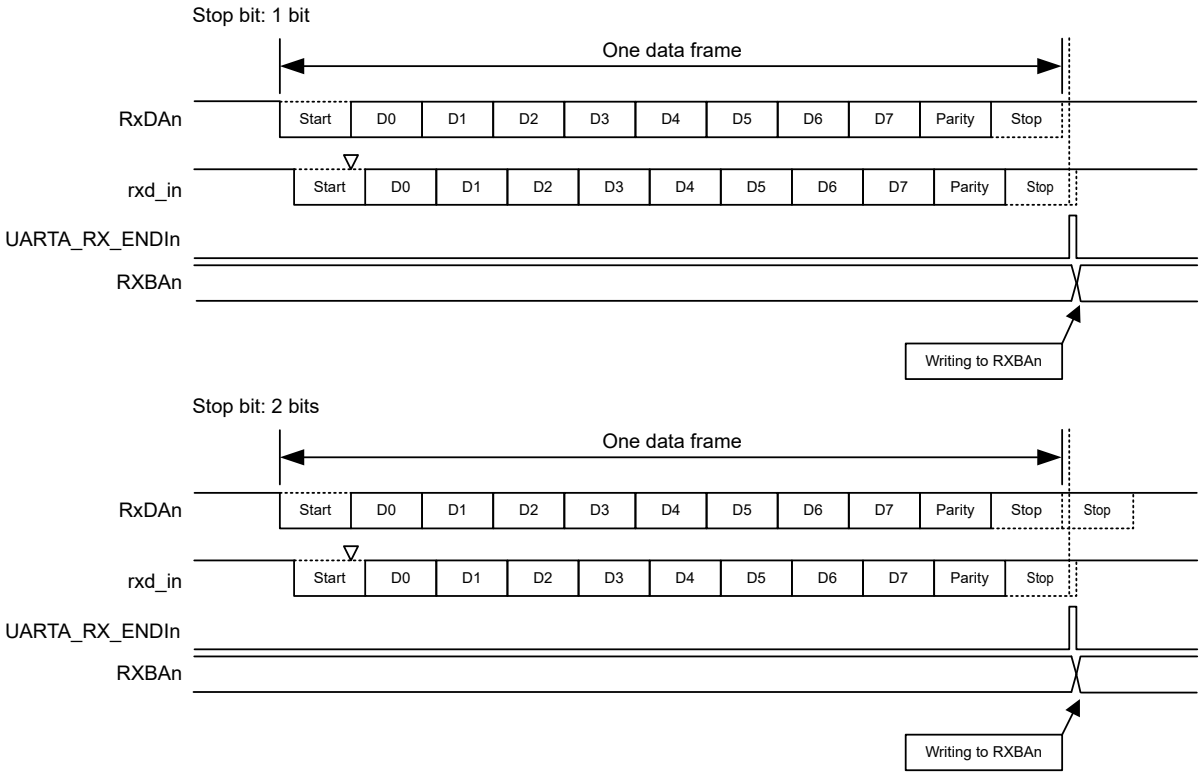
When a parity error (ASISAn.PEA) or a framing error (ASISAn.FEA) occurs during reception, reception continues until a stop bit is received. After completion of the reception, the reception error interrupt (UARTA_RX_ENDIn or UARTA_RX_ERIn) set in the ASIMAn0.ISRMA bit is generated.

When a reception error occurs, read the status register (ASISAn) and then read the receive buffer register (RXBAn) to clear the error flag.

If the receive buffer register (RXBAn) is not read, an overrun error will occur when the next data is received: the reception error state will continue.

Reception is always handled as including a stop bit. Accordingly, the second stop bit is ignored.

Note: n: Unit number (n = 0, 1)



Note: rxd_in: The internal signal generated by latching RxDAn with a noise filter
(rxd_in is delayed relative to RxDAn by maximum of 3 cycles of the UART operation clock.)
Note: The UARTA_RX_ENDIn output timing in the figure is just an example.
The timing relative to RxDAn varies according to the setting of the BRGCAn register.
Note: n: Unit number (n = 0, 1)

Figure 25.7 Timing of UART receive operation

(7) Reception error

Three types of errors may occur during reception; parity error, framing error, and overrun error.

When these errors occur, the corresponding error flag in the status register (ASISAn) is set, and the reception error interrupt request signal (UARTA_RX_ENDIn or UARTA_RX_ERIn) is generated.

The type of the reception error can be identified by the reception error interrupt processing routine, which reads and checks the contents of the status register (ASISAn).

The contents of the ASISAn register is cleared to 0 by setting the corresponding bit of the status clear trigger register (ASCTAn) to 1.

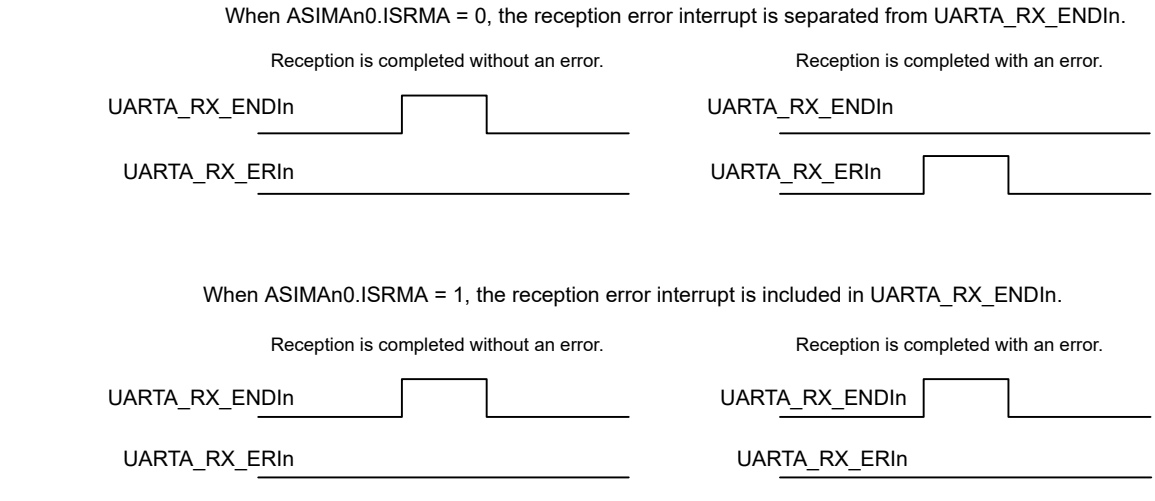
Table 25.8 shows the causes of the reception errors.

Table 25.8 Causes of reception errors

Error flag	Reception error	Cause
ASISAn.PEA	Parity error	The parity specified for reception does not match the parity of receive data.
ASISAn.FEA	Framing error	No stop bit is detected.
ASISAn.OVEA	Overrun error	Before the receive data is read from the receive buffer, the next data reception is completed.

Setting the ISRMA bit of the operation mode setting register 0 (ASIMAn0) to 0 allows the reception error interrupt to be separated from UARTA_RX_ENDIn and allows it to be generated as UARTA_RX_ERIn.

Figure 25.8 shows the interrupt output waveform which varies depending on the setting of the ASIMAn0.ISRMA bit.



Note: n: Unit number (n = 0, 1)

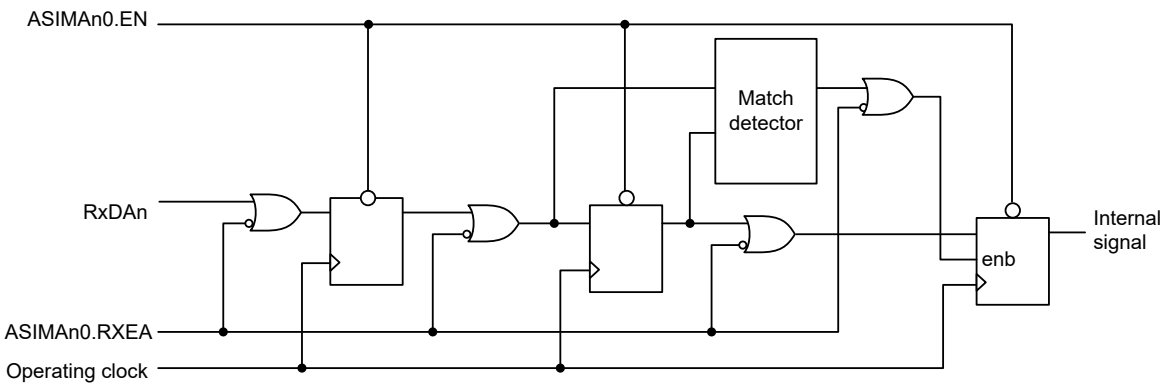
Figure 25.8 Various interrupt output waveforms depending on ASIMAn0.ISRMA setting

25.3.3 Receive Data Noise Filter

This filter samples the receive data (RxDAn), and determines the level when the same level is sampled twice.

The receive data is delayed by a maximum of 3 cycles of the operating clock because of the circuit configuration.

Figure 25.9 shows the noise filter circuit.



Note: When ASIMAn1.ALV = 0 (wait state = high level; start bit = low level), the initial value of the receive data (RxDAn) must be high.

Note: When ASIMAn1.ALV = 1 (wait state = low level; start bit = high level), the initial value of the receive data (RxDAn) must be low.

Note: n: Unit number (n = 0, 1)

Figure 25.9 Noise filter

25.3.4 Baud Rate Generator

The baud rate generator consists of 8-bit programmable counters, and generates a serial clock for transmission and reception of UARTAn.

An 8-bit counter is provided each for transmission and reception.

(1) Configuration of baud rate generator

(a) UARTAn operation clock

When ASIMAn0.EN = 1, the UARTAn operation clock (f_{UTAn}) is supplied to each module. When ASIMAn0.EN = 0, the UARTAn operation clock is fixed to low level.

(b) Transmission counter

This counter is cleared to 0 and stops when ASIMAn0.EN = 0 or ASIMAn0.TXEA = 0. It starts counting when ASIMAn0.EN = 1 and ASIMAn0.TXEA = 1.

The counter is cleared to 0 when the first transmit data is written to the transmit buffer register (TXBAn).

When continuous transmission is performed, the counter is cleared to 0 again when transmission of one frame of data has been completed. If there is no data to be transmitted next, the counter is not cleared to 0 and continues counting until the ASIMAn0.EN or TXEA bit is cleared to 0. When ASIMAn0.EN = 0 or ASIMAn0.TXEA = 0, the counter stops at 0x00.

(c) Reception counter

This counter is cleared to 0 and stops when ASIMAn0.EN = 0 or ASIMAn0.RXEA = 0. It starts counting when the start bit is detected.

The counter stops operation after one frame has been received, until the next start bit is detected. When ASIMAn0.EN = 0 or ASIMAn0.RXEA = 0, the counter stops at 0x00.

Note: n: Unit number (n = 0, 1)

Figure 25.10 shows the configuration of the baud rate generator.

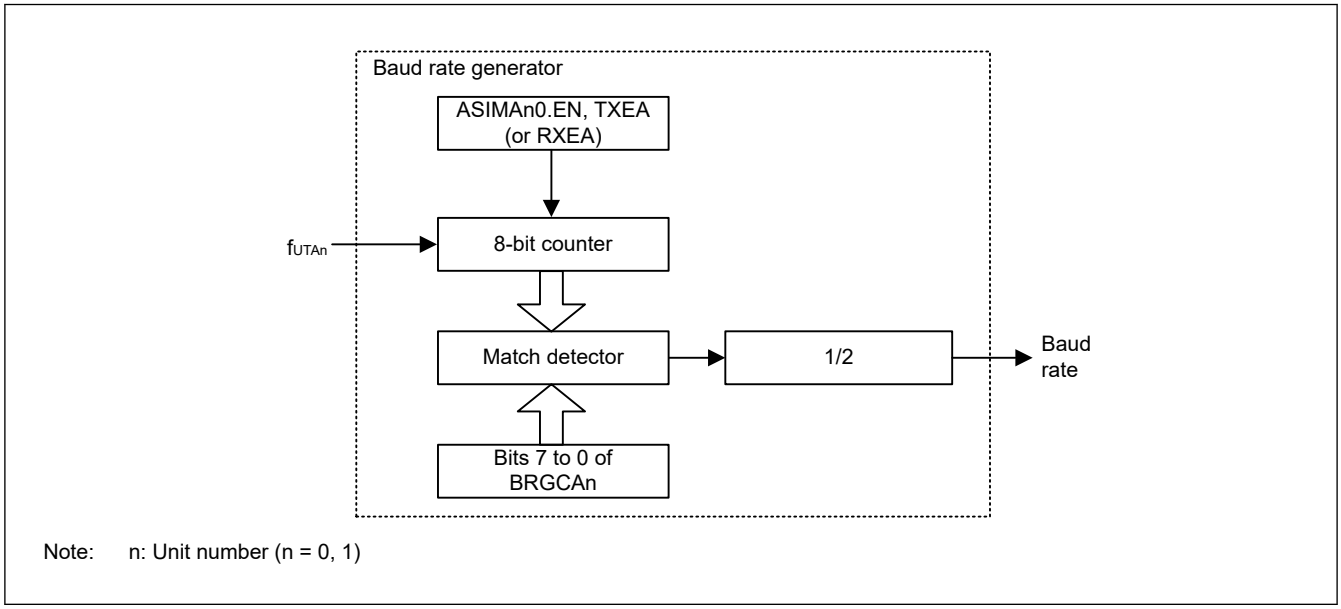


Figure 25.10 Configuration of baud rate generator

(2) Generation of serial clock

A serial clock to be generated can be specified by using the baud rate generator control register (BRGCAn).

The baud rate generator divides the frequency of the input clock signal to the 8-bit counter (f_{UTAn}) by the divisor set by the BRGCAn register. The result of this division is further divided by 2 to produce the serial clock.

(3) Baud rate calculation

(a) Baud rate calculation expression

The baud rate can be calculated by the following expression.

Baud rate = $f_{UTAn} \div (2 \times k)$ [bps]

f_{UTAn}: Frequency of operating clock

k: Value set by bits 7 to 0 of the BRGCAn register (k = 2, 3, 4, …, 255)

(b) Baud rate error

The baud rate error can be calculated by the following expression.

$$\text{Error} = \left[\frac{\text{Actual baud rate (baud rate with error)}}{\text{Desired baud rate (correct baud rate)}} - 1\right] \times 100[\%]$$

Note: Keep the baud rate error during transmission to within the permissible error range on the reception side.

Note: Make sure that the baud rate error during reception satisfies the permissible baud rate error range during reception. Permissible baud rate error during reception is described in (d) Permissible baud rate range during reception.

Note: n: Unit number (n = 0, 1)

(c) Baud rate setting example

Table 25.9 to Table 25.12 show the set data of the baud rate generator.

Table 25.9 Set data of baud rate generator (1/4)

Desired baud rate	In operation with UARTAHCLK = 32 MHz (UTA0CK.SEL[1:0] = 10b)													
	No division		×1/2		×1/4		×1/8		×1/16		×1/32		×1/64	
	(UTAnCK.CK[3:0] = 0x0)		(UTAnCK.CK[3:0] = 0x1)		(UTAnCK.CK[3:0] = 0x2)		(UTAnCK.CK[3:0] = 0x3)		(UTAnCK.CK[3:0] = 0x4)		(UTAnCK.CK[3:0] = 0x5)		(UTAnCK.CK[3:0] = 0x6)	
	k	Error from the desired baud rate	k	Error from the desired baud rate	k	Error from the desired baud rate	k	Error from the desired baud rate	k	Error from the desired baud rate	k	Error from the desired baud rate	k	Error from the desired baud rate
200 bps	Disabled		Disabled		Disabled		Disabled		Disabled		Disabled		Disabled	
300 bps	Disabled		Disabled		Disabled		Disabled		Disabled		Disabled		Disabled	
600 bps	Disabled		Disabled		Disabled		Disabled		Disabled		Disabled		Disabled	
1200 bps	Disabled		Disabled		Disabled		Disabled		Disabled		Disabled		208	0.16%
2400 bps	Disabled		Disabled		Disabled		Disabled		Disabled		208	0.16%	104	0.16%
4800 bps	Disabled		Disabled		Disabled		Disabled		208	0.16%	104	0.16%	52	0.16%
9600 bps	Disabled		Disabled		Disabled		208	0.16%	104	0.16%	52	0.16%	26	0.16%
19200 bps	Disabled		Disabled		208	0.16%	104	0.16%	52	0.16%	26	0.16%	13	0.16%
38400 bps	Disabled		208	0.16%	104	0.16%	52	0.16%	26	0.16%	13	0.16%	Disabled	
76800 bps	208	0.16%	104	0.16%	52	0.16%	26	0.16%	13	0.16%	Disabled		Disabled	
115200 bps	139	−0.08%	69	0.64%	35	−0.79%	17	2.12%	Disabled		Disabled		Disabled	
153600 bps	104	0.16%	52	0.16%	26	0.16%	13	0.16%	Disabled		Disabled		Disabled	

Table 25.10 Set data of baud rate generator (2/4) (1 of 2)

Desired baud rate	In operation with UARTAMOCLK = 4 MHz (UTA0CK.SEL[1:0] = 11b)													
	No division		×1/2		×1/4		×1/8		×1/16		×1/32		×1/64	
	(UTAnCK.CK[3:0] = 0x0)		(UTAnCK.CK[3:0] = 0x1)		(UTAnCK.CK[3:0] = 0x2)		(UTAnCK.CK[3:0] = 0x3)		(UTAnCK.CK[3:0] = 0x4)		(UTAnCK.CK[3:0] = 0x5)		(UTAnCK.CK[3:0] = 0x6)	
	k	Error from the desired baud rate	k	Error from the desired baud rate	k	Error from the desired baud rate	k	Error from the desired baud rate	k	Error from the desired baud rate	k	Error from the desired baud rate	k	Error from the desired baud rate
200 bps	Disabled		Disabled		Disabled		Disabled		Disabled		Disabled		156	0.16%
300 bps	Disabled		Disabled		Disabled		Disabled		Disabled		208	0.16%	104	0.16%
600 bps	Disabled		Disabled		Disabled		Disabled		208	0.16%	104	0.16%	52	0.16%
1200 bps	Disabled		Disabled		Disabled		208	0.16%	104	0.16%	52	0.16%	26	0.16%
2400 bps	Disabled		Disabled		208	0.16%	104	0.16%	52	0.16%	26	0.16%	13	0.16%
4800 bps	Disabled		208	0.16%	104	0.16%	52	0.16%	26	0.16%	13	0.16%	Disabled	

Table 25.10 Set data of baud rate generator (2/4) (2 of 2)

Desired baud rate	In operation with UARTAMOCLK = 4 MHz (UTA0CK.SEL[1:0] = 11b)													
	No division (UTAnCK.CK[3:0] = 0x0)		×1/2 (UTAnCK.CK[3:0] = 0x1)		×1/4 (UTAnCK.CK[3:0] = 0x2)		×1/8 (UTAnCK.CK[3:0] = 0x3)		×1/16 (UTAnCK.CK[3:0] = 0x4)		×1/32 (UTAnCK.CK[3:0] = 0x5)		×1/64 (UTAnCK.CK[3:0] = 0x6)	
	k	Error from the desired baud rate	k	Error from the desired baud rate	k	Error from the desired baud rate	k	Error from the desired baud rate	k	Error from the desired baud rate	k	Error from the desired baud rate	k	Error from the desired baud rate
9600 bps	208	0.16%	104	0.16%	52	0.16%	26	0.16%	13	0.16%	Disabled		Disabled	
19200 bps	104	0.16%	52	0.16%	26	0.16%	13	0.16%	Disabled		Disabled		Disabled	
38400 bps	52	0.16%	26	0.16%	13	0.16%	Disabled		Disabled		Disabled		Disabled	
76800 bps	26	0.16%	13	0.16%	Disabled		Disabled		Disabled		Disabled		Disabled	
115200 bps	17	2.12%	Disabled		Disabled		Disabled		Disabled		Disabled		Disabled	
153600 bps	13	0.16%	Disabled		Disabled		Disabled		Disabled		Disabled		Disabled	

Table 25.11 Set data of baud rate generator (3/4)

Desired baud rate	In operation with UARTAMCLK = 20 MHz (UTA0CK.SEL[1:0] = 01b)													
	No division (UTAnCK.CK[3:0] = 0x0)		×1/2 (UTAnCK.CK[3:0] = 0x1)		×1/4 (UTAnCK.CK[3:0] = 0x2)		×1/8 (UTAnCK.CK[3:0] = 0x3)		×1/16 (UTAnCK.CK[3:0] = 0x4)		×1/32 (UTAnCK.CK[3:0] = 0x5)		×1/64 (UTAnCK.CK[3:0] = 0x6)	
	k	Error from the desired baud rate	k	Error from the desired baud rate	k	Error from the desired baud rate	k	Error from the desired baud rate	k	Error from the desired baud rate	k	Error from the desired baud rate	k	Error from the desired baud rate
200 bps	Disabled		Disabled		Disabled		Disabled		Disabled		Disabled		Disabled	
300 bps	Disabled		Disabled		Disabled		Disabled		Disabled		Disabled		Disabled	
600 bps	Disabled		Disabled		Disabled		Disabled		Disabled		Disabled		255	2.12%
1200 bps	Disabled		Disabled		Disabled		Disabled		Disabled		255	2.12%	130	0.16%
2400 bps	Disabled		Disabled		Disabled		Disabled		255	2.12%	130	0.16%	65	0.16%
4800 bps	Disabled		Disabled		Disabled		255	2.12%	130	0.16%	65	0.16%	33	−1.36%
9600 bps	Disabled		Disabled		255	2.12%	130	0.16%	65	0.16%	33	−1.36%	16	1.73%
19200 bps	Disabled		255	2.12%	130	0.16%	65	0.16%	33	−1.36%	16	1.73%	8	1.73%
38400 bps	255	2.12%	130	0.16%	65	0.16%	33	−1.36%	16	1.73%	8	1.73%	4	1.73%
76800 bps	130	0.16%	65	0.16%	33	−1.36%	16	1.73%	8	1.73%	4	1.73%	Disabled	
115200 bps	87	−0.22%	43	0.94%	22	−1.36%	11	−1.36%	Disabled		Disabled		Disabled	
153600 bps	65	0.16%	33	−1.36%	16	1.73%	8	1.73%	4	1.73%	Disabled		Disabled	

Table 25.12 Set data of baud rate generator (4/4) (1 of 2)

Desired baud rate	In operation with UARTALCLK/UARTASCLK = 32.768 kHz (UTAnCK.CK[3:0] = 0x8)	
	k	Error from the desired baud rate
200 bps	82	−0.10%
300 bps	55	−0.70%
600 bps	27	−1.14%
1200 bps	14	−2.48%

Table 25.12 Set data of baud rate generator (4/4) (2 of 2)

Desired baud rate	In operation with UARTALCLK/UARTASCLK = 32.768 kHz (UTAnCK.CK[3:0] = 0x8)	
	k	Error from the desired baud rate
2400 bps	7	−2.48%
4800 bps	Disabled	
9600 bps	Disabled	
19200 bps	Disabled	
38400 bps	Disabled	
76800 bps	Disabled	
115200 bps	Disabled	
153600 bps	Disabled	

Note:
- UARTAMCLK: UARTA external clock
- UARTASCLK: UARTA sub clock
- UARTAHCLK: UARTA HOCO clock
- UARTAMOCLK: UARTA MOCO clock
- UARTALCLK: UARTA LOCO clock

Note: k: Value set by bits 7 to 0 of the baud rate generator control register (BRGCAn) (k = 2, 3, 4, …, 255)
n: Unit number (n = 0, 1)

(d) Permissible baud rate range during reception

Figure 25.11 shows the permissible error from the baud rate on the transmitting side during reception.

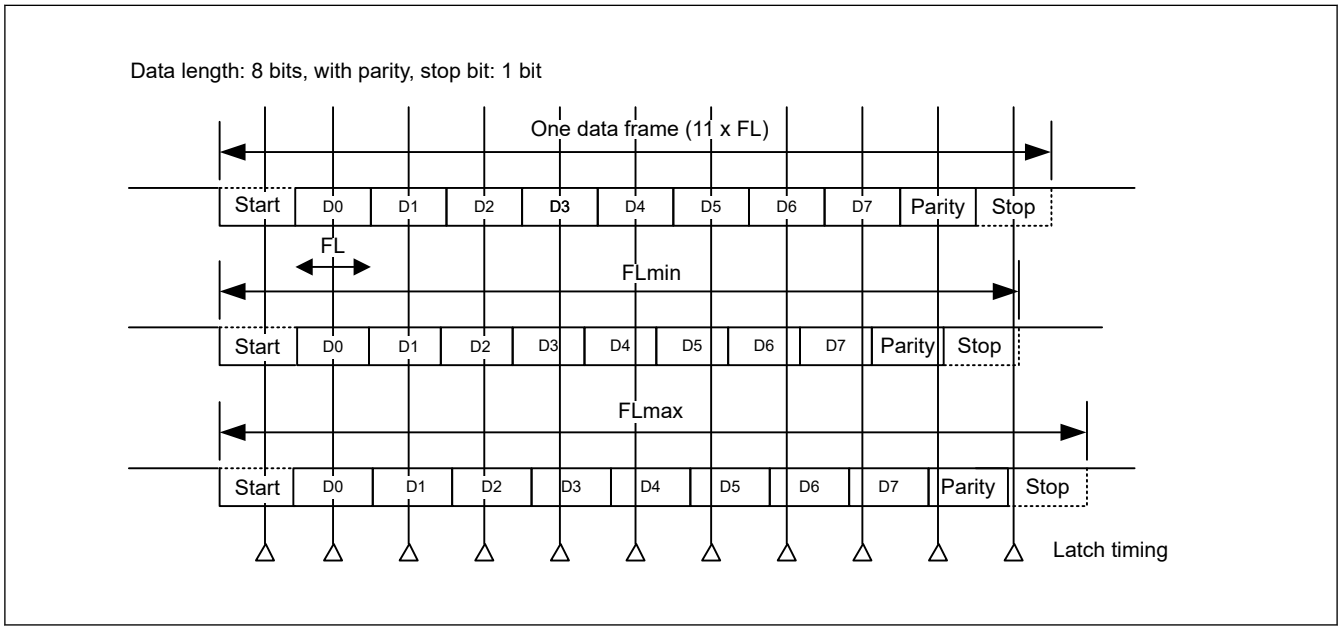


Figure 25.11 Permissible baud rate range during reception

Note: Be sure to make settings so that the baud rate error during reception is within the permissible error range. Use the calculation expression below to check if the error is within the permissible range.

After the start bit is detected, the latch timing of receive data is determined by the counter specified with the baud rate generator control register (BRGCAn). If the whole frame including the stop bit has been received before this latching, reception can proceed correctly.

Assuming that 11 bits of data are received, the theoretical values can be calculated as follows.

- The relation between 1-bit data length and baud rate
  FL = (Brate) - 1
  Brate: Baud rate of UART
  k: Set value of the BRGCAn register

FL: 1-bit data length
Margin of latch timing: 1 clock

- Minimum permissible data frame length (FLmin)
  $FLmin = 11 \times FL - \frac{k-1}{2k} \times FL = \frac{21k+1}{2k}FL$
- Maximum permissible baud rate for reception on the transmitting side (BRmax)
  $BRmax = (FLmin/11)^{-1} = \frac{22k}{21k+1}Brate$
- Maximum permissible data frame length (FLmax)
  $FLmax = \frac{21k+1}{20k}FL \times 11$
- Minimum permissible baud rate for reception on the transmitting side (BRmin)
  $BRmin = (FLmax/11)^{-1} = \frac{20k}{21k-1}Brate$

Table 25.13 shows the permissible baud rate error between UART and the transmitting side that can be calculated from the above minimum and maximum baud rate expressions.

Table 25.13 Maximum and minimum permissible baud rate error

Division ratio (k)	Maximum permissible baud rate error	Minimum permissible baud rate error
2	+2.32%	−2.43%
4	+3.52%	−3.61%
8	+4.14%	−4.19%
20	+4.51%	−4.53%
50	+4.66%	−4.67%
100	+4.71%	−4.71%
255	+4.74%	−4.74%

Note: The permissible error of reception depends on the number of bits in one frame, input clock frequency, and division ratio (k). The higher the input clock frequency and the division ratio (k), the higher the permissible error.
Note: k: Set value of the BRGCAn register
Note: n: Unit number (n = 0, 1)

25.4 Usage Notes

25.4.1 Port Setting for RxDAn Pin

When ASIMAn1.ALV = 0 (wait state = high level, start bit = low level), the initial value of receive data (RxDAn) must be high. When ASIMAn1.ALV = 1 (wait state = low level, start bit = high level), the initial value of receive data (RxDAn) must be low. Accordingly, port setting is required for the RxDAn pin before setting ASIMAn0.EN = 1.

25.4.2 Point for Caution when Selecting the UARTAn Operation Clock (f_{UTAn})

When UARTAMOCLK is selected for f_{UTAn}, communication may not be executed correctly due to the oscillation frequency accuracy of the middle-speed on-chip oscillator. Adjust the accuracy, therefore, by using the MOCO user trimming control register (MOCOUTCR).

When UARTALCLK/UARTASCLK is selected for f_{UTAn} and UARTALCLK is selected for UARTALCLK/UARTASCLK, communication may not be executed correctly due to the oscillation frequency accuracy of the low-speed on-chip oscillator. Adjust the accuracy, therefore, by using the LOCO user trimming control register (LOCOUTCR).

26. Remote Control Signal Receiver (REMC)

26.1 Overview

The remote control signal receiver can receive data by checking the width and period of an external pulse input signal. Table 26.1 lists the specifications of the remote control signal receiver. Figure 26.1 shows a block diagram of the remote control signal receiver.

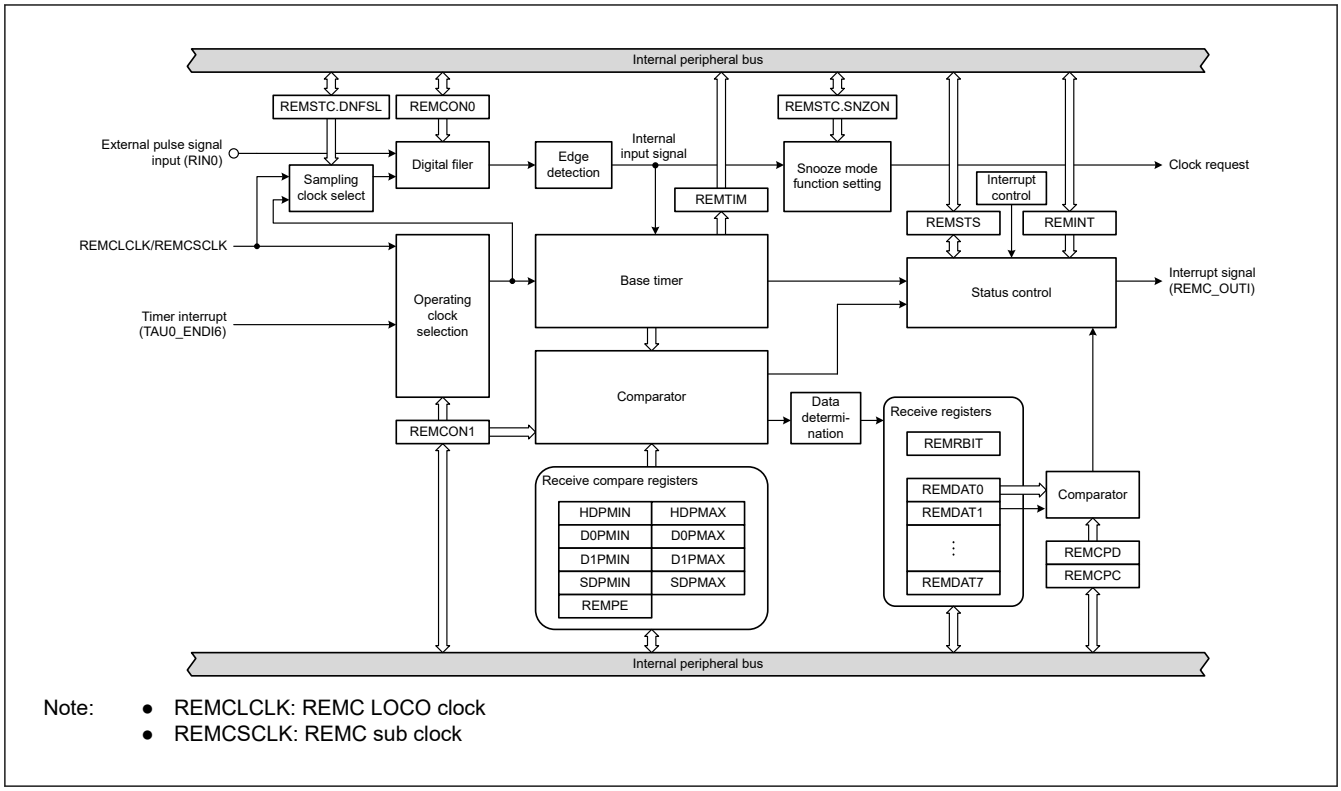
Table 26.1 Specifications of remote control signal receiver

Item	Description
Number of provided units	1
External pulse input	RIN0
Operating clock	REMCLCLK/REMCSCLK Timer interrupt (TAU0_ENDI6)
Detection patterns	Header pattern Data 0 pattern Data 1 pattern Special data pattern
Receive buffer	8 bytes (64 bits)
Compare bit count	1 to 16 bits
Interrupt request signal	REMC_OUTI
Interrupt sources	Header pattern match*1 Compare match*1 Completion of data reception*1 Special data pattern match*1 Data 0 pattern or data 1 pattern match Receive buffer full Receive error
Selectable functions	Input signal inversion Digital filter (matching three or two times)*2 Pattern end setting
Low power consumption	The clock supply can be stopped by setting the MSTPCRC.MSTPC19 bit. Snooze mode can shift to normal operation mode by using the REMC interrupt request signal.

Note:
- REMCLCLK: REMC LOCO clock
- REMCSCLK: REMC sub clock

Note 1. The interrupt mode is selectable from either the normal interrupt mode or the sequential interrupt mode.
In the normal interrupt mode, the OR condition of multiple interrupt sources is applied.
In the sequential interrupt mode, the AND condition of multiple interrupt sources is applied.

Note 2. The sampling clock of the digital filter is an operating clock selected by the REMCON1.CSRC bit or REMCLCLK/REMCSCLK.
When transition to the Snooze mode is enabled, select REMCLCLK/REMCSCLK as the sampling clock of the digital filter.



Note:
- REMCLCLK: REMC LOCO clock
- REMCSCLK: REMC sub clock

Figure 26.1 Block diagram of remote control signal receiver

Table 26.2 shows the input pin used for the remote control signal receiver.

Table 26.2 Pin configuration of remote control signal receiver

Pin name	I/O	Function
RIN0	Input	External pulse signal input

26.2 Register Descriptions

26.2.1 REMCON0 : Function Select Register 0

Base address: REMC = 0x4009_2100

Offset address: 0x00

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	FILSEL	—	EC	INFLG	FIL	INV	ENFLG
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ENFLG*1	Remote Control Status Flag This flag can be used to confirm whether the remote control signal receiver is stopped or operating. This flag changes after 0 to 1 cycle of the operating clock when a value is written to the REMCON1.EN bit. 0: Stopped 1: Operating	R
1	INV*2	Input Signal Inversion 0: Not inverted 1: Inverted	R/W

Bit	Symbol	Function	R/W
2	FIL*2	Digital Filter Enable or Disable Setting 0: Disables the digital filter for matching three or two times 1: Enables the digital filter for matching three or two times	R/W
3	INFLG*1	Input Signal Flag This flag can be used to confirm the level of the internal input signal of the remote control signal receiver. The confirmed level is the result set by the INV and FIL bits. 0: The level of the internal input signal of the remote control signal receiver is low 1: The level of the internal input signal of the remote control signal receiver is high	R
4	EC*2	Receive Error Capture Operation Select This bit can be used to set capture operation to the REMRBIT and REMDATj registers (j = 0 to 7) after an error pattern is received. 0: Captures the data after an error pattern is received 1: Does not capture the data after an error pattern is received	R/W
5	—	This bit is read as 0. The write value should be 0.	R/W
6	FILSEL*2	Digital Filter Function Select 0: Digital filter for matching three times 1: Digital filter for matching two times	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

Note 1. These flags become 0 when the REMCON1.EN bit is set to 0.
Note 2. These bits can be rewritten when the REMCON1.EN bit and the REMCON0.ENFLG flag are both 0 (REMC is stopped).

The REMCON0 register has flags for indicating the operating status of the remote control signal receiver and the level of the input signal, and is used to control the input signal inversion, control the digital filter, and select the receive error capture operation.

Rewriting the REMCON0 register is prohibited when the REMC is in operation (when REMCON1.EN = 1).

26.2.2 REMCON1 : Function Select Register 1

Base address: REMC = 0x4009_2100

Offset address: 0x01

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	CSRC	—	INTMD	EN	TYP[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	TYP[1:0]*1	Receive Mode Select 0 0: Format A 0 1: Format B 1 0: Format C 1 1: Setting prohibited	R/W
2	EN	Remote Control 0: Operation disabled 1: Operation enabled	R/W
3	INTMD*2	Interrupt Mode Select 0: Normal interrupt mode (OR condition) 1: Sequential interrupt mode (AND condition)	R/W
4	—	This bit is read as 0. The write value should be 0.	R/W
5	CSRC*2	Operating Clock Select 0: REMCLCLK/REMCSCLK 1: Timer interrupt (TAU0_ENDI6) Use channel 6 of timer array unit in the interval timer mode	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W

Note: • REMCLCLK: REMC LOCO clock

- REMCSCLK: REMC sub clock

Note 1. To rewrite the TYP[1:0] bits when the REMCON1.EN bit or REMCON0.ENFLG flag is 1 (REMC is operating), change the values of these bits one bit at a time.
Note 2. These bits can be rewritten when the REMCON1.EN bit and the REMCON0.ENFLG flag are both 0 (REMC is stopped).

The REMCON1 register is used to set the operation conditions of the remote control signal receiver such as operating clock, reception mode, and interrupt mode.

TYP[1:0] bits (Receive Mode Select)

These bits can be used to select the format for capturing the remote control signal waveform.

For details of each format, see section 26.3.3. Pattern Setting.

EN bit (Remote Control)

This bit enables or disables REMC operation.

Use the REMCON0.ENFLG flag to confirm whether operation has started or not.

INTMD bit (Interrupt Mode Select)

This bit can be used to select the interrupt mode.

In normal interrupt mode, an interrupt is generated when the OR condition of the interrupt sources enabled (set to 1) in the interrupt control register (REMINT) is met.

In sequential interrupt mode, if all the interrupt generation conditions of the sources (compare match, data reception complete, header pattern match, and special data pattern match) whose interrupt enable bit of the REMINT register is set to 1 are met, an interrupt (REMC_OUTI) is generated.

CSRC bit (Operating Clock Select)

This bit selects the operating clock for the remote control signal receiver.

26.2.3 REMSTS : Status Register

Base address: REMC = 0x4009_2100

Offset address: 0x02

Bit position:	7	6	5	4	3	2	1	0
Bit field:	SDFLG	D1FLG	D0FLG	HDFLG	BFULFLG	DRFLG	REFLG	CPFLG
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CPFLG	Compare Match Flag 0: Mismatch 1: Match	R
1	REFLG	Receive Error Flag 0: No error has occurred 1: An error has occurred	R
2	DRFLG	Data Receiving Flag 0: Waiting for data reception 1: Data is being received	R
3	BFULFLG*1	Receive Buffer Full Flag 0: Receive buffer is empty 1: Receive buffer is full (64 bits received)	R/W
4	HDFLG	Header Pattern Match Flag 0: Mismatch 1: Match	R
5	D0FLG	Data 0 Pattern Match Flag 0: Mismatch 1: Match	R

Bit	Symbol	Function	R/W
6	D1FLG	Data 1 Pattern Match Flag 0: Mismatch 1: Match	R
7	SDFLG	Special Data Pattern Match Flag 0: Mismatch 1: Match	R

Note: If updating and reading data overlap, an undefined value may be read. For details on reading this register, see section 26.4.5. Reading Registers.
Note: This register becomes 0x00 when the REMCON1.EN bit is set to 0.
Note 1. The BFULFLG bit can only be written with 0 to clear the flag. However, if this flag is written when changing the REMCON0.INFLG flag, the value read from this flag may become undefined.

The REMSTS register is used to indicate the reception status and error occurrence status of the remote control signal receiver.

CPFLG flag (Compare Match Flag)

This flag indicates the comparison result between the value of the REMCPD register specified by the REMCPC.CPN[3:0] bits and the data to be stored in the REMDAT1 and REMDAT0 registers.

[Clearing conditions]

- When the DRFLG flag changes from 0 to 1 (next frame reception starts)
- When the HDFLG flag changes from 0 to 1

[Setting condition]

- When the value of the REMCPD register matches the value to be stored in the REMDAT1 and REMDAT0 registers (when the setting value of the REMCPC.CPN[3:0] bits is n, bits n to 0 in the REMCPD register match bits n to 0 in the REMDAT1 and REMDAT0 registers)

REFLG flag (Receive Error Flag)

This flag indicates that a receive error has occurred. The setting conditions differ depending on the value of the REMCON1.TYP[1:0] bits.

[Clearing conditions]

- The header pattern is detected
- When the DRFLG flag changes from 0 to 1 (next frame reception starts)

[Setting condition]

When the REMCON1.TYP[1:0] bits are 00b (format A):

- The data 0, data 1, or special data pattern is detected prior to receiving the header pattern.
- The width between a rising edge and the next rising edge of the input signal is not the header, data 0, data 1, or special data pattern (when the REMCON0.INV bit is 0).
- A conflict occurs between when data reception is completed (timing when the DRFLG flag changes from 1 to 0) and when the input signal level changes (new signal).

When the REMCON1.TYP[1:0] bits are 01b (format B):

- The data 0, data 1, or special data pattern is detected prior to receiving the header pattern.
- The width between a falling edge and the next falling edge of the input signal is not the data 0, data 1, or special data pattern (when the REMCON0.INV bit is 0).
- A conflict occurs between when data reception is completed (timing when the DRFLG flag changes from 1 to 0) and when the input signal level changes (new signal).

When the REMCON1.TYP[1:0] bits are 10b (format C):

- The width between a rising edge and the next rising edge of the input signal is not the header, data 0, data 1, or special data pattern (when the REMCON0.INV bit is 0).

- A conflict occurs between when data reception is completed (timing when the DRFLG flag changes from 1 to 0) and when the input signal level changes (new signal).

DRFLG flag (Data Receiving Flag)

This flag indicates the state of receiving the remote control signal.

[Clearing conditions]

- This flag becomes 0 after one cycle of the operating clock when the value of the base timer is greater than any value of the HDPMAX, D0PMAX, D1PMAX, SDPMAX, and REMPE registers.

[Setting condition]

- Rising edge of REMC internal input signal (when the REMCON0.INV bit is 0)

BFULFLG flag (Receive Buffer Full Flag)

[Clearing conditions]

- When the HDFLG flag changes from 0 to 1
- When the DRFLG flag changes from 0 to 1 (next frame reception starts)
- This flag becomes 0 after one to two cycles of the operating clock when 0 is written to the BFULFLG flag.

[Setting condition]

- When the value of the REMRBIT register becomes 64

HDFLG flag (Header Pattern Match Flag)

[Clearing conditions]

- When the DRFLG flag changes from 0 to 1 (next frame reception starts)
- When the REFLG flag changes from 0 to 1
- See Table 26.3.

[Setting condition]

- See Table 26.3.

D0FLG flag (Data 0 Pattern Match Flag)

[Clearing conditions]

- When the DRFLG flag changes from 0 to 1 (next frame reception starts)
- When the REFLG flag changes from 0 to 1
- See Table 26.3.

[Setting condition]

- See Table 26.3.

D1FLG flag (Data 1 Pattern Match Flag)

[Clearing conditions]

- When the DRFLG flag changes from 0 to 1 (next frame reception starts)
- When the REFLG flag changes from 0 to 1
- See Table 26.3.

[Setting condition]

- See Table 26.3.

SDFLG flag (Special Data Pattern Match Flag)

[Clearing conditions]

- When the DRFLG flag changes from 0 to 1 (next frame reception starts)
- When the REFLG flag changes from 0 to 1
- See Table 26.3.

[Setting condition]

- See Table 26.3.

Table 26.3 shows the measurement results and flags.

Table 26.3 Measurement results and flags

Comparison result between REMTIM register value (measurement result) and each register	Flag value			
	HDFLG	D0FLG	D1FLG	SDFLG
Between HDPMIN and HDPMAX	1	0	0	0
Between D0PMIN and D0PMAX	0	1*1	0	0
Between D1PMIN and D1PMAX	0	0	1*1	0
Between SDPMIN and SDPMAX	0	0	0	1*1
Values not listed above	0	0	0	0

Note 1. When the REMCON1.TYP[1:0] bits are 00b or 01b, the D0FLG, D1FLG, and SDFLG flags remain unchanged until the header pattern is detected.

26.2.4 REMINT : Interrupt Control Register

Base address: REMC = 0x4009_2100

Offset address: 0x03

Bit position:	7	6	5	4	3	2	1	0
Bit field:	SDINT	—	DINT*1	HDINT	BFULINT	DRINT	REINT	CPINT
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CPINT	Compare Match Interrupt Enable 0: Disabled 1: Enabled	R/W
1	REINT	Receive Error Interrupt Enable 0: Disabled 1: Enabled	R/W
2	DRINT*1	Data Receiving Interrupt Enable 0: Disabled 1: Enabled	R/W
3	BFULINT	Receive Buffer Full Interrupt Enable 0: Disabled 1: Enabled	R/W
4	HDINT	Header Pattern Match Interrupt Enable 0: Disabled 1: Enabled	R/W
5	DINT*1	Data 0 Pattern or Data 1 Pattern Match Interrupt Enable 0: Disabled 1: Enabled	R/W
6	—	This bit is read as 0. The write value should be 0.	R/W
7	SDINT	Special Data Pattern Match Interrupt Enable 0: Disabled 1: Enabled	R/W

Note 1. Rewriting this bit is also allowed when REMC is operating (when REMCON1.EN = 1).

The REMINT register is used to enable or disable generation of each interrupt.

Rewriting the REMINT register is prohibited when the REMC is in operation. Rewrite this register while the REMCON1.EN bit and the REMCON0.ENFLG flag are both 0 (REMC is stopped).

26.2.5 REMCPC : Compare Control Register

Base address: REMC = 0x4009_2100

Offset address: 0x05

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

Bit	Symbol	Function	R/W
3:0	CPN[3:0]	Compare Bit Count Specification 0x0: Bit 0 in the REMCPD register is compared with bit 0 in the REMDAT0 register 0x1: Bits 1 and 0 in the REMCPD register are compared with bits 1 and 0 in the REMDAT0 register ⋮ ⋮ 0x7: Bits 7 to 0 in the REMCPD register are compared with bits 7 to 0 in the REMDAT0 register ⋮ ⋮ 0x9: Bits 9 to 0 in the REMCPD register are compared with bits 1 and 0 in the REMDAT1 register and bits 7 to 0 in the REMDAT0 register ⋮ ⋮ 0xF: Bits 15 to 0 in the REMCPD register are compared with bits 7 to 0 in the REMDAT1 register and bits 7 to 0 in the REMDAT0 register	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W

The REMCPC register is used to specify the number of bits to compare the value of the REMCPD register and the data to be stored in the REMDAT1 and REMDAT0 registers when the compare function is used.

Rewriting the REMCPC register is prohibited when the REMC is in operation. This register can be rewritten when the REMCON1.EN bit and the REMCON0.ENFLG flag are both 0 (REMC is stopped).

CPN[3:0] bits (Compare Bit Count Specification)

When the setting value of the CPN[3:0] bits is n, bits n to 0 are compared.

26.2.6 REMCPD : Compare Value Setting Register

Base address: REMC = 0x4009_2100

Offset address: 0x06

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	n/a	Setting value for comparison with received data	R/W

The REMCPD register is used to set the value to be compared with the data in the REMDAT1 and REMDAT0 registers when the compare function is used. The REMCPC.CPN[3:0] bits can be used to set the number of bits to be compared. Rewriting the REMCPD register is prohibited when the REMC is in operation (when REMCON1.EN = 1).

26.2.7 HDPMIN : Header Pattern Minimum Width Setting Register

Base address: REMC = 0x4009_2100

Offset address: 0x08

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	HDMIN[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	HDMIN[10:0]	Set the minimum width of the header pattern	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W

The HDPMIN register is used to set the minimum width of the header pattern. The setting range is from 0x000 to 0x7FF. When the interval between the edges of an external input signal measured using the base timer is between HDPMIN and HDPMAX, the pattern of the signal is detected as a header pattern.

Rewriting the HDPMIN register is prohibited when the REMC is in operation (when REMCON1.EN = 1).

26.2.8 HDPMAX : Header Pattern Maximum Width Setting Register

Base address: REMC = 0x4009_2100

Offset address: 0x0A

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	HDMAX[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	HDMAX[10:0]	Set the maximum width of the header pattern	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W

The HDPMAX register is used to set the maximum width of the header pattern. The setting range is from 0x000 to 0x7FF. When the interval between the edges of an external input signal measured using the base timer is between HDPMIN and HDPMAX, the pattern of the signal is detected as a header pattern.

Rewriting the HDPMAX register is prohibited when the REMC is in operation (when REMCON1.EN = 1).

26.2.9 D0PMIN : Data 0 Pattern Minimum Width Setting Register

Base address: REMC = 0x4009_2100

Offset address: 0x0C

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

Bit	Symbol	Function	R/W
7:0	n/a	Set the minimum width of the data 0 pattern	R/W

The D0PMIN register is used to set the minimum width of the data 0 pattern. The setting range is from 0x00 to 0xFF. When the interval between the edges of an external input signal measured using the base timer is between D0PMIN and D0PMAX, the pattern of the signal is detected as a data 0 pattern.

Rewriting the D0PMIN register is prohibited when the REMC is in operation (when REMCON1.EN = 1).

26.2.10 D0PMAX : Data 0 Pattern Maximum Width Setting Register

Base address: REMC = 0x4009_2100

Offset address: 0x0D

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

Bit	Symbol	Function	R/W
7:0	n/a	Set the maximum width of the data 0 pattern	R/W

The D0PMAX register is used to set the maximum width of the data 0 pattern. The setting range is from 0x00 to 0xFF. When the interval between the edges of an external input signal measured using the base timer is between D0PMIN and D0PMAX, the pattern of the signal is detected as a data 0 pattern.

Rewriting the D0PMAX register is prohibited when the REMC is in operation (when REMCON1.EN = 1).

26.2.11 D1PMIN : Data 1 Pattern Minimum Width Setting Register

Base address: REMC = 0x4009_2100

Offset address: 0x0E

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

Bit	Symbol	Function	R/W
7:0	n/a	Set the minimum width of the data 1 pattern	R/W

The D1PMIN register is used to set the minimum width of the data 1 pattern. The setting range is from 0x00 to 0xFF. When the interval between the edges of an external input signal measured using the base timer is between D1PMIN and D1PMAX, the pattern of the signal is detected as a data 1 pattern.

Rewriting the D1PMIN register is prohibited when the REMC is in operation (when REMCON1.EN = 1).

26.2.12 D1PMAX : Data 1 Pattern Maximum Width Setting Register

Base address: REMC = 0x4009_2100

Offset address: 0x0F

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

Bit	Symbol	Function	R/W
7:0	n/a	Set the maximum width of the data 1 pattern	R/W

The D1PMAX register is used to set the maximum width of the data 1 pattern. The setting range is from 0x00 to 0xFF. When the interval between the edges of an external input signal measured using the base timer is between D1PMIN and D1PMAX, the pattern of the signal is detected as a data 1 pattern.

Rewriting the D1PMAX register is prohibited when the REMC is in operation (when REMCON1.EN = 1).

26.2.13 SDPMIN : Special Data Pattern Minimum Width Setting Register

Base address: REMC = 0x4009_2100

Offset address: 0x10

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	SDMIN[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	SDMIN[10:0]	Set the minimum width of the special data pattern	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W

The SDPMIN register is used to set the minimum width of the special data pattern. The setting range is from 0x000 to 0x7FF.

When the interval between the edges of an external input signal measured using the base timer is between SDPMIN and SDPMAX, the pattern of the signal is detected as a special data pattern.

Rewriting the SDPMIN register is prohibited when the REMC is in operation (when REMCON1.EN = 1).

26.2.14 SDPMAX : Special Data Pattern Maximum Width Setting Register

Base address: REMC = 0x4009_2100

Offset address: 0x12

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	SDMAX[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	SDMAX[10:0]	Set the maximum width of the special data pattern	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W

The SDPMAX register is used to set the maximum width of the special data pattern. The setting range is from 0x000 to 0x7FF.

When the interval between the edges of an external input signal measured using the base timer is between SDPMIN and SDPMAX, the pattern of the signal is detected as a special data pattern.

Rewriting the SDPMAX register is prohibited when the REMC is in operation (when REMCON1.EN = 1).

26.2.15 REMPE : Pattern End Setting Register

Base address: REMC = 0x4009_2100

Offset address: 0x14

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	PE[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	PE[10:0]	Set the width of the pattern end	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W

The REMPE register is used to set the width of the pattern end. The setting range is from 0x000 to 0x7FF. The timing when the REMSTS.DRFLG flag changes from 1 to 0 can be set.

Rewriting the REMPE register is prohibited when the REMC is in operation (when REMCON1.EN = 1).

26.2.16 REMSTC : Receiver Standby Control Register

Base address: REMC = 0x4009_2100

Offset address: 0x16

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

Bit	Symbol	Function	R/W
0	SNZON*1	Snooze Mode Operation Control 0: Disables transition from Software Standby mode to Snooze mode 1: Enables transition from Software Standby mode to Snooze mode	R/W
1	DNFSL*1	Digital Filter Clock Selection 0: REMC operating clock is selected as a sampling clock 1: REMCLCLK/REMCSCLK is selected as a sampling clock	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W

Note:
- REMCLCLK: REMC LOCO clock
- REMCSCLK: REMC sub clock

Note 1. Set this bit to 1 when the REMCON1.EN bit and the REMCON0.ENFLG flag are both 0 (REMC is stopped).

The REMSTC register is used to control the startup of reception (Snooze mode) on receiving the remote control signal in the Software Standby mode and select the sampling clock of the digital filter.

Rewriting the REMSTC register is prohibited when the REMC is in operation (when REMCON1.EN = 1).

SNZON bit (Snooze Mode Operation Control)

- When the input level of the RIN0 pin is changed in the Software Standby mode, the Software Standby mode is exited, and remote control reception is performed without operating the CPU (the Snooze mode).
- Use the Snooze mode only when the timer interrupt (TAU0_ENDI6) is selected as the REMC operating clock. In this case, select a clock that can operate in Snooze mode as the count clock for channel 6 of timer array unit.
- When using Snooze mode, set channel 6 of timer array unit to interval timer mode and enable the count operation before shifting to Software Standby mode.
- Even when Snooze mode is to be used, set the SNZON bit to 0 after return from Snooze mode. Set the SNZON bit to 1 when the transition to Software Standby mode is to be repeated.
- When setting the SNZON bit to 1, set the REMCON0.FIL bit to 1 (enables the digital filter) and set the DNFSL bit to 1 (REMCLCLK/REMCSCLK is selected as a sampling clock).
- After returning from Software Standby mode due to a compare match interrupt or header pattern match interrupt, set the SNZON bit to 0.

DNFSL bit (Digital Filter Clock Selection)

This bit is used to select the sampling clock of the digital filter. Set this bit to 1 when setting the SNZON bit to 1 (transition from Software Standby mode to Snooze mode is enabled).

26.2.17 REMRBIT : Receive Bit Count Register

Base address: REMC = 0x4009_2100

Offset address: 0x17

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	RBIT[5:0]						RBIT0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RBIT0	Receive Bit Count Check (bit 0) 0: This register is cleared (0x00) Others: Setting prohibited	R/W
6:1	RBIT[5:0]	Receive Bit Count Check (bit 6 to 1)	R
7	—	This bit is read as 0. The write value should be 0.	R/W

Note: If updating and reading data overlap, an undefined value may be read. For details on reading this register, see section 26.4.5. Reading Registers.

The REMRBIT register indicates the number of received bits.

All the bits of the REMRBIT register are initialized when the setting of the REMCON1.EN bit is 0.

These bits indicate the bit position of the buffer to be stored by counting the detected data 0 pattern or data 1 pattern.

- When the receive bit count exceeds 64 (0x40), the value returns to 1.
- The header pattern and special data pattern are not counted.
- If an error is detected while the REMCON0.EC bit is 1, the value is not incremented even when the data 0 pattern or data 1 pattern is detected.
- The REMRBIT register becomes 0 when the REMSTS.DRFLG flag changes from 0 to 1.
- The REMRBIT register becomes 0 when the REMSTS.HDFLG flag changes from 0 to 1.

When 0 is written to the RBIT0 bit, the value of the REMRBIT register becomes 0x00 after one to two cycles of the operating clock.

26.2.18 REMDAT0 : Receive Data 0 Register

Base address: REMC = 0x4009_2100

Offset address: 0x18

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DAT[6:0]							DAT0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DAT0	Receive Data 0 Store (bit 0) 0: This register is cleared (0x00) Others: Setting prohibited	R/W
7:1	DAT[6:0]	Receive Data 0 Store (bit 7 to 1)	R

Note: If updating and reading data overlap, an undefined value may be read. For details on reading this register, see section 26.4.5. Reading Registers.

The REMDAT0 register holds received data.

All the bits of the REMDAT0 register are initialized when the setting of the REMCON1.EN bit is 0.

When data 0 pattern or data 1 pattern is detected, the result is sequentially stored bit by bit starting from the REMDAT0.DAT0 bit as received data. For details on storing received data, see section 26.3.8. Receive Data Buffer.

When 0 is written to the DAT0 bit, the value of the REMDAT0 register becomes 0x00 after one to two cycles of the operating clock.

26.2.19 REMDATj : Receive Data j Register (j = 1 to 7)

Base address: REMC = 0x4009_2100

Offset address: 0x18 + 0x1 × j

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

Bit	Symbol	Function	R/W
7:0	DAT[7:0]	Receive Data Store	R

Note: If updating and reading data overlap, an undefined value may be read. For details on reading this register, see section 26.4.5. Reading Registers.

The REMDATj register holds received data.

All the bits of the REMDATj register are initialized when the setting of the REMCON1.EN bit is 0.

DAT[7:0] bits (Receive Data Store)

When data 0 pattern or data 1 pattern is detected, the result is sequentially stored bit by bit starting from the DAT[0] bit as received data. For details on storing received data, see section 26.3.8. Receive Data Buffer.

26.2.20 REMTIM : Measurement Result Register

Base address: REMC = 0x4009_2100

Offset address: 0x20

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	TIM[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	TIM[10:0]	Measurement Result	R
15:11	—	These bits are read as 0. The write value should be 0.	R

Note: If updating and reading data overlap, an undefined value may be read. For details on reading this register, see section 26.4.5. Reading Registers.

The REMTIM register indicates the measurement result of each pattern width.

All the bits of the REMTIM register are initialized when the setting of the REMCON1.EN bit is 0.

TIM[10:0] bits (Measurement Result)

The value of the base timer is captured when any of the following patterns is detected.

- Header pattern
- Data 0 pattern
- Data 1 pattern
- Special data pattern
- Data pattern other than the above (receive error)

26.3 Operation

26.3.1 Overview of REMC Operation

Figure 26.2 shows an example of the remote control signal. The signal begins with a header, followed by a sequence of data. This header differs from the subsequent sequence of data in waveform, allowing the header and the data to be distinguished. The sequence of data contains custom code and data code, and 0 or 1 is distinguished depending on the bit length. After a stop bit, there is an interval during which the signal does not change (frame space), thus constituting a frame.

The time between the edges of the external input signal is measured using the base timer in the REMC. The patterns of the remote control signal are detected and the data is captured according to the measurement results.

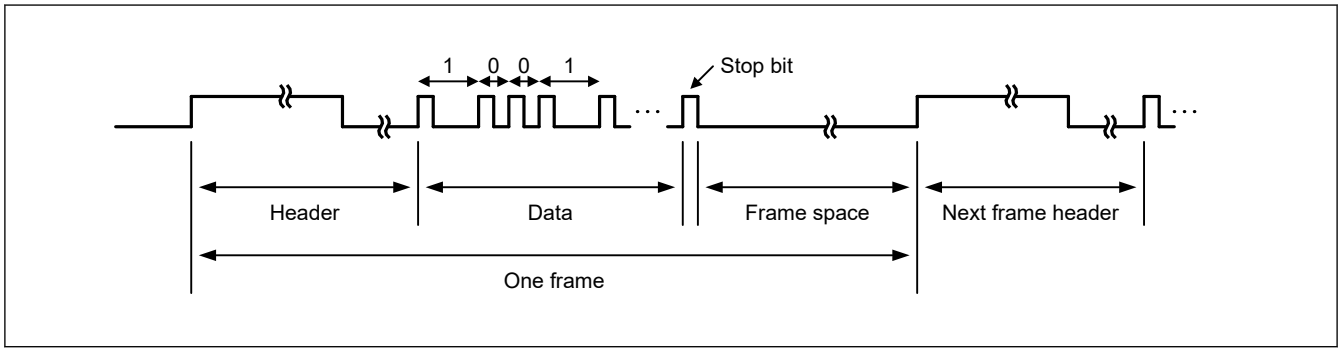


Figure 26.2 Example of remote control signal

26.3.2 Initial Setting

Initialize the REMC according to the procedure shown in Table 26.4 to receive the remote control signal.

Set the REMCON1.EN bit to 0 if the REMC is operating. Then the REMCON0.ENFLG flag becomes 0 and the REMC stops the operation.

Set the format for the remote control signal waveform by the REMCON1.TYP[1:0] bits; select the signal inversion or non-inversion by the REMCON0.INV bit; select the operating clock by the REMCON1.CSRC bit; and set the digital filter by the REMCON0.FIL, REMCON0.FILSEL, and REMSTC.DNFSL bits, while the REMCON0.ENFLG flag is 0. Set the detecting width for each data pattern into the HDPMIN, HDPMAX, D0PMIN, D0PMAX, D1PMIN, D1PMAX, SDPMIN, SDPMAX, and REMPE registers. Make any other settings such as enabling interrupts by the REMINT register and setting of the compare function by the REMCPC and REMCPD registers if required.

After all necessary register settings are completed, set the REMCON1.EN bit to 1 to start REMC operation.

Table 26.4 Example of step for initial settings of REMC

Step		Process	Detail
Example of step for initial settings of REMC	<1>	Start operation.	—
	<2>	Stop the REMC if it is operating.	If REMCON1.EN = 1 or REMCON0.ENFLG = 1, Clear REMCON1.EN bit. Wait until REMCON0.ENFLG is cleared.
	<3>	Select the format for the remote control signal waveform.	Set REMCON1.TYP[1:0] bits.
	<4>	Select whether the input signal to be inverted or not.	If your system wants to invert the signal, set the REMCON0.INV bit.
	<5>	Select the REMC operating clock.	Set the REMCON1.CSRC bit.
	<6>	Configure the digital filter.	Set the REMCON0.FIL and REMCON0.FILSEL bits. Set the REMSTC.DNFSL bit.
	<7>	Set the minimum and maximum values for each data pattern detection width.	Set the HDPMIN and HDPMAX registers. Set the D0PMIN and D0PMAX registers. Set the D1PMIN and D1PMAX registers. Set the SDPMIN and SDPMAX registers. Set the REMPE register.
	<8>	Set any other registers if required.	Set any other registers.
	<9>	Executes next processing.	—

26.3.3 Pattern Setting

The format for capturing the remote control signal reception waveform can be set by setting the REMCON1.TYP[1:0] bits. Figure 26.3 and Figure 26.4 show examples of a remote control signal reception waveform captured by setting the REMCON1.TYP[1:0] bits.

When the REMCON1.TYP[1:0] bits are 00b (format A)

The measured result is determined from the setting value of the header pattern at the rising edge of the internal input signal.

When the header pattern is received, the measured result is determined from the setting values of the data 0, data 1 and special data patterns at the rising edge of the internal input signal.

When the REMCON1.TYP[1:0] bits are 01b (format B)

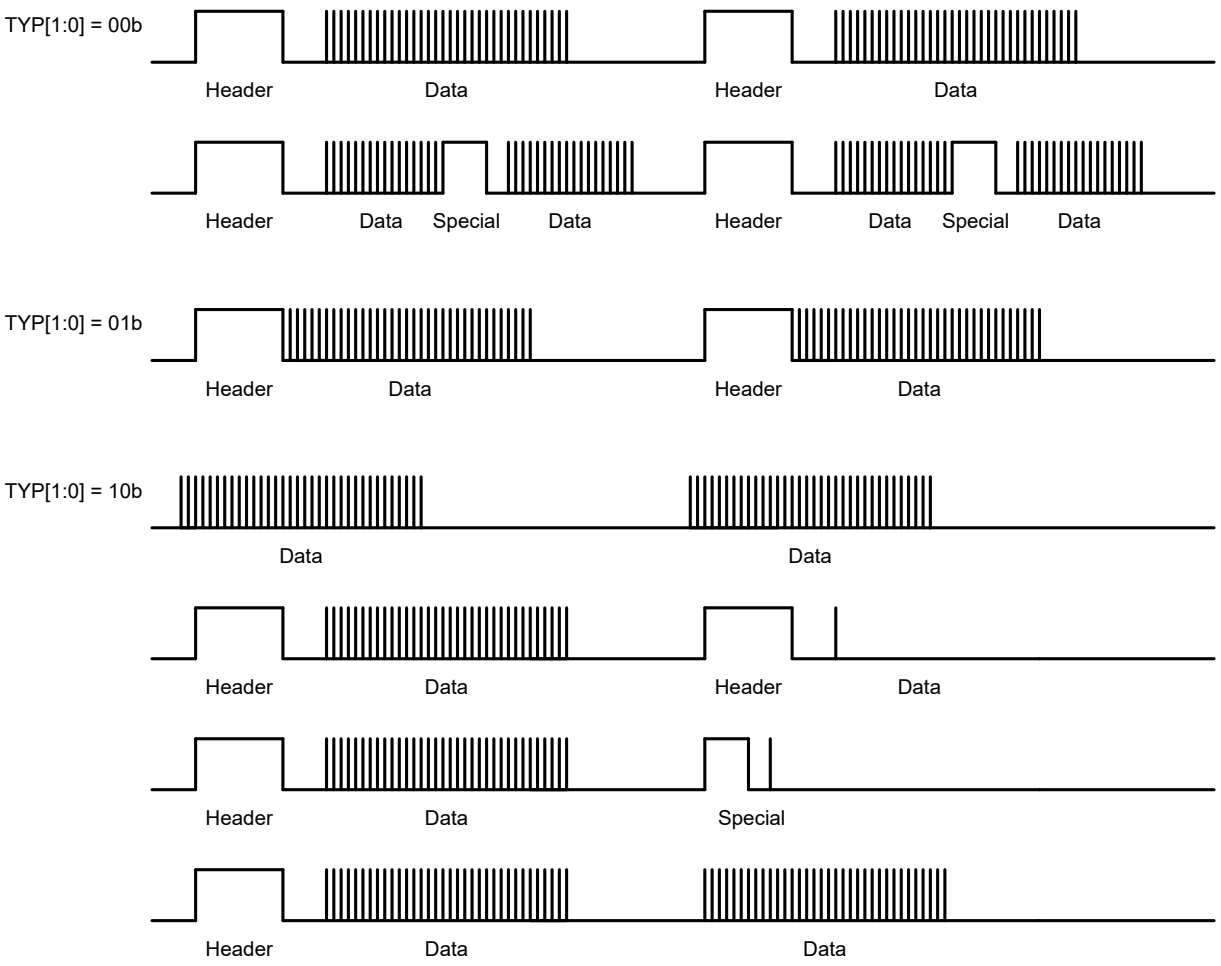
The measured result is determined from the setting value of the header pattern at the falling edge of the internal input signal.

When the header pattern is received, the measured result is determined from the setting values of the data 0, data 1 and special data patterns at the falling edge of the internal input signal.

The header pattern is detected once within one frame.

When the REMCON1.TYP[1:0] bits are 10b (format C)

The measured result is determined from the setting values of the header, data 0, data 1 and special data patterns at the rising edge of the internal input signal.



TYP[1:0]	Header Pattern	Data 0 Pattern	Data 1 Pattern	Special Data Pattern
TYP[1:0] = 00b				
TYP[1:0] = 01b				
TYP[1:0] = 10b				

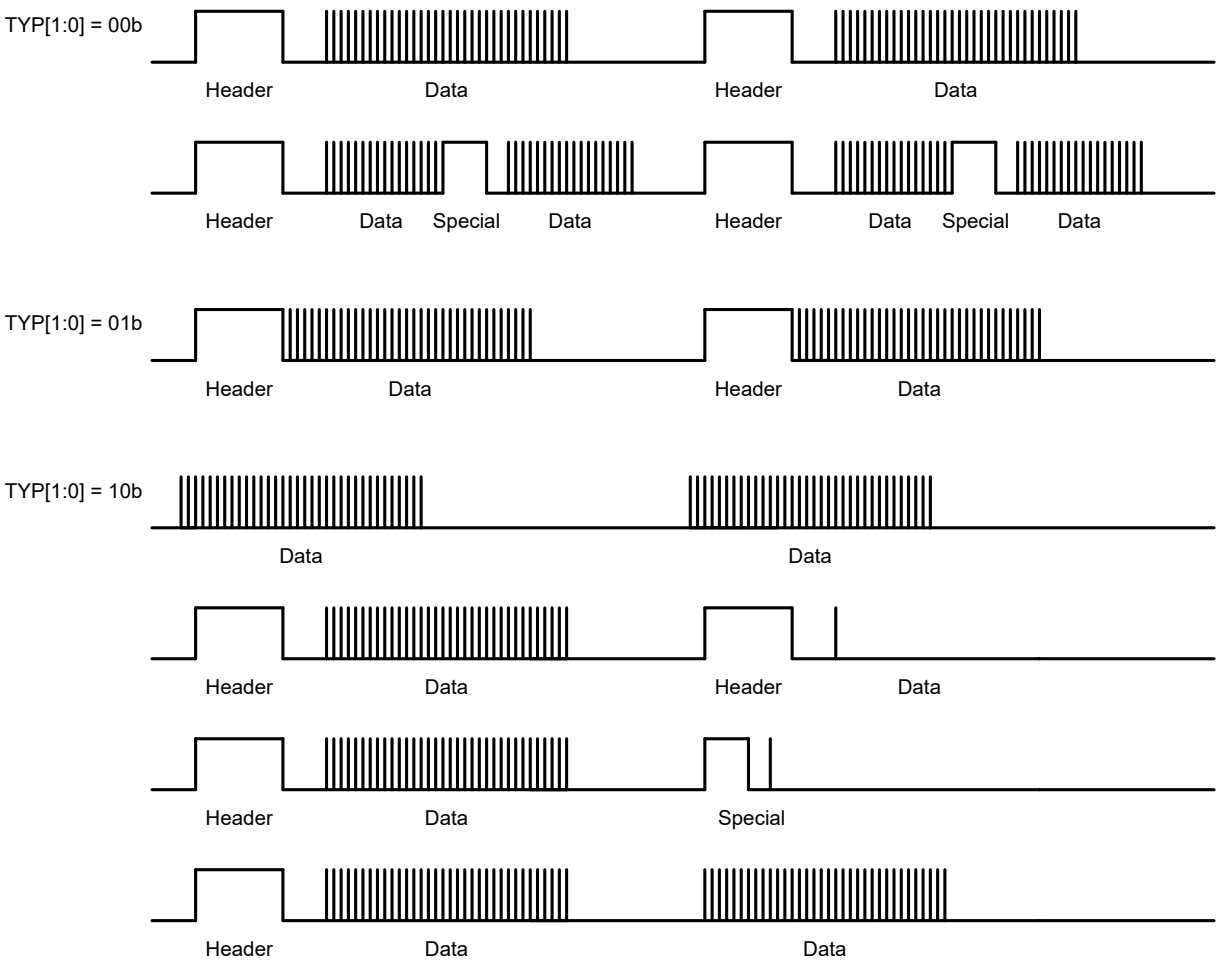
TYP[1:0]: Bits in the REMCON1 register

↑↓ · · · Measurement starts

↑ ↓ · · · Pattern determined

When TYP[1:0] = 01b, the number of detecting the header pattern within the receive frame is one.
The above diagram applies when the REMCON0.INV bit is 0.

Figure 26.3 Example of remote control signal reception waveform captured by setting REMCON1.TYP[1:0] bits (REMCON0.INV = 0)



TYP[1:0]	Header Pattern	Data 0 Pattern	Data 1 Pattern	Special Data Pattern
TYP[1:0] = 00b				
TYP[1:0] = 01b				
TYP[1:0] = 10b				

TYP[1:0]: Bits in the REMCON1 register

↑↓ · · · Measurement starts

↑↓ · · · Pattern determined

When TYP[1:0] = 01b, the number of detecting the header pattern within the receive frame is one.
The above diagram applies when the REMCON0.INV bit is 0.

Figure 26.4 Example of remote control signal reception waveform captured by setting REMCON1.TYP[1:0] bits (REMCON0.INV = 1)

26.3.4 Operating Clock

The REMC can use either of the following clock signals as its operating clock REMCLCLK/REMCSCLK and timer interrupt (TAU0_ENDI6). When transition from the Software Standby mode to the Snooze mode is enabled, select REMCLCLK/REMCSCLK as the clock of the digital filter. When selecting the timer interrupt (TAU0_ENDI6) as the REMC operating clock, see section 18, Timer Array Unit (TAU) to confirm the operating clock setting for channel 6 of timer array unit.

The following sections describe how to supply these clock signals.

26.3.4.1 When using REMCLCLK/REMCSCLK as the REMC operating clock

This section describes the procedure for using REMCLCLK/REMCSCLK as the REMC operating clock. For details, see section 8, Clock Generation Circuit.

1. When using REMCLCLK as REMCLCLK/REMCSCLK
   REMCLCLK is supplied as the REMC operating clock by setting the OSCMR.WUTMMCK0 bit to 1 and setting the REMCON1.CSRC bit to 0.
2. When using REMCSCLK as REMCLCLK/REMCSCLK
   REMCSCLK starts to oscillate by setting the OSCMR.WUTMMCK0 bit to 0 and setting the SOSCCR.SOSTP bit to 0. Then, after the oscillation stabilization time for the sub-clock is secured using the timer function or another function, REMCSCLK is supplied as the REMC operating clock by using software and by setting the REMCON1.CSRC bit to 0.

26.3.4.2 When using TAU0_ENDI6 as the REMC operating clock

The timer interrupt (TAU0_ENDI6) is supplied as the REMC operating clock by setting the RECOM1.CSRC bit to 1. Operate channel 6 of timer array unit in the interval timer mode.

For details about the operation of timer array unit, see section 18, Timer Array Unit (TAU).

26.3.5 RIN0 Input

The options below can be selected in RIN0 input.

- Input polarity
- Digital filter

Figure 26.5 shows the configuration of RIN0 internal input signal generation.

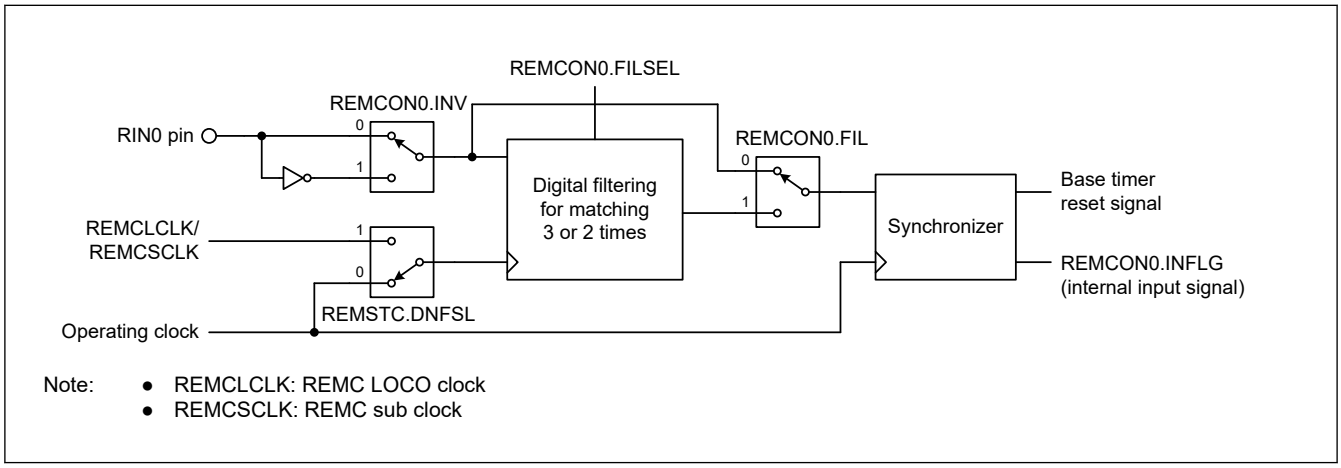
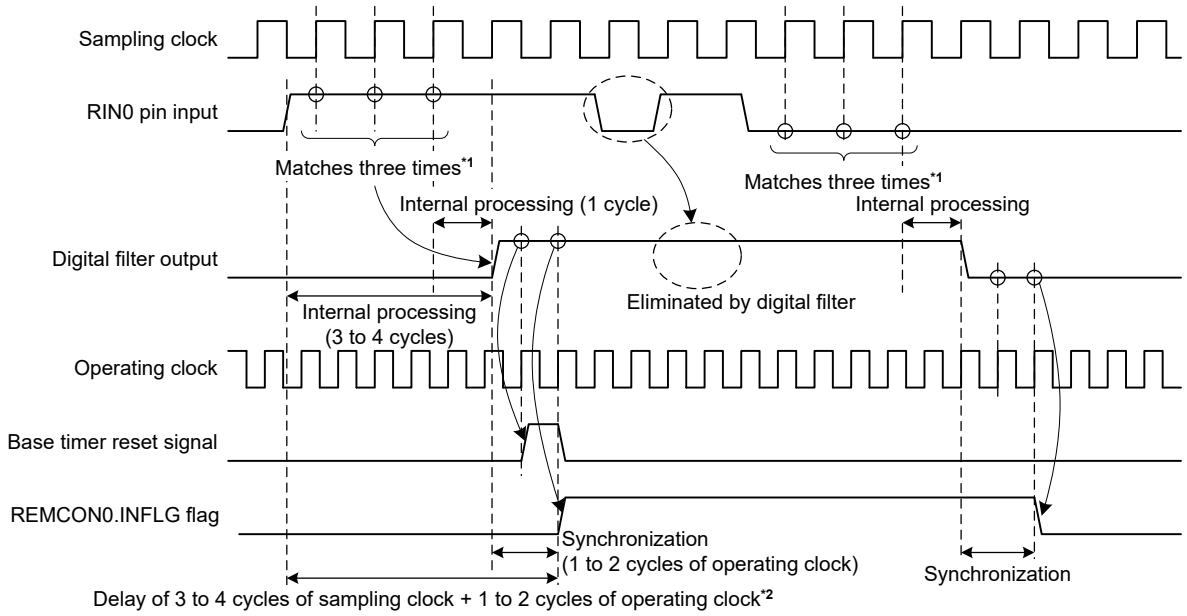


Figure 26.5 RIN0 internal input signal generation configuration

The input polarity of the RIN0 pin can be inverted. Whether to invert or not can be selected by the REMCON0.INV bit. When the REMCON0.FIL bit is 1 (digital filter enabled), if the signal input to the RIN0 pin holds the same level for k sequential cycles (k = 3 or 2; value selected by the REMCON0.FILSEL bit), that level is transferred to the internal circuit. This enables noise to be eliminated from k cycles of the sampling clock. The sampling clock of the digital filter is selectable from the REMC operating clock and REMCLCLK/REMCSCLK by setting the REMSTC.DNFSL bit. When setting the REMSTC.SNZON bit to 1 (transition from Software Standby mode to Snooze mode is enabled), set the REMCON0.FIL bit to 1 (the digital filter is enabled) and the REMSTC.DNFSL bit to 1 (REMCLCLK/REMCSCLK is selected as a sampling clock).

Input to the RIN0 pin is transferred as the REMCON0.INFLG flag (input signal flag) and the base timer reset signal to the internal circuit in synchronization with the operating clock. The base timer reset signal is used to initialize the internal base timer to the pattern detection corresponding to the REMCON1.TYP[1:0] setting. There is a delay caused by internal processing after the input to the RIN0 pin is changed and before these signals are generated. Figure 26.6 shows digital filtering for RIN0 input.



Note 1. The above applies when the REMCON0.FIL bit is 1 (digital filter enabled) and the REMCON0.FILSEL bit is 0 (matches three times). When the FILSEL bit is 1 (matches twice), the signal is output earlier by one cycle of sampling clock than that in this figure.

Note 2. When the FIL bit is 0 (digital filter disabled), a delay becomes one to two cycles of the operating clock for synchronization.

Figure 26.6 Digital filtering for RIN0 input

26.3.6 Pattern Detection

The REMC has a function that detects the following patterns.

- Header pattern
- Data 0 pattern
- Data 1 pattern
- Special data pattern

Using the base timer included in the REMC, the time between the edges of the external input signal is measured to determine which pattern matches the measurement result. This enables detection of the remote control signal and capturing the data. The width for determining each pattern can be set to any value using each pattern setting register. Figure 26.7 shows the waveform of REMC operation.

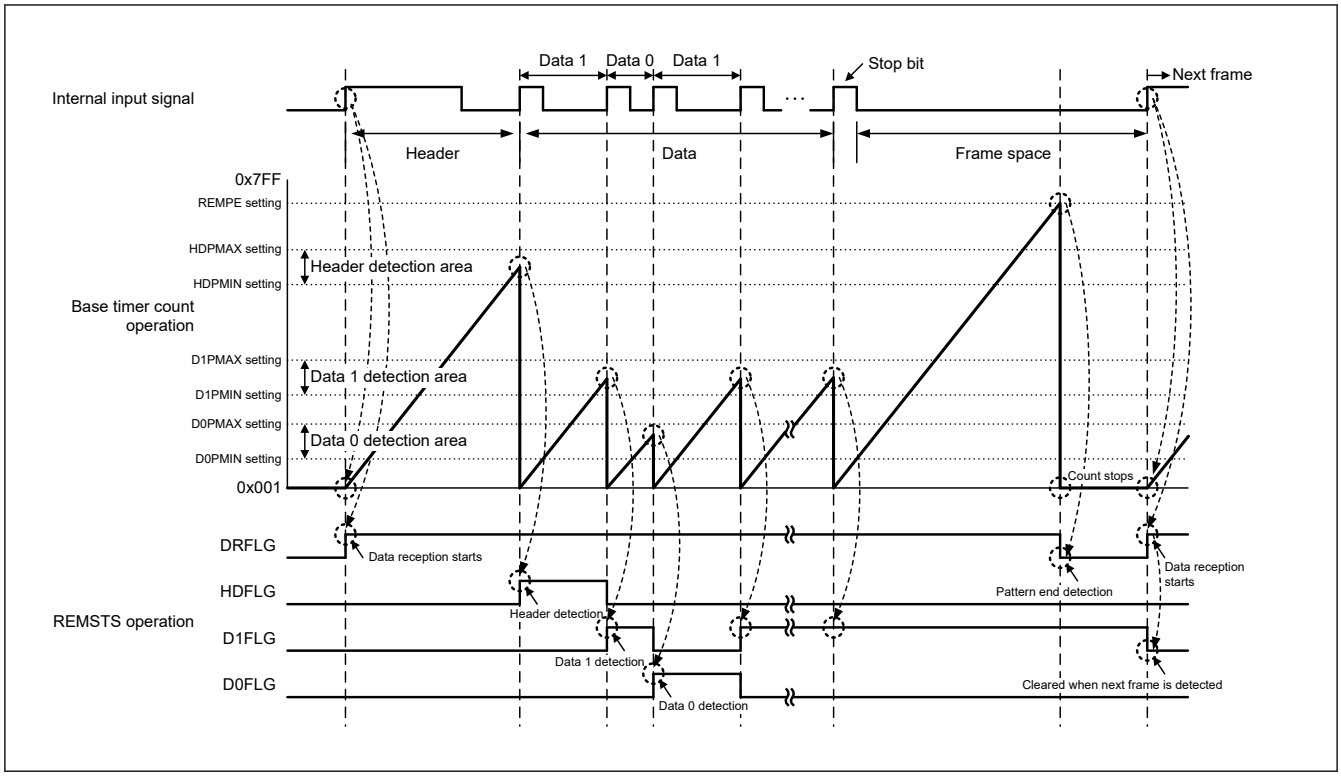


Figure 26.7 Waveform of REMC operation

26.3.6.1 Header pattern detection

The header pattern can be detected by setting the minimum width of the header pattern in the HDPMIN register and the maximum width in the HDPMAX register.

The minimum and maximum widths of the header pattern must be "1 < HDPMIN register value ≤ HDPMAX register value".

$$\text{Setting value n} = \frac{\text{Minimum width (maximum width) of header pattern}}{\text{Operating clock cycle time}}$$

When not using the header pattern, set the HDPMIN and HDPMAX registers to 0x000.

Make sure that the setting value of the header pattern is different from the setting values of data 0, data 1, and special data patterns, and the setting ranges are not overlapped.

When the REMCON1.TYP[1:0] bits are 00b or 01b, if the data 0, data 1, or special data pattern is detected before the header pattern is detected, the following occur:

- The REMSTS.REFLG flag becomes 1 (an error has occurred).
- The REMSTS.D0FLG, REMSTS.D1FLG, and REMSTS.SDFLG flags remain unchanged.
- The REMDAT0 to REMDAT7 registers remain unchanged.

When the REMCON1.TYP[1:0] bits are 01b, the number of detecting the header pattern is one while the DRFLG flag is 1.

26.3.6.2 Data 0 pattern detection

The data 0 pattern can be detected by setting the minimum width of the data 0 pattern in the D0PMIN register and the maximum width in the D0PMAX register.

The minimum and maximum widths of the data 0 pattern must be "1 < D0PMIN register value ≤ D0PMAX register value".

$$\text{Setting value n} = \frac{\text{Minimum width (maximum width) of data 0 pattern}}{\text{Operating clock cycle time}}$$

When not using the data 0 pattern, set the D0PMIN and D0PMAX registers to 0x00.

Make sure that the setting value of the data 0 pattern is different from the setting values of the header, data 1, and special data patterns, and the setting ranges are not overlapped.

When the REMCON1.TYP[1:0] bits are 00b or 01b, if the data 0 pattern or data 1 pattern is detected before the header pattern is detected, the following occur:

- The REMSTS.REFLG flag becomes 1 (an error has occurred).
- The REMSTS.D0FLG, REMSTS.D1FLG, and REMSTS.SDFLG flags remain unchanged.
- The REMDAT0 to REMDAT7 registers remain unchanged.

26.3.6.3 Data 1 pattern detection

The data 1 pattern can be detected by setting the minimum width of the data 1 pattern in the D1PMIN register and the maximum width in the D1PMAX register.

The minimum and maximum widths of the data 1 pattern must be "1 < D1PMIN register value ≤ D1PMAX register value".

$$\text{Setting value n} = \frac{\text{Minimum width (maximum width) of data 1 pattern}}{\text{Operating clock cycle time}}$$

When not using the data 1 pattern, set the D1PMIN and D1PMAX registers to 0x00.

Make sure that the setting value of the data 1 pattern is different from the setting values of the header, data 0, and special data patterns, and the setting ranges are not overlapped.

When the REMCON1.TYP[1:0] bits are 00b or 01b, if the data 0 pattern or data 1 pattern is detected before the header pattern is detected, the following occur:

- The REMSTS.REFLG flag becomes 1 (an error has occurred).
- The REMSTS.D0FLG, REMSTS.D1FLG, and REMSTS.SDFLG flags remain unchanged.
- The REMDAT0 to REMDAT7 registers remain unchanged.

26.3.6.4 Special data pattern detection

The special data pattern can be detected by setting the minimum width of the special data pattern in the SDPMIN register and the maximum width in the SDPMAX register.

The minimum and maximum widths of the special data pattern must be "1 < SDPMIN register value ≤ SDPMAX register value".

$$\text{Setting value n} = \frac{\text{Minimum width (maximum width) of special data pattern}}{\text{Operating clock cycle time}}$$

When not using the special data pattern, set the SDPMIN and SDPMAX registers to 0x000.

Make sure that the setting value of the special data pattern is different from the setting values of the header, data 0, and data 1 patterns, and the setting ranges are not overlapped.

When the REMCON1.TYP[1:0] bits are 00b or 01b, if the special data pattern is detected before the header pattern is detected, the following occur:

- The REMSTS.REFLG flag becomes 1 (an error has occurred).
- The REMSTS.SDFLG flag remains unchanged.
- The REMDAT0 to REMDAT7 registers remain unchanged.

26.3.6.5 Examples of setting pattern setting registers

For the header, data 0, data 1, and special data pattern setting registers, make sure that the minimum to maximum values of each pattern are different, and the setting ranges do not overlap as shown in Figure 26.8.

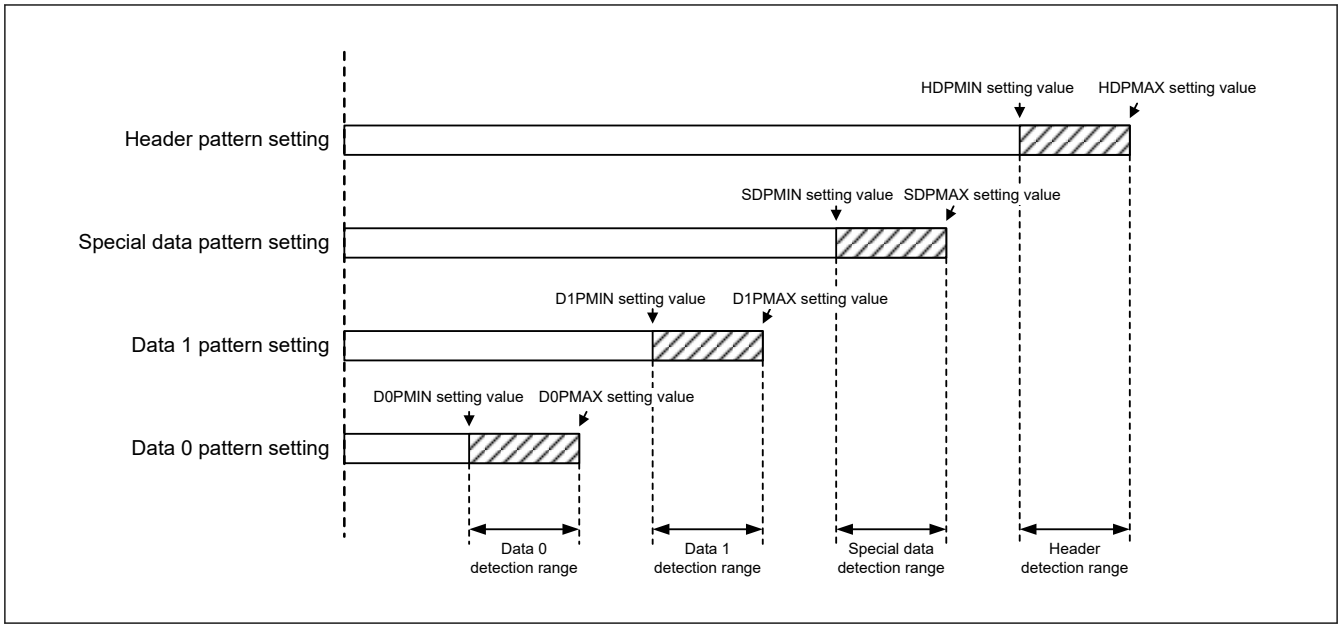


Figure 26.8 Examples of setting pattern setting registers

26.3.6.6 Updating status flags upon pattern detection

The detected patterns can be confirmed by reading the following flags: header pattern match flag (REMSTS.HDFLG), data 0 pattern match flag (REMSTS.D0FLG), data 1 pattern match flag (REMSTS.D1FLG), and special data pattern match flag (REMSTS.SDFLG). These flags are cleared when a different pattern is detected. If a pattern other than the above patterns is detected, it is detected as an error pattern. This can be confirmed by reading the receive error flag (REMSTS.REFLG). This flag is cleared when the next frame is received. Figure 26.9 shows an example of pattern detection and flag operation.

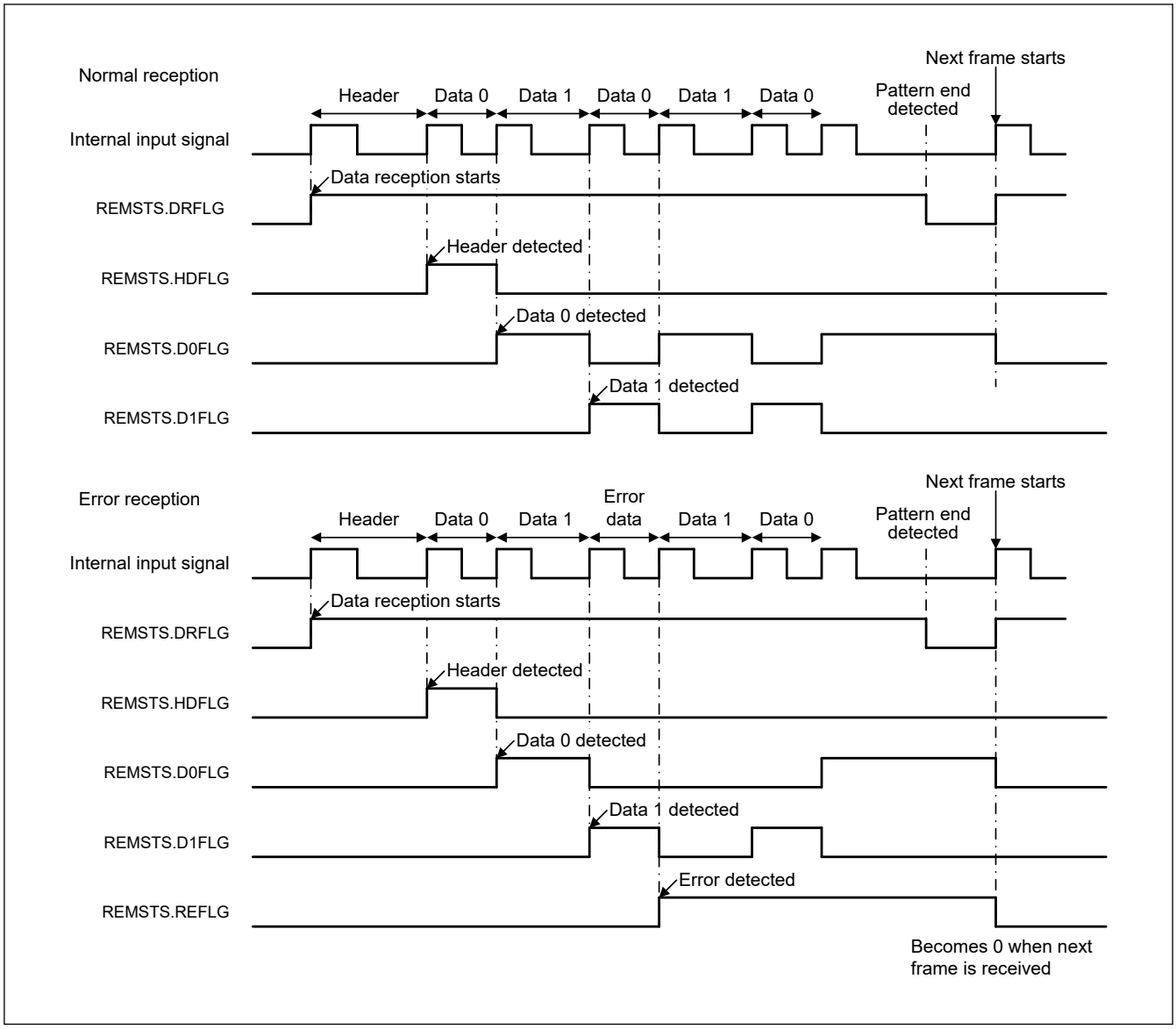


Figure 26.9 Example of flag operation

26.3.7 Pattern End

The timing when the REMSTS.DRFLG flag becomes 0 can be set.

When setting the REMPE register, be sure to set that the REMPE value > HDPMAX, D0PMAX, D1PMAX, or SDPMAX value.

When the REMPE value ≤ HDPMAX, D0PMAX, D1PMAX, or SDPMAX value, the value specified in the REMPE register is not used to set the timing when the REMSTS.DRFLG flag becomes 0. In this case, data reception is completed according to the largest value from among the setting values of the HDPMAX, D0PMAX, D1PMAX, and SDPMAX registers.

Figure 26.10 shows operation of the data reception complete flag for each pattern end setting.

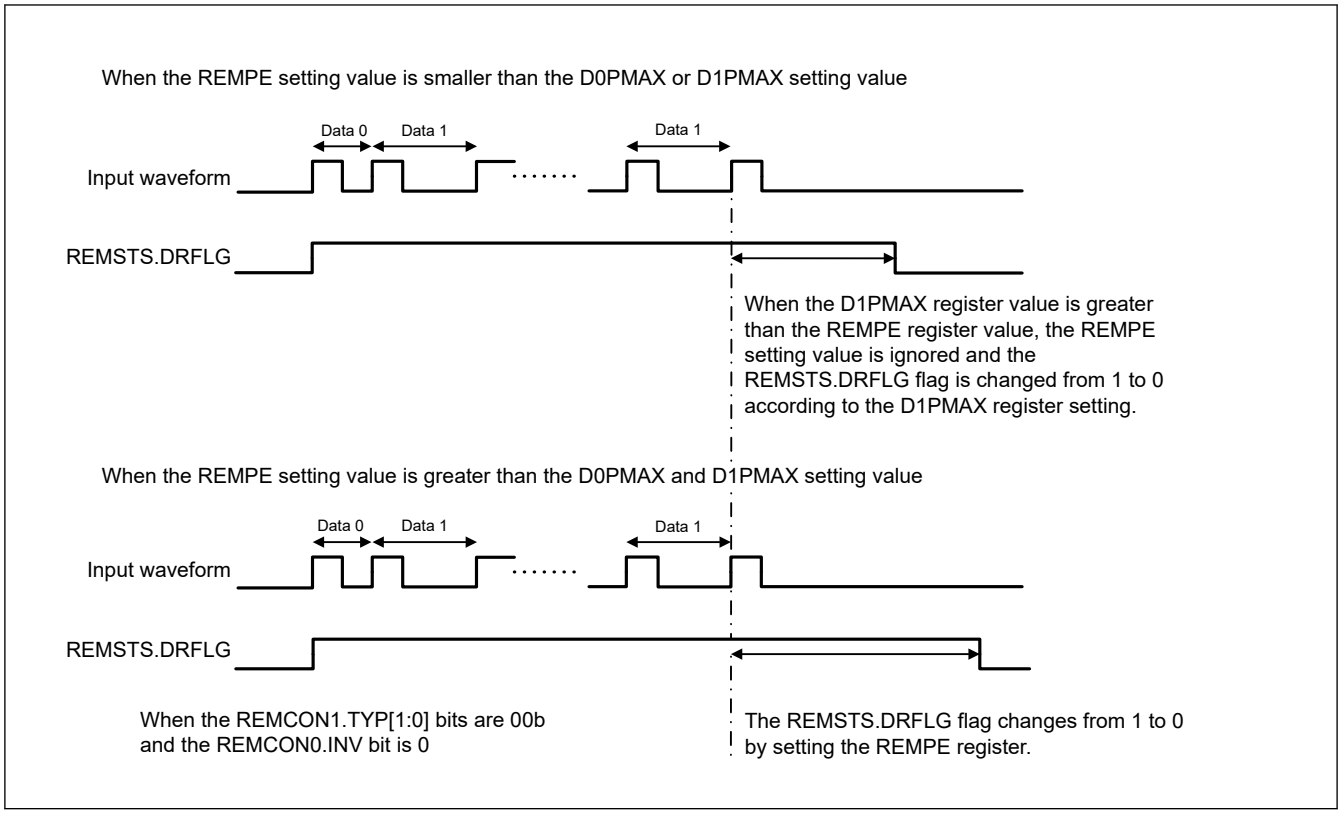


Figure 26.10 Operation of data reception complete flag for each pattern end setting

26.3.8 Receive Data Buffer

The receive data j register (REMDATj) (j = 0 to 7) is an 8-byte (64-bit) buffer for storing received data. When data 0 pattern or data 1 pattern is detected, the detection result is sequentially stored starting from the REMDAT0.DAT0 bit as shown in Figure 26.11. The REMRBIT register is counted up at the same time, so the number of the current received bits can be checked by reading the REMRBIT register. See Table 26.5 for the relationship between the number of received bits and the location where data is stored. The values of the REMDATj and REMRBIT registers do not change even when the header pattern or special data pattern is received. If the REMDATj or REMRBIT register is read while the data is being updated, the value read may be undefined.

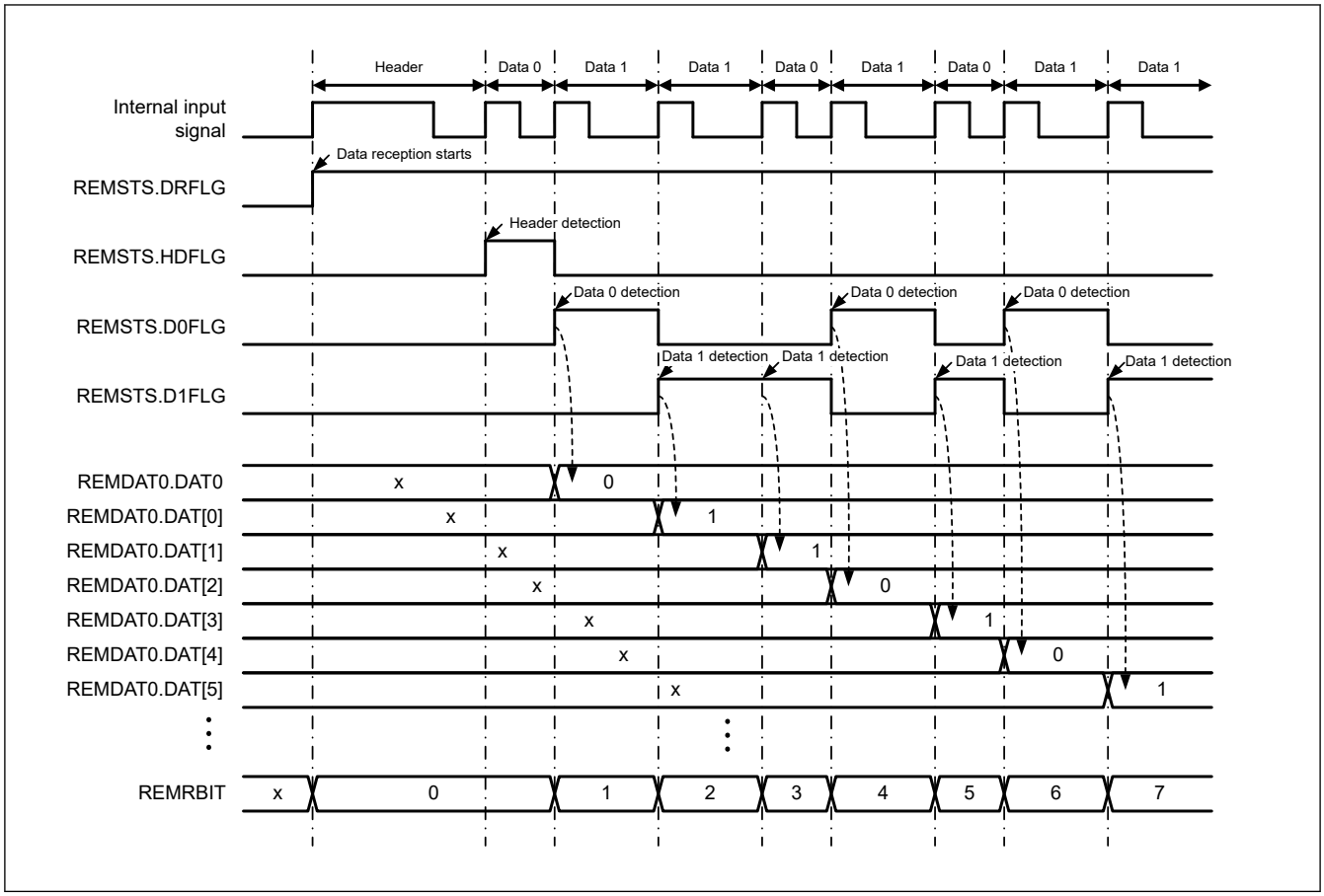


Figure 26.11 Operation of receive data buffer

Table 26.5 Relationship between number of received bits and location where data is stored

Number of received bits	Location where data is stored		Number of received bits	Location where data is stored	
	Register name	Bit name		Register name	Bit name
1	REMDAT0	DAT0	33	REMDAT4	DAT[0]
2		DAT[0]	34		DAT[1]
3		DAT[1]	35		DAT[2]
4		DAT[2]	36		DAT[3]
5		DAT[3]	37		DAT[4]
6		DAT[4]	38		DAT[5]
7		DAT[5]	39		DAT[6]
8		DAT[6]	40		DAT[7]
9	REMDAT1	DAT[0]	41	REMDAT5	DAT[0]
10		DAT[1]	42		DAT[1]
11		DAT[2]	43		DAT[2]
12		DAT[3]	44		DAT[3]
13		DAT[4]	45		DAT[4]
14		DAT[5]	46		DAT[5]
15		DAT[6]	47		DAT[6]
16		DAT[7]	48		DAT[7]
17	REMDAT2	DAT[0]	49	REMDAT6	DAT[0]
18		DAT[1]	50		DAT[1]
19		DAT[2]	51		DAT[2]
20		DAT[3]	52		DAT[3]
21		DAT[4]	53		DAT[4]
22		DAT[5]	54		DAT[5]
23		DAT[6]	55		DAT[6]
24		DAT[7]	56		DAT[7]
25	REMDAT3	DAT[0]	57	REMDAT7	DAT[0]
26		DAT[1]	58		DAT[1]
27		DAT[2]	59		DAT[2]
28		DAT[3]	60		DAT[3]
29		DAT[4]	61		DAT[4]
30		DAT[5]	62		DAT[5]
31		DAT[6]	63		DAT[6]
32		DAT[7]	64		DAT[7]

Note: When the data exceeds 64 bits, the REMDATj register is sequentially overwritten from the first bit.

When 0 is written to the REMDAT0.DAT0 bit, the values of the REMDAT0 to REMDAT7 registers become 0x00 after one to two cycles of the operating clock. Figure 26.12 shows the REMDATj and REMRBIT register operation when 0x00 is written to the REMDAT0 register.

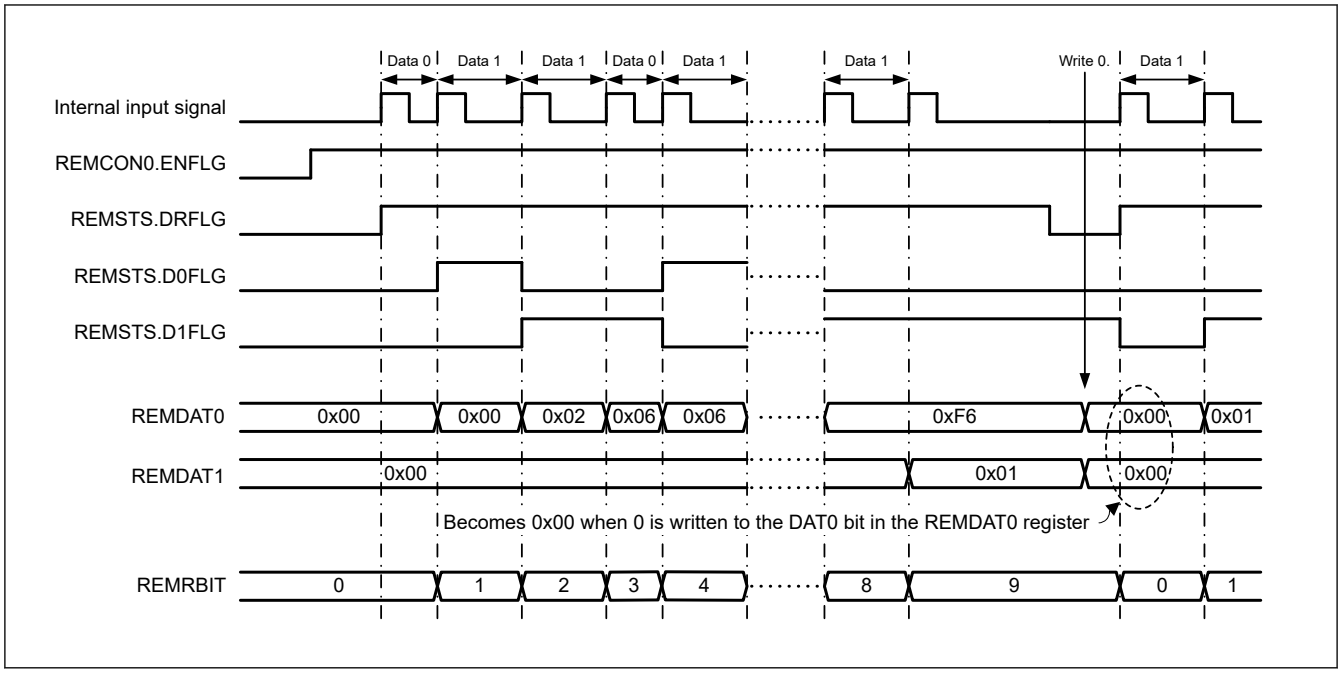


Figure 26.12 REMDATj and REMRBIT register operation (0x00 is written to REMDAT0 register)

When 0 is written to the REMRBIT.RBIT0 bit, the value of the REMRBIT register becomes 0x00 after one to two cycles of the operating clock. When the REMCON1.TYP[1:0] bits are 00b or 01b, if the header pattern is detected during data reception, the value of the REMRBIT register is initialized to 0x00 and the received data is sequentially overwritten from the REMDAT0.DAT0 bit. Figure 26.13 shows operation of header pattern detection during data reception.

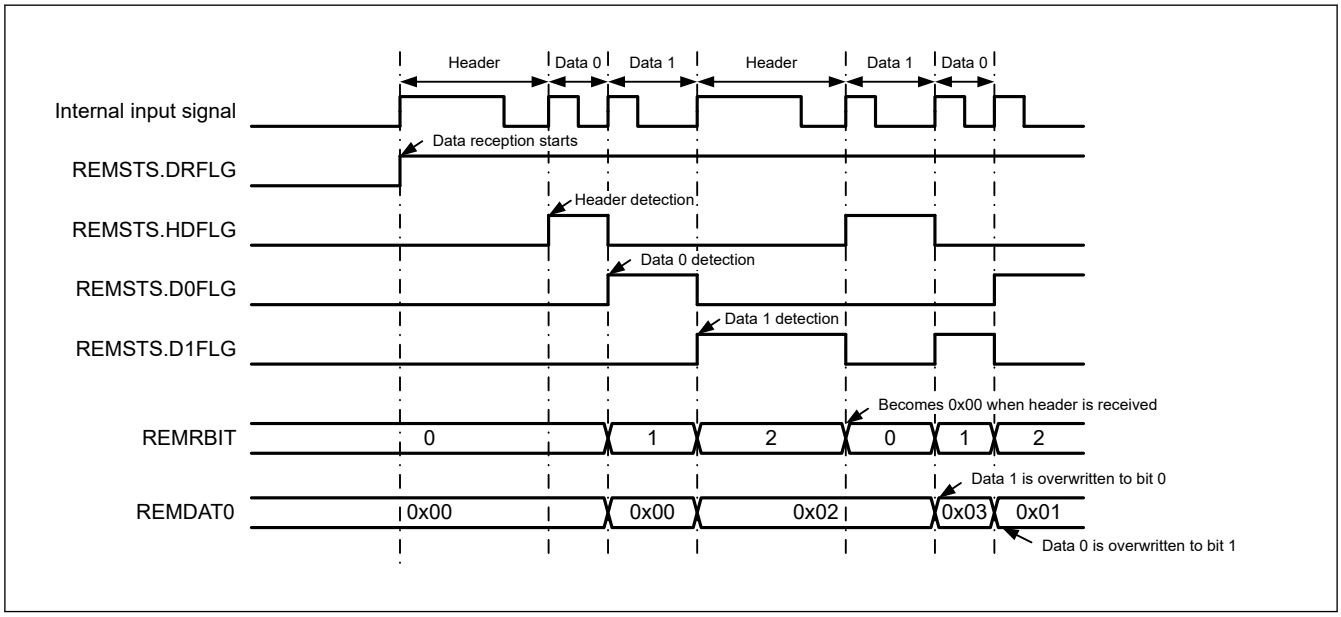


Figure 26.13 Operation of header pattern detection during data reception

When the data exceeds 64 bits, the buffer is sequentially overwritten from the first bit. Figure 26.14 shows the REMRBIT register operation when the REMSTS.BFULFLG flag becomes 1.

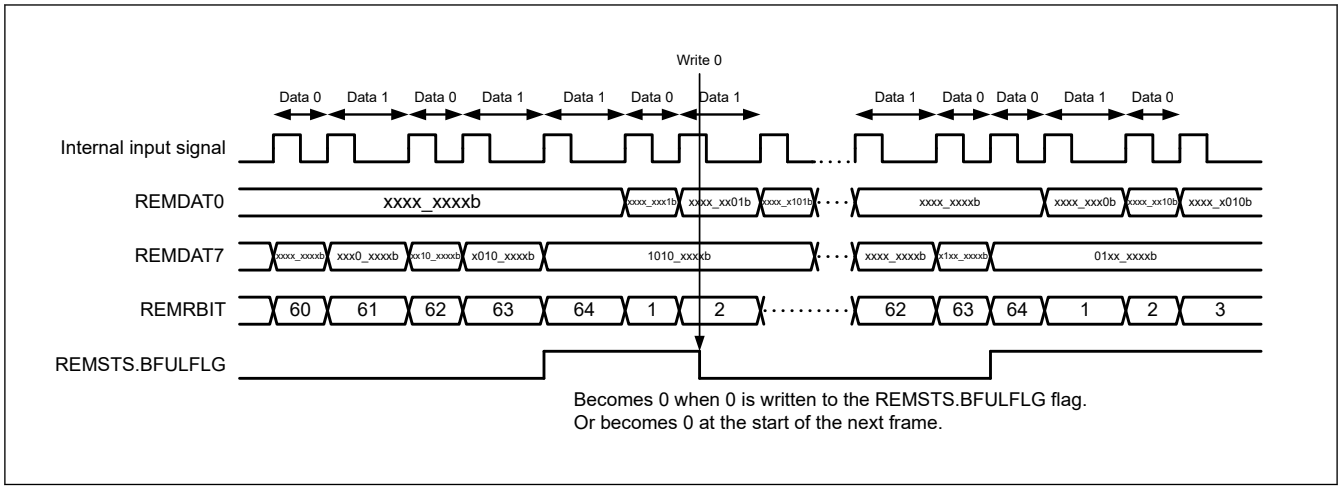


Figure 26.14 REMRBIT register operation (REMSTS.BFULFLG flag = 1)

26.3.9 Compare Function

The REMC has a function to compare the value of the REMCPD register with the value of the REMDAT1 and REMDAT0 registers. As a result of comparison, it can be detected that the first 1 to 16 bits of the remote control signal are the specific values. Figure 26.15 shows the operation timing of the receive buffer and the compare function.

When using the compare function, set registers as shown below:

- Select bits to be compared by setting the REMCPC.CPN[3:0] bits
  (when the setting value is n, bits n to 0 are compared. n: 0 to 15).
- Set the compare data in the REMCPD register.
  When the value of the REMRBIT register becomes the bit count specified by the REMCPC.CPN[3:0] bits, if the value of the REMDAT1 and REMDAT0 registers matches the value of the REMCPD register, the REMSTS.CPFLG flag becomes 1 (compare match).

When the value of the REMRBIT register matches the bit count specified by the REMCPC.CPN[3:0] bits during reception of 64 bits or more, even if the value of the REMDAT1 and REMDAT0 registers matches the value of the REMCPD register, the REMSTS.CPFLG flag does not become 1 (compare match).

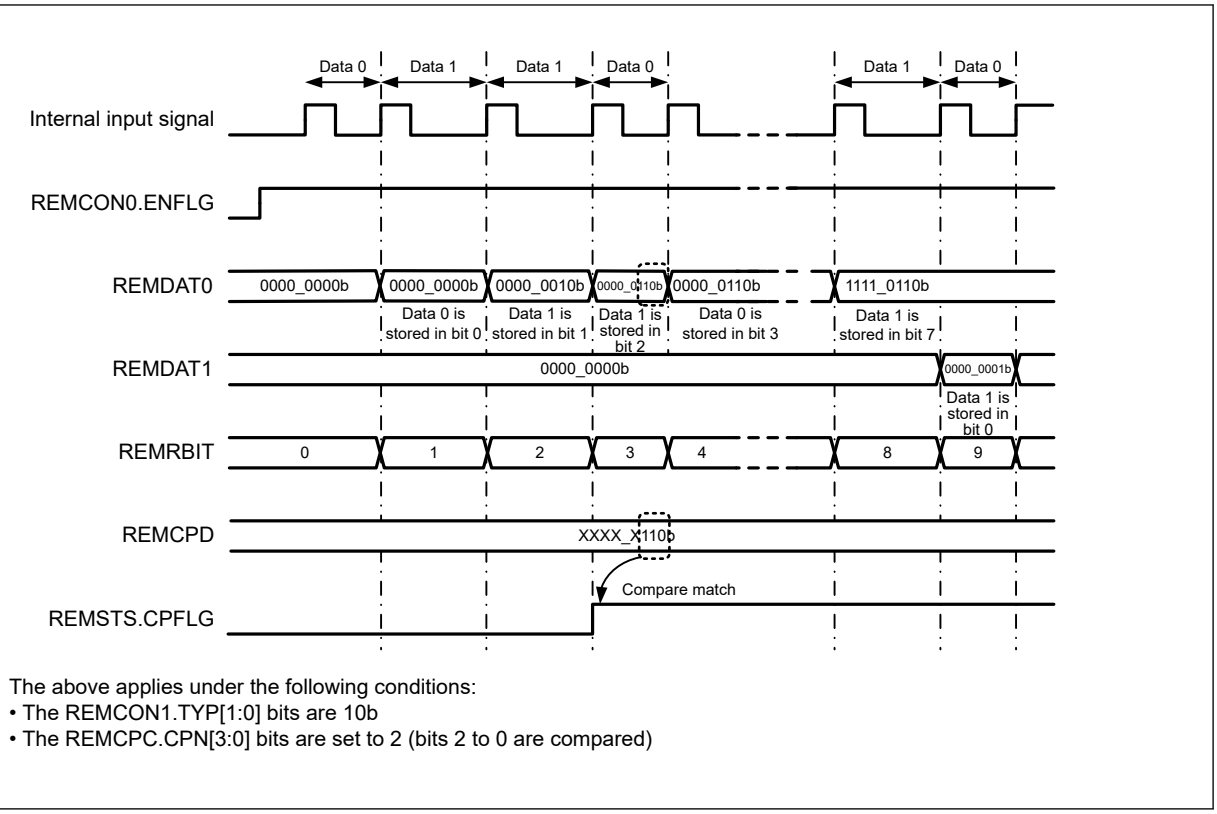


Figure 26.15 Receive buffer and compare function

26.3.10 Error Pattern Reception

When the error pattern is detected during data reception, subsequent operation differs depending on the setting of the REMCON0.EC bit.

Figure 26.16 shows operation of the REMDAT0 and REMRBIT registers when the REMCON0.EC bit is set to 0. If an error is detected while the REMCON0.EC bit is 0, the data when the error is detected is not captured, but data is captured when the data 0 pattern or data 1 pattern is detected later.

Figure 26.17 shows operation of the REMDAT0 and REMRBIT registers when the REMCON0.EC bit is set to 1. If an error is detected while the REMCON0.EC bit is 1, the values of the REMRBIT and REMDAT0 to REMDAT7 registers are not updated even when the data 0 pattern or data 1 pattern is detected later. Once the REMSTS.DRFLG flag is cleared and after data reception is completed, if data reception starts again, the REMSTS.REFLG flag is cleared and data is captured.

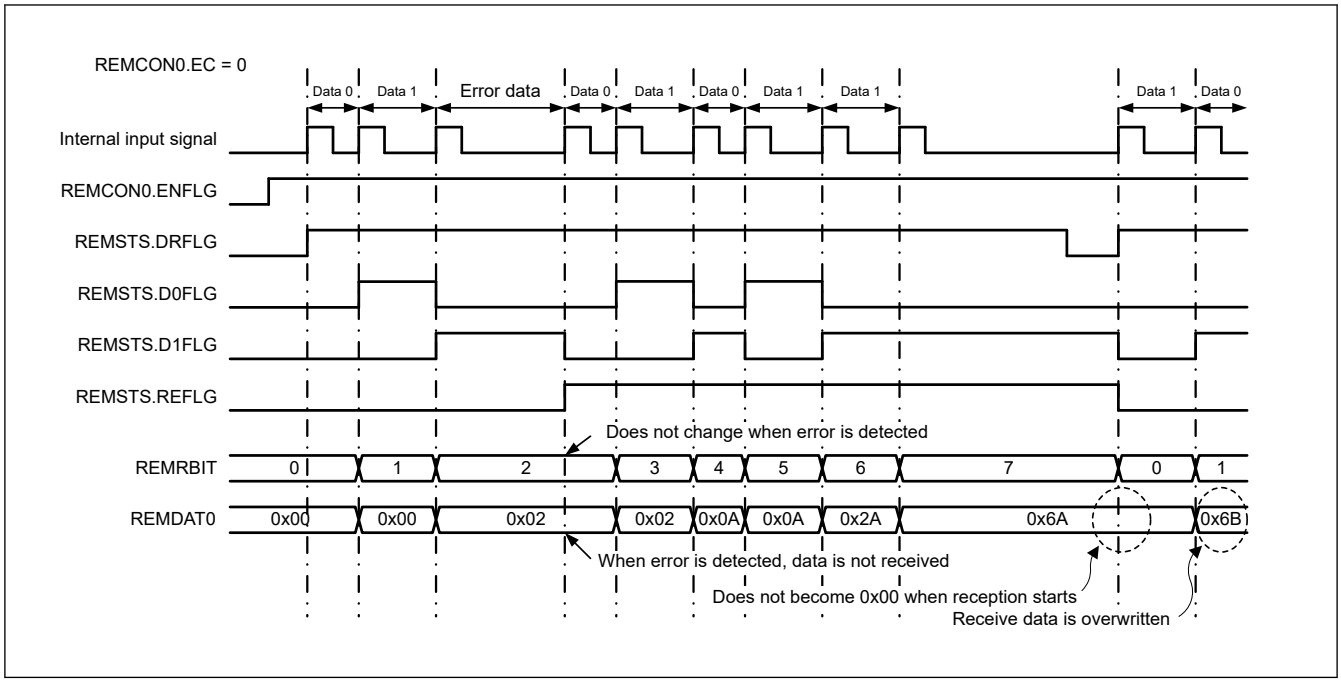


Figure 26.16 REMDAT0 and REMRBIT registers operation on error detection (REMCON0.EC = 0)

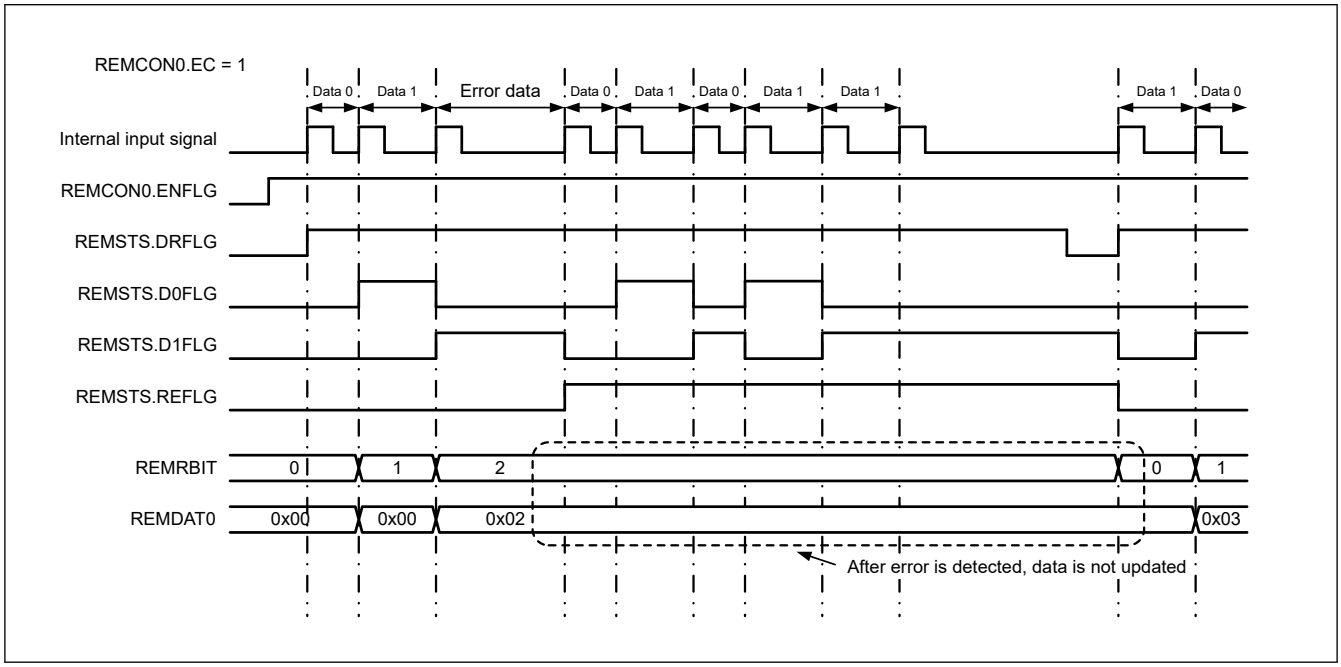


Figure 26.17 REMDAT0 and REMRBIT registers operation on error detection (REMCON0.EC = 1)

26.3.11 Storing Base Timer Value When Pattern is Detected

The measurement result register (REMTIM) holds the base timer value when any of the following patterns is detected. This register allows each pattern width to be measured. Figure 26.18 shows an operation example of the measurement function.

- Header pattern
- Data 0 pattern
- Data 1 pattern
- Special data pattern
- Data pattern other than the above (receive error)

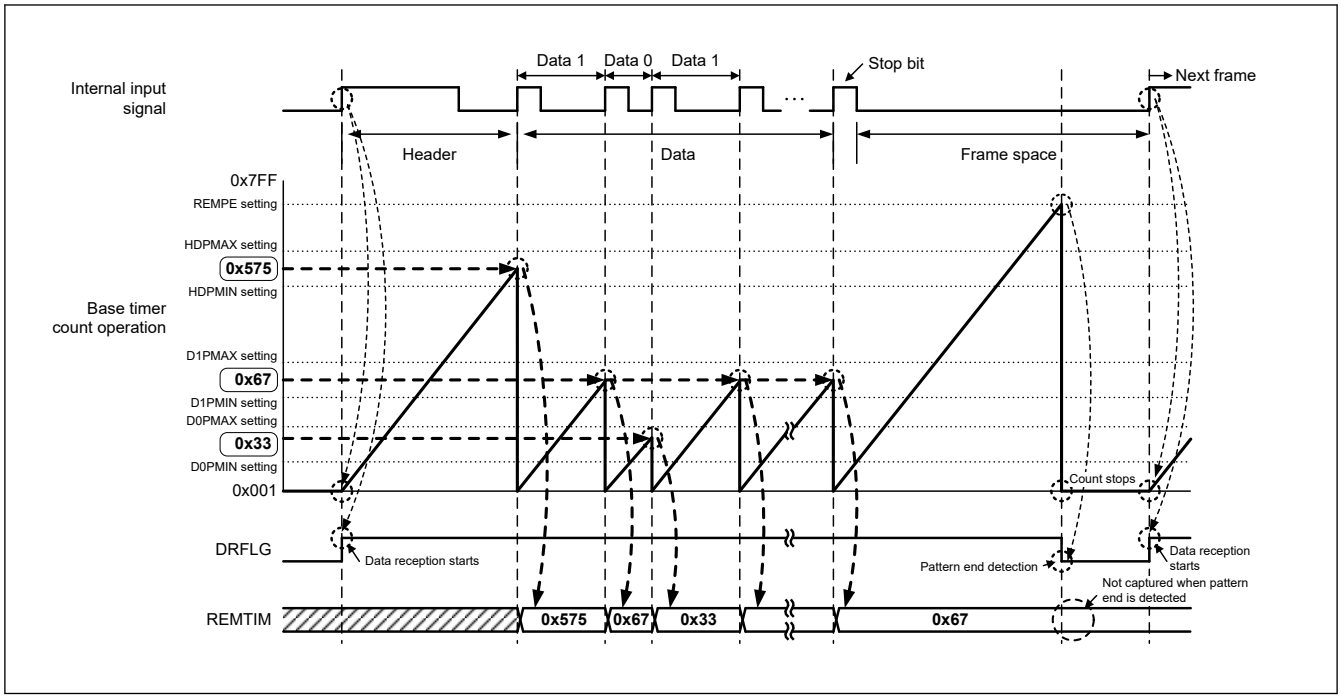


Figure 26.18 Operation example of measurement function

26.3.12 Interrupts

The REMC generates the following interrupt requests: compare match, receive error, data reception complete, receive buffer full, header pattern match, data 0 pattern or data 1 pattern match, and special data pattern match interrupts. An interrupt request is output when the generation conditions for these interrupt requests and the generation conditions for the selected interrupt mode are met.

There are two interrupt modes, normal interrupt mode (OR condition) and sequential interrupt mode (AND condition), which can be selected by using the REMCON1.INTMD bit. In normal interrupt mode, if any of the interrupt generation conditions of sources whose interrupt enable bit of the REMINT register is set to 1 is met, an interrupt request signal (REMC_OUTI) is generated. In sequential interrupt mode, if all the interrupt generation conditions of the sources (compare match, data reception complete, header pattern match, and special data pattern match) whose interrupt enable bit of the REMINT register is set to 1 are met, an interrupt (REMC_OUTI) is generated.

Table 26.6 lists the interrupt sources of the REMC. See section 12, Interrupt Controller Unit (ICU) for details on interrupt control.

Table 26.6 REMC interrupt sources

Interrupt source	Interrupt request generation condition	Interrupt status flag	Interrupt enable bit
Compare match	When the REMSTS.CPFLG flag changes from 0 to 1	REMSTS.CPFLG	REMINT.CPINT
Receive error	When the REMSTS.REFLG flag changes from 0 to 1 (When a receive error is detected)	REMSTS.REFLG	REMINT.REINT
Completion of data reception	When the REMSTS.DRFLG flag changes from 1 to 0	REMSTS.DRFLG	REMINT.DRINT
Receive buffer full	When the REMSTS.BFULFLG flag changes from 0 to 1	REMSTS.BFULFLG	REMINT.BFULINT
Header pattern match	When the REMSTS.HDFLG flag changes from 0 to 1 (When the header pattern is detected)	REMSTS.HDFLG	REMINT.HDINT
Data 0 pattern or data 1 pattern match	• When the REMSTS.D0FLG flag changes from 0 to 1 (When the data 0 pattern is detected) • When the REMSTS.D1FLG flag changes from 0 to 1 (When the data 1 pattern is detected)	REMSTS.D0FLG, REMSTS.D1FLG	REMINT.DINT
Special data pattern match	When the REMSTS.SDFLG flag changes from 0 to 1 (When the special data pattern is detected)	REMSTS.SDFLG	REMINT.SDINT

26.3.13 Snooze Mode Function

In the Snooze mode, the REMC performs remote control data reception operation when detecting a change in the input level of the RIN0 pin. When the timer interrupt (TAU0_ENDI6) is selected as the REMC operating clock, normally the REMC reception operation stops in Software Standby mode. However, using the Snooze mode enables remote control data reception without CPU operation upon detection of a change in the input level of the RIN0 pin.

The Snooze mode can only be used when the timer interrupt (TAU0_ENDI6) is selected as the REMC operating clock. In this case, select a CPU and peripheral hardware clock that can operate in Snooze mode as the count clock for channel 6 of timer array unit.

When using the REMC in Snooze mode, make the following setting before switching to the Software Standby mode.

- Make initial settings of each register immediately before switching to Software Standby mode. To enable remote control data reception in Snooze mode, set the REMCON0.FIL bit to 1 (enables the digital filter) and set the REMSTC.DNFSL bit to 1 (REMCLCLK/REMCSCLK is selected as a sampling clock).
- After the initial settings have been completed, set the REMSTC.SNZON bit to 1, and then set the REMCON1.EN bit to 1.
- The CPU shifts to the Snooze mode on detecting the valid edge of the RIN0 pin input following a transition to the Software Standby mode. When the timer interrupt (TAU0_ENDI6) is supplied as the REMC operating clock, the REMC starts reception.
- After returning to normal operation mode due to generation of an interrupt source specified in the REMINT register such as a compare match interrupt and header pattern match interrupt, set the REMSTC.SNZON bit to 0.

Note: The Snooze mode can only be specified when the high-speed on-chip oscillator clock (HOCO) or middle-speed on-chip oscillator clock (MOCO) is selected for the count clock for channel 6 of timer array unit.

Note: When REMCLCLK/REMCSCLK is selected for the operating clock, remote control data reception operation is available even in Software Standby mode. The current consumption in Software Standby mode is large. Therefore, select a mode in accordance with your system.

Table 26.7 shows the relationship between the conditions for returning from Snooze mode to normal operation mode and interrupt modes.

Table 26.7 Interrupt modes and transition from Snooze mode

	Normal interrupt mode	Sequential interrupt mode
Selectable interrupt sources	Header pattern match Compare match Completion of data reception Special data pattern match Data 0 pattern or data 1 pattern match Receive buffer full Receive error	Header pattern match Compare match Special data pattern match Completion of data reception Data 0 pattern or data 1 pattern match*1 Receive buffer full*1 Receive error*1
Snooze → Normal operation mode returning conditions	• Any of the following interrupts that are enabled in REMINT is generated. Header pattern match Compare match Special data pattern match Completion of data reception Data 0 pattern or data 1 pattern match Receive buffer full Receive error	Either of the following conditions: • Among the following interrupt sources, all of those enabled in REMINT are generated. Header pattern match Compare match Special data pattern match Completion of data reception • Any of the following interrupts that are enabled in REMINT is generated. Data 0 pattern or data 1 pattern match Receive buffer full Receive error
Snooze → Software Standby transition conditions	Data reception is completed while data reception complete interrupt is disabled, and none of the enabled interrupts are generated.	Data reception is completed while data reception complete interrupt is disabled, and not all the enabled interrupts are generated.

Note 1. The AND condition does not apply to these sources in the sequential interrupt mode. As in the normal interrupt mode, the condition is the logical OR of the enabled sources.

Figure 26.19 shows a timing diagram that continues to operate in Snooze mode due to a comparison mismatch, and Figure 26.20 shows a timing diagram that returns from Snooze mode due to a comparison match.

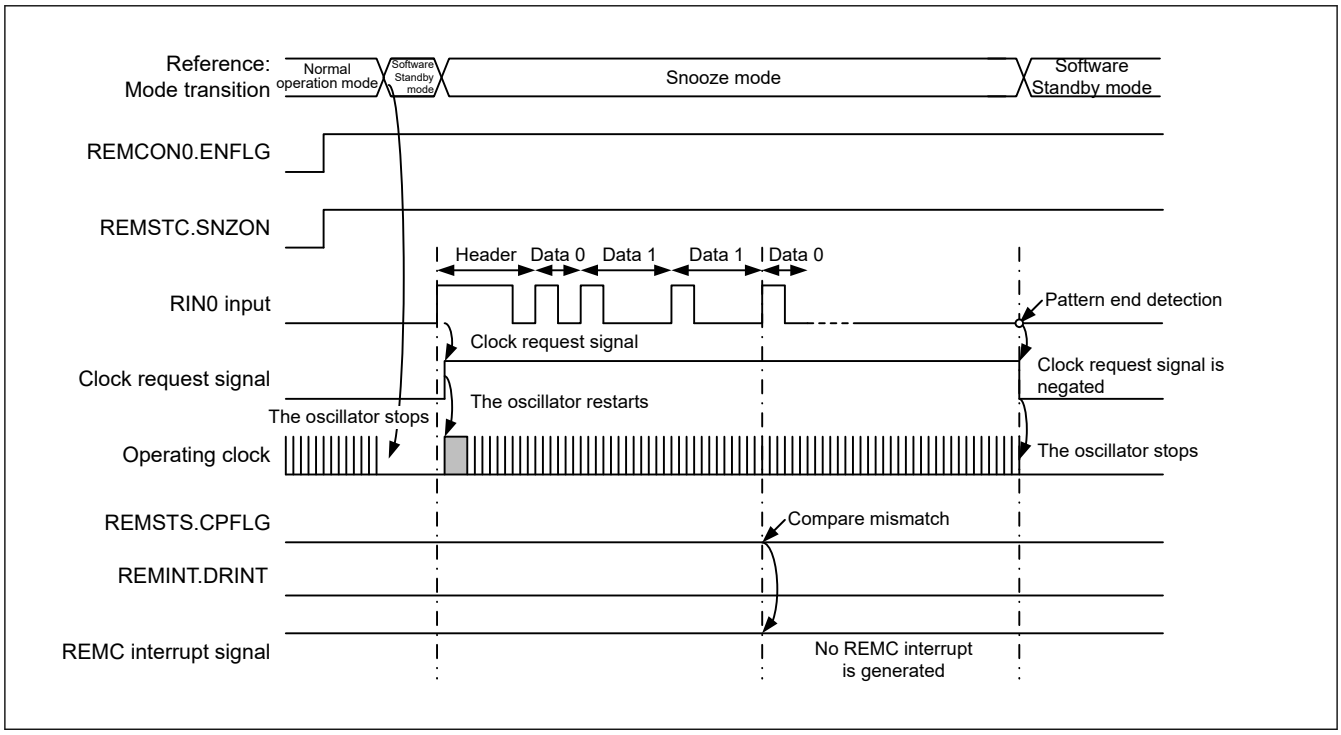


Figure 26.19 Snooze mode continued operation in response to a compare mismatch

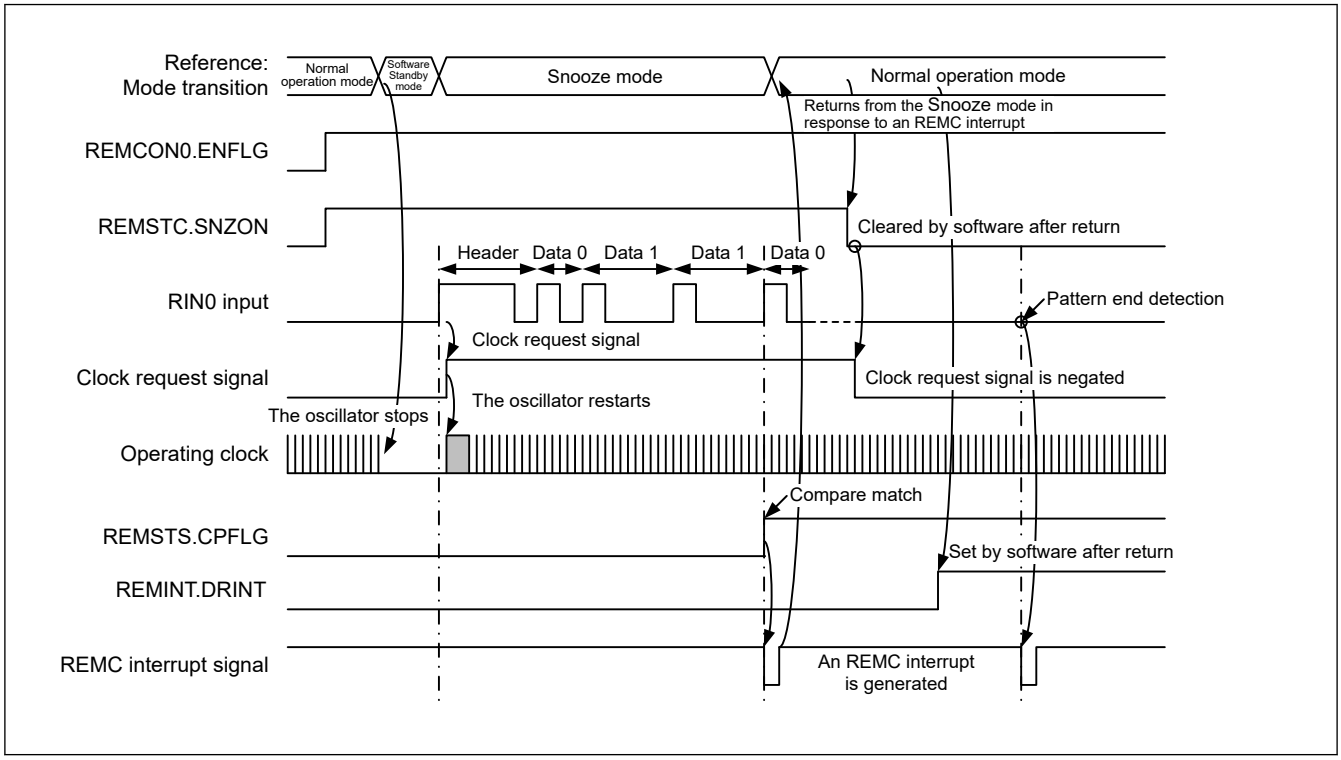


Figure 26.20 Returning from Snooze mode in response to a compare match

Table 26.8 lists the steps for a sample configuration for Snooze mode.

Table 26.8 Example of step for setting up Snooze mode

Step		Process	Detail
CPU is normal operation	<1>	Start operation.	—
	<2>	Disable the remote control reception operation.	If remote control reception operation is not disabled, Clear REMCON1.EN bit. Wait until REMCON0.ENFLG flag is cleared.
	<3>	Initial settings of REMC.	Make initial settings such as the format of the remote control reception waveform and data pattern detection widths.
	<4>	Set TAU0_ENDI6 as the operating clock.	Set REMCON1.CSRC bit.
	<5>	Enable the digital filter.	Set REMCON0.FIL bit.
	<6>	Set REMCLCLK/REMCSCLK as the clock of the digital filter. Enable the Snooze mode operation.	Set the REMSTC.DNFSL bit. Set the REMSTC.SNZON bit.
	<7>	REMINT register setting.	Specify the interrupt to be used to return from Snooze mode to normal operation mode.
	<8>	Initial setting of channel 6 of timer array unit.	Set the timer channel to interval timer mode and select a clock that can operate in Snooze mode as the operating clock. For details, see section 18, Timer Array Unit (TAU).
	<9>	Enable the operation of channel 6 of timer array unit.	Set TS0.TS[6] bit.
	<10>	Enable the remote control reception operation.	Set REMCON1.EN bit.
	<11>	Enable interrupts.	Set the associated IELSRn register and enable interrupt request. For details, see section 12, Interrupt Controller Unit (ICU).
	<12>	Software Standby mode transition.	WFI instruction execution.
CPU is Software Standby mode	<13>	Wait for a valid edge of the RIN0 pin to be detected.	The CPU shifts to the Snooze mode on detecting the valid edge of the RIN0 pin. The clock is supplied to channel 6 of the timer array unit, and TAU0_ENDI6 is generated. The REMC uses TAU0_ENDI6 as the operating clock and performs reception operation.
CPU is Snooze mode	<14>	Check for REMINT to generate an enabled interrupt. If generated, go to <16>.	The mode returns to normal operation mode when the specified interrupt is generated.
	<15>	Completion of reception. Go to <13>.	The mode shifts to Software Standby mode when reception is completed while the specified interrupt is not generated.
CPU is normal operation	<16>	Disable the Snooze mode operation.	Clear REMSTC.SNZON bit.
	<17>	Execute next processing.	—

26.4 Usage Notes

26.4.1 Register Access when Starting Operation of the Remote Control Signal Receiver

The REMCON1.EN bit controls starting and stopping of operation of the remote control signal receiver. The REMCON0.ENFLG flag indicates that the operation is enabled or disabled. After the REMCON1.EN bit is set to 1 (operation enabled), it takes up to zero to one cycle of the operating clock before the REMC circuit starts operating and the REMCON0.ENFLG flag becomes 1. During this period, do not access the REMC related registers (listed in section 26.2.1. REMCON0 : Function Select Register 0 to section 26.2.20. REMTIM : Measurement Result Register) except for the REMCON0.ENFLG flag.

26.4.2 Timing of Changing the Register Values

Change the following registers only when the REMCON1.EN bit and REMCON0.ENFLG flag are both 0 (REMC is stopped).

- REMCON0 register
- REMCON1 register (except for bits 0 to 2)
- REMINT register (except for bits 2 and 5)
- REMCPC register
- REMCPD register
- Pattern width setting registers for header, data 0, data 1, and special data patterns
- Pattern end setting register
- REMSTC register

When rewriting the REMCON1.TYP[1:0] bits while the REMCON1.EN bit or REMCON0.ENFLG flag is 1 (REMC is operating), change the values of these bits one bit at a time. If the REMCON1.TYP[1:0] bits are rewritten when the REMCON0.INFLG flag changes, the signal captured into the remote control signal receiver may be undefined.

After 0 is written to bit 0 in the REMDAT0 or REMRBIT register or the REMSTS.BFULFLG flag, do not write 0 to the same bit again for two cycles of the operating clock. If 0 is written when the REMCON0.INFLG flag changes, the values of the REMDATj and REMRBIT registers and the REMSTS.BFULFLG flag may be undefined.

26.4.3 RIN0 Input Control

If the REMCON0.FILSEL, FIL, or INV bit is rewritten, the signal captured into the remote control signal receiver is undefined for three cycles of the digital filter sampling clock.

26.4.4 Changing the Operating Clock

When the REMCON1.CSRC bit is rewritten, set the following registers again: REMCON0, REMCON1, REMINT, REMCPC, REMCPD, REMPE, and header, data 0, data 1, and special data pattern width setting registers.

26.4.5 Reading Registers

When the following registers are read while data changes, an undefined value may be read.

Flags in the REMCON0 and REMSTS registers (except for the REMSTS.DRFLG flag) and registers REMTIM, REMDAT0 to REMDAT7, and REMRBIT

Follow the procedures below to avoid reading an undefined value.

- Using an interrupt
  Set the REMINT.DRINT bit to 1 (data reception complete interrupt enabled) and read the registers within the REMC interrupt routine.
- Polling by software
  1. Poll the REMSTS.DRFLG flag.
  2. When the REMSTS.DRFLG flag becomes 1, poll this flag until it becomes 0.
  3. Read the necessary content of the registers when the REMSTS.DRFLG flag becomes 0.

27. Cyclic Redundancy Check (CRC)

27.1 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 27.1 lists the CRC calculator specifications and Figure 27.1 shows a block diagram.

Table 27.1 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]*2 • $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$ (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	—

Note 1. This function cannot divide data used in CRC calculations. Write data in 8-bit or 32-bit units.
Note 2. 32-bit CRC is only supported when the snoop function is disabled (CRCCR1.CRCSEN=0).

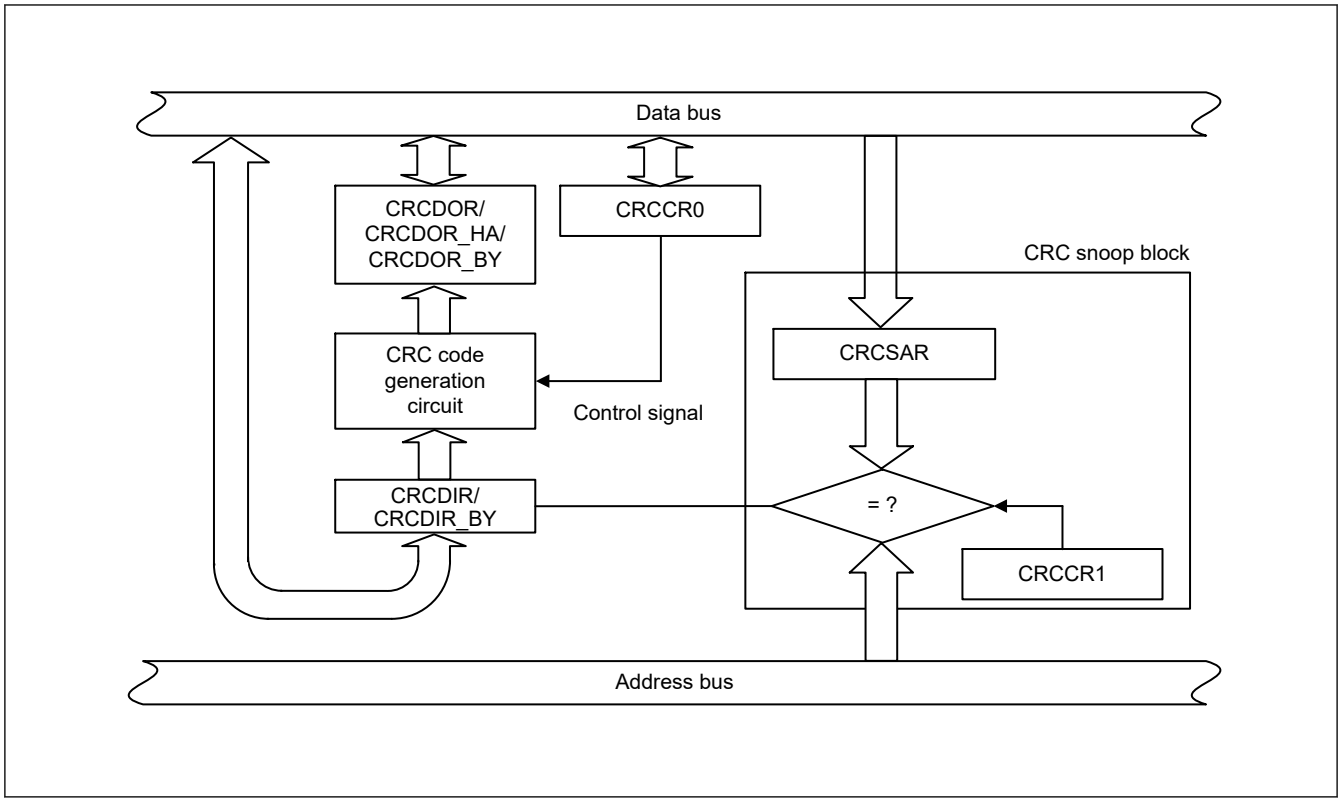


Figure 27.1 CRC calculator block diagram

27.2 Register Descriptions

27.2.1 CRCCR0 : CRC Control Register 0

Base address: CRC = 0x4007_4000

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 27.3. 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.

27.2.2 CRCCR1 : CRC Control Register 1

Base address: CRC = 0x4007_4000

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
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.

27.2.3 CRCDIR/CRCDIR_BY : CRC Data Input Register

Base address: CRC = 0x4007_4000

Offset address: 0x04

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	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

27.2.4 CRCDOR/CRCDOR_HA/CRCDOR_BY : CRC Data Output Register

Base address: CRC = 0x4007_4000

Offset address: 0x08

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	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

27.2.5 CRCSAR : Snoop Address Register

Base address: CRC = 0x4007_4000

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 SDRmn register address in the SAU 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 address can be used for the CRCSA[13:0] bits:

- 0x4009_4000: SAU0.SDR00
- 0x4009_4002: SAU0.SDR01
- 0x4009_4004: SAU0.SDR02
- 0x4009_4006: SAU0.SDR03
- 0x4009_4100: SAU1.SDR10
- 0x4009_4102: SAU1.SDR11

27.3 Operation

27.3.1 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 27.2 and Figure 27.3 show the LSB-first and MSB-first data transmission examples respectively. Figure 27.4 and Figure 27.5 show the LSB-first and MSB-first data reception examples.

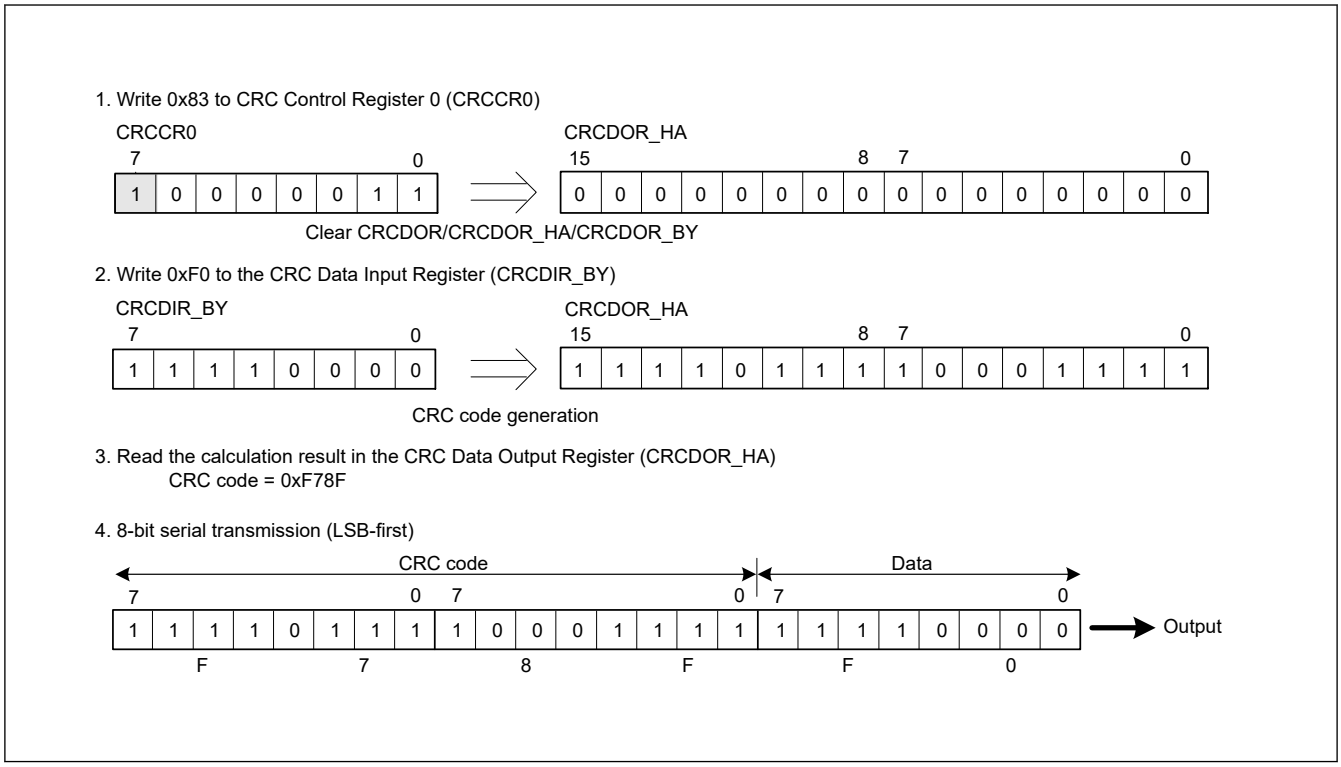


Figure 27.2 LSB-first data transmission

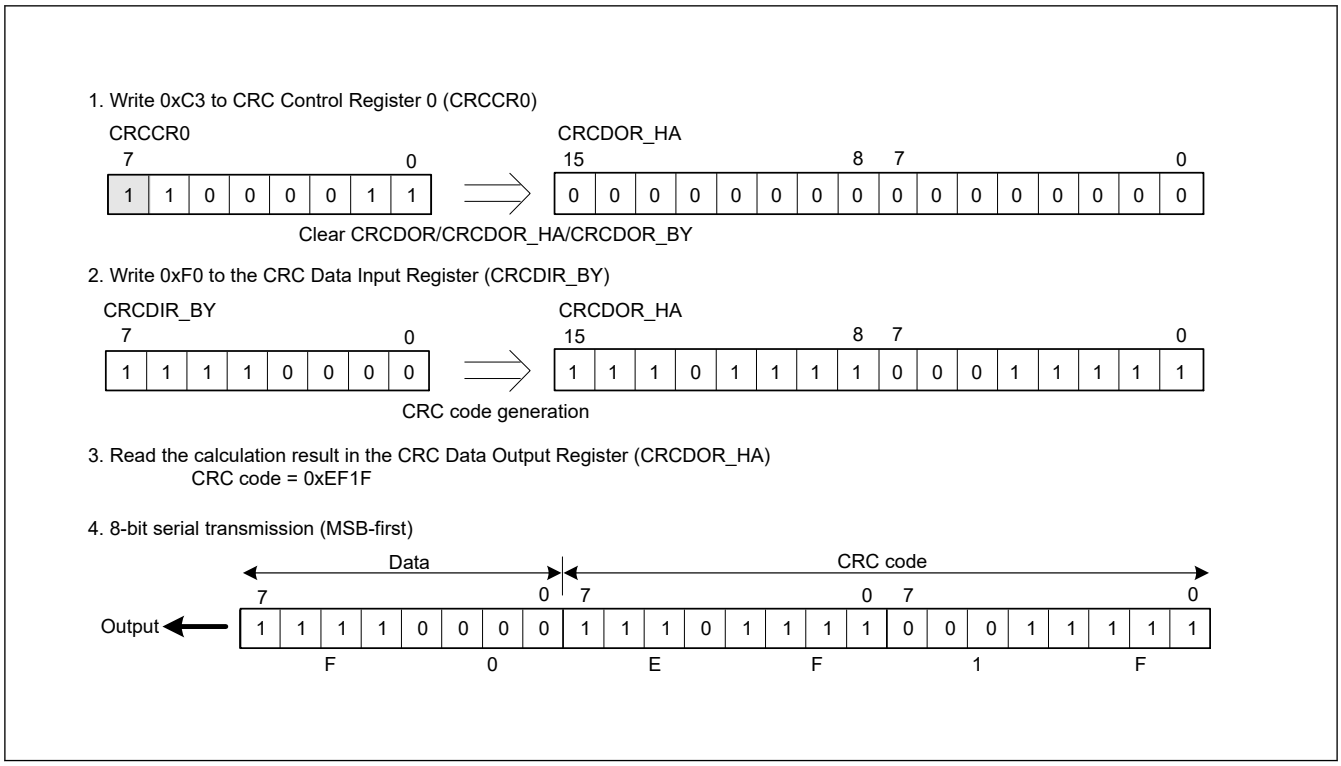


Figure 27.3 MSB-first data transmission

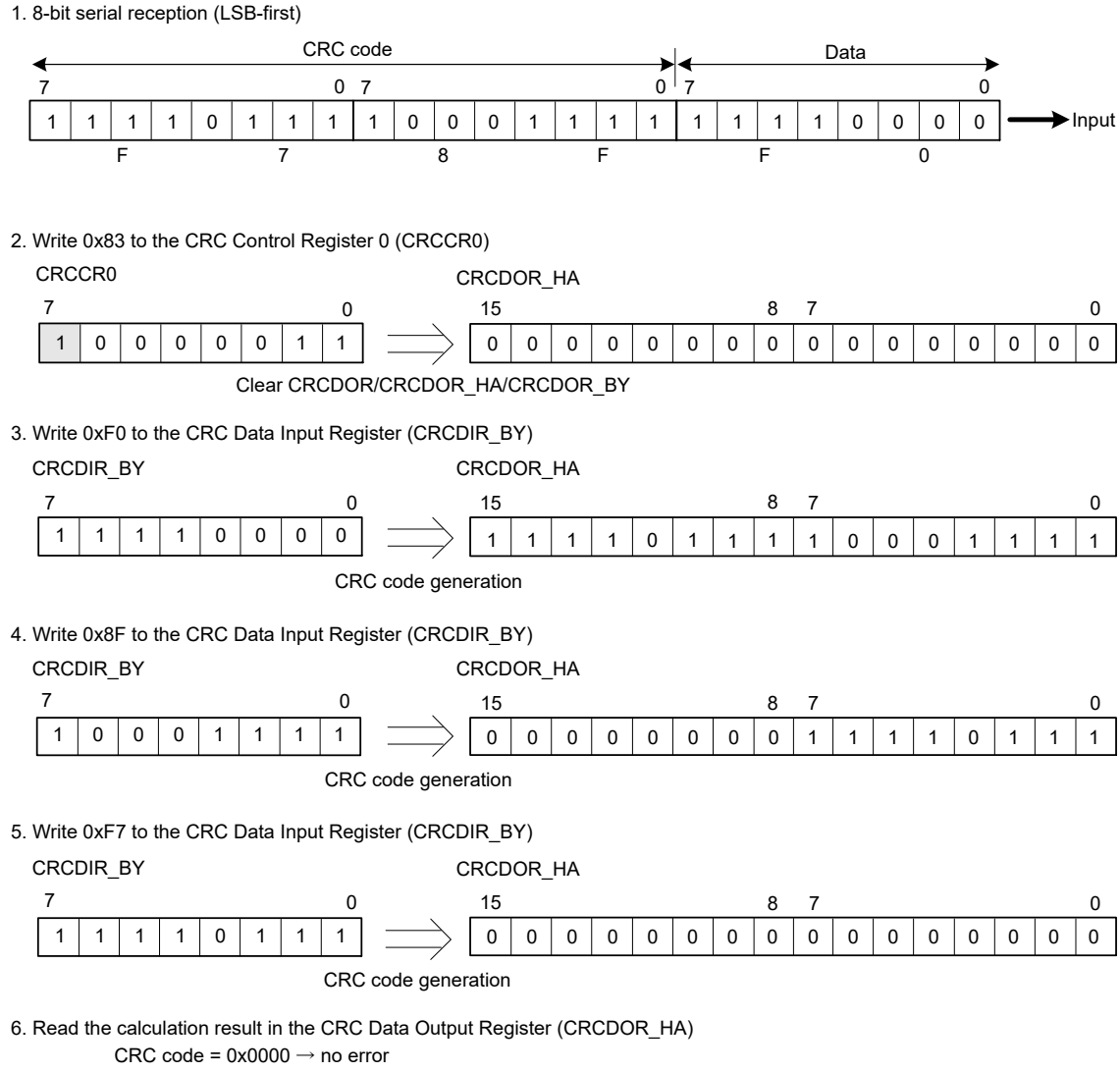


Figure 27.4 LSB-first data reception

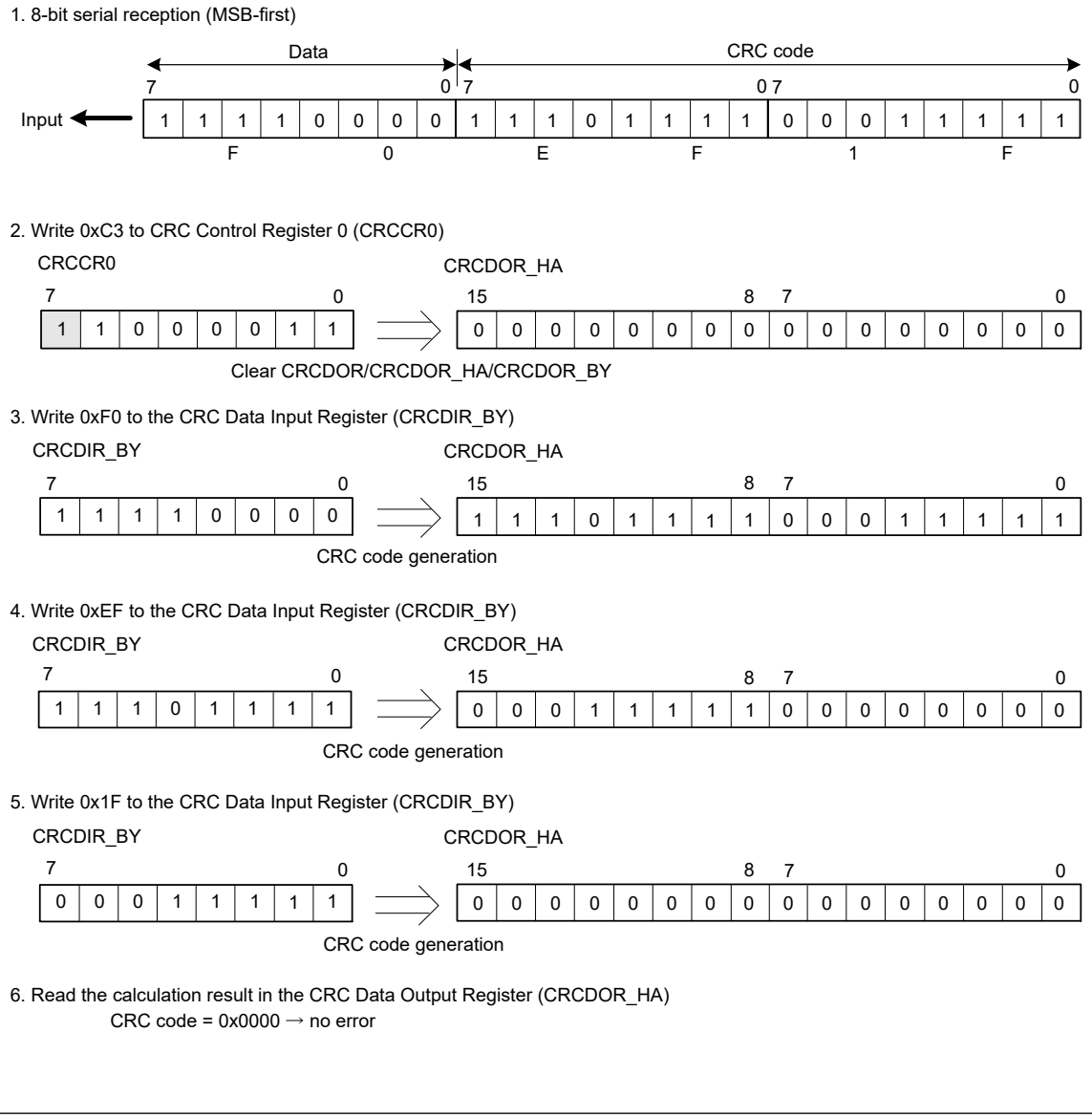


Figure 27.5 MSB-first data reception

27.3.2 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_BY register. All I/O register specified in the section 27.2.5. CRCSAR : Snoop Address Register are subject to the CRC snoop. The CRC snoop function is useful in monitoring writes and reads of the SAUm.SDRmn register.

To use this function, write the lower address 14 bits of a specific register to the CRCSA[13:0] bits 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 and DTC, the CRC calculator stores the data in the CRCDIR_BY 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 and DTC, the CRC calculator stores the data in the CRCDIR_BY 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). Accessing the target register in words (32 bits) by using CRC-32 and CRC-32C generating polynomial is prohibited.

27.4 Usage Notes

27.4.1 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.

27.4.2 Note on Transmission

The transmission sequence for the CRC code differs based on whether the transmission is LSB-first or MSB-first. Figure 27.6 shows an LSB-first and MSB-first data transmission.

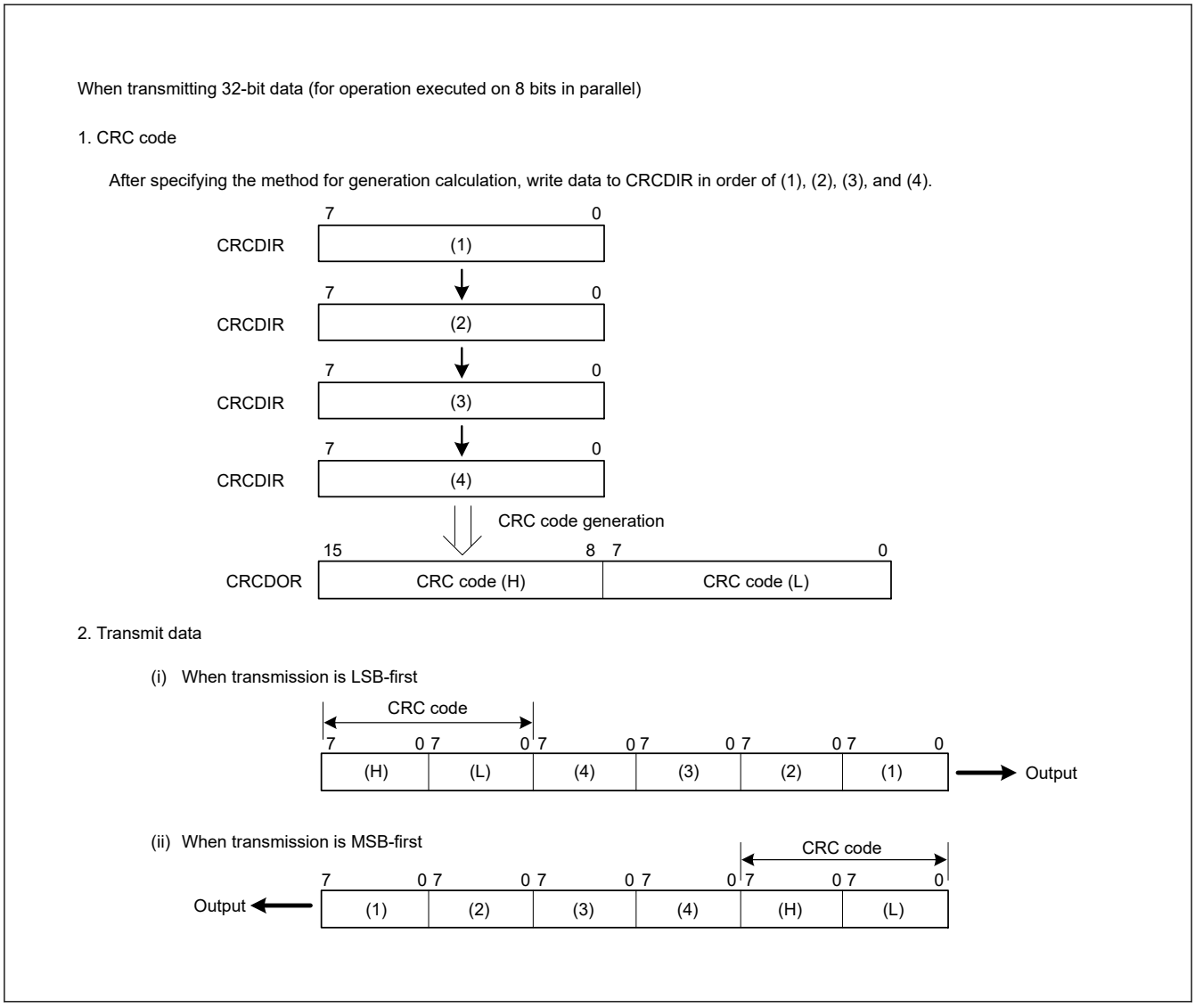


Figure 27.6 LSB-first and MSB-first data transmission

28. True Random Number Generator (TRNG)

28.1 Overview

The true random number generator generates 32-bit random number seeds (which are true random numbers).

28.2 Register Descriptions

28.2.1 TRNGSCR0 : Random Number Seed Command Register 0

Base address: TRNG = 0x4009_1100

Offset address: 0x02

Bit position:	7	6	5	4	3	2	1	0
Bit field:	RDY	—	—	—	EN	ST	—	—
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	ST	Trigger to start generating a random number seed 0: The trigger is inactive. 1: Starts generation of a random number seed.	R/W
3	EN	Control over operation of the true random number generator 0: Stops the true random number generator. 1: Enables the true random number generator.	R/W
6:4	—	These bits are read as 0. The write value should be 0.	R/W
7	RDY	Random number seed generation status flag 0: A random number seed has not been generated or four rounds of reading from the TRNGSDR register have been completed. 1: A random number seed has been generated.	R/W

The TRNGSCR0 register controls operation of the true random number generator. Setting the ST bit to 1 after having set the EN bit to 1 starts the generation of a random number seed. When the true random number generator finishes generating the random number seed, the RDY bit is set to 1.

Since the ST bit serves as the trigger for starting the generation of a random number seed, it is cleared to 0 immediately after 1 having been written to it.

28.2.2 TRNGSCR1 : Random Number Seed Command Register 1

Base address: TRNG = 0x4009_1100

Offset address: 0x03

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

Bit	Symbol	Function	R/W
0	INTEN	TRNG Interrupt 0: Disabled 1: Enabled	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
4	—	This bit is read as 1. The write value should be 1.	R/W

Bit	Symbol	Function	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

28.2.3 TRNGSDR : Random Number Seed Data Register

Base address: TRNG = 0x4009_1100

Offset address: 0x00

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

Bit	Symbol	Function	R/W
7:0	n/a	Random number seed data	R

Note: When the value of the TRNGSCR0.RDY bit is 0, the value in the TRNGSDR register is 0x00.

The TRNGSDR register is an 8-bit register that holds the bytes of random number seeds generated by the true random number generator. The random number seed can be read from this register after the TRNGSCR0.RDY bit is set to 1. As a random number seed consists of 32 bits, four rounds of access to the register are required for each seed. The TRNGSCR0.RDY bit is cleared to 0 following the four rounds of access.

28.3 Operations of the True Random Number Generator

Table 28.1 shows the procedure for using the true random number generator to generate a random number seed.

Table 28.1 Procedure for using the true random number generator to generate a random number seed

Step		Process	Detail
Procedure for using the true random number generator to generate a random number seed	<1>	Start of generation of a random number seed.	—
	<2>	Set the TRNGSCR0 register.	Set the TRNGSCR0.EN bit to 1 to enable the true random number generator.
	<3>	Set the TRNGSCR1 register.	There are two operations for the generation of a random number seed. 1. Polling operation: This process is not necessary. 2. Interrupt operation: Set the TRNGSCR1.INTEN bit to 1 to enable TRNG interrupt output.
	<4>	Set the TRNGSCR0 register.	Set the TRNGSCR0.ST bit to 1 to start generation of a random number seed.
	<5>	Read from the TRNGSDR register.	There are two operations for the generation of a random number seed: 1. Polling operation: Read the TRNGSDR register for 4 times after the TRNGSCR0.RDY bit becomes 1. 2. Interrupt operation: Read the TRNGSDR register for 4 times after TRNG interrupt is generated.
	<6>	Set the TRNGSCR0 register.	Set the TRNGSCR0.EN bit to 0 to disable the true random number generator.
	<7>	Clear the TRNGSCR1 register.	There are two operations for the generation of a random number seed: 1. Polling operation: This process is not necessary. 2. Interrupt operation: Clear the TRNGSCR1.INTEN bit to 0 to disable TRNG interrupt output.
	<8>	End generation of a random number seed.	—

29. 12-bit A/D Converter (ADC12)

The number of analog input channels of the 12-bit A/D converter differs, depending on the product. Table 29.1 shows the number of analog input channels of the 12-bit A/D converter for each product.

Table 29.1 Number of analog input channels of the 12-bit A/D converter for each product

Number of pins	16-pin	24-pin	32-pin	48-pin
Number of analog input channels	4 (ANI0, ANI1, ANI16, ANI17)	6 (ANI0, ANI1, ANI16 to ANI19)	8 (ANI0, ANI1, ANI4, ANI5, ANI16 to ANI19)	10 (ANI0 to ANI5, ANI16 to ANI19)

29.1 Function of 12-bit A/D Converter

The 12-bit A/D converter is used to convert analog input signals into digital values, and is configured to control up to 10 channels of 12-bit A/D converter analog inputs (ANI0 to ANI5 and ANI16 to ANI19). 12-bit, 10-bit, or 8-bit resolution can be selected by the ADTYP[1:0] bits of the 12-bit A/D converter mode register 2 (ADM2). The 12-bit A/D converter has the following function:

- 12-bit, 10-bit or 8-bit resolution A/D conversion
  12-bit, 10-bit, or 8-bit resolution A/D conversion is carried out repeatedly for one analog input channel selected from ANI0 to ANI5 and ANI16 to ANI19. Each time an A/D conversion operation ends, an interrupt request signal (ADC_ENDI) is generated (when in the select mode).

Various A/D conversion modes can be specified by using the mode combinations shown in Table 29.2.

Table 29.2 A/D conversion modes

A/D conversion modes		Specifications
Trigger mode	Software trigger no-wait mode	Conversion is started by setting the ADCE bit to 1 by software, and then setting ADCS to 1 after the A/D power supply stabilization wait time has passed.
	Software trigger wait mode	The power is turned on by setting the ADCS bit to 1 by software while A/D conversion is stopped and conversion is then started automatically after the A/D power supply stabilization wait time has passed.
	Hardware trigger no-wait mode	Conversion is started by detecting a hardware trigger.
	Hardware trigger wait mode	The power to the 12-bit A/D converter is turned on by detecting a hardware trigger while the 12-bit A/D converter is off and in the conversion standby state, and conversion is then started automatically after the stabilization wait time passes. When using the Snooze mode function, specify the hardware trigger wait mode.
Channel selection mode	Select mode	A/D conversion is performed on the analog input of one selected channel.
	Scan mode	A/D conversion is performed on the analog input of four channels in order. Four consecutive channels can be selected from ANI0 to ANI5 as analog input channels.
Conversion operation mode	One-shot conversion mode	A/D conversion is performed on the selected channel once.
	Sequential conversion mode	A/D conversion is sequentially performed on the selected channels until it is stopped by software.

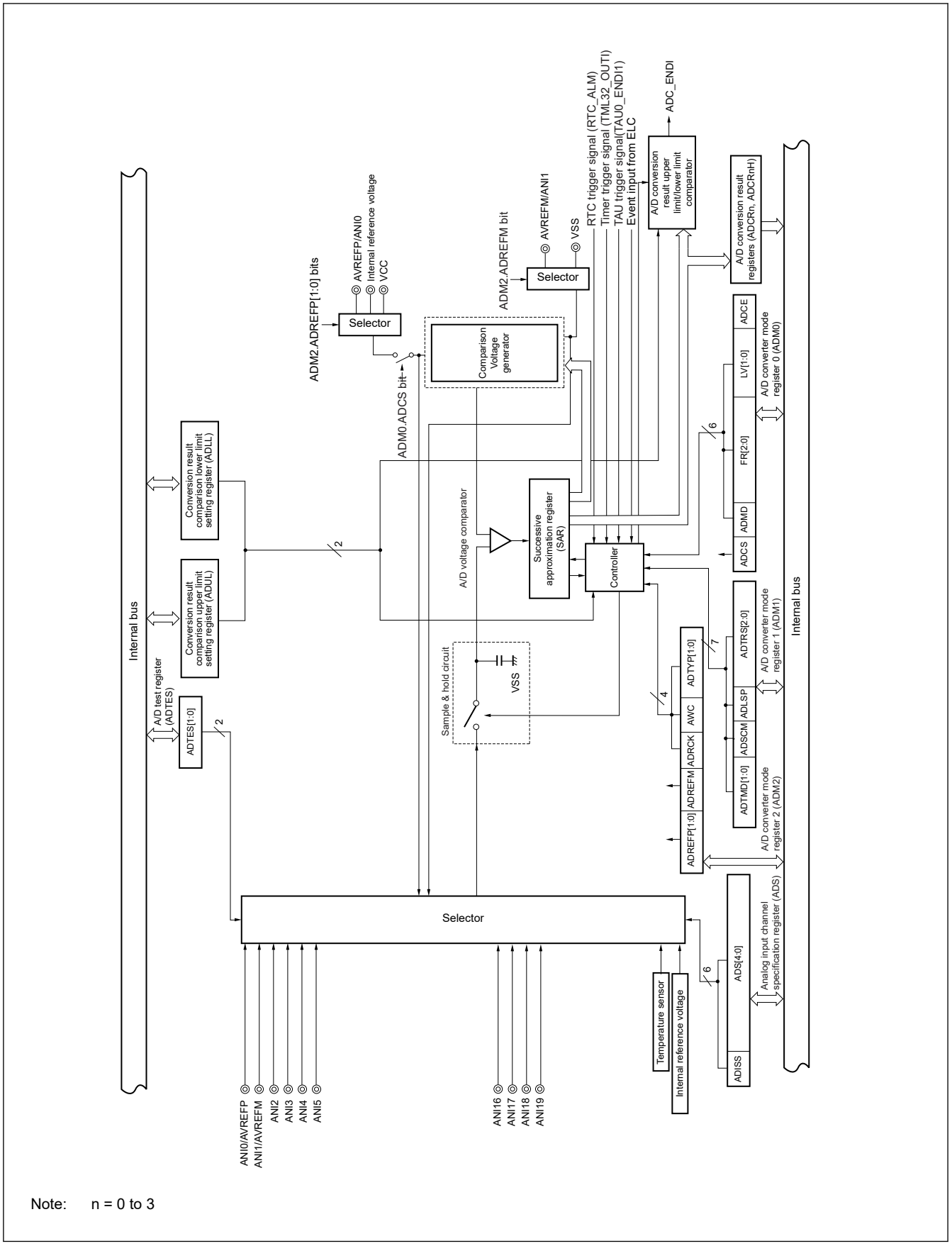
Table 29.3 shows the sampling clock cycle for each operation voltage mode.

Table 29.3 Sampling clock cycle for each operation voltage mode

Operation voltage mode*1	Sampling clock cycles	
Normal mode 1	43 f_{AD}	Set the number of sampling clock cycles so that the sampling capacitor is sufficiently charged according to the output impedance of the analog input source.
Normal mode 2	160 f_{AD}	
Low voltage mode 1	53 f_{AD}	
Low voltage mode 2	80 f_{AD}	

Note 1. The operation mode that can be selected differs depending on the analog input channel, V_{CC} voltage, AV_{REFP} voltage, trigger mode, and PCLKB. See section 29.3.1. ADM0 : A/D Converter Mode Register 0 for details.

Figure 29.1 shows a block diagram of a 12-bit A/D converter.



Note: n = 0 to 3

Figure 29.1 Block diagram of 12-bit A/D converter

29.2 Configuration of 12-bit A/D Converter

The 12-bit A/D converter includes the following hardware.

1. ANI0 to ANI5 and ANI16 to ANI19 pins
   These are the analog input pins of the 10 channels of the 12-bit A/D converter. They input analog signals to be converted into digital signals. Pins other than the one selected as the analog input pin can be used as I/O port pins.
2. Sample & hold circuit
   The sample & hold circuit samples each of the analog input voltages sequentially sent from the input circuit, and sends them to the A/D voltage comparator. This circuit also holds the sampled analog input voltage during A/D conversion.
3. A/D voltage comparator
   This A/D voltage comparator compares the voltage generated from the voltage tap of the comparison voltage generator with the analog input voltage. If the analog input voltage is found to be greater than the reference voltage (1/2 AV_{REF}) as a result of the comparison, the most significant bit (MSB) of the successive approximation register (SAR) is set. If the analog input voltage is less than the reference voltage (1/2 AV_{REF}), the MSB of the SAR is reset. After that, bit 10 of the SAR register is automatically set, and the next comparison is made. The voltage tap of the comparison voltage generator is selected by the value of bit 11, to which the result has already been set.
   Bit 11 = 0: (1/4 AV_{REF})
   Bit 11 = 1: (3/4 AV_{REF})
   The voltage tap of the comparison voltage generator and the analog input voltage are compared and bit 10 of the SAR register is manipulated according to the result of the comparison.
   Analog input voltage ≥ Voltage tap of comparison voltage generator: Bit 10 = 1
   Analog input voltage ≤ Voltage tap of comparison voltage generator: Bit 10 = 0
   Comparison is continued like this to bit 0 of the SAR register.
   AV_{REF}: The '+' side reference voltage of the 12-bit A/D converter. This can be selected from AVREFP, the internal reference voltage[*1], and VCC.

   Note 1. For details about the internal reference voltage, see section 37, Electrical Characteristics.
4. Comparison voltage generator
   The comparison voltage generator generates the voltage to be compared with the input from an analog input pin.
5. Successive approximation register (SAR)
   The SAR is used to set voltage tap data whose values from the comparison voltage generator match the voltage values of the analog input pins, one bit at a time starting from the most significant bit (MSB).
   If data is set in the SAR register all the way to the least significant bit (LSB) (end of A/D conversion), the contents of the SAR register (conversion results) are held in the A/D conversion result register (ADCRn). When all the specified A/D conversion operations have ended, an A/D conversion end interrupt request signal (ADC_ENDI) is generated.
6. 12-bit or 10-bit A/D conversion result register (ADCRn)
   Each time A/D conversion ends, the conversion result is loaded from the successive approximation register, and then operation is performed as follows:
   When A/D conversion with 12-bit resolution is selected, it holds the A/D conversion result in its lower 12 bits (the higher 4 bits are fixed to 0).
   When A/D conversion with 10-bit resolution is selected, it holds the A/D conversion result in its higher 10 bits (the lower 6 bits are fixed to 0).
7. 8-bit A/D conversion result register (ADCRnH)
   The A/D conversion result is loaded from the successive approximation register to this register each time A/D conversion is completed, and the ADCRnH register holds the higher 8 bits of the A/D conversion result.
8. Controller
   This circuit controls the conversion time of an analog input signal that is to be converted into a digital signal, as well as starting and stopping of the conversion operation. When A/D conversion has been completed, this controller generates ADC_ENDI through the A/D conversion result upper limit/lower limit comparator.
9. AV_{REFP} pin
   This pin inputs an external reference voltage (AV_{REFP}).
   If using AV_{REFP} as the '+' side reference voltage of the 12-bit A/D converter, set the ADREFP[1:0] bits of A/D Converter Mode Register 2 (ADM2) to 01b.

The analog signals input to ANI2 to ANI5 and ANI16 to ANI19 are converted to digital signals based on the voltage applied between AV_{REFP} and the '-' side reference voltage (AV_{REFM}/V_{SS}).

In addition to AV_{REFP}, it is possible to select V_{CC} or the internal reference voltage[*1] as the '+' side reference voltage of the 12-bit A/D converter.

10. AV_{REFM} pin

This pin inputs an external reference voltage (AV_{REFM}). To use AV_{REFM} as the '-' side reference voltage of the 12-bit A/D converter, set the ADREFM bit of the ADM2 register to 1.

In addition to AV_{REFM}, it is possible to select V_{SS} as the '-' side reference voltage of the 12-bit A/D converter.

29.3 Registers to Control the 12-bit A/D Converter

The following registers are used to control the 12-bit A/D converter.

- section 29.3.1. ADM0 : A/D Converter Mode Register 0
- section 29.3.2. ADM1 : A/D Converter Mode Register 1
- section 29.3.3. ADM2 : A/D Converter Mode Register 2
- section 29.3.4. ADCR/ADCRn: 12-bit or 10-bit A/D Conversion Result Register n (n = 0 to 3)
- section 29.3.5. ADCRH/ADCRnH : 8-bit A/D Conversion Result Register n (n = 0 to 3)
- section 29.3.6. ADS : Analog Input Channel Specification Register
- section 29.3.7. ADUL : Conversion Result Comparison Upper Limit Setting Register
- section 29.3.8. ADLL : Conversion Result Comparison Lower Limit Setting Register
- section 29.3.9. ADTES : A/D Test Register

29.3.1 ADM0 : A/D Converter Mode Register 0

Base address: ADC120 = 0x4009_C000

Offset address: 0x00

Bit position:	7	6	5	4	3	2	1	0
Bit field:	ADCS	ADMD	FR[2:0]			LV[1:0]		ADCE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ADCE	A/D voltage comparator operation control[*2] 0: Stops A/D voltage comparator operation 1: Enables A/D voltage comparator operation	R/W
2:1	LV[1:0][*1]	Select Operation voltage mode 0 0: Normal mode 1 0 1: Normal mode 2 1 0: Low voltage mode 1 1 1: Low voltage mode 2	R/W
5:3	FR[2:0][*1]	Select Conversion Clock (f_{AD}) 0 0 0: PCLKB/32 0 0 1: PCLKB/16 0 1 0: PCLKB/8 0 1 1: PCLKB/4 1 0 0: PCLKB/2 1 0 1: PCLKB Others: Setting prohibited.	R/W
6	ADMD	Specification of the A/D conversion channel selection mode 0: Select mode 1: Scan mode	R/W

Bit	Symbol	Function	R/W
7	ADCS	A/D conversion operation control 0: Stops conversion operation [When read] ● Conversion is stopped or in standby 1: Enables conversion operation [When read] ● While in the no wait mode (both software and hardware trigger mode): Conversion is enabled ● While in the wait mode (both software and hardware trigger mode): A/D power supply stabilization wait time + conversion	R/W

Note: The ADMD, FR[2:0], and LV[1:0] bits should be changed at least 0.2 µs after conversion stops (ADCS = 0, ADCE = 0).
Note: After changing ADMD, FR[2:0] and LV[1:0] bits, set ADCE = 1 or ADCS = 1 at least 4.8 µs later.
Note: When setting ADCE = 1 or ADCS = 1 from the conversion stop state (ADCS = 0, ADCE = 0), wait at least 5 µs before setting.
Note: Setting change from ADCS = 1 and ADCE = 1 to ADCS = 1 and ADCE = 0 is prohibited.
Note: Do not change the ADCS and ADCE bits from 0 to 1 at the same time by using an 8-bit manipulation instruction. Be sure to follow the procedure described in section 29.7. 12- bit A/D Converter Setup Procedure.
Note 1. For details of the FR[2:0], LV[1:0] bits, and A/D conversion, see Table 29.9 to Table 29.10.
Note 2. While in the software trigger no-wait mode or hardware trigger no-wait mode, the operation of the A/D voltage comparator is controlled by the ADCS and ADCE bits, and it takes 1 µs + 2 cycles of the conversion clock (f_{AD}) from the start of operation for the operation to stabilize. Therefore, immediately after the ADCS bit is set to 1 after at least 1 µs + 2 cycles of the conversion clock (f_{AD}) have elapsed from the time ADCE bit is set to 1, the conversion result becomes valid. When ADCS is set to 1 while ADCE = 0, A/D conversion starts after the stabilization wait time has passed. If ADCS is set before at least 1 µs + 2 cycles of the conversion clock (f_{AD}) have elapsed, ignore data of the first conversion.

This register sets the time for converting analog input to digital data, and starts and stops conversion.

The ADM0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

The value of this register is 0x00 following a reset.

ADCE bit (A/D voltage comparator operation control)

This bit is used for A/D voltage comparator operation control.

LV[1:0] bits (Select Operation voltage mode)

These bits are used to select operation voltage mode.

FR[2:0] bits (Select Conversion Clock (f_{AD}))

These bits are used to select conversion clock (f_{AD}).

ADMD bit (Specification of the A/D conversion channel selection mode)

This bit is used for specification of the A/D conversion channel selection mode.

ADCS bit (A/D conversion operation control)

This bit is used for A/D conversion operation control.

Table 29.4 shows the relationship between ADCS and ADCE bits, including A/D operation status.

Table 29.4 Relationship between ADCS and ADCE bits, including A/D operation status (1 of 2)

ADCS	ADCE	A/D conversion mode	A/D operation state
0	0	All modes	Conversion stopped state
0	1	Hardware trigger wait mode	Trigger standby state
		Other than hardware trigger wait mode	Conversion standby state
1	0	Software trigger wait mode	Conversion operation state
		Other than software trigger wait mode	Conversion stopped state

Table 29.4 Relationship between ADCS and ADCE bits, including A/D operation status (2 of 2)

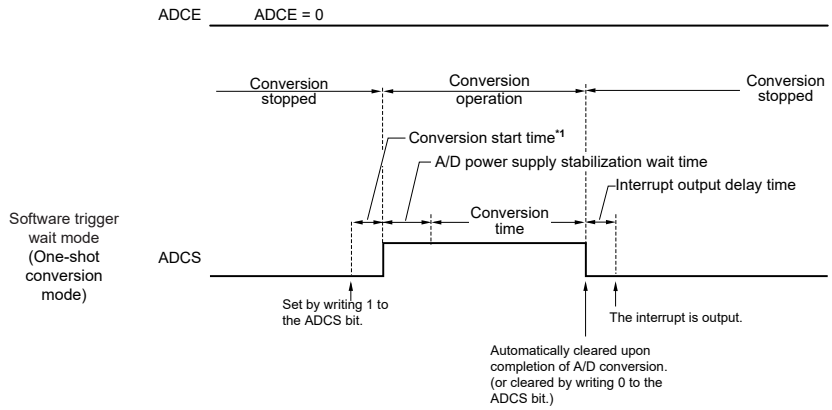
ADCS	ADCE	A/D conversion mode	A/D operation state
1	1	Hardware trigger no-wait mode	Trigger standby state or conversion operation state
		Hardware trigger wait mode or Software trigger no-wait mode	Conversion operation state

Table 29.5 shows the conditions for setting and clearing the ADCS bit.

Table 29.5 Conditions for setting and clearing the ADCS bit

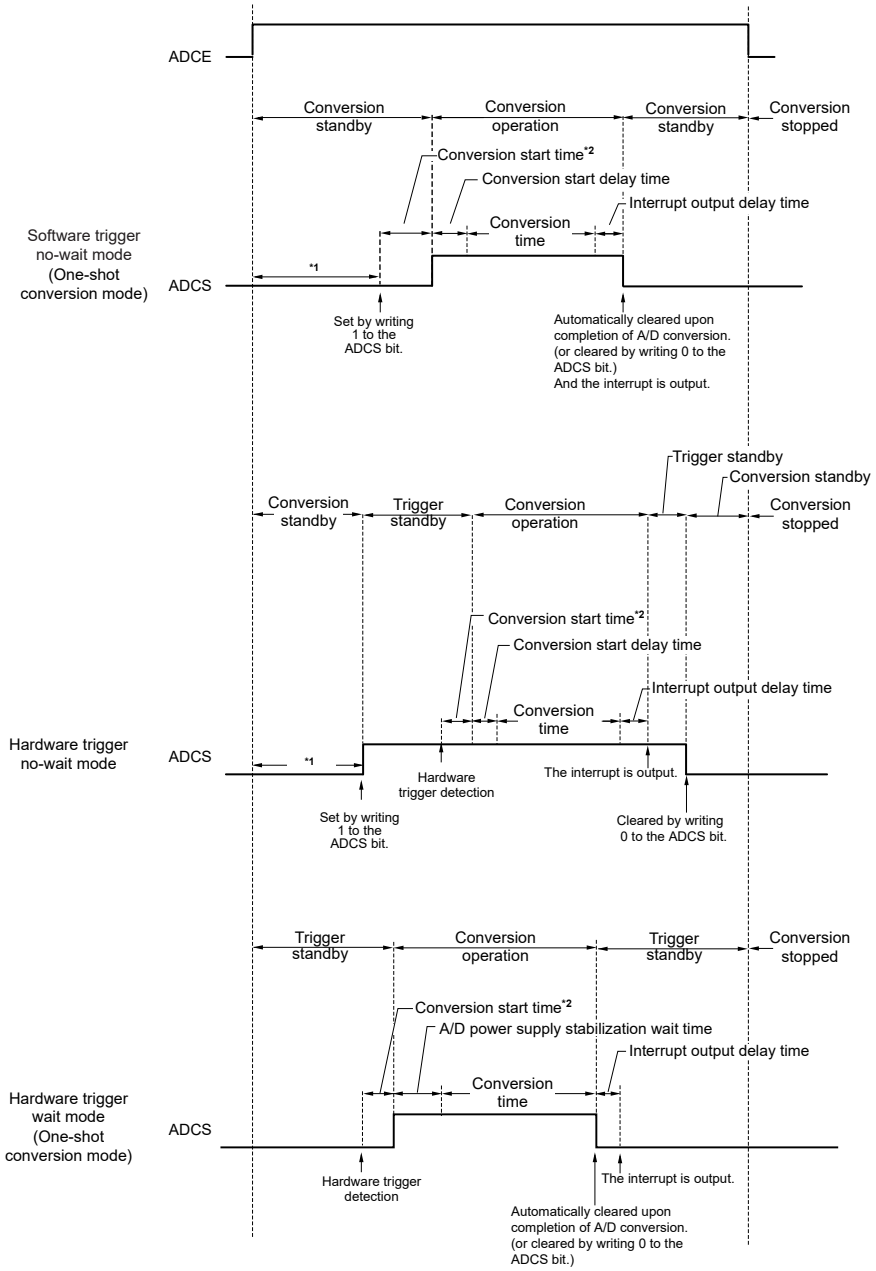
A/D conversion mode			Set conditions	Clear conditions
Software trigger no-wait mode	Select mode	Sequential conversion mode	When 1 is written to ADCS	When 0 is written to ADCS
		One-shot conversion mode		• When 0 is written to ADCS • The bit is automatically cleared to 0 when A/D conversion ends.
	Scan mode	Sequential conversion mode		When 0 is written to ADCS
		One-shot conversion mode		• When 0 is written to ADCS • The bit is automatically cleared to 0 when conversion ends on the specified four channels.
Software trigger wait mode	Select mode	Sequential conversion mode		When 0 is written to ADCS
		One-shot conversion mode		• When 0 is written to ADCS • The bit is automatically cleared to 0 when A/D conversion ends.
	Scan mode	Sequential conversion mode		When 0 is written to ADCS
		One-shot conversion mode		• When 0 is written to ADCS • The bit is automatically cleared to 0 when conversion ends on the specified four channels.
Hardware trigger no-wait mode	Select mode	Sequential conversion mode		When 0 is written to ADCS
		One-shot conversion mode		When 0 is written to ADCS
	Scan mode	Sequential conversion mode		When 0 is written to ADCS
		One-shot conversion mode		When 0 is written to ADCS
Hardware trigger wait mode	Select mode	Sequential conversion mode	When a hardware trigger is input	When 0 is written to ADCS
		One-shot conversion mode		• When 0 is written to ADCS • The bit is automatically cleared to 0 when A/D conversion ends.
	Scan mode	Sequential conversion mode		When 0 is written to ADCS
		One-shot conversion mode		• When 0 is written to ADCS • The bit is automatically cleared to 0 when conversion ends on the specified four channels.

The timing when using the A/D voltage comparator is shown in Figure 29.2 and Figure 29.3.



Note 1. The maximum conversion start time takes the time shown in Table 29.6.

Figure 29.2 Timing when 12-Bit A/D Converter is used (software trigger wait mode)



Note 1. While in the software trigger no-wait mode or hardware trigger no-wait mode, the time from the rising of the ADCE bit to the rising of the ADCS bit must be 1 µs + 2 cycles of the conversion clock (f_{AD}) or longer to stabilize the internal circuit.

Note 2. The maximum conversion start time takes the time shown in Table 29.6.

Figure 29.3 Timing when 12-bit A/D Converter is used (other than software trigger wait mode)

Table 29.6 shows the conversion start time with FR[2:0] and ADLSP bits setting.

Table 29.6 Settings of conversion start time (1 of 2)

ADM1.ADLSP	ADM0.FR[2:0]	Conversion clock (f_{AD})	Conversion start time (number of PCLKB clock)	
			Software trigger no-wait mode/ Hardware trigger no-wait mode	Software trigger wait mode/ Hardware trigger wait mode
0	000b	PCLKB/32	31	1
0	001b	PCLKB/16	15	1
0	010b	PCLKB/8	7	1

Table 29.6 Settings of conversion start time (2 of 2)

ADM1.ADLSP	ADM0.FR[2:0]	Conversion clock (f_{AD})	Conversion start time (number of PCLKB clock)	
			Software trigger no-wait mode/ Hardware trigger no-wait mode	Software trigger wait mode/ Hardware trigger wait mode
0	011b	PCLKB/4	3	1
0	100b	PCLKB/2	1	1
0	101b	PCLKB	1	1
1	011b	PCLKB/4	3	1
1	100b	PCLKB/2	1	1
1	101b	PCLKB	1	1

However, for the second and subsequent conversion in sequential conversion mode and for conversion of the channels specified for scan1, 2, and 3 in scan mode, the conversion start time and stabilization wait time for A/D power supply do not occur after a hardware trigger is detected.

Note: If using the hardware trigger wait mode, setting the ADCS bit to 1 is prohibited (but the bit is automatically switched to 1 when the hardware trigger signal is detected). However, it is possible to clear the ADCS bit to 0 to specify the A/D conversion standby state.

Note: While in the one-shot conversion mode of the hardware trigger no-wait mode, the ADCS bit is not automatically cleared to 0 when A/D conversion ends. Instead, 1 is retained.

Note: Only rewrite the value of the ADCE bit when ADCS = 0 (while in the conversion stopped/conversion standby state).

Note: To complete A/D conversion, specify at least the following time as the hardware trigger interval:
- Hardware trigger no wait mode: 3 PCLKB clock cycles + conversion start time + conversion time
- Hardware trigger wait mode: 3 PCLKB clock cycles + conversion start time + A/D power supply stabilization wait time + conversion time + 5 µs

Table 29.7 shows the relationship between operation voltage mode and conversion time.

Table 29.7 Conversion time in each operation mode

Operation voltage mode	ADM0.LV[1:0]	Conversion time (number of f_{AD} clock) [cycles]	
		Select mode	Scan mode*1
Normal mode 1	00b	64	256
Normal mode 2	01b	181	724
Low voltage mode 1	10b	80	320
Low voltage mode 2	11b	107	428

Note 1. The value in this column is the conversion time for four channels.

Table 29.8 shows the relationship between conversion start delay time, A/D power supply stabilization wait time, and interrupt output delay time.

Table 29.8 Conversion start delay time, A/D power supply stabilization wait time, and interrupt output delay time (1 of 2)

ADM1.ADLSP	ADM0.FR[2:0]	Conversion clock (f_{AD})	Conversion start delay time (number of f_{AD} clock) [cycles]	A/D power supply stabilization wait time (number of f_{AD} clock) [cycles]	Interrupt output delay time (number of f_{AD} clock) [cycles]	
			No-wait mode*1	Wait mode*2	No-wait mode*1	Wait mode*2 *3
0	000b	PCLKB/32	1	4	1	4
0	001b	PCLKB/16	1	4	1	4
0	010b	PCLKB/8	1	6	1	4

Table 29.8 Conversion start delay time, A/D power supply stabilization wait time, and interrupt output delay time (2 of 2)

ADM1.ADLSP	ADM0.FR[2:0]	Conversion clock (f_{AD})	Conversion start delay time (number of f_{AD} clock) [cycles]	A/D power supply stabilization wait time (number of f_{AD} clock) [cycles]	Interrupt output delay time (number of f_{AD} clock) [cycles]	
			No-wait mode*1	Wait mode*2	No-wait mode*1	Wait mode*2 *3
0	011b	PCLKB/4	1	10	1	4
0	100b	PCLKB/2	1	18	1	4
0	101b	PCLKB	1	34	1	4
1	011b	PCLKB/4	1	4	1	4
1	100b	PCLKB/2	1	4	1	4
1	101b	PCLKB	1	6	1	4

Note 1. No-wait mode means either software trigger no-wait mode or hardware trigger no-wait mode.
Note 2. Wait mode means either software trigger wait mode or hardware trigger wait mode.
Note 3. The value in this column is applicable when the one-shot conversion mode is selected. When the sequential conversion mode is selected, the number of clock cycles is shortened by 3 cycles of the conversion clock (f_{AD}).

Table 29.9 to Table 29.10 show the A/D conversion time with FR[2:0], LV[1:0], and ADLSP bits setting.

Table 29.9 A/D conversion time in Normal mode 1 and 2 (1 of 2)

ADM0.LV[1:0]	ADM1.ADLSP	ADM0.FR[2:0]	Conversion clock (f_{AD})	PCLKB condition [MHz]	Voltage condition*4	A/D conversion time [µs]*1			
						Select mode		Scan mode	
						No-wait mode*2	Wait mode*3 *5	No-wait mode*2	Wait mode*3 *5
00b (Normal mode 1)	0	000b	PCLKB/32	PCLKB = 48, 32	2.4 V ≤ AVREFP ≤ VCC ≤ 5.5 V	66 x 32/ PCLKB	72 x 32/ PCLKB	258 x 32/ PCLKB	264 x 32/ PCLKB
	0	001b	PCLKB/16	16 ≤ PCLKB ≤ 32		66 x 16/ PCLKB	72 x 16/ PCLKB	258 x 16/ PCLKB	264 x 16/ PCLKB
				PCLKB = 48			Setting prohibited		Setting prohibited
	0	010b	PCLKB/8	8 ≤ PCLKB ≤ 32		66 x 8/ PCLKB	74 x 8/ PCLKB	258 x 8/ PCLKB	266 x 8/ PCLKB
				PCLKB = 48			Setting prohibited		Setting prohibited
	0	011b	PCLKB/4	4 < PCLKB ≤ 32		66 x 4/ PCLKB	78 x 4/ PCLKB	258 x 4/ PCLKB	270 x 4/ PCLKB
				PCLKB = 48			Setting prohibited		Setting prohibited
	0	100b	PCLKB/2	4 < PCLKB ≤ 32		66 x 2/ PCLKB	86 x 2/ PCLKB	258 x 2/ PCLKB	278 x 2/ PCLKB
				PCLKB = 48			Setting prohibited		Setting prohibited
	0	101b	PCLKB	4 < PCLKB ≤ 32		66 x 1/ PCLKB	102 x 1/ PCLKB	258 x 1/ PCLKB	294 x 1/ PCLKB
				PCLKB = 48	2.7 V ≤ AVREFP ≤ VCC ≤ 5.5 V		Setting prohibited		Setting prohibited
	1	011b	PCLKB/4	PCLKB = 4	2.4 V ≤ AVREFP ≤ VCC ≤ 5.5 V	66 x 4/ PCLKB	72 x 4/ PCLKB	258 x 4/ PCLKB	264 x 4/ PCLKB
	1	100b	PCLKB/2	2 ≤ PCLKB ≤ 4		66 x 2/ PCLKB	72 x 2/ PCLKB	258 x 2/ PCLKB	264 x 2/ PCLKB
	1	101b	PCLKB	1 ≤ PCLKB ≤ 4		66 x 1/ PCLKB	74 x 1/ PCLKB	258 x 1/ PCLKB	266 x 1/ PCLKB
	Other than the above, setting prohibited		—	—	—	—	—	—	—

Table 29.9 A/D conversion time in Normal mode 1 and 2 (2 of 2)

ADM0.LV[1:0]	ADM1.ADLSP	ADM0.FR[2:0]	Conversion clock (f_{AD})	PCLKB condition [MHz]	Voltage condition*4	A/D conversion time [µs]*1			
						Select mode		Scan mode	
						No-wait mode*2	Wait mode*3 *5	No-wait mode*2	Wait mode*3 *5
01b (Normal mode 2)	0	000b	PCLKB/32	PCLKB = 48, 32	2.4 V ≤ AVREFP ≤ VCC ≤ 5.5 V	183 x 32/ PCLKB	189 x 32/ PCLKB	726 x 32/ PCLKB	732 x 32/ PCLKB
	0	001b	PCLKB/16	16 ≤ PCLKB ≤ 32		183 x 16/ PCLKB	189 x 16/ PCLKB	726 x 16/ PCLKB	732 x 16/ PCLKB
				PCLKB = 48			Setting prohibited		Setting prohibited
	0	010b	PCLKB/8	8 ≤ PCLKB ≤ 32		183 x 8/ PCLKB	191 x 8/ PCLKB	726 x 8/ PCLKB	734 x 8/ PCLKB
				PCLKB = 48			Setting prohibited		Setting prohibited
	0	011b	PCLKB/4	4 < PCLKB ≤ 32		183 x 4/ PCLKB	195 x 4/ PCLKB	726 x 4/ PCLKB	738 x 4/ PCLKB
				PCLKB = 48			Setting prohibited		Setting prohibited
	0	100b	PCLKB/2	4 < PCLKB ≤ 32		183 x 2/ PCLKB	203 x 2/ PCLKB	726 x 2/ PCLKB	746 x 2/ PCLKB
				PCLKB = 48			Setting prohibited		Setting prohibited
	0	101b	PCLKB	4 < PCLKB ≤ 32		183 x 1/ PCLKB	219 x 1/ PCLKB	726 x 1/ PCLKB	762 x 1/ PCLKB
				PCLKB = 48	2.7 V ≤ AVREFP ≤ VCC ≤ 5.5 V		Setting prohibited		Setting prohibited
	1	011b	PCLKB/4	PCLKB = 4	2.4 V ≤ AVREFP ≤ VCC ≤ 5.5 V	183 x 4/ PCLKB	189 x 4/ PCLKB	726 x 4/ PCLKB	732 x 4/ PCLKB
	1	100b	PCLKB/2	2 ≤ PCLKB ≤ 4		183 x 2/ PCLKB	189 x 2/ PCLKB	726 x 2/ PCLKB	732 x 2/ PCLKB
	1	101b	PCLKB	1 ≤ PCLKB ≤ 4		183 x 1/ PCLKB	191 x 1/ PCLKB	726 x 1/ PCLKB	734 x 1/ PCLKB
	Other than the above, setting prohibited		—	—	—	—	—	—	—

Note: The A/D conversion time must also be within the relevant range of conversion times described in section 37.4. ADC12 Characteristics.

Note: Rewrite the FR[2:0], LV[1:0] bits to different values while conversion is stopped (ADCS = 0, ADCE = 0). The FR[2:0], and LV[1:0] bits should be changed at least 0.2 µs after conversion stops (ADCS = 0, ADCE = 0).

Note: The above A/D conversion time does not include the conversion start time. Add the conversion start time to obtain the time for the first conversion. Additionally, the A/D conversion time does not include clock frequency errors. Consider clock frequency errors when selecting the A/D conversion time.

Note: When the internal reference voltage or temperature sensor output voltage is selected as the target for A/D conversion, use normal mode 2 and use a conversion clock (f_{AD}) with a frequency no greater than 32 MHz.

Note: When the internal reference voltage is selected as the positive reference voltage, normal mode 1 and mode 2 cannot be used. Use low voltage mode 1 or 2.

Note 1. A/D conversion time consists of conversion start delay time, A/D power supply stabilization wait time, conversion time, and interrupt output delay time.
See Figure 29.2, Figure 29.3, Table 29.7, and Table 29.8.

Note 2. No-wait mode means software trigger no-wait mode or hardware trigger no-wait mode.

Note 3. Wait mode means software trigger wait mode or hardware trigger wait mode. For the second and subsequent conversion in sequential conversion mode and for conversion of the channels specified for scan 1, 2, and 3 in scan mode, the conversion start time and A/D power supply stabilization wait time do not occur after a software trigger or a hardware trigger is detected.
Note 4. For ICLK frequency and VCC conditions, see section 10.5.2. Operating Range. Set the frequency and VCC to satisfy this condition.
Note 5. The value in this column is applicable when the one-shot conversion mode is selected. When the sequential conversion mode is selected, the number of clock cycles is shortened by 3 cycles of the conversion clock (f_{AD}).

Table 29.10 A/D conversion time in Low voltage mode 1 and 2 (1 of 2)

ADM0.LV[1:0]	ADM1.ADLSP	ADM0.FR[2:0]	Conversion clock (f_{AD})	PCLKB condition [MHz]*4	Voltage condition*4	A/D conversion time [µs]*1			
						Select mode		Scan mode	
						No-wait mode*2	Wait mode*3 *5	No-wait mode*2	Wait mode*3 *5
10b (Low Voltage mode 1)	0	000b	PCLKB/32	PCLKB = 48, 32	1.6 V ≤ AVREFP ≤ VCC ≤ 5.5 V	82 x 32/ PCLKB	88 x 32/ PCLKB	322 x 32/ PCLKB	328 x 32/ PCLKB
	0	001b	PCLKB/16	16 ≤ PCLKB ≤ 32		82 x 16/ PCLKB	88 x 16/ PCLKB	322 x 16/ PCLKB	328 x 16/ PCLKB
				PCLKB = 48			Setting prohibited		Setting prohibited
	0	010b	PCLKB/8	8 ≤ PCLKB ≤ 32		82 x 8/ PCLKB	90 x 8/ PCLKB	322 x 8/ PCLKB	330 x 8/ PCLKB
				PCLKB = 48	1.8 V ≤ AVREFP ≤ VCC ≤ 5.5 V		Setting prohibited		Setting prohibited
	0	011b	PCLKB/4	4 < PCLKB ≤ 32	1.8 V ≤ AVREFP ≤ VCC ≤ 5.5 V	82 x 4/ PCLKB	94 x 4/ PCLKB	322 x 4/ PCLKB	334 x 4/ PCLKB
				PCLKB = 48	2.4 V ≤ AVREFP ≤ VCC ≤ 5.5 V		Setting prohibited		Setting prohibited
	0	100b	PCLKB/2	4 < PCLKB ≤ 16	1.8 V ≤ AVREFP ≤ VCC ≤ 5.5 V	82 x 2/ PCLKB	102 x 2/ PCLKB	322 x 2/ PCLKB	342 x 2/ PCLKB
				4 < PCLKB ≤ 32	2.4 V ≤ AVREFP ≤ VCC ≤ 5.5 V				
				PCLKB = 48	2.7 V ≤ AVREFP ≤ VCC ≤ 5.5 V		Setting prohibited		Setting prohibited
	0	101b	PCLKB	4 < PCLKB ≤ 8	1.8 V ≤ AVREFP ≤ VCC ≤ 5.5 V	82 x 1/ PCLKB	118 x 1/ PCLKB	322 x 1/ PCLKB	358 x 1/ PCLKB
				4 < PCLKB ≤ 16	2.4 V ≤ AVREFP ≤ VCC ≤ 5.5 V				
				4 < PCLKB ≤ 24	2.7 V ≤ AVREFP ≤ VCC ≤ 5.5 V				
	1	011b	PCLKB/4	PCLKB = 4	1.6 V ≤ AVREFP ≤ VCC ≤ 5.5 V	82 x 4/ PCLKB	88 x 4/ PCLKB	322 x 4/ PCLKB	328 x 4/ PCLKB
	1	100b	PCLKB/2	2 ≤ PCLKB ≤ 4		82 x 2/ PCLKB	88 x 2/ PCLKB	322 x 2/ PCLKB	328 x 2/ PCLKB
	1	101b	PCLKB	1 ≤ PCLKB ≤ 4		82 x 1/ PCLKB	90 x 1/ PCLKB	322 x 1/ PCLKB	330 x 1/ PCLKB
	Other than the above, setting prohibited		—	—	—	—	—	—	—

Table 29.10 A/D conversion time in Low voltage mode 1 and 2 (2 of 2)

ADM0.LV[1:0]	ADM1.ADLSP	ADM0.FR[2:0]	Conversion clock (f_{AD})	PCLKB condition [MHz]*4	Voltage condition*4	A/D conversion time [µs]*1			
						Select mode		Scan mode	
						No-wait mode*2	Wait mode*3 *5	No-wait mode*2	Wait mode*3 *5
11b (Low Voltage mode 2)	0	000b	PCLKB/32	PCLKB = 48, 32	1.6 V ≤ AVREFP ≤ VCC ≤ 5.5 V	109 x 32/ PCLKB	115 x 32/ PCLKB	430 x 32/ PCLKB	436 x 32/ PCLKB
	0	001b	PCLKB/16	16 ≤ PCLKB ≤ 32		109 x 16/ PCLKB	115 x 16/ PCLKB	430 x 16/ PCLKB	436 x 16/ PCLKB
				PCLKB = 48			Setting prohibited		Setting prohibited
	0	010b	PCLKB/8	8 ≤ PCLKB ≤ 32		109 x 8/ PCLKB	117 x 8/ PCLKB	430 x 8/ PCLKB	438 x 8/ PCLKB
				PCLKB = 48	1.8 V ≤ AVREFP ≤ VCC ≤ 5.5 V		Setting prohibited		Setting prohibited
	0	011b	PCLKB/4	4 < PCLKB ≤ 32	1.8 V ≤ AVREFP ≤ VCC ≤ 5.5 V	109 x 4/ PCLKB	121 x 4/ PCLKB	430 x 4/ PCLKB	442 x 4/ PCLKB
				PCLKB = 48	2.4 V ≤ AVREFP ≤ VCC ≤ 5.5 V		Setting prohibited		Setting prohibited
	0	100b	PCLKB/2	4 < PCLKB ≤ 16	1.8 V ≤ AVREFP ≤ VCC ≤ 5.5 V	109 x 2/ PCLKB	129 x 2/ PCLKB	430 x 2/ PCLKB	450 x 2/ PCLKB
				4 < PCLKB ≤ 32	2.4 V ≤ AVREFP ≤ VCC ≤ 5.5 V				
				PCLKB = 48	2.7 V ≤ AVREFP ≤ VCC ≤ 5.5 V		Setting prohibited		Setting prohibited
	0	101b	PCLKB	4 < PCLKB ≤ 8	1.8 V ≤ AVREFP ≤ VCC ≤ 5.5 V	109 x 1/ PCLKB	145 x 1/ PCLKB	430 x 1/ PCLKB	466 x 1/ PCLKB
				4 < PCLKB ≤ 16	2.4 V ≤ AVREFP ≤ VCC ≤ 5.5 V				
				4 < PCLKB ≤ 24	2.7 V ≤ AVREFP ≤ VCC ≤ 5.5 V				
	1	011b	PCLKB/4	PCLKB = 4	1.6 V ≤ AVREFP ≤ VCC ≤ 5.5 V	109 x 4/ PCLKB	115 x 4/ PCLKB	430 x 4/ PCLKB	436 x 4/ PCLKB
	1	100b	PCLKB/2	2 ≤ PCLKB ≤ 4		109 x 2/ PCLKB	115 x 2/ PCLKB	430 x 2/ PCLKB	436 x 2/ PCLKB
	1	101b	PCLKB	1 ≤ PCLKB ≤ 4		109 x 1/ PCLKB	117 x 1/ PCLKB	430 x 1/ PCLKB	438 x 1/ PCLKB
	Other than the above, setting prohibited		—	—	—	—	—	—	—

Note: The A/D conversion time must also be within the relevant range of conversion times described in section 37.4. ADC12 Characteristics.

Note: Rewrite the FR[2:0], LV[1:0] bits to different values while conversion is stopped (ADCS = 0, ADCE = 0). The FR[2:0], and LV[1:0] bits should be changed at least 0.2 µs after conversion stops (ADCS = 0, ADCE = 0).

Note: The above A/D conversion time does not include the conversion start time. Add the conversion start time to obtain the time for the first conversion. Additionally, the A/D conversion time does not include clock frequency errors. Consider clock frequency errors when selecting the A/D conversion time.
Note: When the internal reference voltage or temperature sensor output voltage is selected as the target for A/D conversion, use low voltage mode 2 and use a conversion clock (f_{AD}) with a frequency no greater than 16 MHz.
Note: When the internal reference voltage is selected as the positive reference voltage, the conversion clock (f_{AD}) must be in the range from 1 to 2 MHz.
Note 1. A/D conversion time consists of conversion start delay time, A/D power supply stabilization wait time, conversion time, and interrupt output delay time.
See Figure 29.2, Figure 29.3, Table 29.7, and Table 29.8.
Note 2. No-wait mode means software trigger no-wait mode or hardware trigger no-wait mode.
Note 3. Wait mode means software trigger wait mode or hardware trigger wait mode. For the second and subsequent conversion in sequential conversion mode and for conversion of the channels specified for scan 1, 2, and 3 in scan mode, the conversion start time and A/D power supply stabilization wait time do not occur after a software trigger or a hardware trigger is detected.
Note 4. For ICLK frequency and VCC conditions, see section 10.5.2. Operating Range. Set the frequency and VCC to satisfy this condition.
Note 5. The value in this column is applicable when the one-shot conversion mode is selected. When the sequential conversion mode is selected, the number of clock cycles is shortened by 3 cycles of the conversion clock (f_{AD}).

Figure 29.4 shows the timing of sampling and A/D conversion.

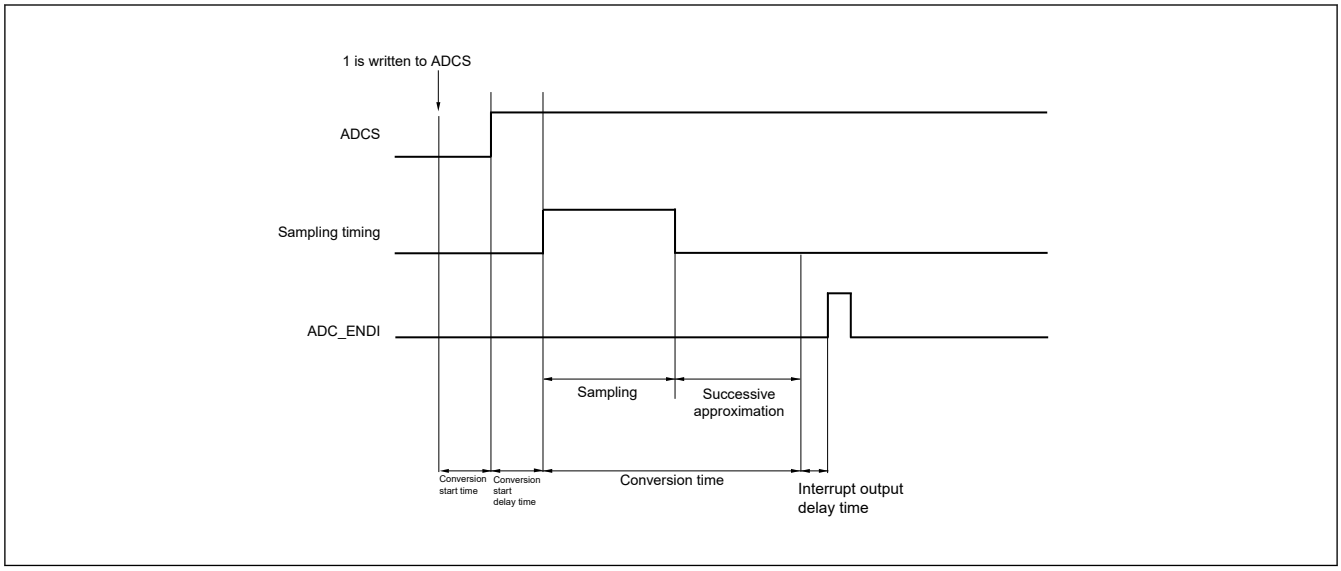


Figure 29.4 12-Bit A/D converter sampling and A/D conversion timing (example for software trigger no-wait mode, select mode, and one-shot conversion mode)

29.3.2 ADM1 : A/D Converter Mode Register 1

Base address: ADC120 = 0x4009_C000
Offset address: 0x02

Bit position:	7	6	5	4	3	2	1	0
Bit field:	ADTMD[1:0]		ADSCM	—	ADLSP	ADTRS[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	ADTRS[2:0]	Selection of the hardware trigger signal 0 0 0: Timer Array Unit channel 1 count or capture end interrupt signal (TAU0_ENDI1) 0 1 0: Realtime clock interrupt signal (RTC_ALM) 0 1 1: 32-bit interval timer interrupt signal (TML32_OUTI) 1 0 0: Event input from ELC Others: Setting prohibited.	R/W
3	ADLSP	PCLKB input frequency setting 0: 4 MHz < PCLKB ≤ 48 MHz 1: 1 MHz ≤ PCLKB ≤ 4 MHz	R/W

Bit	Symbol	Function	R/W
4	—	This bit is read as 0. The write value should be 0.	R/W
5	ADSCM	Specification of the A/D conversion mode 0: Sequential conversion mode 1: One-shot conversion mode	R/W
7:6	ADTMD[1:0]	Selection of the A/D conversion trigger mode 1 0: Hardware trigger no-wait mode 1 1: Hardware trigger wait mode Others: Software trigger no-wait mode or software trigger wait mode	R/W

Note: Only rewrite the value of the ADM1 register while conversion operation is stopped (ADCS = 0, ADCE = 0).
Note: To complete A/D conversion, specify at least the following time as the hardware trigger interval:
Hardware trigger no wait mode: 3 PCLKB clock cycles + conversion start time + A/D conversion time
Hardware trigger wait mode: 3 PCLKB clock cycles + conversion start time + A/D power supply stabilization wait time + A/D conversion time + 5 µs
Note: In modes other than Snooze mode, input of the next RTC_ALM or TML32_OUTI is not recognized as a valid hardware trigger for up to 4 PCLKB cycles after the first RTC_ALM or TML32_OUTI is input.

This register is used to specify the A/D conversion trigger, conversion mode, and hardware trigger signal. The ADM1 register can be set by a 1-bit or 8-bit memory manipulation instruction.

The value of this register is 0x00 following a reset.

ADTRS[2:0] bits (Selection of the hardware trigger signal)

These bits are used for selection of the hardware trigger signal.

ADLSP bit (PCLKB input frequency setting)

This bit is used for PCLKB input frequency setting.

ADSCM bit (Specification of the A/D conversion mode)

This bit is used for specification of the A/D conversion mode.

ADTMD[1:0] bits (Selection of the A/D conversion trigger mode)

These bits are used for selection of the A/D conversion trigger mode.

29.3.3 ADM2 : A/D Converter Mode Register 2

Base address: ADC120 = 0x4009_C000

Offset address: 0x90

Bit position:	7	6	5	4	3	2	1	0
Bit field:	ADREFP[1:0]		ADREFM	—	ADRCK	AWC	ADTYP[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	ADTYP[1:0]	Selection of the hardware trigger signal 0 0: 10-bit resolution 0 1: 8-bit resolution 1 0: 12-bit resolution Others: Setting prohibited.	R/W
2	AWC	Specification of the Snooze mode 0: Do not use the Snooze mode function. 1: Use the Snooze mode function.	R/W
3	ADRCK	Checking the upper limit and lower limit conversion result values 0: The interrupt signal (ADC_ENDI) is output when the ADLL register ≤ the ADCRn register ≤ the ADUL register (AREA 1). 1: The interrupt signal (ADC_ENDI) is output when the ADCRn register ≤ the ADLL register (AREA 2) or the ADUL register < the ADCRn register (AREA 3).	R/W
4	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
5	ADREFM	Selection of the '-' side reference voltage of the 12-bit A/D converter 0: Supplied from V_{SS} 1: Supplied from AVREFM	R/W
7:6	ADREFP[1:0]	Selection of the '+' side reference voltage source of the 12-bit A/D converter 0 0: Supplied from V_{CC} 0 1: Supplied from AVREFP 1 0: Supplied from the internal reference voltage*1 1 1: Discharge the internal circuitry	R/W

Note: Only rewrite the value of the ADM2 register while conversion operation is stopped (ADCS = 0, ADCE = 0).
Note: Do not set the ADREFP[1] bit to 1b when shifting to Software Standby mode mode, or to Sleep mode while the CPU is operating on the subsystem clock. When the internal reference voltage is selected (ADREFP[1:0] = 10b), the 12-bit A/D converter internal reference voltage current (IADREF) indicated in section 37.2.5. Operating and Standby Current is added to ICC.
Note: When using AV_{REFP} and AV_{REFM}, specify ANI0 and ANI1 as the analog input channels and specify input mode by using the Port mn Pin Function Select Register.
Note 1. For details about the internal reference voltage, see section 37, Electrical Characteristics.

This register is used to select the '+' side and '-' side reference voltages of the 12-bit A/D converter, check the upper limit and lower limit A/D conversion result values, select the resolution, and specify whether to use the Snooze mode.

The ADM2 register can be set by a 1-bit or 8-bit memory manipulation instruction.

The value of this register is 0x00 following a reset.

ADTYP[1:0] bits (Selection of the hardware trigger signal)

These bits are used for selection of the hardware trigger signal.

AWC bit (Specification of the Snooze mode)

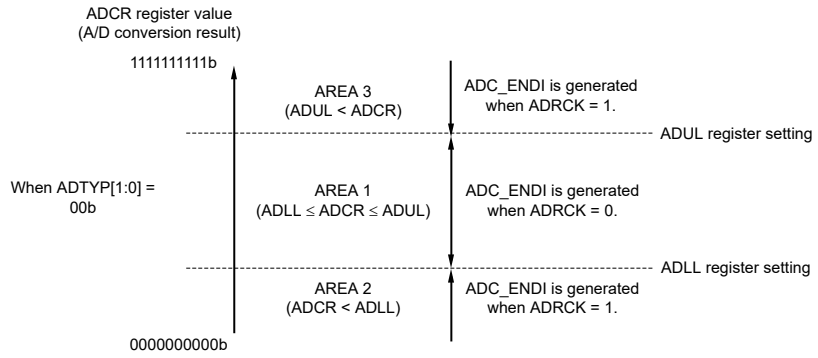
This bit is used for specification of the Snooze mode.

- When using the Snooze mode function, set AWC to 1.
- Using the Snooze mode function in the software trigger no-wait mode, software trigger wait mode, or hardware trigger no-wait mode is prohibited.
- Using the Snooze mode function in hardware trigger wait mode in sequential conversion mode is prohibited.
- When using the Snooze mode function, specify a hardware trigger interval of at least "conversion start time + A/D power supply stabilization wait time + A/D conversion time + 3 PCLKB clock cycles + 5 µs".
- Even when using Snooze mode, be sure to set the AWC bit to 0 in normal operation and change it to 1 just before shifting to Software Standby mode.
  Also, be sure to change the AWC bit to 0 after returning from Software Standby mode to normal operation.
  If the AWC bit is left set to 1, A/D conversion will not start normally in spite of the subsequent Snooze mode or normal operation.

ADRCK bit (Checking the upper limit and lower limit conversion result values)

This bit is used for checking the upper limit and lower limit conversion result values.

Figure 29.5 shows the generation range of the interrupt signal (ADC_ENDI) for AREA 1 to AREA 3.



Note: If ADC_ENDI does not occur, the A/D conversion result is not stored in the ADCRn or ADCRnH register.

Figure 29.5 ADRCK bit interrupt signal generation range (in 10-bit resolution mode)

ADREFM bit (Selection of the '-' side reference voltage of the 12-bit A/D converter)

This bit is used for selection of the '-' side reference voltage of the 12-bit A/D converter.

ADREFP[1:0] bits (Selection of the '+' side reference voltage source of the 12-bit A/D converter)

These bits are used for selection of the '+' side reference voltage source of the 12-bit A/D converter.

Use Table 29.11 procedure to rewrite the ADREFP[1:0] bits.

Table 29.11 Register settings for ADREFP[1:0] rewrite

Step		Process	Detail
Register settings for ADREFP[1:0] Rewrite	<1>	Set ADM0.ADCE = 0	—
	<2>	Wait 0.2 µs or more	—
	<3>	Set ADREFP[1:0] = 11b	This step is only necessary when the values of ADREFP[1:0] are changed to 10b, respectively.
	<4>	Reference voltage discharge time: 1 µs	
	<5>	Change the values of ADREFP[1:0]	Setting '+' side reference.
	<6>	Reference voltage stabilization wait time (A)	ADREFP[1:0] = 10b: A = 5 µs ADREFP[1:0] = 00b or 01b: A = 4.8 µs
	<7>	Set ADM0.ADCE = 1	—
	<8>	Reference voltage stabilization wait time (B)	B = 1 µs + 2 cycles of conversion clock (f_{AD})
	<9>	Start the A/D conversion	—

29.3.4 ADCR/ADCRn: 12-bit or 10-bit A/D Conversion Result Register n (n = 0 to 3)

Base address: ADC120 = 0x4009_C000

Offset address: 0x06 (ADCR)
0xA0 (ADCR0)
0xA2 (ADCR1)
0xA4 (ADCR2)
0xA6 (ADCR3)

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	n/a	The A/D conversion result is stored. When A/D conversion with 12-bit resolution is selected, the higher 4 bits are fixed to 0. When A/D conversion with 10-bit resolution is selected, the lower 6 bits are fixed to 0. When A/D conversion with 8-bit resolution is selected, the lower 8 bits are fixed to 0.	R

Note: The contents of the ADCR register are stored in the ADCR0 register.
Note: When 8-bit resolution A/D conversion is selected (when the ADTYP[1:0] bits of A/D converter mode register 2 (ADM2) are respectively set to 01b) and the ADCRn register is read, 0 is read from the bits other than the higher 8 bits.
Note: When the ADCRn register is accessed in 16-bit units, and A/D conversion with 10-bit resolution is selected, the higher 10 bits of the conversion result are read in order starting at bit 15 of the ADCRn register.
When A/D conversion with 12-bit resolution is selected, the higher 12 bits of the conversion result are read in order starting at bit 11 of the ADCRn register.

Note: The contents of the ADCRnH register may become undefined when writing to any of the following registers.

- A/D converter mode register 0 (ADM0)
- Analog input channel specification register (ADS)

Read the conversion result following conversion completion before writing to any of these registers. Otherwise, the correct conversion result may not be obtained.

ADCRn is a 16-bit register that holds the A/D conversion result.

When A/D conversion with 12-bit resolution is selected, the higher 4 bits are fixed to 0.

When A/D conversion with 10-bit resolution is selected, the lower 6 bits are fixed to 0.

Each time A/D conversion ends, the conversion result is loaded from the successive approximation register (SAR). The ADCRn register can be read by a 16-bit memory manipulation instruction.

The value of this register is 0x0000 following a reset.

In select mode, the conversion results are stored in the ADCR and ADCR0 registers[*1]. In scan mode, the conversion results of scan 0 are stored in the ADCR and ADCR0 registers, and the conversion results of scan 1 to 3 are stored in the ADCR1 to ADCR3 registers.[*1]

Note 1. If the A/D conversion result is outside the range specified by using the A/D conversion comparison function (set up by the ADRCK bit of the ADM2 register, ADUL register, and ADLL register; see Figure 29.5), the result is not stored.

29.3.5 ADCRH/ADCRnH : 8-bit A/D Conversion Result Register n (n = 0 to 3)

Base address: ADC120 = 0x4009_C000

Offset address: 0x07 (ADCRH)
0xA1 (ADCR0H)
0xA3 (ADCR1H)
0xA5 (ADCR2H)
0xA7 (ADCR3H)

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

Bit	Symbol	Function	R/W
7:0	n/a	The A/D conversion result is stored. The higher 8 bits of 12-bit resolution are stored.	R

Note: The contents of the ADCRH register are stored in theADCR0H register.
Note: The contents of the ADCRnH register may become undefined when writing to any of the following registers.

- A/D converter mode register 0 (ADM0)
- Analog input channel specification register (ADS)

Read the conversion result following conversion completion before writing to any of these registers. Otherwise, the correct conversion result may not be obtained.

ADCRnH is an 8-bit register that holds the A/D conversion result. The higher 8 bits of 12-bit resolution are stored[*1]. The ADCRnH register can be read by an 8-bit memory manipulation instruction.

The value of this register is 0x00 following a reset.

Note 1. If the A/D conversion result is outside the range specified by using the A/D conversion comparison function (setup by the ADRCK bit of the ADM2 register, ADUL register, and ADLL register; see Figure 29.5), the result is not stored.

29.3.6 ADS : Analog Input Channel Specification Register

Base address: ADC120 = 0x4009_C000

Offset address: 0x01

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

Bit	Symbol	Function	R/W
4:0	ADS[4:0]	Selection of the analog input channel (See Table 29.12 to Table 29.13)	R/W
6:5	—	These bits are read as 0. The write value should be 0.	R/W
7	ADISS	Select internal or external of analog input (See Table 29.12 to Table 29.13) 0: External input 1: Internal circuit input	R/W

Note: Rewrite the value of the ADISS bit while conversion is stopped (ADCS = 0, ADCE = 0).

Note: If using AV_{REFP} as the '+' side reference voltage of the 12-bit A/D converter, do not select ANI0 as an A/D conversion channel.

Note: If using AV_{REFM} as the '-' side reference voltage of the 12-bit A/D converter, do not select ANI1 as an A/D conversion channel.

Note: When the setting of the ADISS bit is 1, the internal reference voltage cannot be used for the '+' side reference voltage. After the ADISS bit is set to 1, the initial conversion result cannot be used. For the setting flow, see section 29.7.5. Example of Using the ADC when Selecting the Temperature Sensor Output Voltage or Internal Reference Voltage, and Software Trigger No-wait Mode and One-shot Conversion Mode.
For details about the internal reference voltage, see section 37, Electrical Characteristics.

Note: Do not set the ADISS bit to 1 when shifting to Software Standby mode, or to Sleep mode while the CPU is operating on the subsystem clock. When the ADISS bit is set to 1, the 12-bit A/D converter internal reference voltage current (IADREF) indicated in section 37.2.5. Operating and Standby Current is added to ICC.

Note: When the setting of the ADISS bit is 1, the hardware trigger wait mode and one-shot conversion mode cannot be used at the same time.

This register specifies the input channel of the analog voltage to be A/D converted.

The ADS register can be set by a 1-bit or 8-bit memory manipulation instruction.

The value of this register is 0x00 following a reset.

ADS[4:0] bits (Selection of the analog input channel)

These bits are used for selection of the analog input channel.

ADISS bit (Select internal or external of analog input)

This bit is used for select internal or external of analog input.

Table 29.12 and Table 29.13 show the input sources that can be selected for ADS[4:0] bits and ADISS bit in each operating mode.

<Select mode (ADMD = 0)>

Table 29.12 Input source selection by ADS[4:0] bits and ADISS bit in select mode (1 of 2)

ADISS	ADS[4:0]	Analog input channel	Input source
0	00000b	ANI0	P002
0	00001b	ANI1	P003
0	00010b	ANI2	P400
0	00011b	ANI3	P401
0	00100b	ANI4	P006
0	00101b	ANI5	P007

Table 29.12 Input source selection by ADS[4:0] bits and ADISS bit in select mode (2 of 2)

ADISS	ADS[4:0]	Analog input channel	Input source
0	10000b	ANI16	P000
0	10001b	ANI17	P001
0	10010b	ANI18	P106
0	10011b	ANI19	P105
1	00000b	—	Temperature sensor output voltage
1	00001b	—	Internal reference voltage*1
Other than the above		Setting prohibited	

Note 1. For details about the internal reference voltage, see section 37, Electrical Characteristics.

<Scan mode (ADMD = 1)>

Table 29.13 Input source selection by ADS[4:0] bits and ADISS bit in scan mode

ADISS	ADS[4:0]	Analog input channel			
		Scan 0	Scan 1	Scan 2	Scan 3
0	00000b	ANI0	ANI1	ANI2	ANI3
0	00001b	ANI1	ANI2	ANI3	ANI4
0	00010b	ANI2	ANI3	ANI4	ANI5
Other than the above		Setting prohibited			

29.3.7 ADUL : Conversion Result Comparison Upper Limit Setting Register

Base address: ADC120 = 0x4009_C000

Offset address: 0x91

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

Bit	Symbol	Function	R/W
7:0	n/a	Setting the upper limit for A/D conversion results	R/W

This register is used to specify the setting for checking the upper limit of the A/D conversion results.

The A/D conversion results and ADUL register value are compared, and interrupt signal (ADC_ENDI) generation is controlled in the range specified by the ADRCK bit of A/D converter mode register 2 (ADM2) (shown in Figure 29.5).

The ADUL register can be set by an 8-bit memory manipulation instruction.

The value of this register is 0xFF following a reset.

29.3.8 ADLL : Conversion Result Comparison Lower Limit Setting Register

Base address: ADC120 = 0x4009_C000

Offset address: 0x92

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

Bit	Symbol	Function	R/W
7:0	n/a	Setting the lower limit for A/D conversion results	R/W

This register is used to specify the setting for checking the lower limit of the A/D conversion results.

The A/D conversion results and ADLL register value are compared, and interrupt signal (ADC_ENDI) generation is controlled in the range specified by the ADRCK bit of A/D converter mode register 2 (ADM2) (shown in Figure 29.5).

The ADLL register can be set by an 8-bit memory manipulation instruction.

The value of this register is 0x00 following a reset.

Note: When A/D conversion with 10-bit resolution is selected, the A/D conversion result register ADCRn[15:8] value is compared with the values in the ADUL and ADLL registers. When A/D conversion with 12-bit resolution is selected, the A/D conversion result register ADCRn[11:4] value is compared with the values in the ADUL and ADLL registers.

Note: Only write new values to the ADUL and ADLL registers while conversion is stopped (ADCS = 0, ADCE = 0).

Note: The setting of the ADUL register must be greater than that of the ADLL register.

29.3.9 ADTES : A/D Test Register

Base address: ADC120 = 0x4009_C000

Offset address: 0x93

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

Bit	Symbol	Function	R/W
1:0	ADTES[1:0]	Selection of A/D conversion target for testing 0 0: ANIxx, temperature sensor output voltage or internal reference voltage*1 (Set by analog input channel specification register (ADS)) 1 0: The '-' side reference voltage (selected by the ADREFM bit of the ADM2 register) 1 1: The '+' side reference voltage (selected by the ADREFP[1:0] bits of the ADM2 register) Others: Setting prohibited.	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. For details about the internal reference voltage, see section 37, Electrical Characteristics.

This register is used to select A/D conversion target. The '+' side reference voltage, '-' side reference voltage, analog input channel, temperature sensor, and internal reference voltage*1 can be selected as the A/D conversion target.

When using this register to test the converter, set as follows.

- For zero-scale measurement, select the '-' side reference voltage as the target for conversion.
- For full-scale measurement, select the '+' side reference voltage as the target for conversion.

The ADTES register can be set by an 8-bit memory manipulation instruction.

The value of this register is 0x00 following a reset.

ADTES[1:0] bits (Selection of A/D conversion target for testing)

These bits are used for selection of A/D conversion target for testing.

29.4 12-bit A/D Converter Operations

The 12-bit A/D converter conversion operations are described below.

<1> The voltage input to the selected analog input channel is sampled by the sample & hold circuit.

<2> When sampling has been done for a certain time, the sample & hold circuit is placed in the hold state and the sampled voltage is held until the A/D conversion operation has ended.

<3> Bit 11 of the successive approximation register (SAR) is set to 1. The series resistor string voltage tap is set to 1/2 AVREF by the tap selector.

<4> The voltage difference between the series resistor string voltage tap and sampled voltage is compared by the voltage comparator. If the analog input is greater than 1/2 AVREF, the MSB of the SAR register remains set to 1. If the analog input is smaller than 1/2 AVREF, the MSB is reset to 0.

<5> Next, bit 10 of the SAR register is automatically set to 1, and the operation proceeds to the next comparison. The series resistor string voltage tap is selected according to the preset value of bit 11, as described below.

- Bit 11 = 1: (3/4) AV_{REF}
- Bit 11 = 0: (1/4) AV_{REF}

The voltage tap and sampled voltage are compared and bit 10 of the SAR register is manipulated as follows.

- Sampled voltage ≥ Voltage tap: Bit 10 = 1
- Sampled voltage < Voltage tap: Bit 10 = 0

<6> Comparison is continued in this way up to bit 0 of the SAR register.

<7> Upon completion of the comparison of 12 bits, an effective digital result value remains in the SAR register, and the result value is transferred to the A/D conversion result register (ADCRn, ADCRnH) and then latched[*1].

At the same time, the A/D conversion end interrupt request signal (ADC_ENDI) can also be generated[*1].

<8> Repeat steps <1> to <7>, until the ADCS bit is cleared to 0[*2].

To stop the 12-bit A/D converter, clear the ADCS bit to 0.

Note: Two types of the A/D conversion result registers are available.
- ADCRn register (16 bits): Store 12-bit or 10-bit A/D conversion value
- ADCRnH register (8 bits): Store 8-bit A/D conversion value

Note: AV_{REF}: The '+' side reference voltage of the 12-bit A/D converter. This can be selected from AV_{REFP}, the internal reference voltage, and V_{CC}.

For details about the internal reference voltage, see section 37, Electrical Characteristics.

Note: n = 0 to 3

Note 1. If the A/D conversion result is outside the A/D conversion result range specified by the ADRCK bit and the ADUL and ADLL registers (see Figure 29.5), the A/D conversion end interrupt request signal is not generated and no A/D conversion results are stored in the ADCRn and ADCRnH registers.

Note 2. While in the sequential conversion mode, the ADCS flag is not automatically cleared to 0. This flag is not automatically cleared to 0 while in the one-shot conversion mode of the hardware trigger no-wait mode, either. Instead, 1 is retained.

Figure 29.6 shows the conversion operation of the 12-bit A/D converter during software trigger no-wait mode.

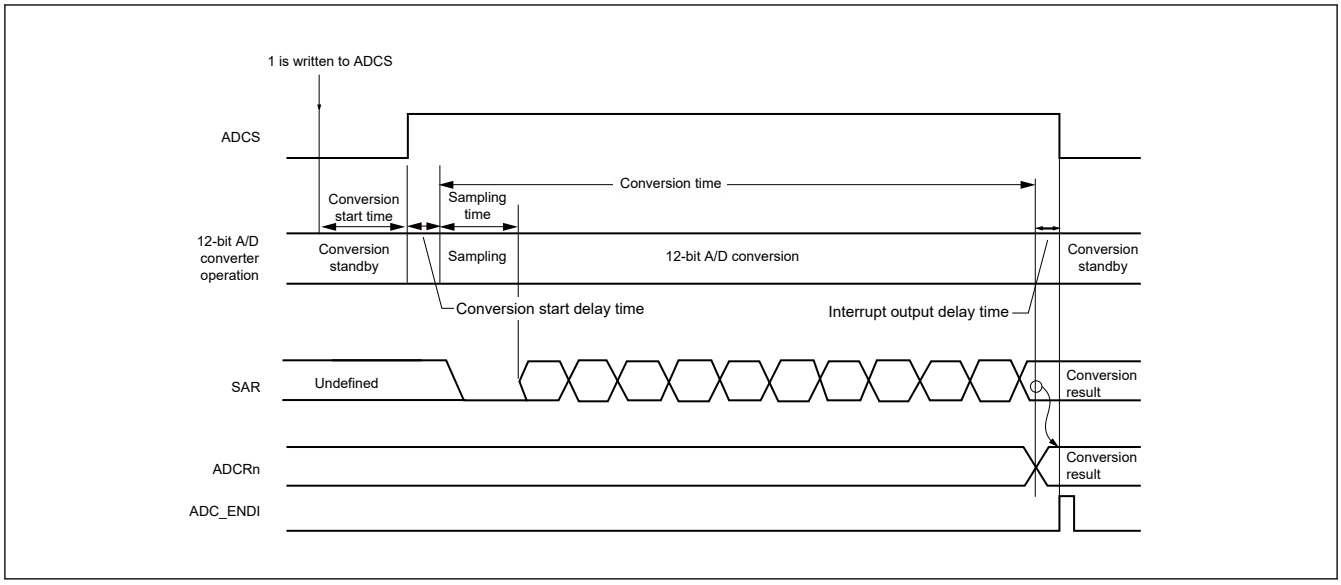


Figure 29.6 Conversion operation of 12-bit A/D converter (software trigger no-wait mode)

In one-shot conversion mode, the ADCS bit is automatically cleared to 0 after completion of A/D conversion.

In sequential conversion mode, A/D conversion operations proceed continuously until the software clears bit 7 (ADCS) of the 12-bit A/D converter mode register 0 (ADM0) to 0.

Writing to the analog input channel specification register (ADS) during A/D conversion interrupts the current conversion after which A/D conversion of the analog input specified by the ADS register proceeds. Data from the A/D conversion that was in progress are discarded.

The value of the A/D conversion result register (ADCRn, ADCRnH) is 0x00 or 0x0000 following a reset.

29.5 Input Voltage and Conversion Results

The relationship between the analog voltage input to the analog input pins (ANI0 to ANI5, ANI16 to ANI19) and the theoretical A/D conversion result (stored in the 12-bit or 10-bit A/D conversion result register (ADCRn)) is shown by the following expression.

$$\mathrm{ADCRn} = \mathrm{INT}\left(\frac{V_{\mathrm{AIN}}}{\mathrm{AV_{REF}}} \times 4096 + 0.5\right)$$

or

$$(\mathrm{ADCRn} - 0.5) \times \frac{\mathrm{AV_{REF}}}{4096} \leq V_{\mathrm{AIN}} < (\mathrm{ADCRn} + 0.5) \times \frac{\mathrm{AV_{REF}}}{4096}$$

where,

INT (): Function which returns integer part of value in parentheses

V_{AIN}: Analog input voltage

AV_{REF}: AV_{REF} pin voltage

ADCRn: 12-bit or 10-bit A/D conversion result register (ADCRn) value

Figure 29.7 shows relationship between analog input voltage and A/D conversion result.

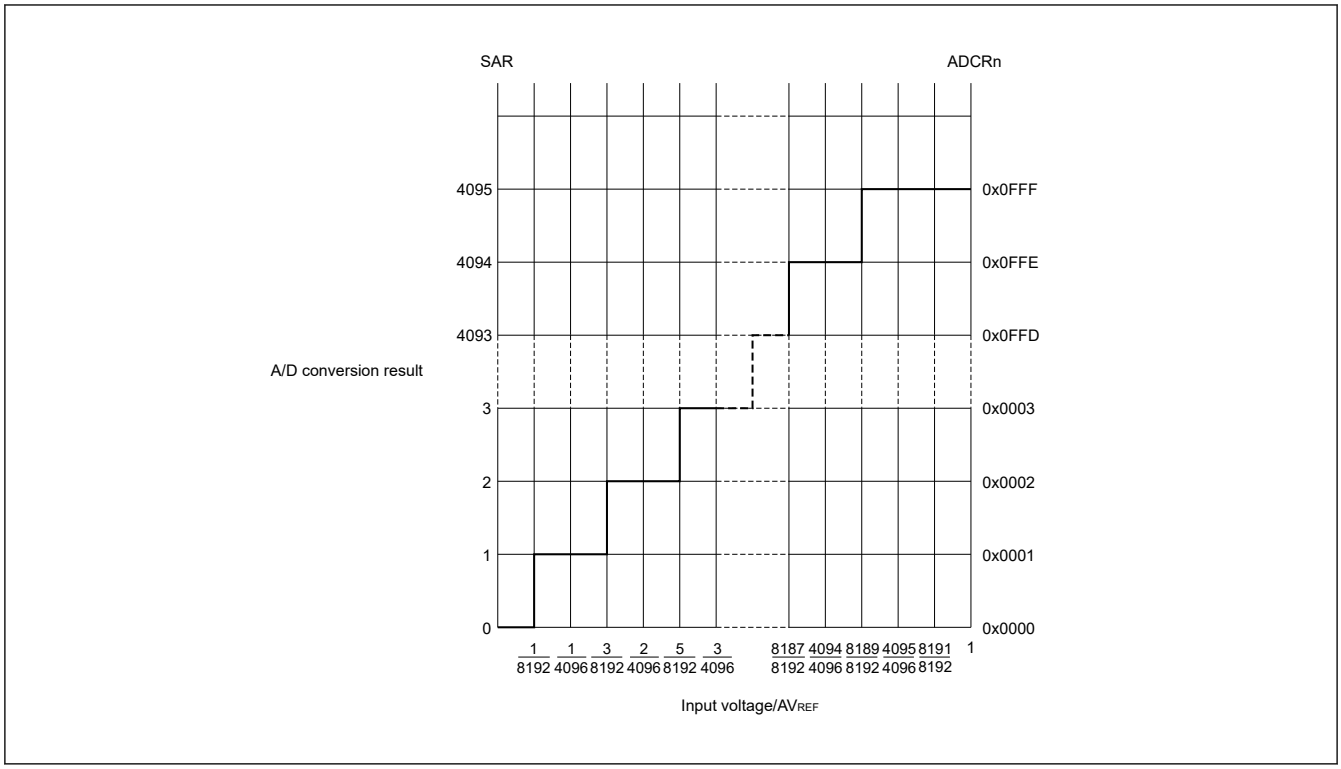


Figure 29.7 Relationship between analog input voltage and A/D conversion result

AV_{REF}: The '+' side reference voltage of the 12-bit A/D converter. This can be selected from AV_{REFP}, the internal reference voltage*1, and V_{CC}.

Note 1. For details about the internal reference voltage, see section 37, Electrical Characteristics.

29.6 12-bit A/D Converter Operation Modes

The operation of each 12-bit A/D converter mode is described below. In addition, the procedure for specifying each mode is described in section 29.7. 12- bit A/D Converter Setup Procedure.

29.6.1 Software Trigger No-wait Mode (Select Mode, Sequential Conversion Mode)

<1> In the conversion stop state, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the 12-bit A/D converter enters the conversion standby state.

<2> After the software counts up to the stabilization wait time (1 µs + 2 cycles of the conversion clock (f_{AD})), the ADCS bit of the ADM0 register is set to 1 to perform the A/D conversion of the analog input specified by the analog input channel specification register (ADS).

<3> When A/D conversion ends, the conversion result is stored in the A/D conversion result registers (ADCR, ADCRH, ADCR0, and ADCR0H), and the A/D conversion end interrupt request signal (ADC_ENDI) is generated. After A/D conversion ends, the next A/D conversion immediately starts.

<4> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<5> When the value of the ADS register is written during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<6> Even if a hardware trigger is input during conversion operation, A/D conversion does not start.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the 12-bit A/D converter enters the conversion standby state.

<8> When ADCE is cleared to 0 while in the A/D conversion standby state, the 12-bit A/D converter enters the conversion stop state.

Figure 29.8 shows the example of software trigger no-wait mode (select mode, sequential conversion mode) operation timing.

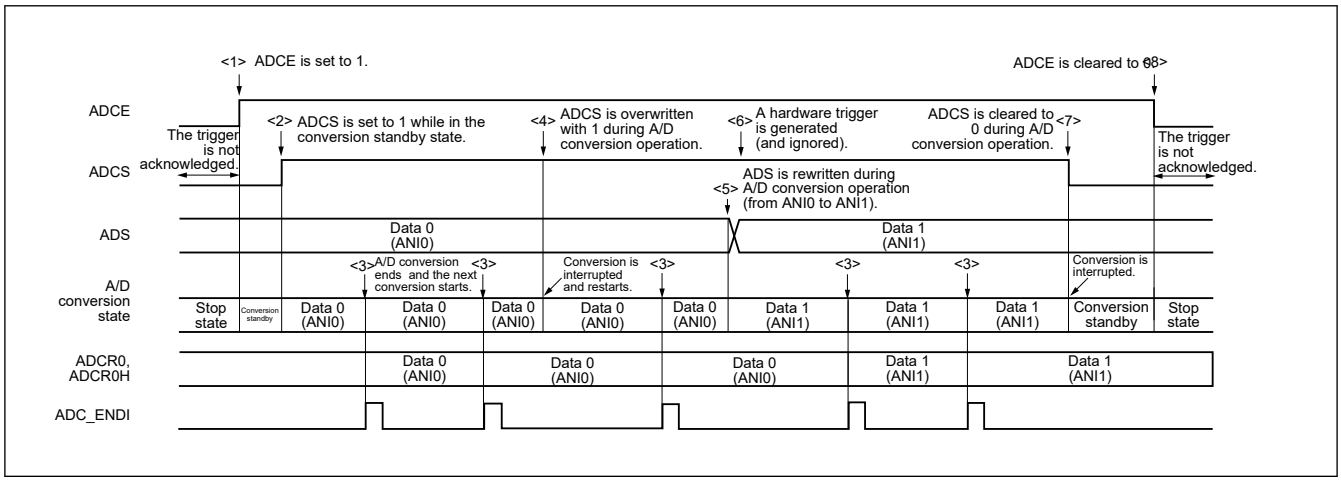


Figure 29.8 Example of software trigger no-wait mode (select mode, sequential conversion mode) operation timing

Note: When <4> or <5> is detected while conversion is in progress, conversion is automatically restarted from the rising edge of the next cycle of the conversion clock (f_{AD}). The conversion time at the first conversion operation restarted is the same as that when there is A/D power supply stabilization wait time in software trigger wait mode or hardware trigger wait mode. See Table 29.9 and Table 29.10.

29.6.2 Software Trigger No-wait Mode (Select Mode, One-shot Conversion Mode)

<1> In the conversion stop state, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the 12-bit A/D converter enters the conversion standby state.

<2> After the software counts up to the stabilization wait time (1 µs + 2 cycles of the conversion clock (f_{AD})), the ADCS bit of the ADM0 register is set to 1 to perform the A/D conversion of the analog input specified by the analog input channel specification register (ADS).

<3> When A/D conversion ends, the conversion result is stored in the A/D conversion result registers (ADCR, ADCRH, ADCR0, and ADCR0H), and the A/D conversion end interrupt request signal (ADC_ENDI) is generated.

<4> After A/D conversion ends, the ADCS bit is automatically cleared to 0, and the 12-bit A/D converter enters the conversion standby state.

<5> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the 12-bit A/D converter enters the conversion standby state.

<8> When ADCE is cleared to 0 while in the A/D conversion standby state, the 12-bit A/D converter enters the conversion stop state. In addition, A/D conversion does not start even if a hardware trigger is input while in the A/D conversion standby state.

Figure 29.9 shows the example of software select no-wait mode (select mode, one-shot conversion mode) operation timing.

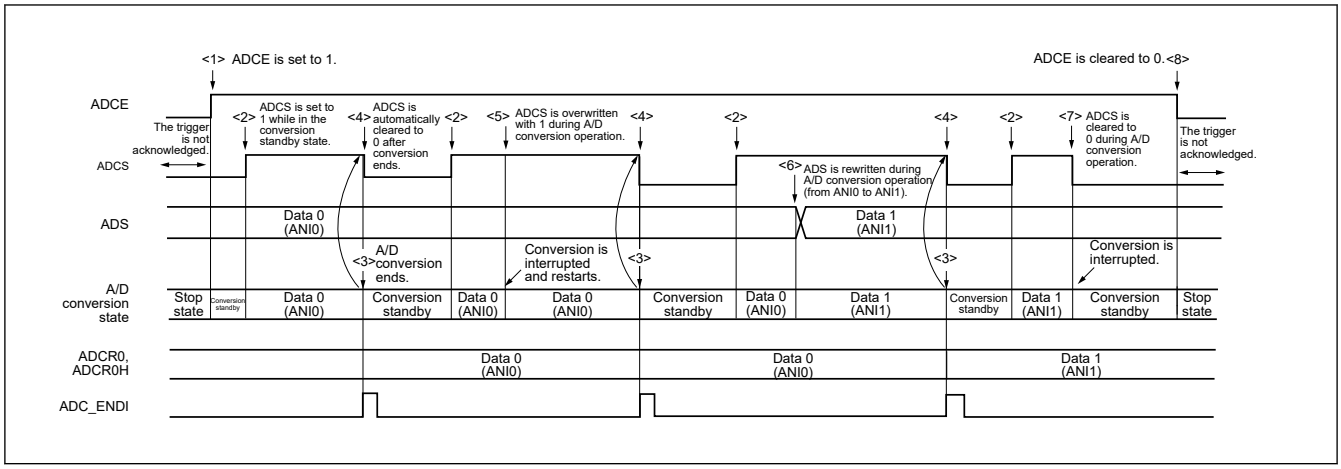


Figure 29.9 Example of software select no-wait mode (select mode, one-shot conversion mode) operation timing

Note: When <5> or <6> is detected while conversion is in progress, conversion is automatically restarted from the rising edge of the next cycle of the conversion clock (f_{AD}). The conversion time at the first conversion operation restarted is the same as that when there is A/D power supply stabilization wait time in software trigger wait mode or hardware trigger wait mode. See Table 29.9 and Table 29.10.

29.6.3 Software Trigger No-wait Mode (Scan Mode, Sequential Conversion Mode)

<1> In the conversion stop state, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the 12-bit A/D converter enters the conversion standby state.

<2> After the software counts up to the stabilization wait time (1 µs + 2 cycles of the conversion clock (f_{AD})), the ADCS bit of the ADM0 register is set to 1 to perform A/D conversion on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<3> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCRn, ADCRnH) each time conversion ends, and the A/D conversion end interrupt request signal (ADC_ENDI) is generated immediately after A/D conversion of the four channels ends. After A/D conversion of the four channels ends, the next A/D conversion of the specified channels automatically starts (until all four channels are finished).

<4> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

<5> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<6> Even if a hardware trigger is input during conversion operation, A/D conversion does not start.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the 12-bit A/D converter enters the conversion standby state.

<8> When ADCE is cleared to 0 while in the A/D conversion standby state, the 12-bit A/D converter enters the conversion stop state.

Figure 29.10 shows the example of software trigger no-wait mode (scan mode, sequential conversion mode) operation timing.

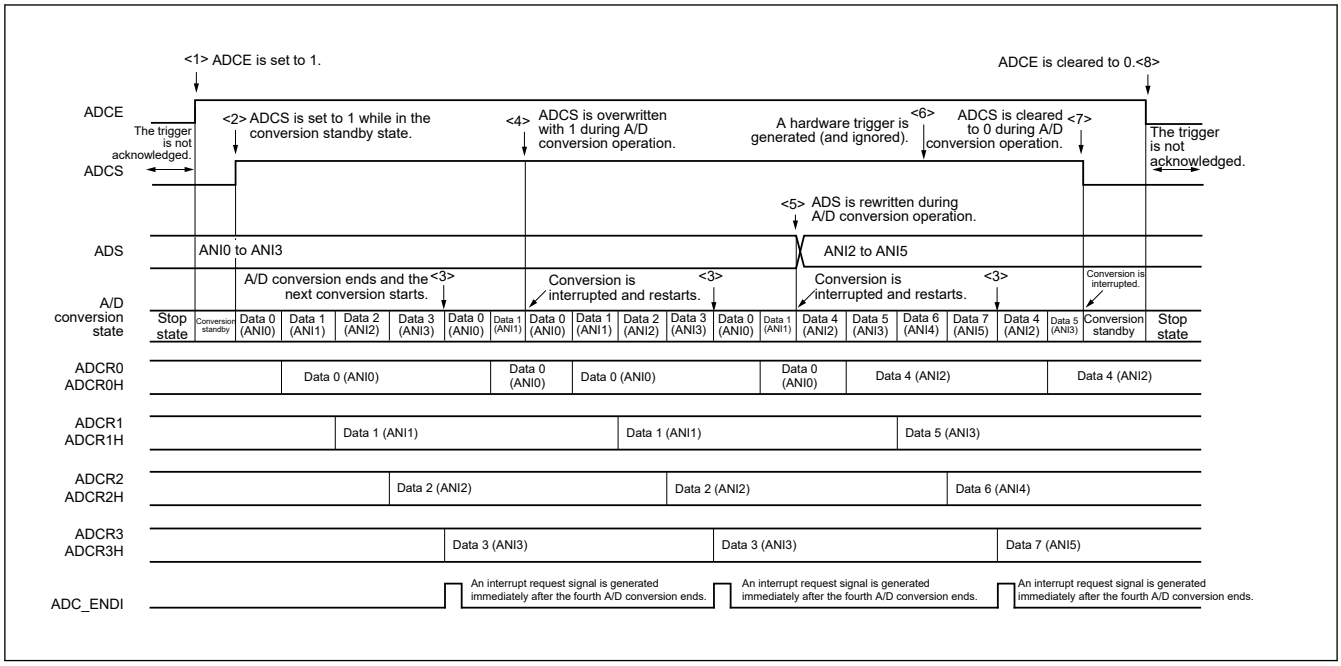


Figure 29.10 Example of software trigger no-wait mode (scan mode, sequential conversion mode) operation timing

Note: When <4> or <5> is detected while conversion is in progress, conversion is automatically restarted from the rising edge of the next cycle of the conversion clock (f_{AD}). The conversion time at the first conversion operation restarted is the same as that when there is A/D power supply stabilization wait time in software trigger wait mode or hardware trigger wait mode. See Table 29.9 and Table 29.10.

29.6.4 Software Trigger No-wait Mode (Scan Mode, One-shot Conversion Mode)

<1> In the conversion stop state, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the 12-bit A/D converter enters the conversion standby state.

<2> After the software counts up to the stabilization wait time (1 µs + 2 cycles of the conversion clock (f_{AD})), the ADCS bit of the ADM0 register is set to 1 to perform A/D conversion on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<3> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCRn, ADCRnH) each time conversion ends, and the A/D conversion end interrupt request signal (ADC_ENDI) is generated immediately after A/D conversion of the four channels ends.

<4> After A/D conversion of the four channels ends, the ADCS bit is automatically cleared to 0, and the 12-bit A/D converter enters the conversion standby state.

<5> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the 12-bit A/D converter enters the conversion standby state.

<8> When ADCE is cleared to 0 while in the A/D conversion standby state, the 12-bit A/D converter enters the conversion stop state. In addition, A/D conversion does not start even if a hardware trigger is input while in the A/D conversion standby state.

Figure 29.11 shows the example of software trigger no-wait mode (scan mode, one-shot conversion mode) operation timing.

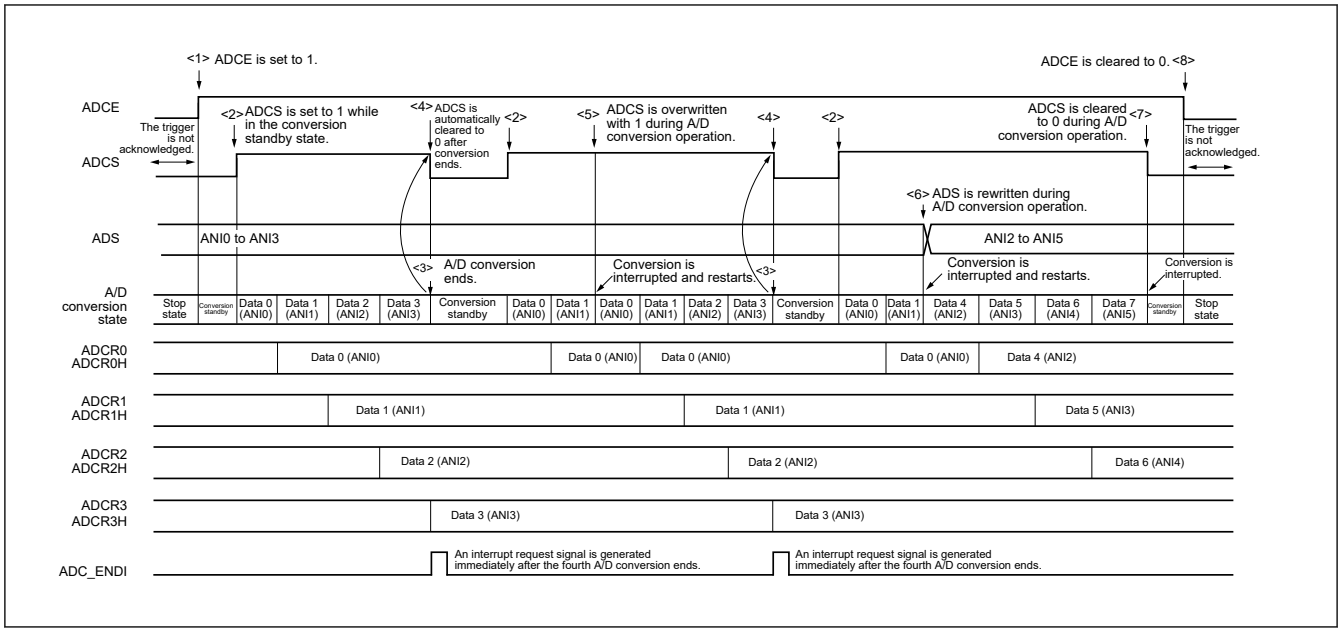


Figure 29.11 Example of software trigger no-wait mode (scan mode, one-shot conversion mode) operation timing

Note: When <5> or <6> is detected while conversion is in progress, conversion is automatically restarted from the rising edge of the next cycle of the conversion clock (f_{AD}). The conversion time at the first conversion operation restarted is the same as that when there is A/D power supply stabilization wait time in software trigger wait mode or hardware trigger wait mode. See Table 29.9 and Table 29.10.

29.6.5 Software Trigger Wait Mode (Select Mode, Sequential Conversion Mode)

<1> To shift to software trigger wait mode, the ADCE bit of A/D converter mode register 0 (ADM0) must be set to 0 (conversion stop state).

<2> If ADCS is set to 1 in the conversion stop state, A/D conversion is performed on the analog input specified by the analog input channel specification register (ADS) (software trigger wait mode).

<3> When A/D conversion ends, the conversion result is stored in the A/D conversion result registers (ADCR, ADCRH, ADCR0, and ADCR0H), and the A/D conversion end interrupt request signal (ADC_ENDI) is generated. After A/D conversion ends, the next A/D conversion immediately starts.

<4> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<5> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<6> Even if a hardware trigger is input during conversion operation, A/D conversion does not start.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the 12-bit A/D converter enters the conversion stop state.

Figure 29.12 shows the example of software trigger wait mode (select mode, sequential conversion mode) operation timing.

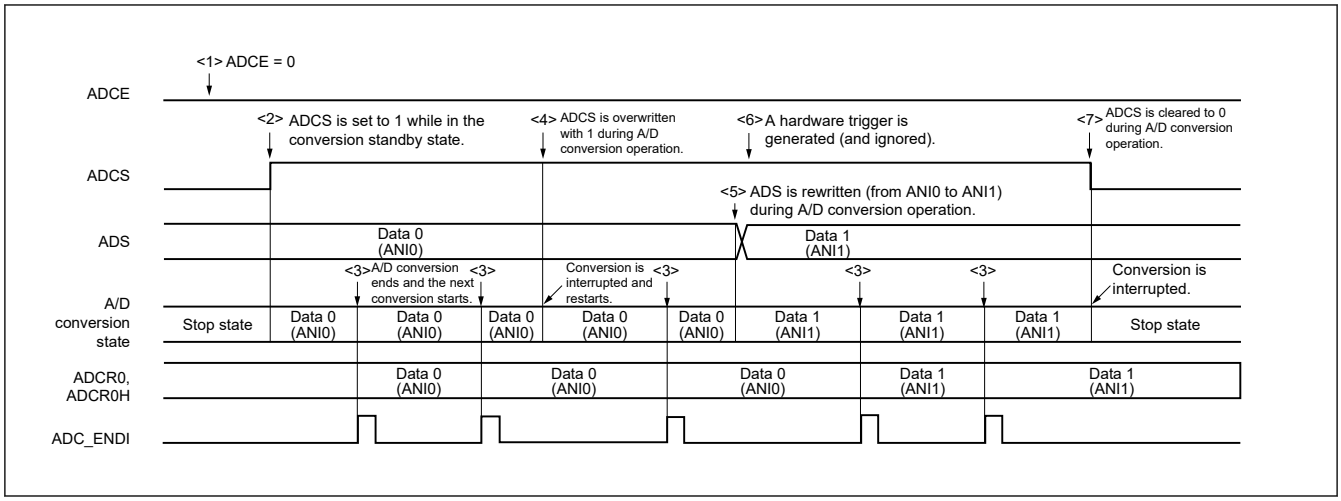


Figure 29.12 Example of software trigger wait mode (select mode, sequential conversion mode) operation timing

Note: When <4> or <5> is detected during conversion operation, conversion is restarted automatically after the stabilization wait time has passed since the rising edge of the next conversion clock (f_{AD}). The conversion time at the first conversion operation restarted is the same as that when there is A/D power supply stabilization wait time in software trigger wait mode or hardware trigger wait mode. See Table 29.9 and Table 29.10.

29.6.6 Software Trigger Wait Mode (Select Mode, One-shot Conversion Mode)

<1> To shift to software trigger wait mode, the ADCE bit of A/D converter mode register 0 (ADM0) must be set to 0 (conversion stop state).

<2> If ADCS is set to 1 in the conversion stop state, A/D conversion is performed on the analog input specified by the analog input channel specification register (ADS) (software trigger wait mode).

<3> When A/D conversion ends, the conversion result is stored in the A/D conversion result registers (ADCR, ADCRH, ADCR0, and ADCR0H), and the A/D conversion end interrupt request signal (ADC_ENDI) is generated.

<4> After A/D conversion ends, the ADCS bit is automatically cleared to 0, and the 12-bit A/D converter enters the conversion stop state.

<5> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is initialized.

<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the 12-bit A/D converter enters the conversion stop state.

<8> When a hardware trigger is input during conversion operation, the trigger is not accepted.

Figure 29.13 shows the example of software trigger wait mode (select mode, one-shot conversion mode) operation timing.

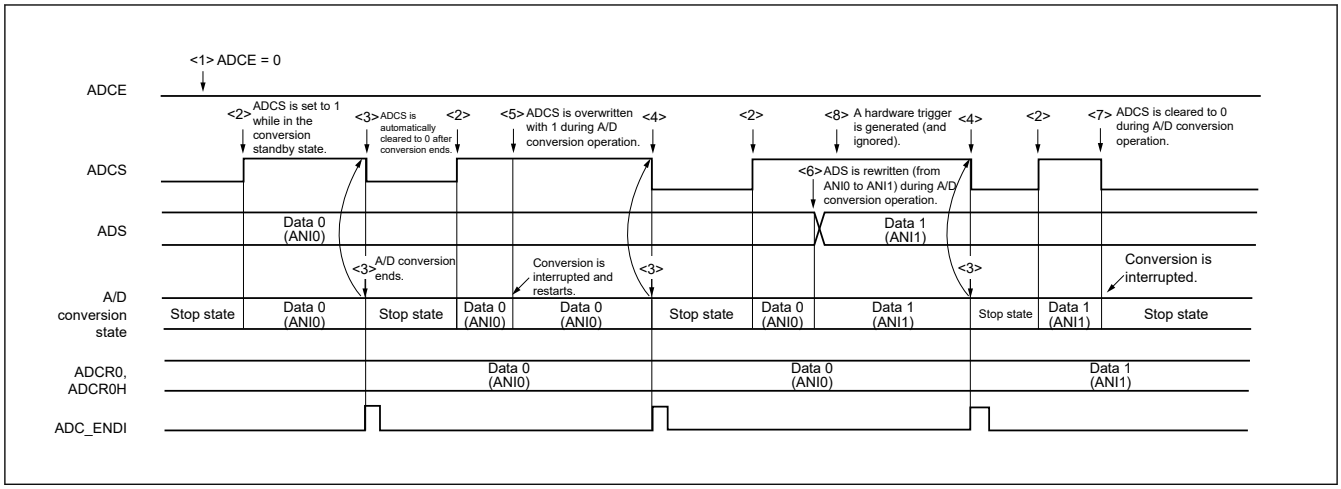


Figure 29.13 Example of software trigger wait mode (select mode, one-shot conversion mode) operation timing

Note: When <5> or <6> is detected during conversion operation, conversion is restarted automatically after the stabilization wait time has passed since the rising edge of the next conversion clock (f_{AD}). The conversion time at the first conversion operation restarted is the same as that when there is A/D power supply stabilization wait time in software trigger wait mode or hardware trigger wait mode. See Table 29.9 and Table 29.10.

29.6.7 Software Trigger Wait Mode (Scan Mode, Sequential Conversion Mode)

<1> To shift to software trigger wait mode, the ADCE bit of A/D converter mode register 0 (ADM0) must be set to 0 (conversion stop state).

<2> If ADCS is set to 1 in the conversion stop state, A/D conversion is performed on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS) (software trigger wait mode).

A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<3> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCRn, ADCRnH) each time conversion ends, and the A/D conversion end interrupt request signal (ADC_ENDI) is generated immediately after A/D conversion of the four channels ends. After A/D conversion of the four channels ends, the next A/D conversion of the specified channels automatically starts.

<4> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<5> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<6> When a hardware trigger is input during conversion operation, the trigger is not accepted.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the 12-bit A/D converter enters the conversion stop state.

Figure 29.14 shows the example of software trigger wait mode (scan mode, sequential conversion mode) operation timing.

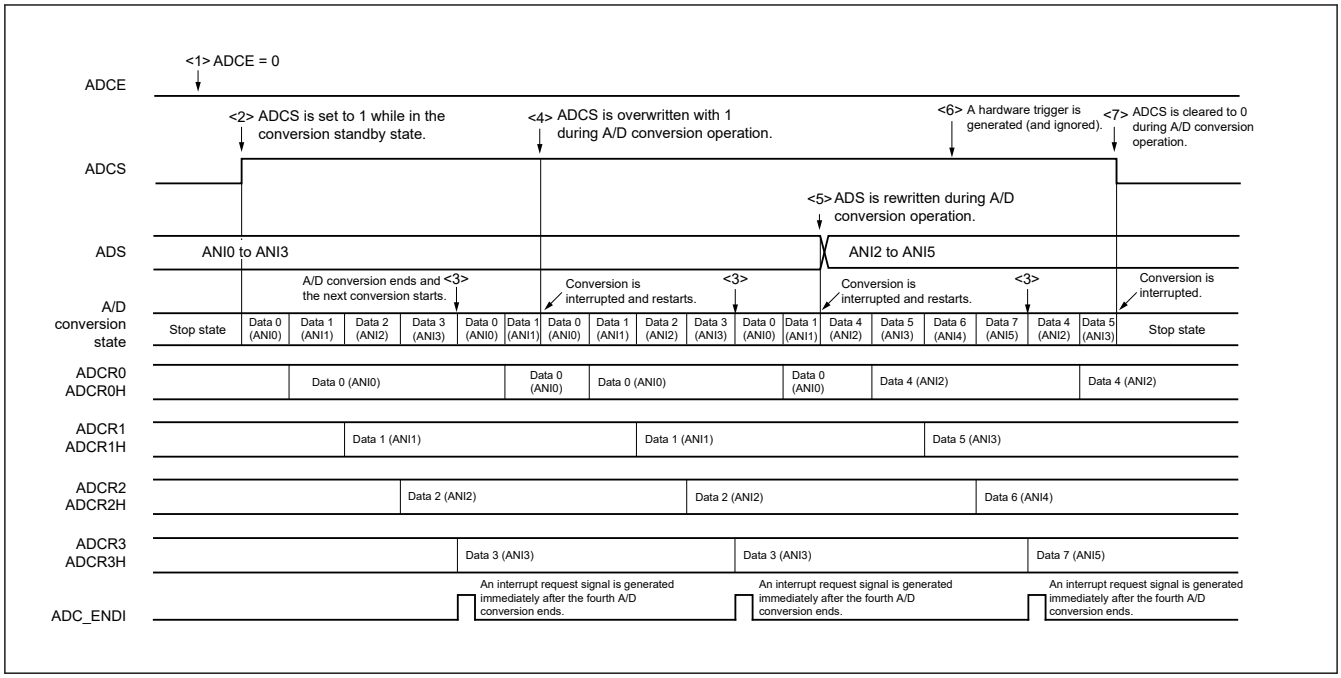


Figure 29.14 Example of software trigger wait mode (scan mode, sequential conversion mode) operation timing

Note: When <4> or <5> is detected during conversion operation, conversion is restarted automatically after the stabilization wait time has passed since the rising edge of the next conversion clock (f_{AD}). The conversion time at the first conversion operation restarted is the same as that when there is A/D power supply stabilization wait time in software trigger wait mode or hardware trigger wait mode. See Table 29.9 and Table 29.10.

29.6.8 Software Trigger Wait Mode (Scan Mode, One-shot Conversion Mode)

<1> To shift to software trigger wait mode, the ADCE bit of A/D converter mode register 0 (ADM0) must be set to 0 (conversion stop state).

<2> If ADCS is set to 1 in the conversion stop state, A/D conversion is performed on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS) (software trigger wait mode).

A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<3> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCRn, ADCRnH) each time conversion ends, and the A/D conversion end interrupt request signal (ADC_ENDI) is generated immediately after A/D conversion of the four channels ends.

<4> After A/D conversion ends, the ADCS bit is automatically cleared to 0, and the 12-bit A/D converter enters the conversion stop state.

<5> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the 12-bit A/D converter enters the conversion stop state.

<8> When a hardware trigger is input during conversion operation, the trigger is not accepted.

Figure 29.15 shows the example of software trigger wait mode (scan mode, one-shot conversion mode) operation timing.

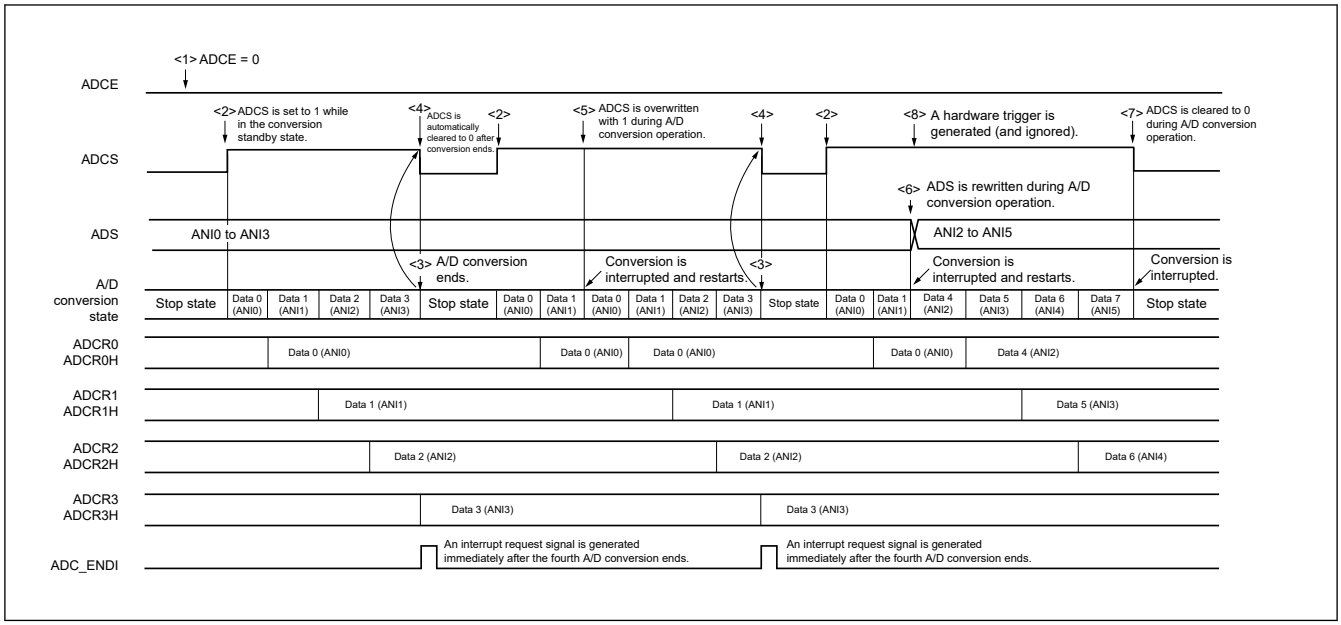


Figure 29.15 Example of software trigger wait mode (scan mode, one-shot conversion mode) operation timing

Note: When <5> or <6> is detected during conversion operation, conversion is restarted automatically after the stabilization wait time has passed since the rising edge of the next conversion clock (f_{AD}). The conversion time at the first conversion operation restarted is the same as that when there is A/D power supply stabilization wait time in software trigger wait mode or hardware trigger wait mode. See Table 29.9 and Table 29.10.

29.6.9 Hardware Trigger No-wait Mode (Select Mode, Sequential Conversion Mode)

<1> In the conversion stop state, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the 12-bit A/D converter enters the conversion standby state.

<2> After the software counts up to the stabilization wait time (1 µs + 2 cycles of the conversion clock (f_{AD})), the ADCS bit of the ADM0 register is set to 1 to place the 12-bit A/D converter in the hardware trigger standby state (and conversion does not start at this stage). Note that, while in this state, A/D conversion does not start even if ADCS is set to 1.

<3> If a hardware trigger is input while ADCS = 1, A/D conversion is performed on the analog input specified by the analog input channel specification register (ADS).

<4> When A/D conversion ends, the conversion result is stored in the A/D conversion result registers (ADCR, ADCRH, ADCR0, and ADCR0H), and the A/D conversion end interrupt request signal (ADC_ENDI) is generated. After A/D conversion ends, the next A/D conversion immediately starts.

<5> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<7> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<8> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the 12-bit A/D converter enters the conversion standby state. However, the 12-bit A/D converter does not stop in this state.

<9> When ADCE is cleared to 0 while in the A/D conversion standby state, the 12-bit A/D converter enters the conversion stop state.

When ADCS = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 29.16 shows the example of hardware trigger no-wait mode (select mode, sequential conversion mode) operation timing.

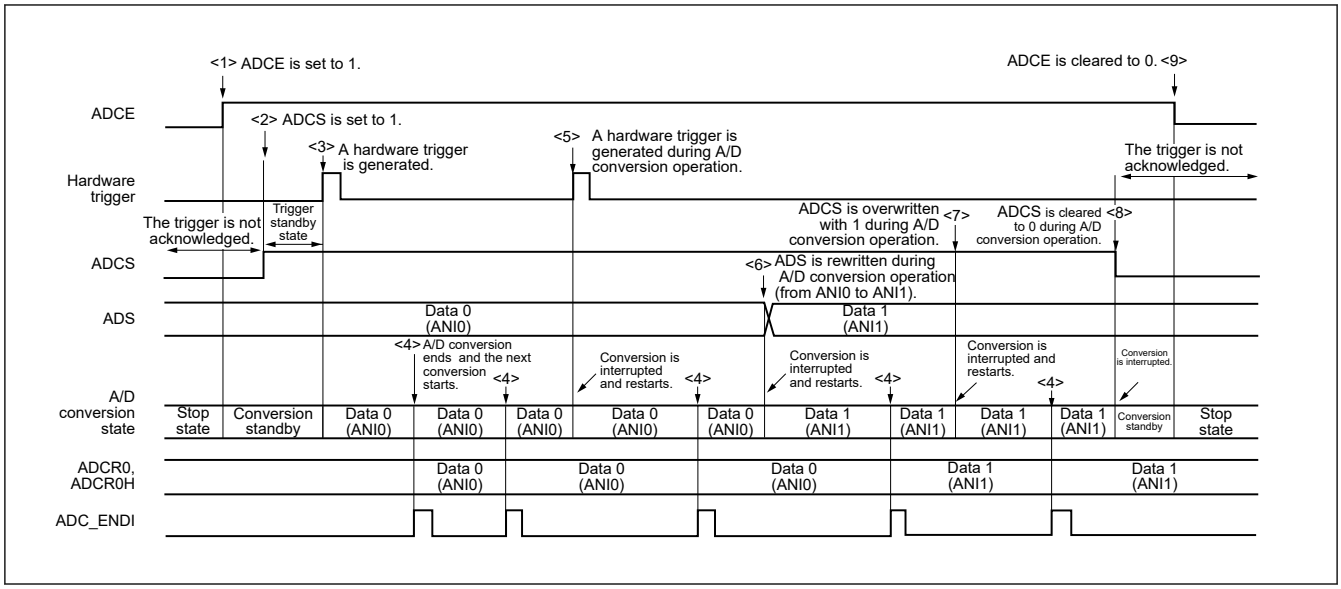


Figure 29.16 Example of hardware trigger no-wait mode (select mode, sequential conversion mode) operation timing

Note: When <5>, <6>, or <7> is detected while conversion is in progress, conversion is automatically restarted from the rising edge of the next cycle of the conversion clock (f_{AD}). The conversion time at the first conversion operation restarted is the same as that when there is A/D power supply stabilization wait time in software trigger wait mode or hardware trigger wait mode. See Table 29.9 and Table 29.10.

29.6.10 Hardware Trigger No-wait Mode (Select Mode, One-shot Conversion Mode)

<1> In the conversion stop state, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the 12-bit A/D converter enters the conversion standby state.

<2> After the software counts up to the stabilization wait time (1 μs + 2 cycles of the conversion clock (f_{AD})), the ADCS bit of the ADM0 register is set to 1 to place the 12-bit A/D converter in the hardware trigger standby state (and conversion does not start at this stage). Note that, while in this state, A/D conversion does not start even if ADCS is set to 1.

<3> If a hardware trigger is input while ADCS = 1, A/D conversion is performed on the analog input specified by the analog input channel specification register (ADS).

<4> When A/D conversion ends, the conversion result is stored in the A/D conversion result registers (ADCR, ADCRH, ADCR0, and ADCR0H), and the A/D conversion end interrupt request signal (ADC_ENDI) is generated.

<5> After A/D conversion ends, the ADCS bit remains set to 1, and the 12-bit A/D converter enters the hardware trigger standby state.

<6> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<7> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<8> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<9> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the 12-bit A/D converter enters the conversion standby state. However, the 12-bit A/D converter does not stop in this state.

<10>When ADCE is cleared to 0 while in the A/D conversion standby state, the 12-bit A/D converter enters the conversion stop state.

When ADCS = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 29.17 shows the example of hardware trigger no-wait mode (select mode, one-shot conversion mode) operation timing.

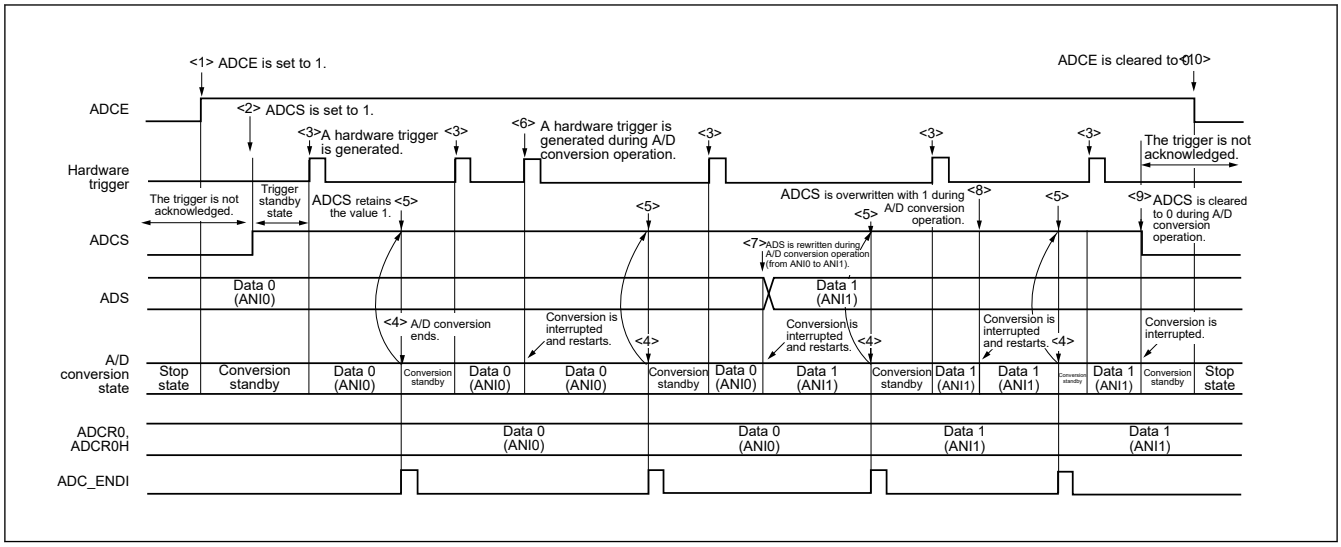


Figure 29.17 Example of hardware trigger no-wait mode (select mode, one-shot conversion mode) operation timing

Note: When <6>, <7>, or <8> is detected while conversion is in progress, conversion is automatically restarted from the rising edge of the next cycle of the conversion clock (f_{AD}). The conversion time at the first conversion operation restarted is the same as that when there is A/D power supply stabilization wait time in software trigger wait mode or hardware trigger wait mode. See Table 29.9 and Table 29.10.

29.6.11 Hardware Trigger No-wait Mode (Scan Mode, Sequential Conversion Mode)

<1> In the conversion stop state, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the 12-bit A/D converter enters the conversion standby state.

<2> After the software counts up to the stabilization wait time (1 μs + 2 cycles of the conversion clock (f_{AD})), the ADCS bit of the ADM0 register is set to 1 to place the 12-bit A/D converter in the hardware trigger standby state (and conversion does not start at this stage). Note that, while in this state, A/D conversion does not start even if ADCS is set to 1.

<3> If a hardware trigger is input while ADCS = 1, A/D conversion is performed on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<4> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCRn, ADCRnH) each time conversion ends, and the A/D conversion end interrupt request signal (ADC_ENDI) is generated immediately after A/D conversion of the four channels ends. After A/D conversion of the four channels ends, the next A/D conversion of the specified channels automatically starts.

<5> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<7> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<8> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the 12-bit A/D converter enters the conversion standby state. However, the 12-bit A/D converter does not stop in this state.

<9> When ADCE is cleared to 0 while in the A/D conversion standby state, the 12-bit A/D converter enters the conversion stop state.

When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start.

Figure 29.18 shows the example of hardware trigger no-wait mode (scan mode, sequential conversion mode) operation timing.

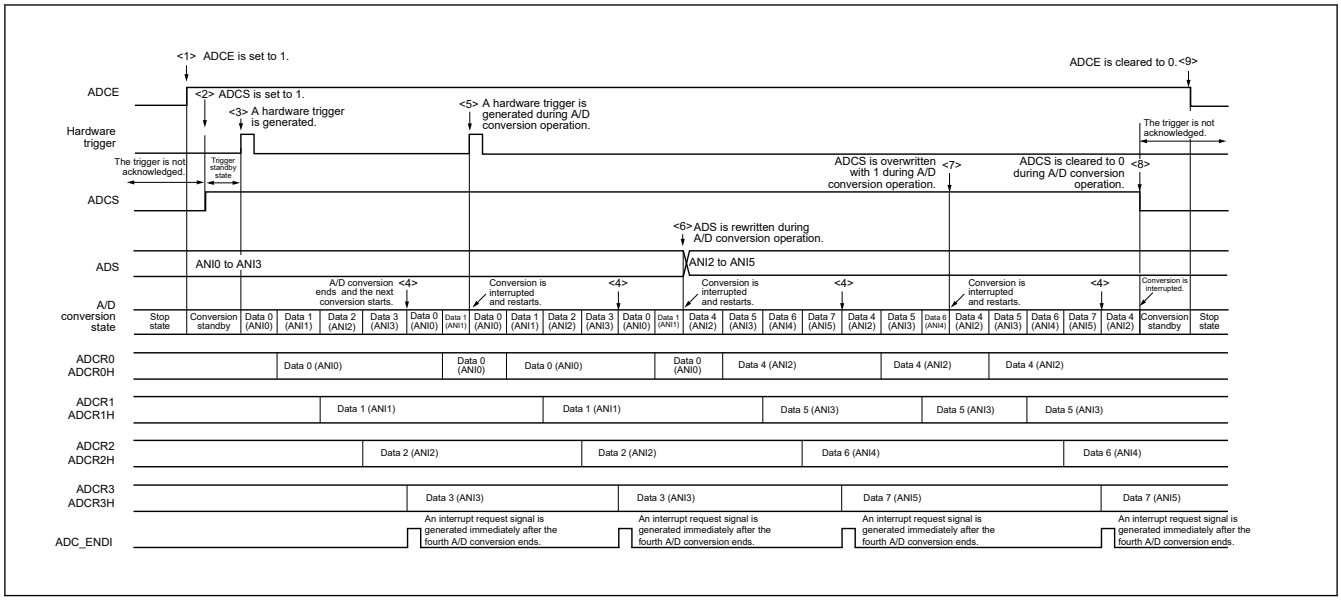


Figure 29.18 Example of hardware trigger no-wait mode (scan mode, sequential conversion mode) operation timing

Note: When <5>, <6>, or <7> is detected while conversion is in progress, conversion is automatically restarted from the rising edge of the next cycle of the conversion clock (f_{AD}). The conversion time at the first conversion restarted operation is the same as that when there is A/D power supply stabilization wait time in software trigger wait mode or hardware trigger wait mode. See Table 29.9 and Table 29.10.

29.6.12 Hardware Trigger No-wait Mode (Scan Mode, One-shot Conversion Mode)

<1> In the conversion stop state, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the 12-bit A/D converter enters the conversion standby state.

<2> After the software counts up to the stabilization wait time (1 μs + 2 cycles of the conversion clock (f_{AD})), the ADCS bit of the ADM0 register is set to 1 to place the 12-bit A/D converter in the hardware trigger standby state (and conversion does not start at this stage). Note that, while in this state, A/D conversion does not start even if ADCS is set to 1.

<3> If a hardware trigger is input while ADCS = 1, A/D conversion is performed on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<4> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCRn, ADCRnH) each time conversion ends, and the A/D conversion end interrupt request signal (ADC_ENDI) is generated immediately after A/D conversion of the four channels ends.

<5> After A/D conversion of the four channels ends, the ADCS bit remains set to 1, and the 12-bit A/D converter enters the hardware trigger standby state.

<6> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

<7> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<8> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

<9> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the 12-bit A/D converter enters the conversion standby state. However, the 12-bit A/D converter does not stop in this state.

<10>When ADCE is cleared to 0 while in the A/D conversion standby state, the 12-bit A/D converter enters the conversion stop state.

When ADCS = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 29.19 shows the example of hardware trigger no-wait mode (scan mode, one-shot conversion mode) operation timing.

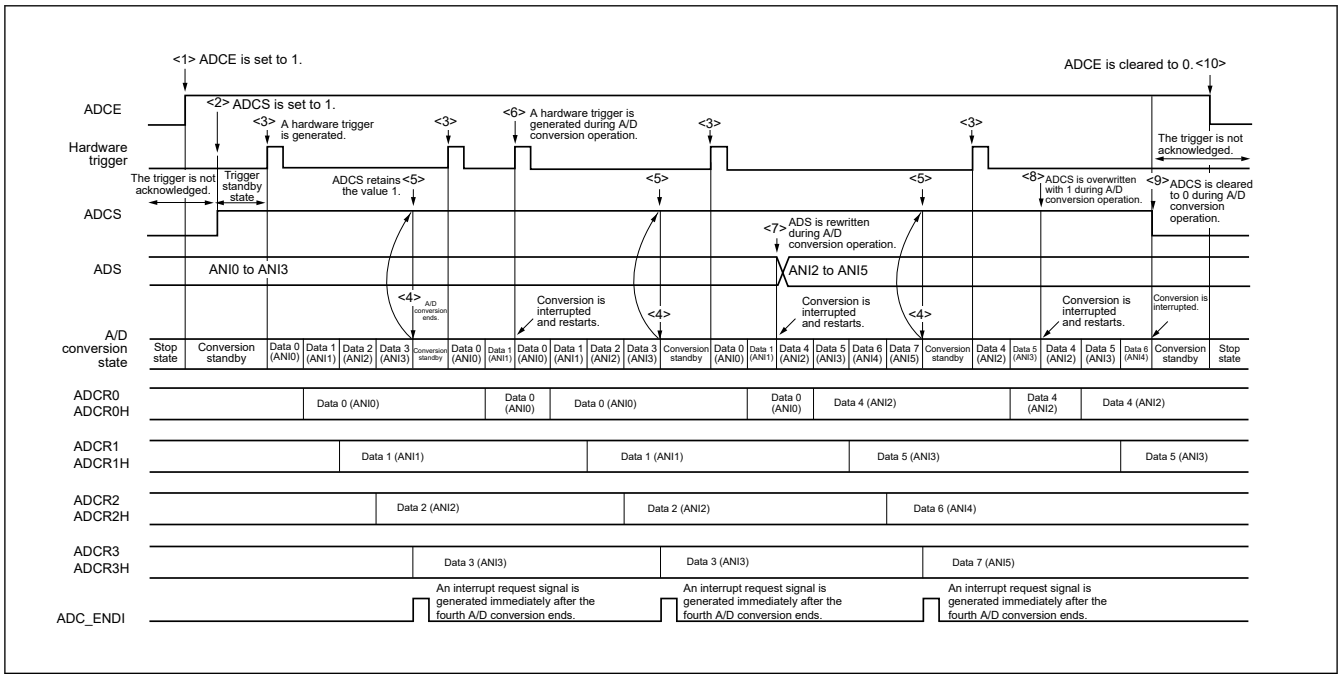


Figure 29.19 Example of hardware trigger no-wait mode (scan mode, one-shot conversion mode) operation timing

Note: When <6>, <7>, or <8> is detected while conversion is in progress, conversion is automatically restarted from the rising edge of the next cycle of the conversion clock (f_{AD}). The conversion time at the first conversion operation restarted is the same as that when there is A/D power supply stabilization wait time in software trigger wait mode or hardware trigger wait mode. See Table 29.9 and Table 29.10.

29.6.13 Hardware Trigger Wait Mode (Select Mode, Sequential Conversion Mode)

<1> In the conversion stop state, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the 12-bit A/D converter enters the hardware trigger standby state.

<2> If a hardware trigger is input while in the hardware trigger standby state, A/D conversion is performed on the analog input specified by the analog input channel specification register (ADS). The ADCS bit of the ADM0 register is automatically set to 1 according to the hardware trigger input.

<3> When A/D conversion ends, the conversion result is stored in the A/D conversion result registers (ADCR, ADCRH, ADCR0, and ADCR0H), and the A/D conversion end interrupt request signal (ADC_ENDI) is generated. After A/D conversion ends, the next A/D conversion immediately starts. (At this time, no hardware trigger is necessary.)

<4> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<5> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<6> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, the 12-bit A/D converter enters the hardware trigger standby state, and the 12-bit A/D converter enters the conversion stop state. When ADCE = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 29.20 shows the example of hardware trigger wait mode (select mode, sequential conversion mode) operation timing.

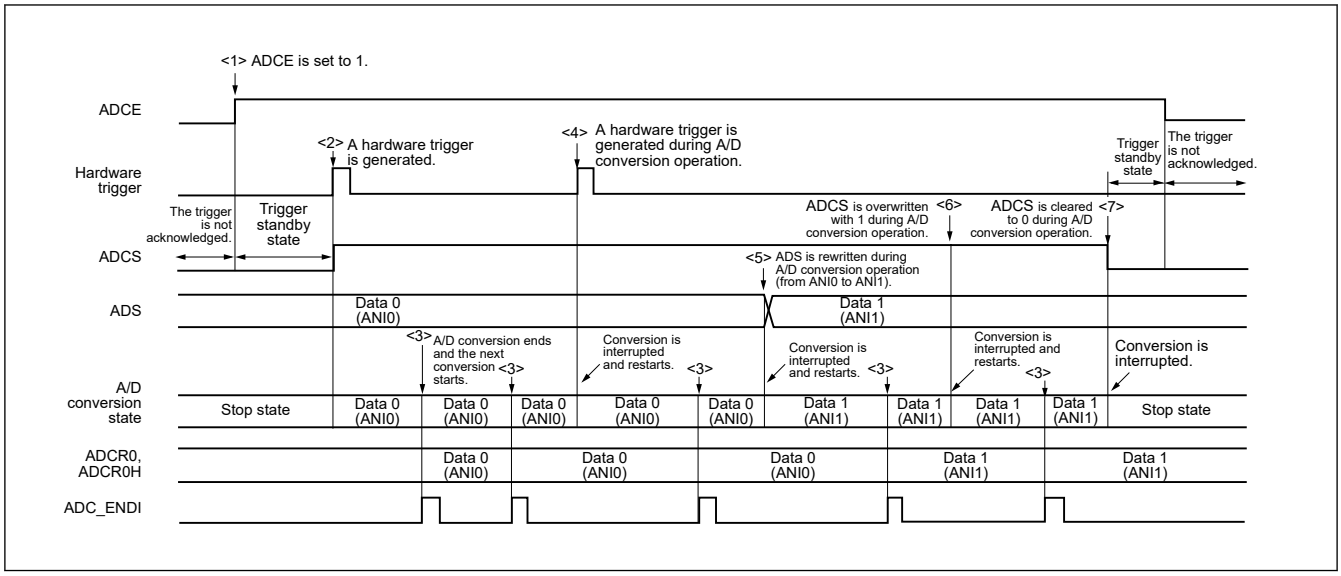


Figure 29.20 Example of hardware trigger wait mode (select mode, sequential conversion mode) operation timing

Note: When <4>, <5>, or <6> is detected during conversion operation, conversion is restarted automatically after the stabilization wait time has passed since the rising edge of the next conversion clock (f_{AD}). The conversion time at the first conversion operation restarted is the same as that when there is A/D power supply stabilization wait time in software trigger wait mode or hardware trigger wait mode. See Table 29.9 and Table 29.10.

29.6.14 Hardware Trigger Wait Mode (Select Mode, One-shot Conversion Mode)

<1> In the conversion stop state, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the 12-bit A/D converter enters the hardware trigger standby state.

<2> If a hardware trigger is input while in the hardware trigger standby state, A/D conversion is performed on the analog input specified by the analog input channel specification register (ADS). The ADCS bit of the ADM0 register is automatically set to 1 according to the hardware trigger input.

<3> When A/D conversion ends, the conversion result is stored in the A/D conversion result registers (ADCR, ADCRH, ADCR0, and ADCR0H), and the A/D conversion end interrupt request signal (ADC_ENDI) is generated.

<4> After A/D conversion ends, the ADCS bit is automatically cleared to 0, and the 12-bit A/D converter enters the conversion stop state.

<5> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<7> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is initialized.

<8> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, the 12-bit A/D converter enters the hardware trigger standby state, and the 12-bit A/D converter enters the conversion stop state. When ADCE = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 29.21 shows the example of hardware trigger wait mode (select mode, one-shot conversion mode) operation timing.

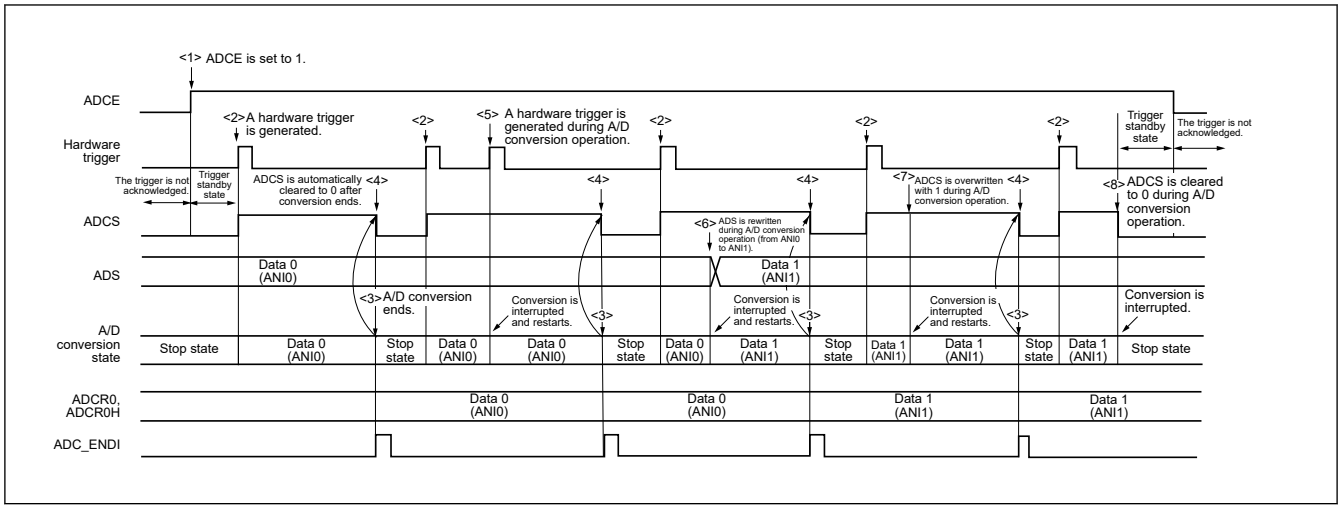


Figure 29.21 Example of hardware trigger wait mode (select mode, one-shot conversion mode) operation timing

Note: When <5>, <6>, or <7> is detected during conversion operation, conversion is restarted automatically after the stabilization wait time has passed since the rising edge of the next conversion clock (f_{AD}).The conversion time at the first conversion operation restarted is the same as that when there is A/D power supply stabilization wait time in software trigger wait mode or hardware trigger wait mode. See Table 29.9 and Table 29.10.

Note: The setting of ADISS being 1 (the input source is temperature sensor output voltage or internal reference voltage) cannot be used in the hardware trigger wait mode (select mode and one-shot conversion mode).

29.6.15 Hardware Trigger Wait Mode (Scan Mode, Sequential Conversion Mode)

<1> In the conversion stop state, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the 12-bit A/D converter enters the hardware trigger standby state.

<2> If a hardware trigger is input while in the hardware trigger standby state, A/D conversion is performed on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). The ADCS bit of the ADM0 register is automatically set to 1 according to the hardware trigger input. A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<3> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCRn, ADCRnH) each time conversion ends, and the A/D conversion end interrupt request signal (ADC_ENDI) is generated immediately after A/D conversion of the four channels ends. After A/D conversion of the four channels ends, the next A/D conversion of the specified channels automatically starts.

<4> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

<5> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<6> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, the 12-bit A/D converter enters the hardware trigger standby state, and the 12-bit A/D converter enters the conversion stop state. When ADCE = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 29.22 shows the example of hardware trigger wait mode (scan mode, sequential conversion mode) operation timing.

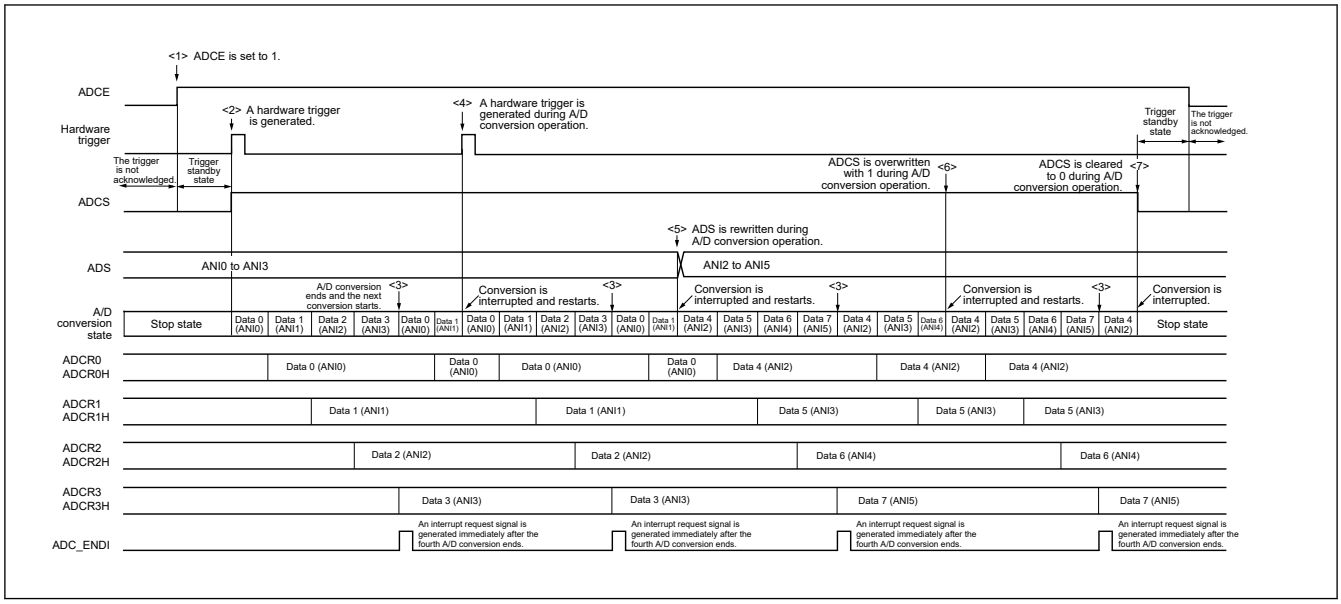


Figure 29.22 Example of hardware trigger wait mode (scan mode, sequential conversion mode) operation timing

Note: When <4>, <5>, or <6> is detected during conversion operation, conversion is restarted automatically after the stabilization wait time has passed since the rising edge of the next conversion clock (f_{AD}). The conversion time at the first conversion operation restarted is the same as that when there is A/D power supply stabilization wait time in software trigger wait mode or hardware trigger wait mode. See Table 29.9 and Table 29.10.

29.6.16 Hardware Trigger Wait Mode (Scan Mode, One-shot Conversion Mode)

<1> In the conversion stop state, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the 12-bit A/D converter enters the hardware trigger standby state.

<2> If a hardware trigger is input while in the hardware trigger standby state, A/D conversion is performed on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). The ADCS bit of the ADM0 register is automatically set to 1 according to the hardware trigger input. A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<3> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCRn, ADCRnH) each time conversion ends, and the A/D conversion end interrupt request signal (ADC_ENDI) is generated immediately after A/D conversion of the four channels ends.

<4> After A/D conversion ends, the ADCS bit is automatically cleared to 0, and the 12-bit A/D converter enters the conversion stop state.

<5> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<7> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<8> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, the 12-bit A/D converter enters the hardware trigger standby state, and the 12-bit A/D converter enters the conversion stop state. When ADCE = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 29.23 shows the example of hardware trigger wait mode (scan mode, one-shot conversion mode) operation timing.

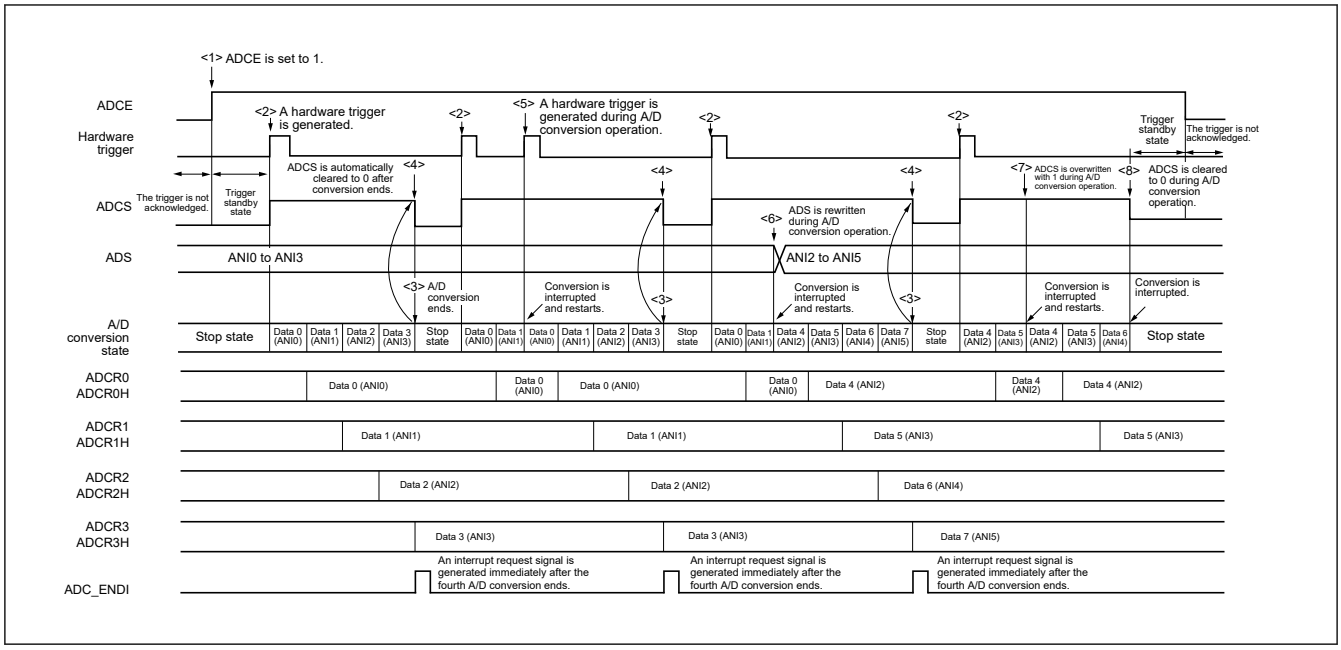


Figure 29.23 Example of hardware trigger wait mode (scan mode, one-shot conversion mode) operation timing

Note: When <5>, <6>, or <7> is detected during conversion operation, conversion is restarted automatically after the stabilization wait time has passed since the rising edge of the next conversion clock (f_{AD}). The conversion time at the first conversion operation restarted is the same as that when there is A/D power supply stabilization wait time in software trigger wait mode or hardware trigger wait mode. See Table 29.9 and Table 29.10.

29.7 12- bit A/D Converter Setup Procedure

The 12-bit A/D converter setup procedure in each operation mode is described in the following section.

29.7.1 Setting up Software Trigger No-wait Mode

Table 29.14 shows the setup steps in software trigger no-wait mode.

Table 29.14 Setting up software trigger no-wait mode

Step		Process	Detail
Setting up Software Trigger No-Wait Mode	<1>	• ADM0 register setting • ADM1 register setting • ADM2 register setting • ADUL and ADLL register setting • ADS register setting (The order of the settings is irrelevant.)	• ADM0 register FR[2:0], LV[1:0]: bits: These are used to specify the A/D conversion time . ADMD bit: Select mode or scan mode • ADM1 register ADTMD[1:0] bits: These are used to specify the software trigger no-wait mode. ADSCM bit: Sequential conversion mode or one-shot conversion mode • ADM2 register ADRCK bit: This is used to select the range for the A/D conversion result comparison value for generating the interrupt signal from AREA1, AREA3, and AREA 2. ADTYP[1:0] bits: 12-bit, 10-bit, or 8-bit resolution • ADUL and ADLL register These are used to specify the upper limit and lower limit A/D conversion result comparison values. • ADS register ADS[4:0] bits: These are used to select the analog input channels.
	<2>	Supplied from the internal reference voltage?	• Supplied from an internal reference voltage – Setting ADM2 register: ADREFP[1:0] bits to 11b – Reference voltage discharge time: 1 µs wait • Supplied from other voltage source This step is through.
	<3>	Setting ADM2 register	• ADM2 register ADREFM bit: This is used to select the '-' side reference voltage source ADREFP[1:0] bits: These are used to select the '+' side reference voltage source. Before the supply setting of the internal reference voltage (ADREFP[1:0] = 10b), the reference voltage discharge time (1 µs) is required.
	<4>	Reference voltage stabilization wait time count A	The reference voltage stabilization wait time count indicated by A below may be required if the values of the ADREFP[1:0] bits are changed. If the values of ADREFP[1:0] are changed to 10b, respectively: A = 5 µs A wait is not required if the values of ADREFP[1:0] are changed to 00b or 01b, respectively.
	<5>	ADCE bit setting	The ADCE bit of the ADM0 register is set to 1, and the 12-bit A/D converter enters the conversion standby state.
	<6>	Reference voltage stabilization wait time count B	Use software to control waiting until reference voltage stabilization wait time count B (1 µs + 2 cycles of the conversion clock (fAD)) elapses.
	<7>	ADCS bit setting	After reference voltage stabilization wait time count B elapses, the ADCS bit of the ADM0 register is set to 1, and A/D conversion starts.
	<8>	Start of A/D conversion	—
	—	⋮	The A/D conversion operations are performed.
	<9>	End of A/D conversion	The A/D conversion end interrupt (ADC_ENDI) is generated.*1
	<10>	Storage of conversion results in the ADCRn or ADCRnH register	The conversion results are stored in the ADCRn or ADCRnH register.

Note 1. Depending on the settings of the ADRCK bit, ADUL and ADLL registers, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCRn or ADCRnH register.

29.7.2 Setting up Software Trigger Wait Mode

Table 29.15 shows the setup steps in software trigger wait mode.

Table 29.15 Setting up software trigger wait mode

Step		Process	Detail
Setting up Software Trigger Wait Mode	<1>	• ADM0 register setting • ADM1 register setting • ADM2 register setting • ADUL and ADLL register setting • ADS register setting (The order of the settings is irrelevant.)	• ADM0 register FR[2:0], LV[1:0]: bits: These are used to specify the A/D conversion time . ADMD bit: Select mode or scan mode • ADM1 register ADTMD[1:0] bits: These are used to specify the software trigger wait mode. ADSCM bit: Sequential conversion mode or one-shot conversion mode • ADM2 register ADRCK bit: This is used to select the range for the A/D conversion result comparison value for generating the interrupt signal from AREA1, AREA3, and AREA 2. ADTYP[1:0] bits: 12-bit, 10-bit, or 8-bit resolution • ADUL and ADLL register These are used to specify the upper limit and lower limit A/D conversion result comparison values. • ADS register ADS[4:0] bits: These are used to select the analog input channels.
	<2>	Supplied from the internal reference voltage?	• Supplied from an internal reference voltage – Setting ADM2 register: ADREFP[1:0] bits to 11b – Reference voltage discharge time: 1 µs wait • Supplied from other voltage source This step is through.
	<3>	Setting ADM2 register	• ADM2 register ADREFM bit: This is used to select the '-' side reference voltage source ADREFP[1:0] bits: These are used to select the '+' side reference voltage source. Before the supply setting of the internal reference voltage (ADREFP[1:0] = 10b), the reference voltage discharge time (1 µs) is required.
	<4>	Reference voltage stabilization wait time count A	The reference voltage stabilization wait time count indicated by A below may be required if the values of the ADREFP[1:0] bits are changed. If the values of ADREFP[1:0] are changed to 10b, respectively: A = 5 µs. A wait is not required if the values of ADREFP[1:0] are changed to 00b or 01b, respectively.
	<5>	ADCE bit setting	Do not set the ADCE bit of the ADM0 register. The 12-bit A/D converter must remain in the stopped state.
	<6>	ADCS bit setting	Do not set the ADCE bit of the ADM0 register. The 12-bit A/D converter enters the conversion standby state.
	<7>	Stabilization wait time for A/D power supply	The 12-bit A/D converter automatically counts up to the stabilization wait time for A/D power supply.
	<8>	Start of A/D conversion	After counting up to the stabilization wait time for A/D power supply ends, A/D conversion starts.
	—	⋮	The A/D conversion operations are performed.
	<9>	End of A/D conversion	The A/D conversion end interrupt (ADC_ENDI) is generated.*1
	<10>	Storage of conversion results in the ADCRn or ADCRnH register	The conversion results are stored in the ADCRn or ADCRnH register.

Note 1. Depending on the settings of the ADRCK bit, ADUL and ADLL registers, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCRn or ADCRnH register.

29.7.3 Setting up Hardware Trigger No-wait Mode

Table 29.16 shows the setup steps in hardware trigger no-wait mode.

Table 29.16 Setting up hardware trigger no-wait mode

Step		Process	Detail
Setting up Hardware Trigger No-Wait Mode	<1>	• ADM0 register setting • ADM1 register setting • ADM2 register setting • ADUL and ADLL register setting • ADS register setting (The order of the settings is irrelevant.)	• ADM0 register FR[2:0], LV[1:0]: bits: These are used to specify the A/D conversion time . ADMD bit: Select mode or scan mode • ADM1 register ADTMD[1:0] bits: These are used to specify the hardware trigger no-wait mode. ADSCM bit: Sequential conversion mode or one-shot conversion mode • ADM2 register ADRCK bit: This is used to select the range for the A/D conversion result comparison value for generating the interrupt signal from AREA1, AREA3, and AREA 2. ADTYP[1:0] bits: 12-bit, 10-bit, or 8-bit resolution • ADUL and ADLL register These are used to specify the upper limit and lower limit A/D conversion result comparison values. • ADS register ADS[4:0] bits: These are used to select the analog input channels.
	<2>	Supplied from the internal reference voltage?	• Supplied from an internal reference voltage – Setting ADM2 register: ADREFP[1:0] bits to 11b – Reference voltage discharge time: 1 µs wait • Supplied from other voltage source This step is through.
	<3>	Setting ADM2 register	• ADM2 register ADREFM bit: This is used to select the '-' side reference voltage source ADREFP[1:0] bits: These are used to select the '+' side reference voltage source. Before the supply setting of the internal reference voltage (ADREFP[1:0] = 10b), the reference voltage discharge time (1 µs) is required.
	<4>	Reference voltage stabilization wait time count A	The reference voltage stabilization wait time count indicated by A below may be required if the values of the ADREFP[1:0] bits are changed. If the values of ADREFP[1:0] are changed to 10b, respectively: A = 5 µs A wait is not required if the values of ADREFP[1:0] are changed to 00b or 01b, respectively.
	<5>	ADCE bit setting	The ADCE bit of the ADM0 register is set to 1, and the 12-bit A/D converter enters the conversion standby state.
	<6>	Reference voltage stabilization wait time count B	Use software to control waiting until reference voltage stabilization wait time count B (1 µs + 2 cycles of the conversion clock (f_{AD})) elapses.
	<7>	ADCS bit setting	After reference voltage stabilization wait time count B elapses, the ADCS bit of the ADM0 register is set to 1, and A/D converter enters the hardware trigger standby state.
	<8>	Start of A/D conversion	When a hardware trigger occurs, 12-bit conversion is started.
	—	⋮	The A/D conversion operations are performed.
	<9>	End of A/D conversion	The A/D conversion end interrupt (ADC_ENDI) is generated.*1
	<10>	Storage of conversion results in the ADCRn or ADCRnH register	The conversion results are stored in the ADCRn or ADCRnH register.

Note 1. Depending on the settings of the ADRCK bit, ADUL and ADLL registers, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCRn or ADCRnH register.

29.7.4 Setting up Hardware Trigger Wait Mode

Table 29.17 shows the setup steps in hardware trigger wait mode.

Table 29.17 Setting up hardware trigger wait mode

Step		Process	Detail
Setting up Hardware Trigger Wait Mode	<1>	• ADM0 register setting • ADM1 register setting • ADM2 register setting • ADUL and ADLL register setting • ADS register setting (The order of the settings is irrelevant.)	• ADM0 register FR[2:0], LV[1:0]: bits: These are used to specify the A/D conversion time . ADMD bit: Select mode or scan mode • ADM1 register ADTMD[1:0] bits: These are used to specify the hardware trigger wait mode. ADSCM bit: Sequential conversion mode or one-shot conversion mode • ADM2 register ADRCK bit: This is used to select the range for the A/D conversion result comparison value for generating the interrupt signal from AREA1, AREA3, and AREA 2. ADTYP[1:0] bits: 12-bit, 10-bit, or 8-bit resolution • ADUL and ADLL register These are used to specify the upper limit and lower limit A/D conversion result comparison values. • ADS register ADS[4:0] bits: These are used to select the analog input channels.
	<2>	Supplied from the internal reference voltage?	• Supplied from an internal reference voltage – Setting ADM2 register: ADREFP[1:0] bits to 11b – Reference voltage discharge time: 1 µs wait • Supplied from other voltage source This step is through.
	<3>	Setting ADM2 register	• ADM2 register ADREFM bit: This is used to select the '-' side reference voltage source ADREFP[1:0] bits: These are used to select the '+' side reference voltage source. Before the supply setting of the internal reference voltage (ADREFP[1:0] = 10b), the reference voltage discharge time (1 µs) is required.
	<4>	Reference voltage stabilization wait time count A	The reference voltage stabilization wait time count indicated by A below may be required if the values of the ADREFP[1:0] bits are changed. If the values of ADREFP[1:0] are changed to 10b, respectively: A = 5 µs A wait is not required if the values of ADREFP[1:0] are changed to 00b or 01b, respectively.
	<5>	ADCE bit setting	The ADCE bit of the ADM0 register is set to 1, and the 12-bit A/D converter enters the hardware trigger standby state.
	<6>	Hardware trigger generation	—
	<7>	Stabilization wait time for A/D power supply	The 12-bit A/D converter automatically counts up to the stabilization wait time for A/D power supply.
	<8>	Start of A/D conversion	After counting up to the stabilization wait time for A/D power supply ends, A/D conversion starts.
	—	⋮	The A/D conversion operations are performed.
	<9>	End of A/D conversion	The A/D conversion end interrupt (ADC_ENDI) is generated.*1
	<10>	Storage of conversion results in the ADCRn or ADCRnH register	The conversion results are stored in the ADCRn or ADCRnH register.

Note 1. Depending on the settings of the ADRCK bit, ADUL and ADLL registers, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCRn or ADCRnH register.

29.7.5 Example of Using the ADC when Selecting the Temperature Sensor Output Voltage or Internal Reference Voltage, and Software Trigger No-wait Mode and One-shot Conversion Mode

Table 29.18 shows the setup steps When Temperature Sensor Output Voltage and Internal Reference Voltage Is Selected.

Table 29.18 Setup when temperature sensor output voltage and internal reference voltage is selected

Step		Process	Detail
Setup when Temperature Sensor Output Voltage and Internal Reference Voltage Is Selected	<1>	• ADM0 register setting • ADM1 register setting • ADM2 register setting • ADUL and ADLL register setting • ADS register setting	• ADM0 register FR[2:0], LV[1:0]: bits: These are used to specify the A/D conversion time . ADMD bit: This is used to specify the select mode. • ADM1 register ADTMD[1:0] bits: These are used to specify the software trigger no-wait mode. ADSCM bit: One-shot conversion mode • ADM2 register ADREFP[1;0] and ADREFM bits: These are used to select the reference voltage. ADRCK bit: This is used to select the range for the A/D conversion result comparison value for generating the interrupt signal from AREA1, AREA3, and AREA 2. ADTYP[1:0] bits: 12-bit, 10-bit, or 8-bit resolution • ADUL and ADLL register These are used to specify the upper limit and lower limit A/D conversion result comparison values. • ADS register ADISS and ADS[4:0] bits: These are used to select the temperature sensor output voltage or internal reference voltage.
	<2>	Reference voltage stabilization wait time count A	The reference voltage stabilization wait time count A may be required if the values of the ADREFP[1:0] bits are changed. A wait is not required if the values of ADREFP[1:0] are changed to 00b or 01b, respectively. Setting the values of ADREFP[1:0] to 10b, respectively is prohibited.
	<3>	ADCE bit setting	The ADCE bit of the ADM0 register is set to 1, and the 12-bit A/D converter enters the conversion standby state.
	<4>	Reference voltage stabilization wait time count B	Use software to control waiting until reference voltage stabilization wait time count B (1 µs + 2 cycles of the conversion clock (fAD)) elapses.
	<5>	ADCS bit setting	After reference voltage stabilization wait time count B elapses, the ADCS bit of the ADM0 register is set to 1, and A/D conversion starts.
	<6>	Start of A/D conversion	—
	<7>	End of A/D conversion	The A/D conversion end interrupt (ADC_ENDI) will be generated. After ADISS is set to 1, the initial conversion result cannot be used.
	<8>	ADCS bit setting	The ADCS bit of the ADM0 register is set to 1, and A/D conversion starts.
	<9>	Start of A/D conversion	—
	<10>	End of A/D conversion	The A/D conversion end interrupt (ADC_ENDI) is generated.*1
	<11>	Storage of conversion results in the ADCRn or ADCRnH register	The conversion results are stored in the ADCRn or ADCRnH register.

Note 1. Depending on the settings of the ADRCK bit, ADUL and ADLL registers, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCRn or ADCRnH register.

29.7.6 Setting up Test Mode

Table 29.19 shows the setup steps in test mode.

Table 29.19 Setting up test mode

Step		Process	Detail
Setting up Test Mode	<1>	• ADM0 register setting • ADM1 register setting • ADM2 register setting • ADUL and ADLL register setting • ADS register setting • ADTES register setting (The order of the settings is irrelevant.)	• ADM0 register FR[2:0], LV[1:0]: bits: These are used to specify the A/D conversion time . ADMD bit: This is used to specify the select mode. • ADM1 register ADTMD[1:0] bits: These are used to specify the software trigger no-wait mode. ADSCM bit: This is used to specify the one-shot conversion mode. • ADM2 register ADRCK bit: This is used to select the range for the A/D conversion result comparison value for generating the interrupt signal to AREA 2. ADTYP[1:0] bits: 12-bit, 10-bit, or 8-bit resolution • ADUL and ADLL register These set ADUL to 0xFF and ADLL to 0x00 (initial values). • ADS register ADS[4:0] bits: These are used to set to ANI0. • ADTES register ADTES[1:0] bits: AVREFM or AVREFP.
	<2>	Supplied from the internal reference voltage?	• Supplied from an internal reference voltage – Setting ADM2 register: ADREFP[1:0] bits to 11b – Reference voltage discharge time: 1 µs wait • Supplied from other voltage source This step is through.
	<3>	Setting ADM2 register	• ADM2 register ADREFM bit: This is used to select the '-' side reference voltage source ADREFP[1:0] bits: These are used to select the '+' side reference voltage source. Before the supply setting of the internal reference voltage (ADREFP[1:0] = 10b), the reference voltage discharge time (1 µs) is required.
	<4>	Reference voltage stabilization wait time count A	The reference voltage stabilization wait time count indicated by A below may be required if the values of the ADREFP[1:0] bits are changed. If the values of ADREFP[1:0] are changed to 10b, respectively: A = 5 µs A wait is not required if the values of ADREFP[1:0] are changed to 00b or 01b, respectively.
	<5>	ADCE bit setting	The ADCE bit of the ADM0 register is set to 1, and the 12-bit A/D converter enters the conversion standby state.
	<6>	Reference voltage stabilization wait time count B	Use software to control waiting until reference voltage stabilization wait time count B (1 µs + 2 cycles of the conversion clock (f_{AD})) elapses.
	<7>	ADCS bit setting	After reference voltage stabilization wait time count B elapses, the ADCS bit of the ADM0 register is set to 1, and A/D conversion starts.
	<8>	Start of A/D conversion	—
	<9>	End of A/D conversion	The A/D conversion end interrupt (ADC_ENDI) is generated.*1
	<10>	Storage of conversion results in the ADCRn or ADCRnH register	The conversion results are stored in the ADCRn or ADCRnH register.

Note 1. Depending on the settings of the ADRCK bit, ADUL and ADLL registers, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCRn or ADCRnH register.

29.8 Snooze Mode Function

The snooze mode function allows A/D conversion to be performed without running the CPU. This is effective for reducing the operating current.

29.8.1 A/D Conversion by Inputting a Hardware Trigger

In Snooze mode, A/D conversion is triggered by inputting a hardware trigger.

When performing A/D conversion by inputting a hardware trigger in Snooze mode, only the following two conversion modes can be used:

- Hardware trigger wait mode (select mode, one-shot conversion mode)
- Hardware trigger wait mode (scan mode, one-shot conversion mode)

If the A/D conversion result range is specified using the ADUL and ADLL registers, A/D conversion results can be determined at a certain interval of time. Using this function enables power supply voltage monitoring and input key determination based on A/D inputs.

When using the Snooze mode function, the initial setting of each register is specified before switching to the Software Standby mode (for details about these settings, see Table 29.20). Just before moving to Software Standby mode, set bit 2 (AWC) of A/D converter mode register 2 (ADM2) to 1. After the initial settings are specified, set bit 0 (ADCE) of A/D converter mode register 0 (ADM0) to 1.

If a hardware trigger is input after switching to the Snooze mode, the 12-bit A/D converter automatically counts up to the A/D power supply stabilization wait time, and then A/D conversion starts.

The Snooze mode operation after A/D conversion ends differs depending on whether an interrupt signal is generated and the settings of the SELSR0 and SNZEDCR0 registers.*1

Note: 12-bit A/D converter can only be triggered by the ELC in Snooze mode.

Note 1. Depending on the setting of the A/D conversion result comparison function (ADRCK bit, ADUL and ADLL registers), there is a possibility of no interrupt signal being generated.

(1) If an interrupt is generated after A/D conversion ends

If the A/D conversion result value is inside the range of values specified by the A/D conversion result comparison function (which is set up by using the ADRCK bit, ADUL and ADLL registers), the A/D conversion end interrupt request signal (ADC_ENDI) is generated.

In select mode, an A/D conversion end interrupt request signal (ADC_ENDI) is generated when A/D conversion ends.

In scan mode, if even one value of the A/D conversion results of the four channels falls within the range specified by the A/D conversion result comparison function, and A/D conversion end interrupt request signal (ADC_ENDI) is generated.

- When an A/D conversion end interrupt is used as a snooze cancel factor
  The MCU returns to normal mode from Snooze mode.
  At this time, be sure to clear bit 2 (AWC = 0) of the 12-bit A/D converter mode register 2 (ADM2).
  If the AWC bit is left set to 1, A/D conversion does not start normally in the subsequent Snooze or normal operation mode.
- When A/D compare mismatch is used as a software standby transition factor
  Since no mismatch occurs, the MCU does not move to Software Standby mode.
  The state of the MCU after an A/D conversion end interrupt is generated depends on the subsequent processing.

Figure 29.24 shows an operation example when interrupt is generated after A/D conversion ends (while in scan mode).

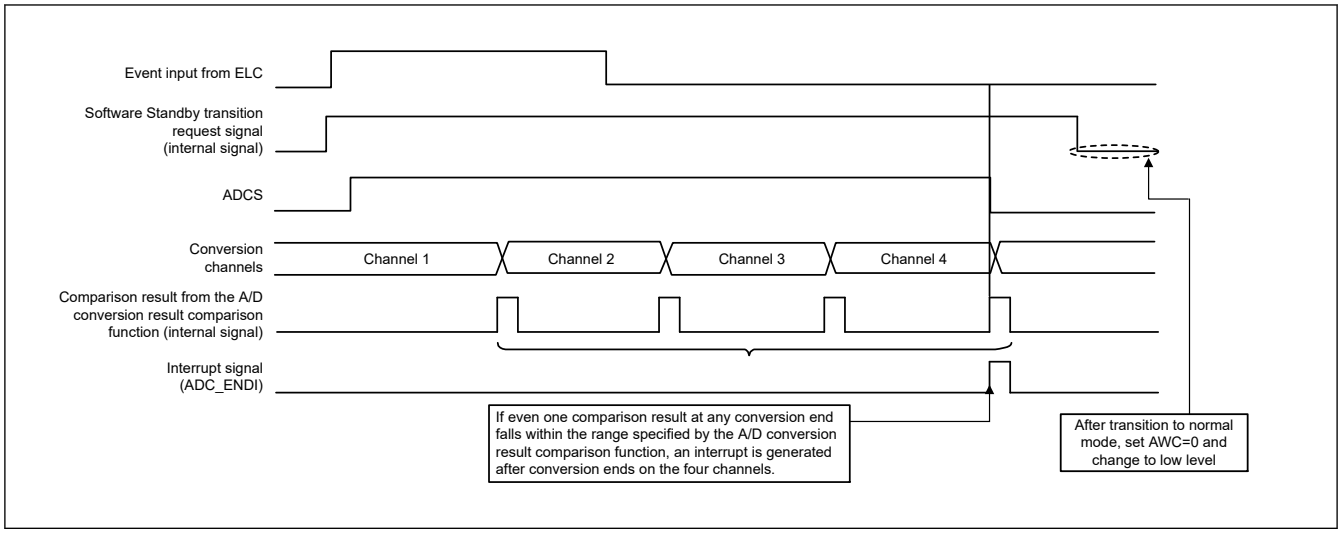


Figure 29.24 Operation example when an A/D conversion end interrupt is used as a snooze cancel factor (while in scan mode)

(2) If no interrupt is generated after A/D conversion ends

If the A/D conversion result value is outside the range of values specified by the A/D conversion result comparison function (which is set up by using the ADRCK bit, ADUL and ADLL registers), the A/D conversion end interrupt request signal (ADC_ENDI) is not generated.

- When an A/D conversion end interrupt is used as a snooze cancel factor
  Since an A/D conversion end interrupt is not generated, it remains in Snooze mode and waits for the next hardware trigger input.
- When A/D compare mismatch is used as a software standby transition factor
  The MCU moves to Software Standby mode. For details, see Table 10.8.

Figure 29.25 shows an operation example when no interrupt is generated after A/D conversion ends (while in scan mode).

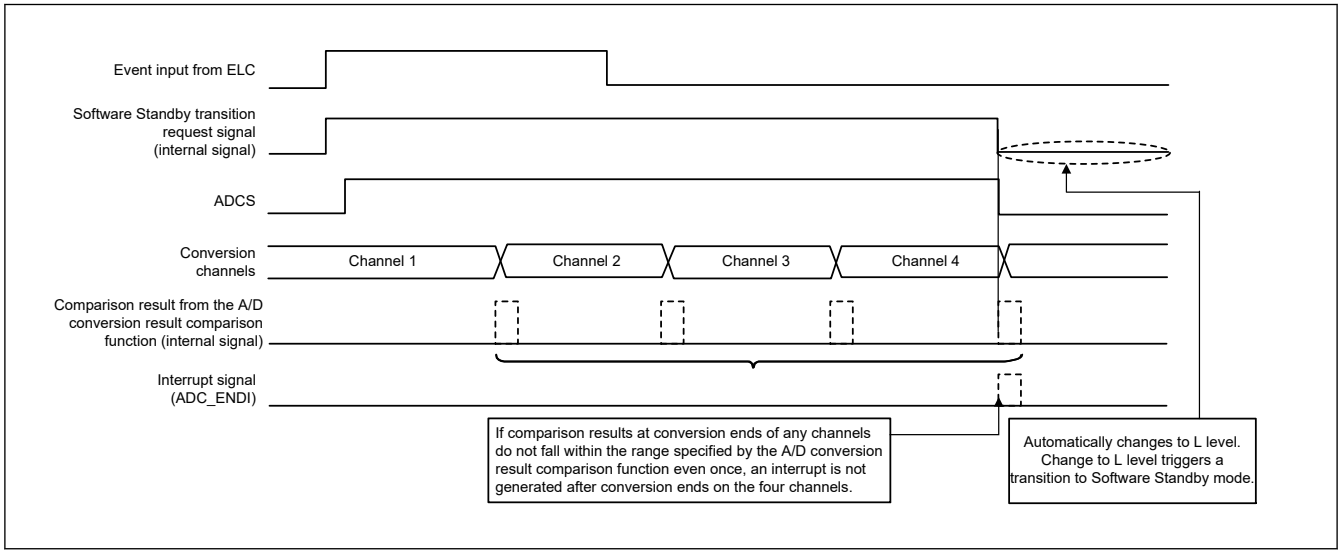


Figure 29.25 Operation example when A/D compare mismatch is used as a software standby transition factor (while in scan mode)

Table 29.20 shows procedure for setting up Snooze mode (hardware trigger)

Table 29.20 Procedure for setting up Snooze mode (hardware trigger)

Step		Process	Detail
Normal operation	<1>	• ADM0 register setting • ADM1 register setting • ADM2 register setting • ADUL and ADLL register setting • ADS register setting (The order of the settings is irrelevant.)	• ADM0 register FR[2:0], LV[1:0]: bits: These are used to specify the A/D conversion time. ADMD bit: Select mode or scan mode • ADM1 register ADTMD[1:0] bits: These are used to specify the hardware trigger wait mode. ADSCM bit: One-shot conversion mode • ADM2 register ADREFP[1:0] and ADREFM bits: These are used to select the reference voltage. ADRCK bit: This is used to select the range for the A/D conversion result comparison value for generating the interrupt signal from AREA1, AREA3, and AREA 2. ADTYP[1:0] bits: 12-bit, 10-bit, or 8-bit resolution • ADUL and ADLL register These are used to specify the upper limit and lower limit A/D conversion result comparison values. • ADS register ADS[4:0] bits: These are used to select the analog input channels.
	<2>	Reference voltage stabilization wait time count A	The reference voltage stabilization wait time count indicated by A below may be required if the values of the ADREFP[1:0] bits are changed. If the values of ADREFP[1:0] are changed to 10b, respectively: A = 5 µs Before changing as above, perform reference supply discharge (1 µs) by setting ADREFP[1:0] = 11b. A wait is not required if the values of ADREFP[1:0] are changed to 00b or 01b, respectively.
	<3>	AWC bit setting	Immediately before entering the Stop mode, enable the Snooze mode by setting the AWC bit of the ADM2 register to 1.
Software Standby mode	<4>	ADCE bit setting	The ADCE bit of the ADM0 register is set to 1, and the 12-bit A/D converter enters the conversion standby state.
	<5>	Enter the Software Standby mode	—
Snooze mode	<6>	Enter the Snooze mode	Snooze mode is entered at the snooze request of another module.
	<7>	Hardware trigger generation	After hardware trigger is generated, the 12-bit A/D converter automatically counts up to the stabilization wait time for A/D power supply and A/D conversion is started in the Snooze mode.
	—	⋮	The A/D conversion operations are performed.
	<8>	End of A/D conversion	—
	<9>	Generate ADC_ENDI?	• ADC_ENDI is generated : Go to step <10> • ADC_ENDI is not generated : Go to step <7>*1.
Normal operation	<10>	Storage of conversion results in the ADCRn or ADCRnH register	The conversion results are stored in the ADCRn or ADCRnH register.
	<11>	AWC bit setting	Clearing the AWC bit of the ADM2 register to 0*2.
	<12>	Normal operation ⋮	—

Note 1. If the A/D conversion end interrupt request signal (ADC_ENDI) is not generated depending on the settings of the ADRCK bit, ADUL and ADLL registers, the result is not stored in the ADCRn or ADCRnH register. The MCU remains in Snooze mode and waits for the next hardware trigger input. If a hardware trigger is input later, A/D conversion operation is again performed in the Snooze mode.

Note 2. If the AWC bit is left set to 1, A/D conversion will not start normally in the subsequent Snooze or normal operation mode. Be sure to clear the AWC bit to 0.

29.9 How to Read the 12-bit A/D Converter Characteristics Table

Here, special terms unique to the 12-bit A/D converter are explained.

(1) Resolution

This is the minimum analog input voltage that can be identified. That is, the percentage of the analog input voltage per bit of digital output is called 1LSB (Least Significant Bit). The percentage of 1LSB with respect to the full scale is expressed by %FSR (Full Scale Range).

1LSB is as follows when the resolution is 12 bits.

$$1\ \text{LSB} = 1/2^{12} = 1/4096$$
$$\approx 0.024\ \%\text{FSR}$$

Accuracy has no relation to resolution, but is determined by absolute accuracy.

(2) Absolute accuracy

This shows the maximum error value between the actual measured value and the theoretical value.

Zero-scale error, full-scale error, integral linearity error, differential linearity errors, and combinations of these express the absolute accuracy.

Note that the quantization error is not included in the absolute accuracy in the characteristics table.

(3) Quantization error

When analog values are converted to digital values, a ±1/2LSB error naturally occurs. In a 12-bit converter, an analog input voltage in a range of ±1/2LSB is converted to the same digital code, so a quantization error cannot be avoided.

Note that the quantization error is not included in the absolute accuracy, zero-scale error, full-scale error, integral linearity error, and differential linearity error in the characteristics table.

Figure 29.26 shows the absolute accuracy, and Figure 29.27 shows the quantization error.

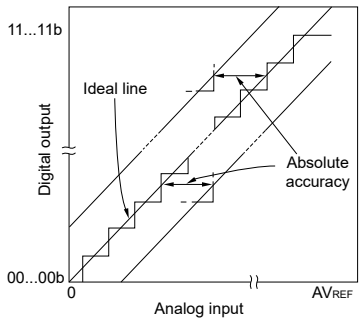


Figure 29.26 Absolute accuracy

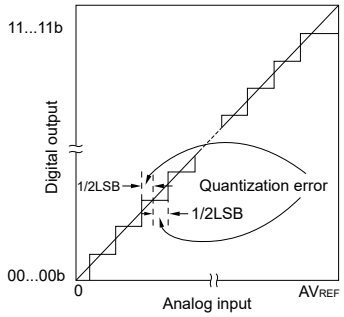


Figure 29.27 Quantization error

(4) Zero-scale error

This shows the difference between the actual measurement value of the analog input voltage and the theoretical value (1/2LSB) when the digital output changes from 0……000b to 0……001b.

If the actual measurement value is greater than the theoretical value, it shows the difference between the actual measurement value of the analog input voltage and the theoretical value (3/2LSB) when the digital output changes from 0……001b to 0……010b.

(5) Full-scale error

This shows the difference between the actual measurement value of the analog input voltage and the theoretical value (Full-scale – 3/2LSB) when the digital output changes from 1……110b to 1……111b.

(6) Integral linearity error

This shows the degree to which the conversion characteristics deviate from the ideal linear relationship. It expresses the maximum value of the difference between the actual measurement value and the ideal straight line when the zero-scale error and full-scale error are 0.

(7) Differential linearity error

While the ideal width of code output is 1LSB, this indicates the difference between the actual measurement value and the ideal value of the width of output code.

The zero-scale error is shown in Figure 29.28, the full-scale error is shown in Figure 29.29, the integral linearity error is shown in Figure 29.30, and the differential linearity error is shown in Figure 29.31.

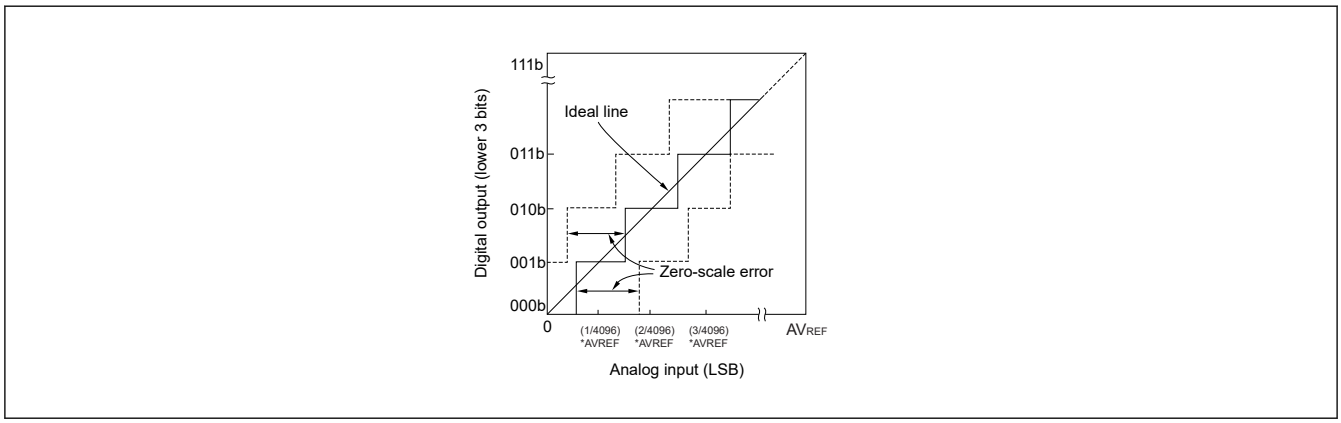


Figure 29.28 Zero-scale error

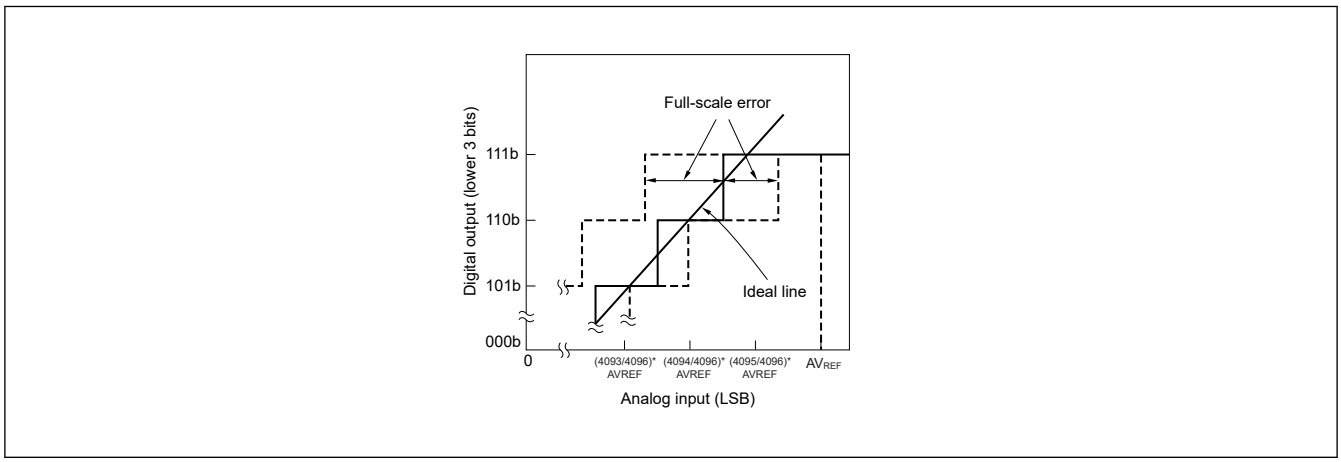


Figure 29.29 Full-scale error

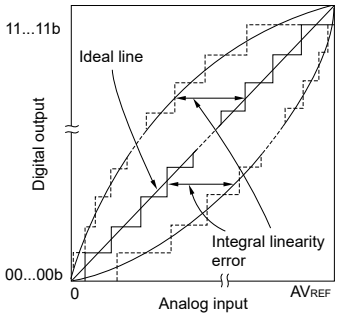


Figure 29.30 Integral linearity error

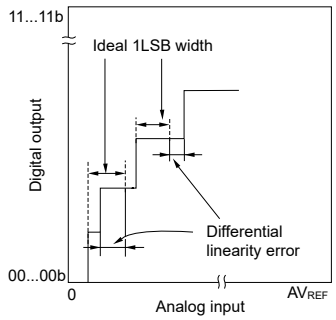


Figure 29.31 Differential linearity error

(8) Conversion time

This expresses the time from the start of sampling to when the digital output is obtained. The sampling time is included in the conversion time in the characteristics table.

(9) Sampling time

This is the time the analog switch is turned on for the analog voltage to be sampled by the sample & hold circuit.

Figure 29.32 shows the sampling time at the A/D conversion time.

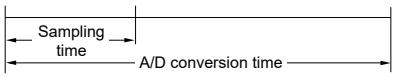


Figure 29.32 Sampling time at the A/D conversion time

29.10 Points for Caution when the 12-bit A/D Converter is to be Used

(1) Operating current in Software Standby mode

Shift to Software Standby mode after stopping the 12-bit A/D converter (by setting bit 7 (ADCS) of A/D converter mode register 0 (ADM0) to 0). The operating current can be reduced by setting bit 0 (ADCE) of the ADM0 register to 0 at the same time.

To restart from the software standby state, clear the IR bit in the corresponding IELSRn register and start operation.

(2) Input range of ANI0 to ANI5 and ANI16 to ANI19 pins

Observe the rated range of the ANI0 to ANI5 and ANI16 to ANI19 pins input voltage. If a voltage exceeding VCC and AVREFP or a voltage lower than VSS and AVREFM (even in the range of absolute maximum ratings) is input to an

analog input channel, the converted value of that channel becomes undefined. In addition, the converted values of the other channels may also be affected.

When internal reference voltage is selected as the reference voltage for the '+' side of the 12-bit A/D converter, do not input a voltage equal to or higher than the internal reference voltage to a pin selected by the ADS register. However, it is no problem that a voltage equal to or higher than the internal reference voltage is input to a pin not selected by the ADS register.

Note: For details about the internal reference voltage, see section 37, Electrical Characteristics.

(3) Conflicting operations

<1> Conflict between the conversion result being stored in the A/D conversion result register (ADCRn or ADCRnH) at the end of conversion and the read access to the ADCRn or ADCRnH register by instruction.

The ADCRn or ADCRnH register read has priority. After the read operation, the new conversion result is written to the ADCRn or ADCRnH registers.

<2> Conflict between the conversion result being stored in the A/D conversion result register (ADCRn or ADCRnH) at the end of conversion and the write access to the A/D converter mode register 0 (ADM0) or analog input channel specification register (ADS) by instruction.

The ADM0 and ADS registers write have priority. The ADCRn and ADCRnH registers write is not performed, nor is the conversion end interrupt signal (ADC_ENDI) generated.

(4) Noise countermeasures

To maintain the 12-bit or 10-bit resolution, attention must be paid to noise input to the AVREFP, VCC, ANI0 to ANI5, and ANI16 to ANI19 pins.

<1> Connect a capacitor with a low equivalent resistance and a good frequency response (capacitance of about 0.1 µF) via the shortest possible run of relatively thick wiring to the V_{CC} and AV_{REFP} pins.

<2> The higher the output impedance of the analog input source, the greater the influence. To reduce the noise, connecting an external capacitor as shown in Figure 29.33 is recommended.

<3> Do not switch these pins with other pins during conversion.

Figure 29.33 shows connections of V_{CC}, AV_{REFP}, and analog input pins.

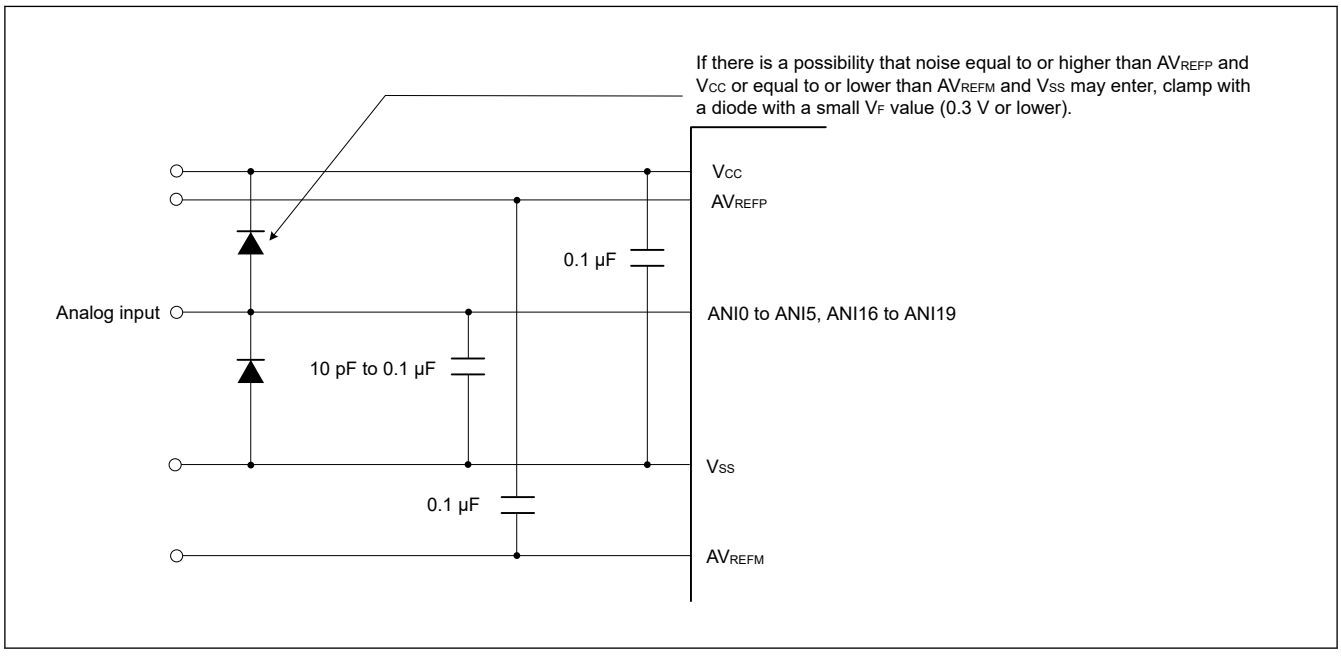


Figure 29.33 Connections of V_{CC}, AV_{REFP}, and analog input pins

(5) Analog input (ANIxx) pins

<1> The analog input pins (ANI0 to ANI5 and ANI16 to ANI19) are also used as input port pins (P000 to P003, P006, P007, P105, P106, P400, P401). When 12-bit A/D conversion is performed with any of the ANI0 to ANI5 and ANI16 to ANI19 pins selected, do not change to output value P000 to P003, P006, P007, P105, P106, P400, P401 while conversion is in progress; otherwise the conversion resolution may be degraded.

<2> If a pin adjacent to a pin that is being A/D converted is used as a digital I/O port pin, the A/D conversion result might differ from the expected value due to a coupling noise. Be sure to avoid the input or output of digital signals and signals with similarly sharp transitions during conversion.

(6) Input impedance of analog input (ANIxx) pins

This A/D converter charges a sampling capacitor for sampling during sampling time.

Therefore, only a leakage current flows when sampling is not in progress, and a current that charges the capacitor flows during sampling. Consequently, the input impedance fluctuates depending on whether sampling is in progress.

To make sure that sampling is effective, however, we recommend using the converter with analog input sources that have output impedances no greater than 1 kΩ. If a source has a higher output impedance, lengthen the sampling time or connect a larger capacitor (with a value of about 0.1 µF) to the pin from among ANI0 to ANI5 and ANI16 to ANI19 to which the source is connected (see Figure 29.33). The sampling capacitor may be being charged while the setting of the ADCS bit is 0 and immediately after sampling is restarted and so is not defined at these times. Accordingly, the state of conversion is undefined after charging starts in the next round of conversion after the value of the ADCS bit has been 1 or when conversion is repeated. Thus, to secure full charging regardless of the size of fluctuations in the analog signal, ensure that the output impedances of the sources of analog inputs are low or secure sufficient time for the completion of sampling.

(7) Interrupt status flag (IR)

The interrupt status flag (IR) is not cleared even if the analog input channel specification register (ADS) is changed.

Therefore, if an analog input pin is changed during A/D conversion, the A/D conversion result and IR flag for the pre-change analog input may have been set before the ADS register is rewritten. When reading the IR flag immediately after rewriting to the ADS register, note that the IR flag is set although A/D conversion for the post- change analog input has not ended.

When A/D conversion is stopped and then resumed, clear IR flag before the A/D conversion operation is resumed.

Figure 29.34 shows timing of A/D conversion end interrupt request generation.

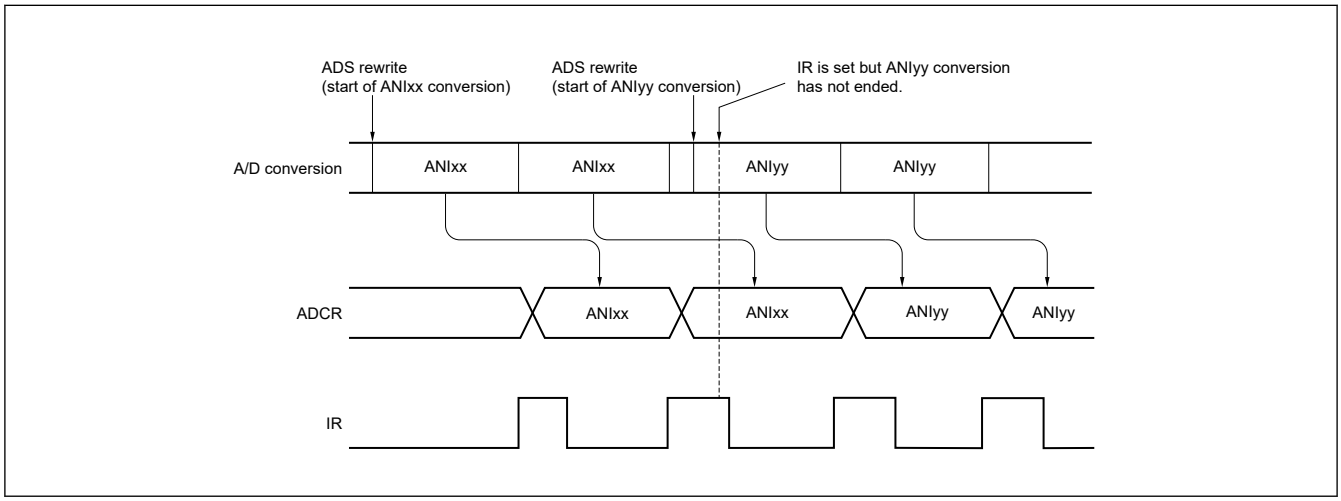


Figure 29.34 Timing of A/D conversion end interrupt request generation

(8) Conversion results just after A/D conversion start

While in the software trigger no-wait mode or hardware trigger no-wait mode, the first A/D conversion value immediately after A/D conversion starts may not fall within the rating range if the ADCS bit is set to 1 within 1 µs + 2 cycles of the conversion clock (f_{AD}) after the ADCE bit was set to 1. Take measures such as polling the A/D conversion end interrupt request signal (ADC_ENDI) and removing the first conversion result.

(9) A/D conversion result register (ADCRn, ADCRnH) read operation

When a write operation is performed to A/D converter mode register 0 (ADM0), analog input channel specification register (ADS), Port Control Register 0 to 4, or Port mn Pin Function Select Register, the contents of the ADCRn and ADCRnH registers may become undefined. After the completion of conversion, read the conversion result before writing to the ADM0, ADS, Port Control Register 0 to 4, or Port mn Pin Function Select Register, otherwise, an incorrect conversion result may be read.

(10) Internal equivalent circuit

Figure 29.35 shows the equivalent circuit of the analog input block, and Table 29.21 shows the equivalent circuit constants for each voltage.

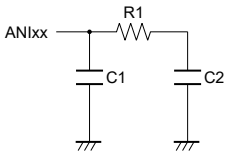


Figure 29.35 Internal equivalent circuit of ANIxx pin

Table 29.21 Resistance and capacitance values of equivalent circuit (reference values)

AV_{REFP}, V_{CC}	ANIxx pin	R1 [kΩ]	C1 [pF]	C2 [pF]
2.4 V ≤ VCC ≤ 5.5 V	ANI0 to ANI5	11	8	9
	ANI16 to ANI19	12	8	10
1.8 V ≤ VCC < 2.4 V	ANI0 to ANI5	55	8	9
	ANI16 to ANI19	60	8	10
1.6 V ≤ VCC < 1.8 V	ANI0 to ANI5	110	8	9
	ANI16 to ANI19	120	8	10

Note: The resistance and capacitance values shown in Table 29.21 are not guaranteed values.

(11) Starting the 12-bit A/D converter

Start the 12-bit A/D converter after the AV_{REFP} and V_{CC} voltages stabilize.

(12) Note on when power is applied or power off

When AVREFP is used as a reference voltage, AVREFP ≤ VCC should be maintained when power is applied or when power is off. If this condition is not maintained, it may lead to permanent destruction of the device.

30. 8-Bit D/A Converter (DAC8)

30.1 Overview

The MCU provides a 8-bit D/A Converter (DAC8). The D/A converters can output to the analog output pin but are mainly used for generating reference input voltage of Comparator (CMP). Table 30.1 lists the DAC8 specifications, Figure 30.1 shows a block diagram, and Table 30.2 lists the I/O pins.

Table 30.1 DAC8 specifications

Item	Descriptions
Resolution	8 bits
Output channels	Two channels
Module-stop function	Module-stop state can be set to reduce power consumption
Event link function (input)	The DA0 and DA1 conversion can be started on input of event signal

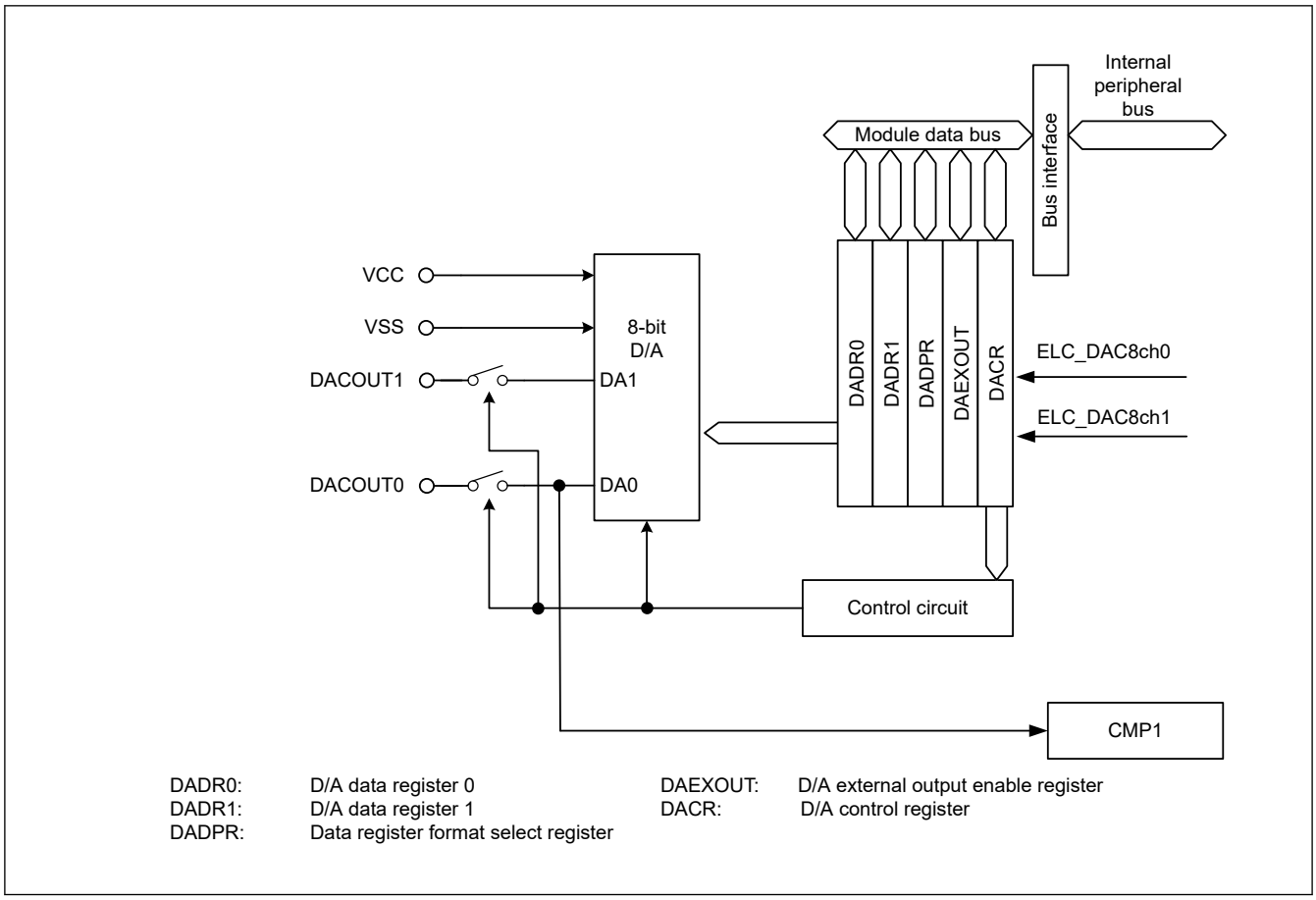


Figure 30.1 DAC8 block diagram

Table 30.2 lists the pin configuration of the DAC8.

Table 30.2 DAC8 I/O pins

Pin name	I/O	Function
VCC	Input	Analog block power supply pin
VSS	Input	Analog block power supply ground pin
DACOUT0	Output	Channel 0 output pin for the analog signals processed by the DAC8
DACOUT1	Output	Channel 1 output pin for the analog signals processed by the DAC8

30.2 Register Descriptions

30.2.1 DADRn : D/A Data Register n (n = 0, 1)

Base address: DAC8 = 0x4005_E000

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.

8-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 8 bits, [7:0], are valid. In left-justified format (DADPR.DPSEL = 1), the upper 8 bits, [15:8], are valid.

30.2.2 DACR : D/A Control Register

Base address: DAC8 = 0x4005_E000

Offset address: 0x04

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DAOE1	DAOE0	—	—	—	—	—	—
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	—	This bit is read as 0. The write value should be 0.	R/W
6	DAOE0	D/A Output Enable 0 0: Analog output of channel 0 (DA0) is disabled. 1: D/A conversion of channel 0 is enabled. Analog output of channel 0 (DA0) is enabled.	R/W
7	DAOE1	D/A Output Enable 1 0: Analog output of channel 1 (DA1) is disabled. 1: D/A conversion of channel 1 is enabled. Analog output of channel 1 (DA1) is enabled.	R/W

DAOE0 bit (D/A Output Enable 0)

The DAOE0 bit controls D/A conversion, and analog output.

DAOE1 bit (D/A Output Enable 1)

The DAOE1 bit controls D/A conversion, and analog output.

30.2.3 DADPR : DADRn Format Select Register

Base address: DAC8 = 0x4005_E000

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

30.2.4 DAEXOUT : D/A External Output Enable Register

Base address: DAC8 = 0x4005_E000

Offset address: 0x700

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	DAEXO1	DAEXO0	—	—	—	—	—	—
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	—	These bits are read as 0. The write value should be 0.	R/W
6	DAEXO0	D/A External Pin Output Enable 0 0: DA0 output to DACOUT0 pin is disabled. 1: DA0 output to DACOUT0 pin is enabled.	R/W
7	DAEXO1	D/A External Pin Output Enable 1 0: DA1 output to DACOUT1 pin is disabled. 1: DA1 output to DACOUT1 pin is enabled.	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W

DAEXO0 bit (D/A External Pin Output Enable 0)

The DAEXO0 bit controls the D/A conversion output (DA0) to external pin DACOUT0.

DAEXO1 bit (D/A External Pin Output Enable 1)

The DAEXO1 bit controls the D/A conversion output (DA1) to external pin DACOUT1.

30.3 Operation

The DAC8 includes D/A conversion circuits for two channels, each of which can operate independently. When the DAOEn bit (n = 0, 1) in DACR is set to 1, DAC8 is enabled and the conversion result is output.

This following example shows D/A conversion on channel 0. Figure 30.2 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:

$$\frac{\text{Setting in DADRm}}{256} \times \text{VCC}$$

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.

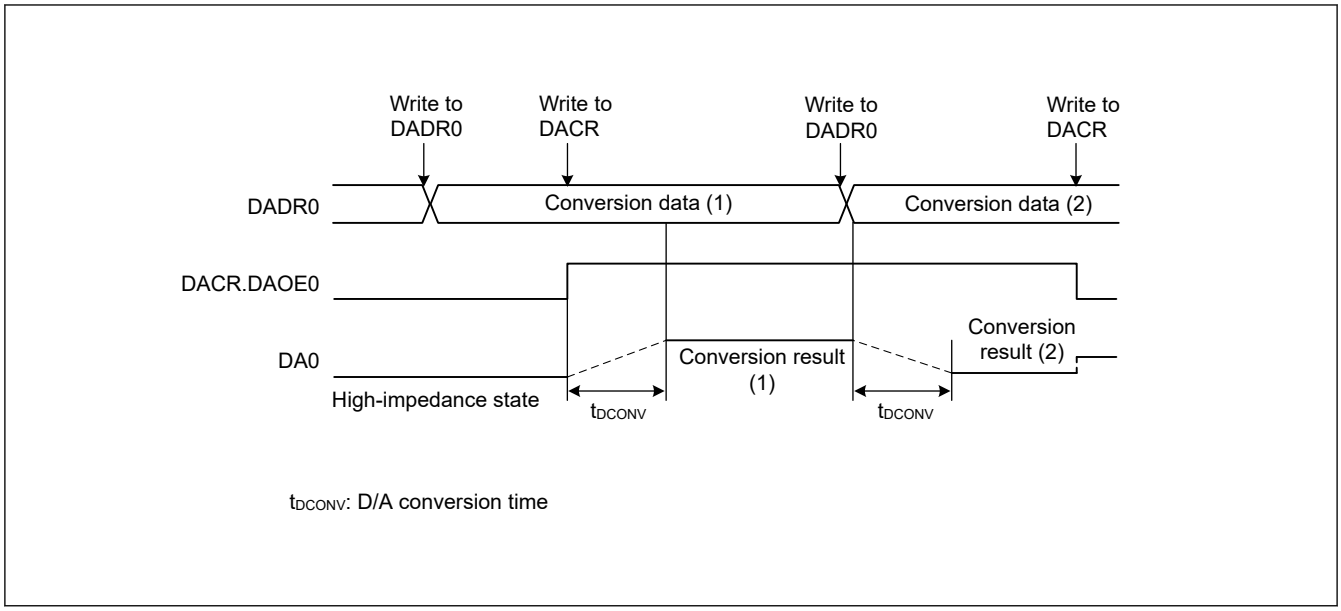


Figure 30.2 Example of DAC8 operation

30.4 Event Link Operation Setting Procedure

This section describes the procedures used in event link operation.

30.4.1 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_DAC8ch0 event signal to be linked to each peripheral module in the ELSR20 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 ELSR20 register to 0x0000 to stop event link operation of DAC8 channel 0. All event link operation is stopped when the ELCR.ELCON bit is set to 0.

30.4.2 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_DAC8ch1 event signal to be linked to each peripheral module in the ELSR19 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 1.
5. Set the ELSR19 register to 0x0000 to stop event link operation on DAC8 channel 1. All event link operation is stopped when the ELCR.ELCON bit is set to 0.

30.4.3 Usage Notes on Event Link Operation

- When the event specified for the ELC_DAC8ch0 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_DAC8ch1 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.

30.5 Usage Notes

30.5.1 Module Stop Function Setting

Operation of the 8-bit D/A converter can be disabled or enabled using the module stop control register. The initial setting is for operation of the 8-bit D/A converter to be stopped. Register access is enabled by releasing the module stop state. For details, see section 10, Low Power Modes.

30.5.2 Operation of the D/A Converter in 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 and DAOE0 bits to 0.

30.5.3 Operation of the D/A Converter in Software Standby Mode

When the MCU enters software standby mode with D/A conversion enabled, the D/A converter 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, and DAOE0 bits to 0.

30.5.4 Setting the D/A Converter

When using the D/A converter output voltage as a reference input voltage of the Comparator (CMP), set the D/A converter and wait for the D/A converter output settling time (t_{DCONV}) before enabling the comparator. Similarly, before making any changes to the settings of the D/A converter stop the comparator temporarily, and after the changes are made, wait for the D/A converter output settling time before enabling the comparator.

30.5.5 D/A Converter output

Currents on DACOUT0 and DACOUT1 pins cannot be obtained because the output impedance of the D/A converter is high. If the load input impedance is low, insert a follower amplifier between the load and DACOUT0 , DACOUT1 pins. Also make the wiring to the follower amplifier and the load as short as possible because the output impedance is high. If the wiring becomes long, consider shielding the wiring with the ground pattern. In case of using D/A converter as reference of Comparator, do not use DACOUT0 and DACOUT1 pins, because the noise from these pins affects the function of Comparator.

31. Comparator (CMP)

31.1 Functions of Comparator

The comparator has the following functions:

- Comparator high-speed mode or comparator low-speed mode
- External reference voltage input and internal reference voltage[*1] or D/A converter output[*1] are selectable as the reference voltage
- Canceling width of the noise canceling digital filter
- An interrupt signal that can be generated by detecting an active edge of the comparator output
- An event signal that can be output to the logic and event link controller (ELC) by detecting an active edge of the comparator output.

Note: Comparator 0 is not available in R9A02G0214CBY.

Note 1. The internal reference voltage and D/A converter output 0 are selectable for comparators 0 and 1, respectively.

31.2 Configuration of Comparator

Figure 31.1 shows a block diagram of the Comparator.

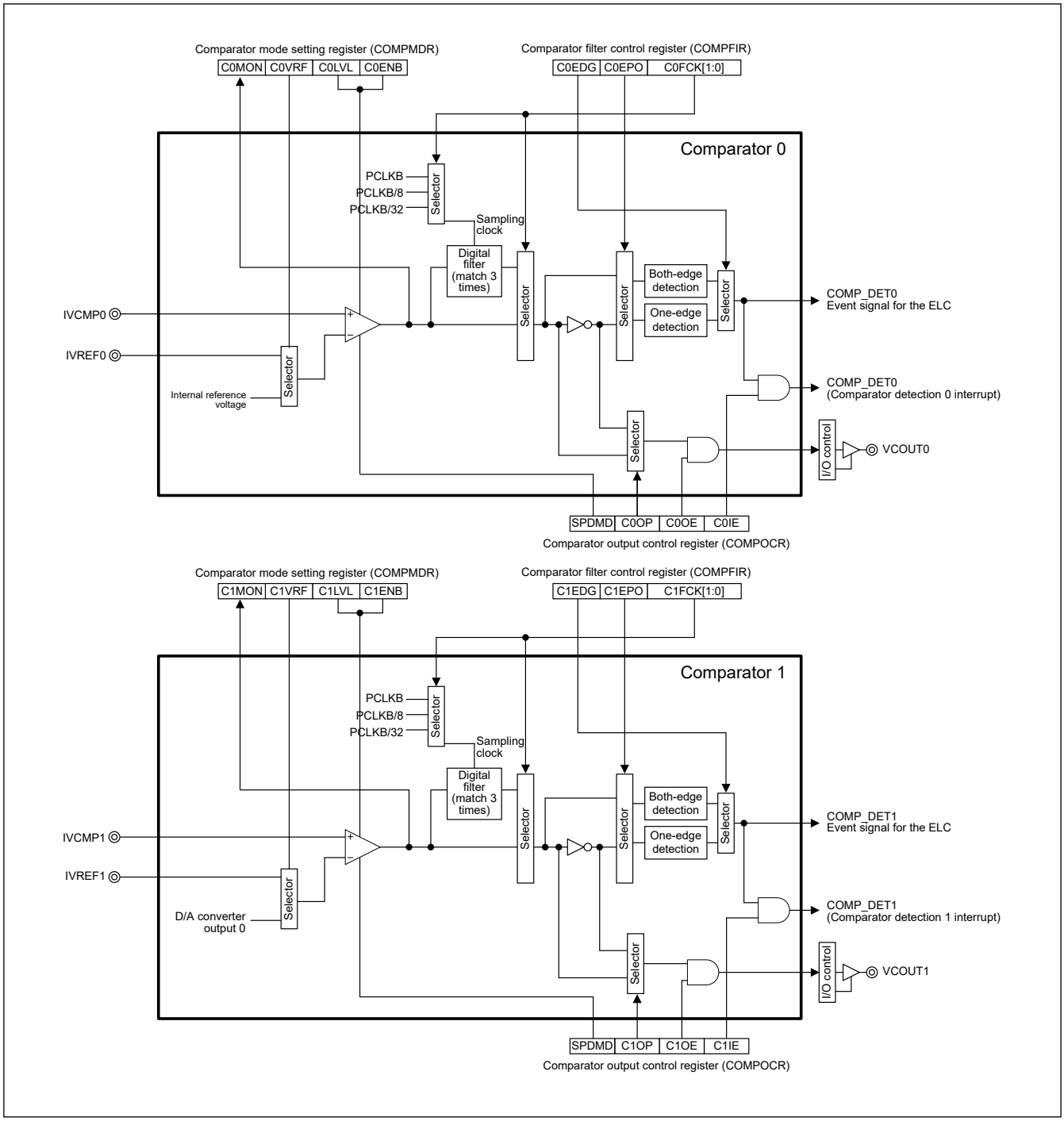


Figure 31.1 Comparator block diagram

31.3 Registers to Control the Comparator

The following registers are used to control the comparator.

- COMPMDR : Comparator mode setting register
- COMPFIR : Comparator filter control register
- COMPOCR : Comparator output control register

31.3.1 COMPMDR : Comparator Mode Setting Register

Base address: CMP = 0x4009_1200

Offset address: 0x00

Bit position:	7	6	5	4	3	2	1	0
Bit field:	C1MON	C1VRF*4	C1LVL	C1ENB	C0MON	C0VRF	C0LVL	C0ENB
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	C0ENB*1 *3	Comparator 0 operation enable 0: Comparator 0 operation disabled 1: Comparator 0 operation enabled	R/W
1	C0LVL*2 *3	Selection of comparator 0 reference voltage range 0: 0 to VCC - 1.4 V 1: 1.4 V to VCC	R/W
2	C0VRF*2 *3	Selection of comparator 0 reference voltage 0: Supply through the IVREF0 pin 1: Supply through the internal reference voltage	R/W
3	C0MON*3	Comparator 0 monitor flag 0: IVCMP0 < comparator 0 reference voltage (IVREF0 or internal reference voltage) 1: IVCMP0 > comparator 0 reference voltage (IVREF0 or internal reference voltage)	R
4	C1ENB	Comparator 1 operation enable 0: Comparator 1 operation disabled 1: Comparator 1 operation enabled	R/W
5	C1LVL	Selection of comparator 1 reference voltage range 0: 0 to VCC - 1.4 V 1: 1.4 V to VCC	R/W
6	C1VRF*4	Selection of comparator 1 reference voltage 0: Supply through the IVREF1 pin 1: Supply through D/A converter output 0	R/W
7	C1MON	Comparator 1 monitor flag 0: IVCMP1 < comparator 1 reference voltage (IVREF1 or D/A converter output 0) 1: IVCMP1 > comparator 1 reference voltage (IVREF1 or D/A converter output 0)	R

Note 1. The initial value is 0 immediately after a reset is released. However, the value is undefined when C0ENB is set to 0 and C1ENB is set to 0 after operation of the comparator is enabled once.
Note 2. When the internal reference voltage (C0VRF = 1) is to be used as the comparator 0 reference voltage, C0LVL = 0 and VCC ≥ (internal reference voltage (max.) + 1.4 V) must be satisfied.
Note 3. Comparator 0 is not available in R9A02G0214CBY. The write value should be 0.
Note 4. IVREF1 pin is not available in R9A02G0214CBY. The reference voltage should be set to DAC.

The COMPMDR register is used to make various settings of comparators 0 and 1, such as selection of the reference voltage, starting or stopping comparing operation, and also indicates the results of comparison.

The COMPMDR register can be set by a 1-bit or 8-bit memory manipulation instruction. Note that the CnMON bit is read-only.

The value of this register is 0x00 following a reset.

31.3.2 COMPFIR : Comparator Filter Control Register

Base address: CMP = 0x4009_1200

Offset address: 0x01

Bit position:	7	6	5	4	3	2	1	0
Bit field:	C1EDG	C1EPO	C1FCK[1:0]		C0EDG	C0EPO	C0FCK[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	C0FCK[1:0]*2 *3 *4	Comparator 0 digital filter selection 0 0: No comparator 0 filter 0 1: Comparator 0 filter enabled, sampling at PCLKB 1 0: Comparator 0 filter enabled, sampling at PCLKB/8 1 1: Comparator 0 filter enabled, sampling at PCLKB/32	R/W
2	C0EPO*2 *4	Comparator 0 edge polarity switching 0: Interrupt request at comparator 0 rising edge 1: Interrupt request at comparator 0 falling edge	R/W
3	C0EDG*2 *4	Comparator 0 edge detection selection 0: Interrupt request by comparator 0 one-edge detection 1: Interrupt request by comparator 0 both-edge detection	R/W
5:4	C1FCK[1:0]*1 *3	Comparator 1 digital filter selection 0 0: No comparator 1 filter 0 1: Comparator 1 filter enabled, sampling at PCLKB 1 0: Comparator 1 filter enabled, sampling at PCLKB/8 1 1: Comparator 1 filter enabled, sampling at PCLKB/32	R/W
6	C1EPO*1	Comparator 1 edge polarity switching 0: Interrupt request at comparator 1 rising edge 1: Interrupt request at comparator 1 falling edge	R/W
7	C1EDG*1	Comparator 1 edge detection selection 0: Interrupt request by comparator 1 one-edge detection 1: Interrupt request by comparator 1 both-edge detection	R/W

Note 1. Changing the C1FCK[1:0] bits field or the C1EPO or C1EDG bit may lead to the generation of an event signal for the ELC and a comparator 1 interrupt request. Only change these bits while the output from comparator 1 is not selected as an input signal to the ELC and the comparator 1 interrupt is masked. After that, to use the comparator 1 interrupt, clear the IR flag in the IELSRn register to clear the interrupt status.
If the C1FCK[1:0] bits are changed from 00b (no comparator 1 filter) to a value other than 00b (comparator 1 filter enabled), allow the time for sampling four times to elapse until the filter output is updated, and then use the comparator 1 interrupt request or the event signal for the ELC.

Note 2. Changing the C0FCK[1:0] bits field or the C0EPO or C0EDG bit may lead to the generation of an event signal for the ELC and a comparator 0 interrupt request. Only change these bits while the output from comparator 0 is not selected as an input signal to the ELC and the comparator 0 interrupt is masked. After that, to use the comparator 0 interrupt, clear the IR flag in the IELSRn register to clear the interrupt status.
If the C0FCK[1:0] bits are changed from 00b (no comparator 0 filter) to a value other than 00b (comparator 0 filter enabled), allow the time for sampling four times to elapse until the filter output is updated, and then use the comparator 0 interrupt request or the event signal for the ELC.

Note 3. When using the comparators in Software Standby mode, make the settings for non-use of the filters (C0FCK[1:0] = 00b, C1FCK[1:0] = 00b).

Note 4. Comparator 0 is not available in R9A02G0214CBY. The write value should be 0.

The COMPFIR register is used to select the valid edge for use with the comparator interrupt signals and whether or not to use the digital filter. If rejecting noise is required, set the CnFCK[1:0] bits to enable the digital filters. While a digital filter is enabled, the filter conveys the input level after detecting matches for three cycles of the sampling clock for use with the filter.

The COMPFIR register can be set by a 1-bit or 8-bit memory manipulation instruction.

The value of this register is 0x00 following a reset.

31.3.3 COMPOCR : Comparator Output Control Register

Base address: CMP = 0x4009_1200

Offset address: 0x02

Bit position:	7	6	5	4	3	2	1	0
Bit field:	SPDMD	C1OP	C1OE	C1IE	—	C0OP	C0OE	C0IE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	C0IE*3 *4	Comparator 0 interrupt request enable 0: Comparator 0 interrupt request disabled 1: Comparator 0 interrupt request enabled	R/W
1	C0OE*4	VCOUT0 pin output enable 0: Comparator 0 VCOUT0 pin output disabled 1: Comparator 0 VCOUT0 pin output enabled	R/W
2	C0OP*4	VCOUT0 output polarity selection 0: Comparator 0 output is output to VCOUT0 1: Inverted comparator 0 output is output to VCOUT0	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	C1IE*2	Comparator 1 interrupt request enable 0: Comparator 1 interrupt request disabled 1: Comparator 1 interrupt request enabled	R/W
5	C1OE	VCOUT1 pin output enable 0: Comparator 1 VCOUT1 pin output disabled 1: Comparator 1 VCOUT1 pin output enabled	R/W
6	C1OP	VCOUT1 output polarity selection 0: Comparator 1 output is output to VCOUT1 1: Inverted comparator 1 output is output to VCOUT1	R/W
7	SPDMD*1	Comparator speed selection 0: Comparator low-speed mode 1: Comparator high-speed mode	R/W

Note 1. When rewriting the SPDMD bit, be sure to set the CiENB bit (i = 0, 1) in the COMPMDR register to 0 in advance.
Note 2. If the C1IE bit is changed from 0 (interrupt request disabled) to 1 (interrupt request enabled), the comparator interrupt request might be generated. Before changing these bits, set the ELSRn register to 0 (the CMP 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 C0IE bit is changed from 0 (interrupt request disabled) to 1 (interrupt request enabled), the comparator interrupt request might be generated. Before changing these bits, set the ELSRn register to 0 (the CMP output is not linked). After changing these bits, clear the IR flag in the IELSRn register to 0 to clear the interrupt status.
Note 4. Comparator 0 is not available in R9A02G0214CBY. The write value should be 0.

The COMPOCR register is used to select items including the response speed of the comparators, control of the VCOUTn outputs, and enabling or disabling the interrupt request signals.

The COMPOCR register can be set by a 1-bit or 8-bit memory manipulation instruction.

The value of this register is 0x00 following a reset.

31.3.4 Registers Controlling Port Functions of Analog Input Pins

When using the IVCMP0, IVCMP1, IVREF0, IVREF1, VCOUT0 and VCOUT1 pins, set the corresponding bits of the Port mn Pin Function Select Register (PmnPFS).

31.4 Operation

Comparator 0 and comparator 1 operate independently. Their setting methods and operations are the same. Table 31.1 lists the Procedure for Setting Comparator Associated Registers.

Table 31.1 Procedure for setting comparator associated registers (1 of 2)

Step	Register	Bit	Setting value
1	Associated MSTPCRD register	MSTPD28	0: Cancel the module-stop state
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 converter, set the D/A converter register.
4	COMPOCR	SPDMD	Select the comparator response speed (0: Low-speed mode, 1: High-speed mode).*1

Table 31.1 Procedure for setting comparator associated registers (2 of 2)

Step	Register	Bit	Setting value
5	COMPMDR	CiVRF	Select the reference voltage. (For i = 0, 0: Input to the IVREF0 pin, 1: Internal reference voltage) (For i = 1, 0: Input to the IVREF1 pin, 1: D/A converter output 0)
		CiLVL	Select the reference input voltage range (0: 0 to V_{CC} - 1.4 V, 1: 1.4 V to V_{CC})
6	COMPMDR	CiENB	1 (operation enabled)
7	Wait for comparator stabilization time t_{CMP}.		
8	COMPFIR	CiFCK[1:0]	Select the sampling clock and whether the digital filter is used or not.
		CiEPO, CiEDG	Select the edge detection condition for an interrupt request (rising edge, falling edge or both edges).
9	COMPOCR	CiOP, CiOE	Set the VCOUTi output (select the polarity and set output enabled or disabled).
		CiIE	Set the interrupt request output enabled or disabled.
10	Associated pin function control register (PFS)	PSEL, PMR	Select the VCOUTi port function.
11	IELSRn	IR, IELS[4:0]	When using an interrupt, select the interrupt status flag and the ICU event link.*2

Note: i = 0, 1
Note 1. Comparator 0 and comparator 1 cannot be set independently.
Note 2. After the setting of the comparator, an unnecessary interrupt may occur until operation becomes stable, so initialize the interrupt flag.

Figure 31.2 shows an operation example of comparator i (i = 0, 1). In both low-speed mode and high-speed mode, the CiMON bit in the COMPMDR register is set to 1 when the analog input voltage is higher than the reference input voltage, and the CiMON bit is set to 0 when the analog input voltage is lower than the reference input voltage. The input voltage to the IVREF0 pin is selected for use as the reference voltage.

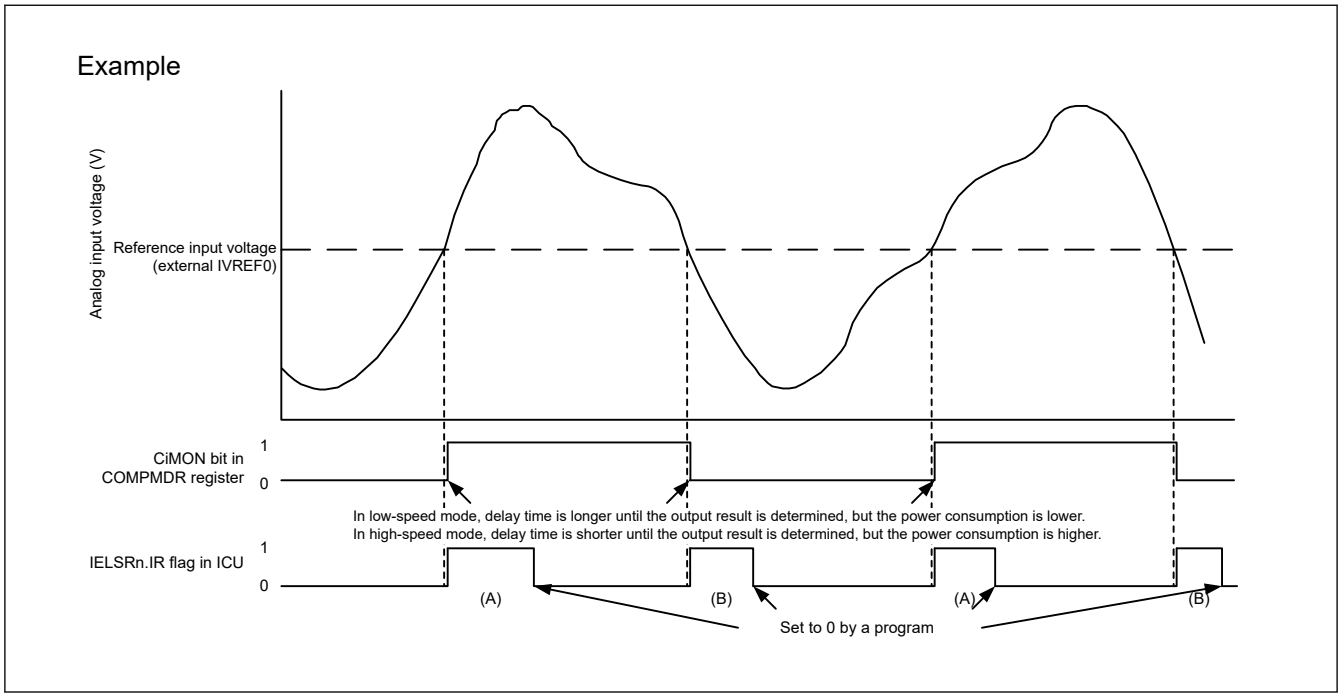


Figure 31.2 Comparator i (i = 0, 1) operation example

Note: The above diagram applies when the CiFCK[1:0] bits in the COMPFIR register = 00b (no filter) and CiEDG = 1 (both edges). When CiEDG = 0 and CiEPO = 0 (rising edge), the IELSR.IR flag changes as shown by (A) only. When CiEDG = 0 and CiEPO = 1 (falling edge), the IELSR.IR flag changes as shown by (B) only.

31.4.1 Comparator i Digital Filter (i = 0, 1)

Comparator i contains a digital filter. The sampling clock can be selected by the CiFCK[1:0] bits in the COMPFIR register.

The comparator i output signal is sampled every sampling clock, and when the level matches three times, that value is determined as the digital filter output at the next sampling clock.

Figure 31.3 shows the Comparator i (i = 0, 1) Digital Filter and Interrupt Operation Example.

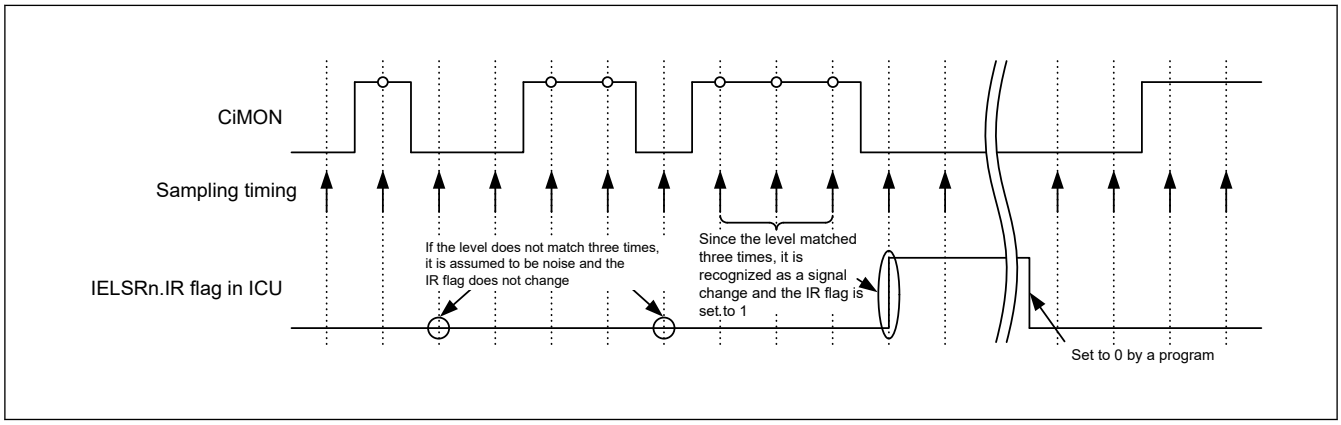


Figure 31.3 Comparator i (i = 0, 1) digital filter and interrupt operation example

Note: The above operation example applies when the CiFCK[1:0] bits in the COMPFIR register is 01b, 10b, or 11b (digital filter enabled).

31.4.2 Comparator i (i = 0, 1) Interrupts

The comparator generates interrupt requests from two sources, comparator 0 and comparator 1.

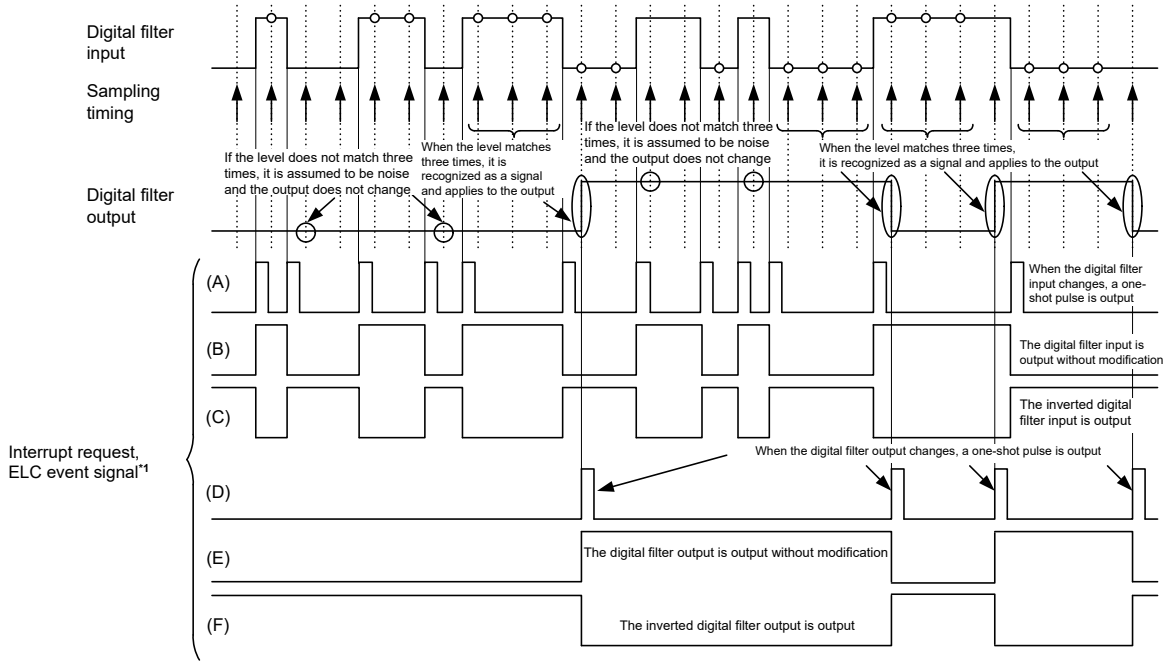
When using the comparator i interrupt, set the CiIE bit in the COMPOCR register to 1 (interrupt request output enabled). Additionally, set the IELSR register of the Interrupt Controller Unit (ICU). The condition for interrupt request generation can be set by the COMPFIR register. The comparator outputs can also be passed through the digital filter. Three different sampling clocks can be selected for the digital filter.

For details on the register setting and interrupt request generation, refer to section 31.3.2. COMPFIR : Comparator Filter Control Register and section 31.3.3. COMPOCR : Comparator Output Control Register.

31.4.3 Event Signal Output for the Event Link Controller (ELC)

An event signal for the ELC is generated by detecting the edge for the digital filter output set by the COMPFIR register, which is the same as the condition for interrupt request generation. However, unlike interrupt requests, the event signal for the ELC is always output regardless of the setting of the CilE bit in the COMPOCR register.

Figure 31.4 shows the Interrupt Request and Event Signal Output for ELC Generated by Digital Filter.



Note 1. When the CiIE bit (i = 0, 1) is 1, the same waveform is generated for an interrupt request and an ELC event signal. When the CiIE bit (i = 0, 1) is 0, the value is fixed at 0 for an interrupt request only.

Figure 31.4 Interrupt request and event signal output for ELC generated by digital filter

The waveforms of (A), (B), and (C) are shown for an operation example when the CiFCK[1:0] bits (i = 0, 1) in the COMPFIR register are 00b (no digital filter). The waveforms (D), (E), and (F) are shown for an operation example when the CiFCK[1:0] bits (i = 0, 1) in the COMPFIR register are 01b, 10b, or 11b (digital filter enabled).

(A) and (D) apply when the CiEDG bit is set to 1 (both edges), (B) and (E) when the CiEDG bit is 0 and the CiEPO bit is 0 (rising edge), and (C) and (F) when the CiEDG bit is 0 and the CiEPO bit is 1 (falling edge).

31.4.4 Comparator i Output (i = 0, 1)

The comparison result from the comparator can be output to external pins. Bits CiOP and CiOE in the COMPOCR register can be used to set the output polarity (non-inverted output or inverted output) and output enabled or disabled.

For the correspondence between the register setting and the comparator output, refer to section 31.3.3. COMPOCR : Comparator Output Control Register.

To output the comparator comparison result to the VCOUTi output pin, set the associated Port mn Pin Function Control register (PmnPFS) in the I/O register.

31.5 Usage Notes

31.5.1 About Activating DTC

When DTC activation is enabled under either of the following conditions, a DTC transfer is started and an interrupt is generated after completion of the transfer. Therefore, check the monitor flag (CnMON) of the comparator before enabling DTC activation as required (n = 0, 1).

- The comparator is set to an interrupt request on one-edge detection (CnEDG = 0), an interrupt request at the rising edge for the comparator (CnEPO = 0), and IVCMP > comparator n reference voltage
- The comparator is set to an interrupt request on one-edge detection (CnEDG = 0), an interrupt request at the falling edge for the comparator (CnEPO = 1), and IVCMP < comparator n reference voltage

31.5.2 Settings for the Module-Stop Function

CMP operation can be disabled or enabled using the Module Stop Control Register. The CMP is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 10, Low Power Modes.

31.5.3 CMP Operation in Module-Stop State

When the module-stop state is entered while CMP is operating, analog circuits in the CMP are not stopped and the analog power supply current is the same as that when CMP is being used. If the analog power supply current needs to be reduced in the module-stop state, set the COMPMDR.CiENB bit to 0 to stop the CMP.

31.5.4 CMP Operation in Software Standby Mode State

When the Software Standby mode is entered while CMP is operating, analog circuits in the CMP are not stopped and the analog power supply current is the same as that when CMP is being used. If the analog power supply current needs to be reduced in the module-stop state, set the COMPMDR.CiENB bit to 0 to stop the CMP.

31.5.5 Setting the D/A Converter for Generating Reference Voltage

Set the D/A converter to generate reference voltage and wait for the D/A converter conversion time before enabling the comparator. Similarly, before making any changes to the settings of the D/A converter, stop the comparator temporarily. After the changes are made, wait for the D/A converter conversion time before enabling the comparator.

32. Temperature Sensor (TSN)

32.1 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 ADC12 for conversion and can be further used by the end application.

Table 32.1 lists the TSN specifications, and Figure 32.1 shows a block diagram.

Table 32.1 TSN specifications

Item	Description
Temperature sensor voltage output	Temperature sensor outputs a voltage to the 12-bit A/D converter

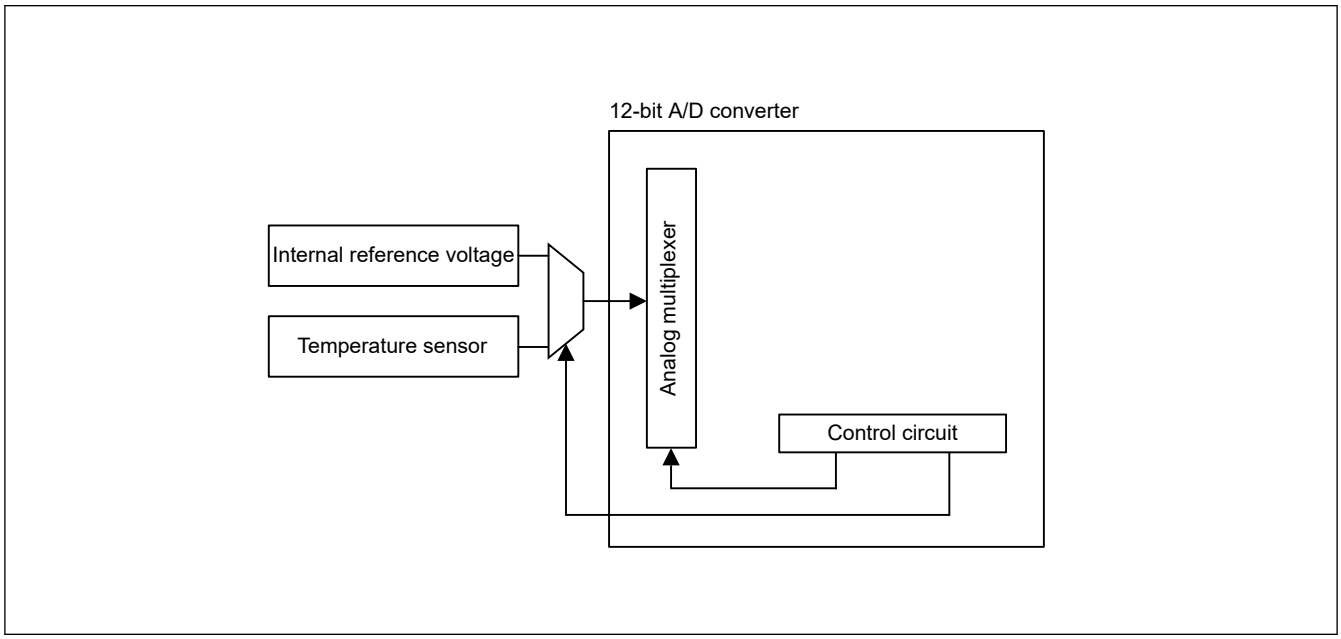


Figure 32.1 TSN block diagram

32.2 Register Descriptions

32.2.1 TSCDR : Temperature Sensor Calibration Data Register

Base address: FLCN = 0x407E_C000

Offset Address: 0x0228

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

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 = 140°C and VCC = 3.3 V converted to a digital value by the 12-bit A/D converter.

Temperature sensor calibration data is stored in the lower 12 bits of the TSCDR register.

32.3 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 12-bit A/D converter. To obtain the die temperature, convert this value into the temperature.

32.3.1 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 12-bit A/D converter to measure the voltage V1 output by the temperature sensor at temperature T1.
2. Again use the 12-bit 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 37, 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 (CAL140) of the temperature sensor measured under the condition Ta = Tj = 140°C and VCC = 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 CAL140:

V1 = 3.3 × CAL140 / 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 + 140 [°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 = 140°C and VCC = 3.3 V (V)
- Slope: Temperature gradient of the temperature sensor*1 / 1000 (V/°C)

Note 1. See section 37, Electrical Characteristics.

32.3.2 Procedures for Using the Temperature Sensor

For details, see section 29, 12-bit A/D Converter (ADC12).

33. Data Operation Circuit (DOC)

33.1 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 33.1 lists the data operation circuit specifications and Figure 33.1 shows a block diagram of the data operation circuit.

Table 33.1 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_0000 (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)

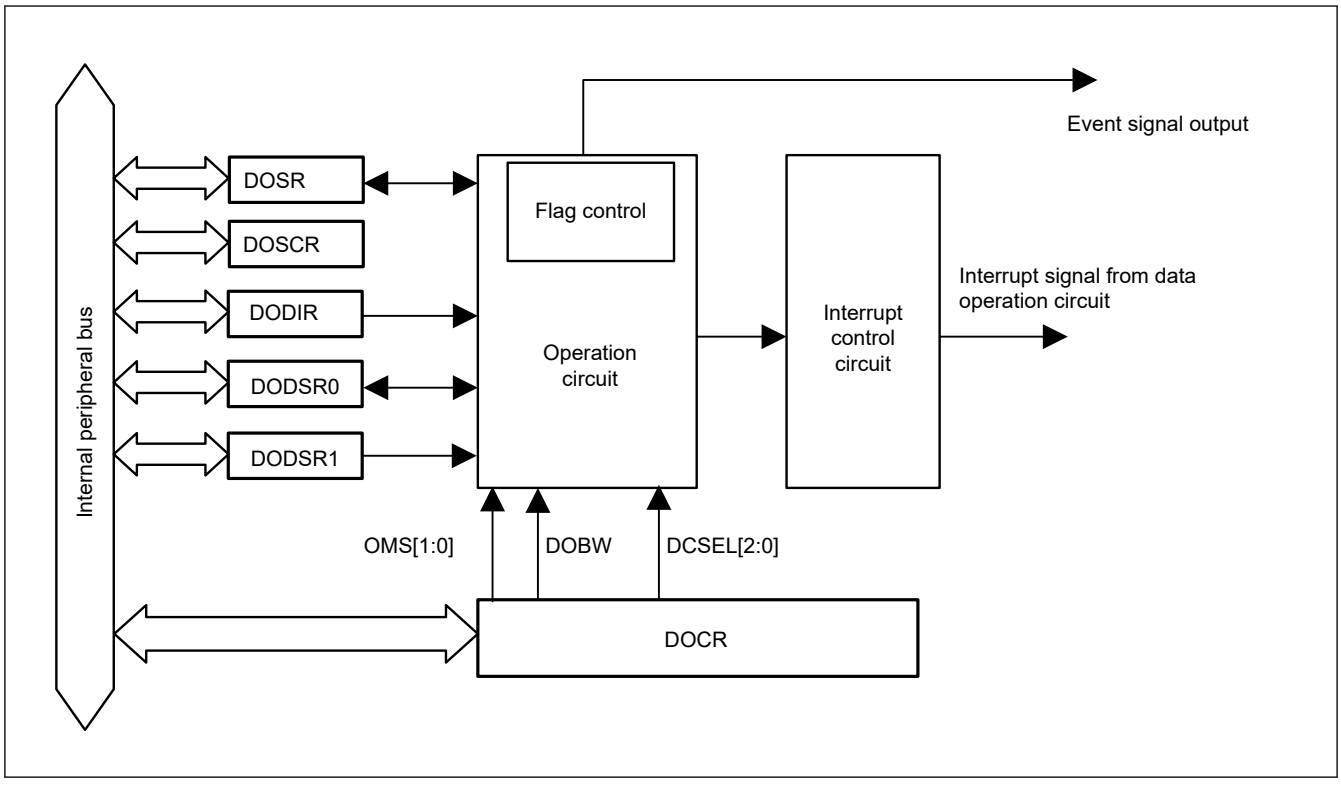


Figure 33.1 DOC block diagram

33.2 Register Descriptions

33.2.1 DOCR : DOC Control Register

Base address: DOC = 0x4008_5F00

Offset address: 0x00

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	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
7	—	This bit is read as 0. The write value should be 0.	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.

33.2.2 DOSR : DOC Flag Status Register

Base address: DOC = 0x4008_5F00

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] = 00b (Data comparison mode): The compared value matches the detection condition selected by the DOCR.DCSEL[2:0] bits
- DOCR.OMS[1:0] = 01b (Data addition mode): The result of data addition is greater than 0xFFFF (DOCR.DOBW = 0) or 0xFFFF_FFFF (DOCR.DOBW = 1)
- DOCR.OMS[1:0] = 10b (Data subtraction mode): The result of data subtraction is less than 0x0000 (DOCR.DOBW = 0) or 0x0000_0000 (DOCR.DOBW = 1)

[Clearing condition]

- Writing 1 to the DOSCR.DOPCFCL bit

33.2.3 DOSCR : DOC Flag Status Clear Register

Base address: DOC = 0x4008_5F00

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.

33.2.4 DODIR : DOC Data Input Register

Base address: DOC = 0x4008_5F00

Offset address: 0x0C

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:																																
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

33.2.5 DODSR0 : DOC Data Setting Register 0

Base address: DOC = 0x4008_5F00

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

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

33.2.6 DODSR1 : DOC Data Setting Register 1

Base address: DOC = 0x4008_5F00

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

33.3 Operation

33.3.1 Data Comparison Mode

Figure 33.2 to Figure 33.7 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] bits 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 the DOCR.DCSEL[2:0] bits, the DOCR.DOPCF flag is set to 1 and an ELC event and a data operation circuit interrupt are 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).

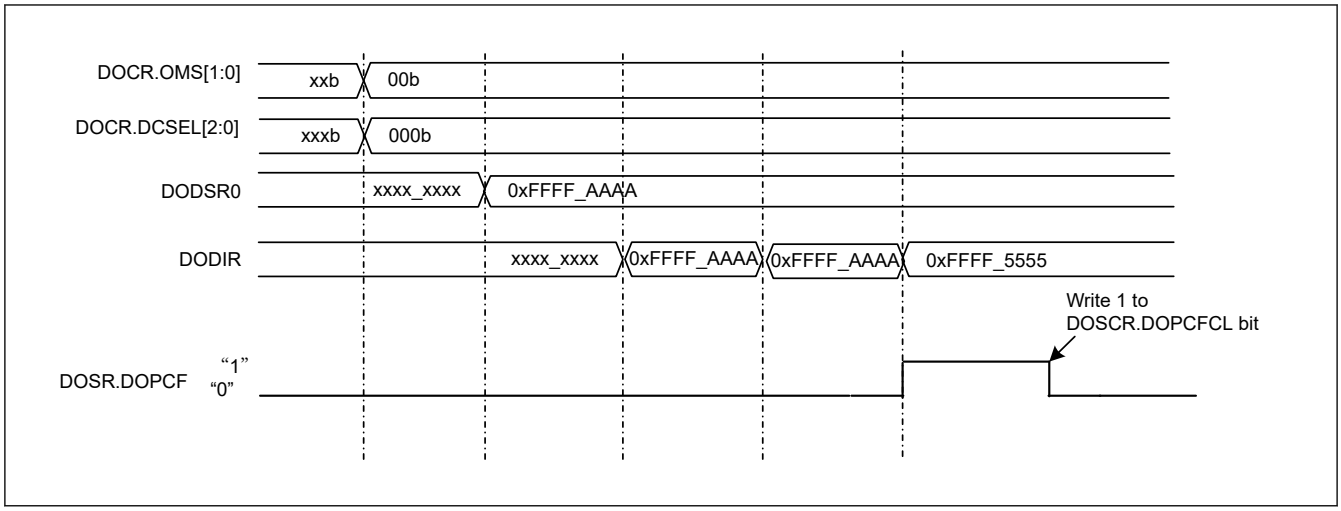


Figure 33.2 Example of Operation in Data Comparison Mode (Detection condition: Mismatch)

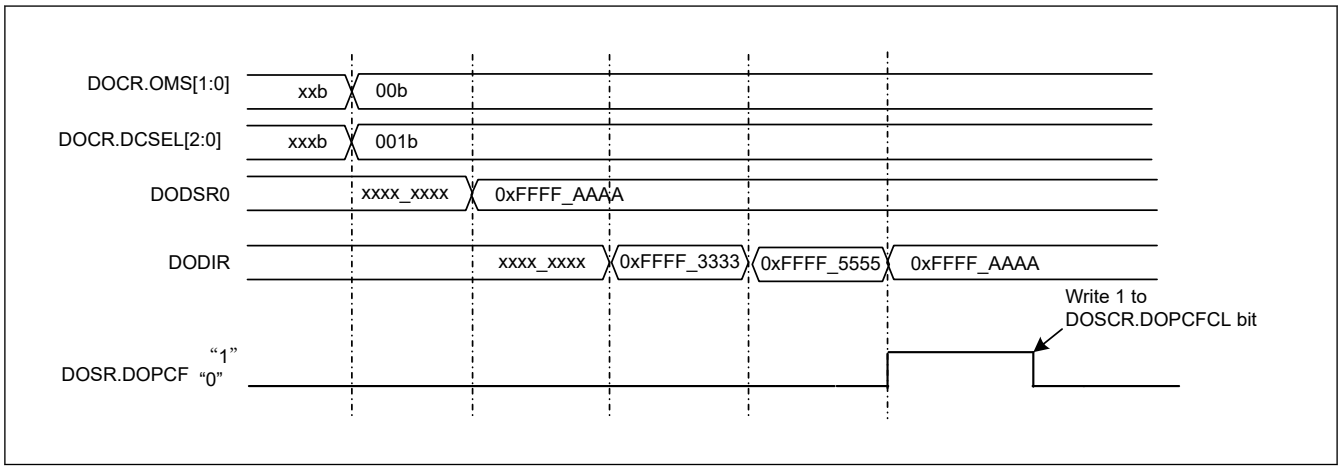


Figure 33.3 Example of Operation in Data Comparison Mode (Detection condition: Match)

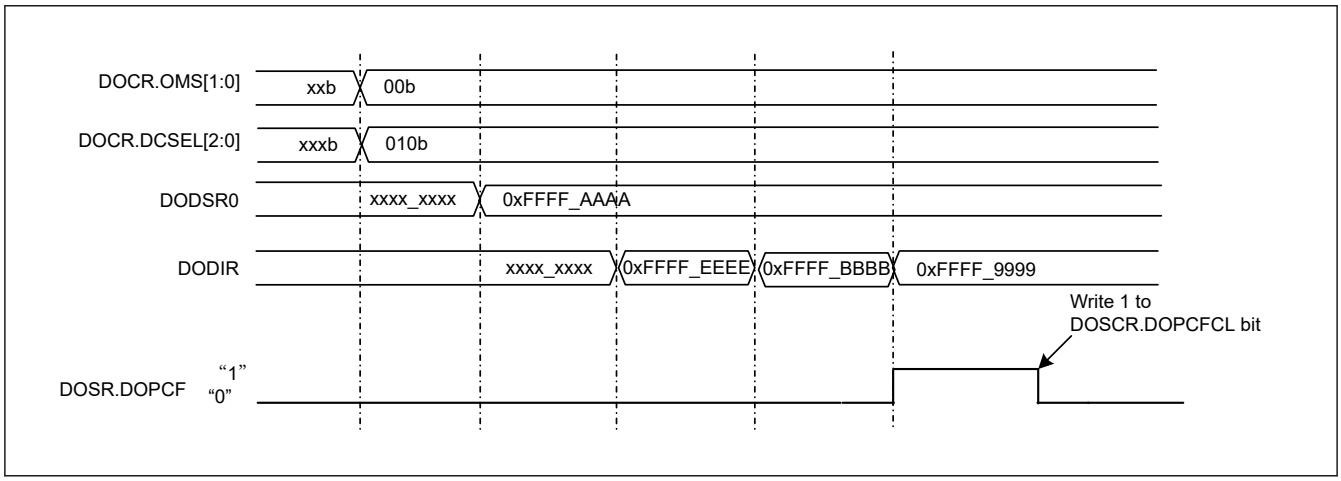


Figure 33.4 Example of Operation in Data Comparison Mode (Detection condition: Lower)

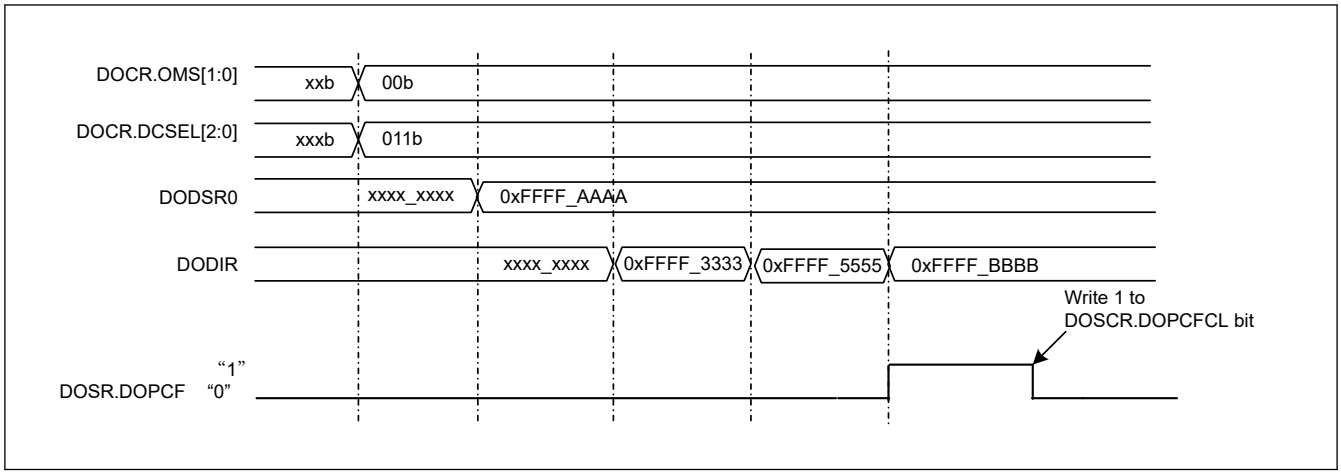


Figure 33.5 Example of Operation in Data Comparison Mode (Detection condition: Upper)

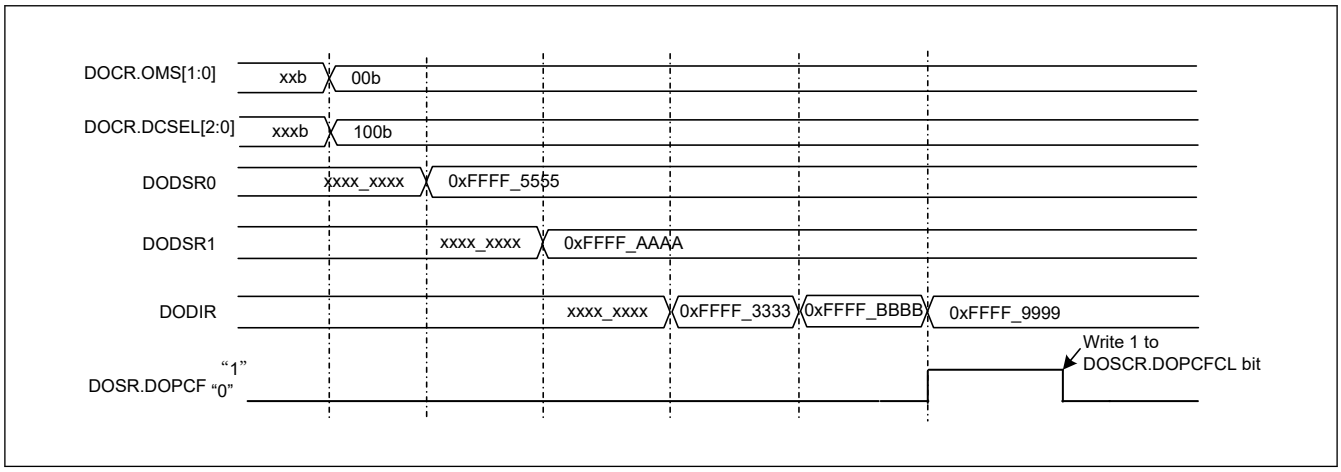


Figure 33.6 Example of Operation in Data Comparison Mode (Detection condition: Inside window)

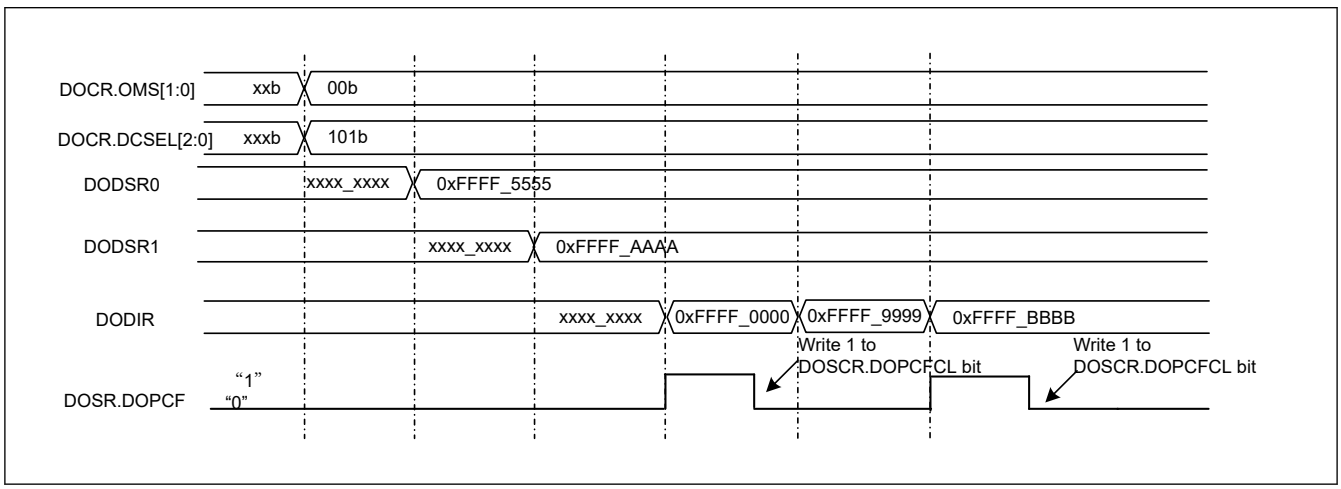


Figure 33.7 Example of Operation in Data Comparison Mode (Detection condition: Outside window)

33.3.2 Data Addition Mode

Figure 33.8 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 and a data operation circuit interrupt are generated.

Note 1. Addition is executed only by writing to the DODIR.

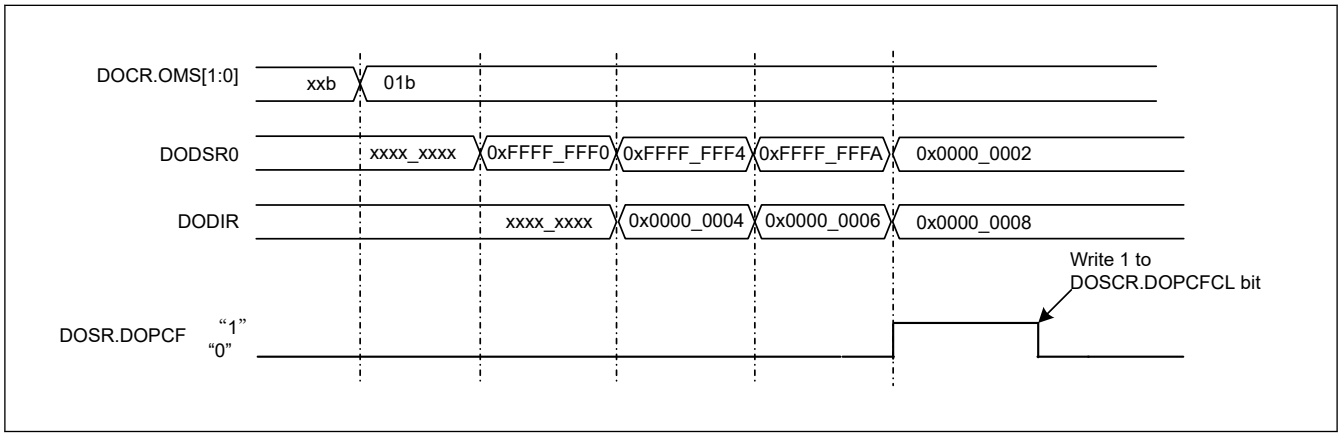


Figure 33.8 Example of Operation in Data Addition Mode

33.3.3 Data Subtraction Mode

Figure 33.9 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 and a data operation circuit interrupt are generated.

Note 1. Subtraction is executed only by writing to the DODIR.

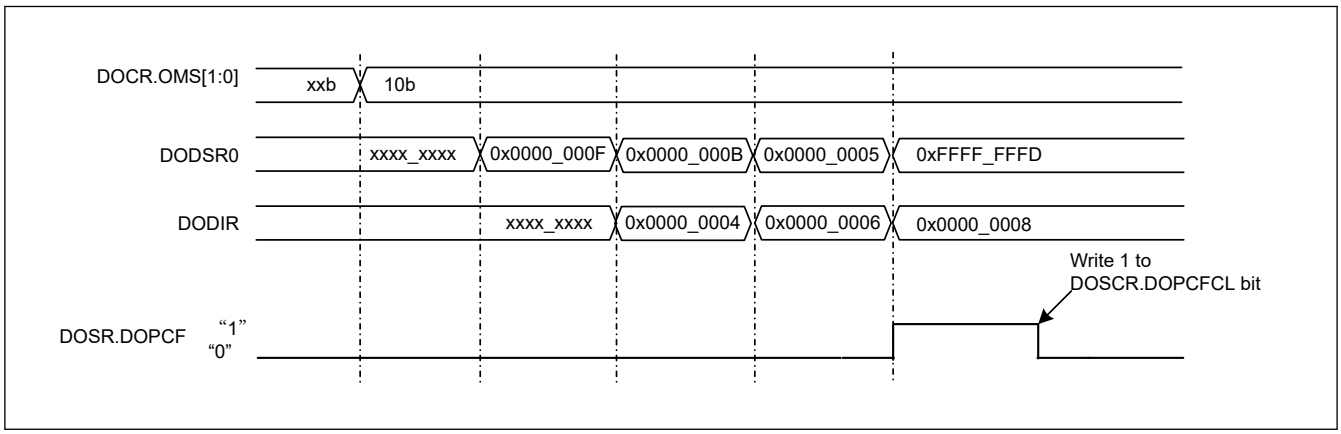


Figure 33.9 Example of Operation in Data Subtraction Mode

33.4 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, and then interrupt request signal is generated. Table 33.2 describes the interrupt request.

Table 33.2 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).

33.5 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)

33.6 Usage Notes

33.6.1 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.

34. SRAM

34.1 Overview

The MCU provides an on-chip, high-density SRAM module with either parity-bit checking or Error Correction Code (ECC). The first 4 KB area of the SRAM0 is the ECC. Parity check is performed on the other areas.

Table 34.1 lists the SRAM specifications.

Table 34.1 SRAM specifications

Parameter	Without ECC	With ECC
SRAM capacity	SRAM0: 12 KB	SRAM0: 4 KB
SRAM address	SRAM0: 0x2000_4000 to 0x2000_6FFF	SRAM0: 0x2000_0000 to 0x2000_0FFF
Access	0 wait For details, see section 34.3.6. Access Cycle	
Parity	Even parity with 8-bit data and 1-bit parity	No parity
Error checking	Even parity error check	1-bit error correction and up to 2-bit error detection

34.2 Register Descriptions

34.2.1 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

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 SRAM0 (without ECC).

34.2.2 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
7:1	KW[6:0]	Write Key Code These bits enable or disable writes to the SRAMPRCR bit	W

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.

34.2.3 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

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 access mode to the ECC area in SRAM0.

34.2.4 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 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 the ECC area of 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 ECC area in SRAM0 while writing 0 to this register.

34.2.5 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

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 (ECC area).

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 SRAM (ECC area) 1-Bit Error Status Register (ECC1STS) in response to a 1-bit error in the ECC area of SRAM0. This register also functions as an interrupt or a reset mask.

34.2.6 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 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 ECC area of 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 ECC area in SRAM0 while writing 0 to this register.

34.2.7 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

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.

34.2.8 ECCPRCR2 : ECC Protection Register 2

Base address: SRAM = 0x4000_2000

Offset address: 0xD0

Bit position:	7	6	5	4	3	2	1	0
Bit field:	KW2[6:0]							ECCPRCR2
Value after reset:	0	0	0	0	0	0	0	0

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

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.

34.2.9 ECCETST : ECC Test Control Register

Base address: SRAM = 0x4000_2000

Offset address: 0xD4

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

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

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 34.3.3. ECC Decoder Testing.

34.2.10 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

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 (ECC area).

34.3 Operation

34.3.1 ECC Function

You can enable or disable the ECC function by setting the ECCMODE register. By default, the ECC function is disabled and the ECC check type is SEC-DED (Single-Error Correction and Double-Error Detection).

When the ECC function is enabled, 7-bit check bits are appended to the 32-bit data for writes. For reads, 39-bit data (32- bit data and 7-bit check bits) is read from the SRAM (ECC area).

When the ECC function and error checking are both enabled, an error correction is performed 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, the error is detected without error correction, and the ECC2ERR bit in the ECC2STS register is set to 1.

When the ECC function is enabled and error checking is disabled, error correction is performed if a 1-bit error occurs but the ECC1ERR bit in the ECC1STS register is not updated even if the E1STSEN bit in the ECC1STSEN register is 1. If a 2-bit error occurs, the error is detected but the ECC2ERR bit in the ECC2STS register is not updated, and error correction is not performed.

When the ECC function is disabled, neither error correction nor error detection is performed even when a 1-bit or 2-bit error occurs. Therefore, the ECC1ERR bit or ECC2ERR bit is not updated.

It is not possible to confirm the location where the error is detected. Therefore, after an error occurred, update all the data by writing 32-bit data to the SRAM.

The SRAM data is undefined after a power on. If the SRAM is accessed when the ECC function and error checking are both enabled, an ECC error might occur. To avoid this, write 32-bit data to the area used in the SRAM before using the ECC function.

When a read access is performed consecutively after a write access, the read access has priority. Therefore, during initialization, do not perform a read access successively after a write access.

34.3.2 ECC Error Generation

When the ECC function is enabled and error checking is applied to SRAM0 (ECC area), 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 a 2-bit error or a 1-bit error has occurred.

To mask ECC 1-bit errors, set the ECC1STSEN.E1STSEN bit to 0 to disable ECC1ERR bit update. An ECC error is not generated when the ECC function is disabled or enabled without error checking.

An ECC error can generate a non-maskable interrupt or a reset, as selected in the ECCOAD register. When the OAD bit in the ECCOAD register is set to 1, an ECC error is output to the reset function. When the OAD bit in the ECCOAD register is set to 0, an ECC error is output to the ICU as a non-maskable interrupt.

34.3.3 ECC Decoder Testing

Figure 34.1 shows the ECC decoder testing.

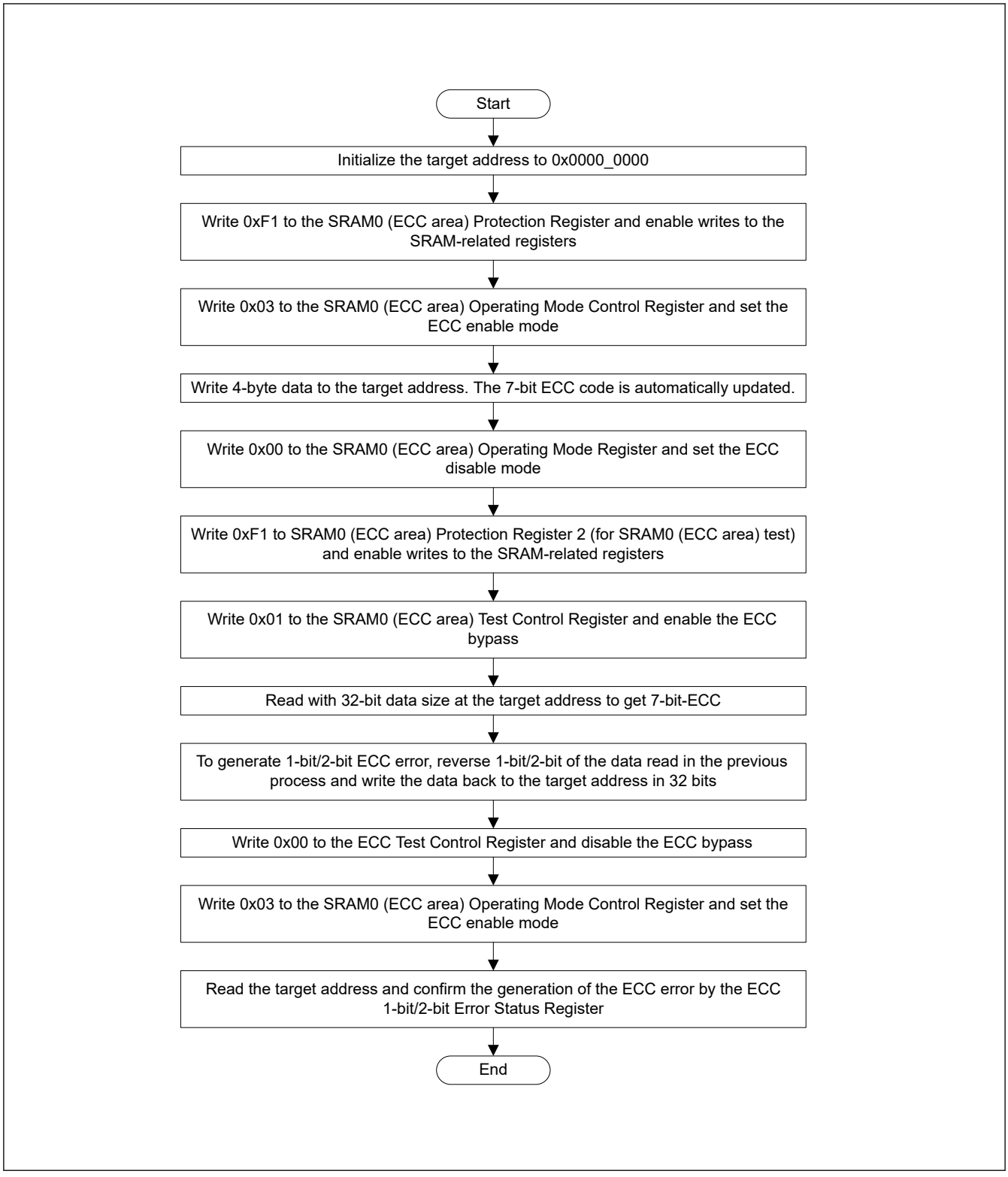


Figure 34.1 ECC decoder testing

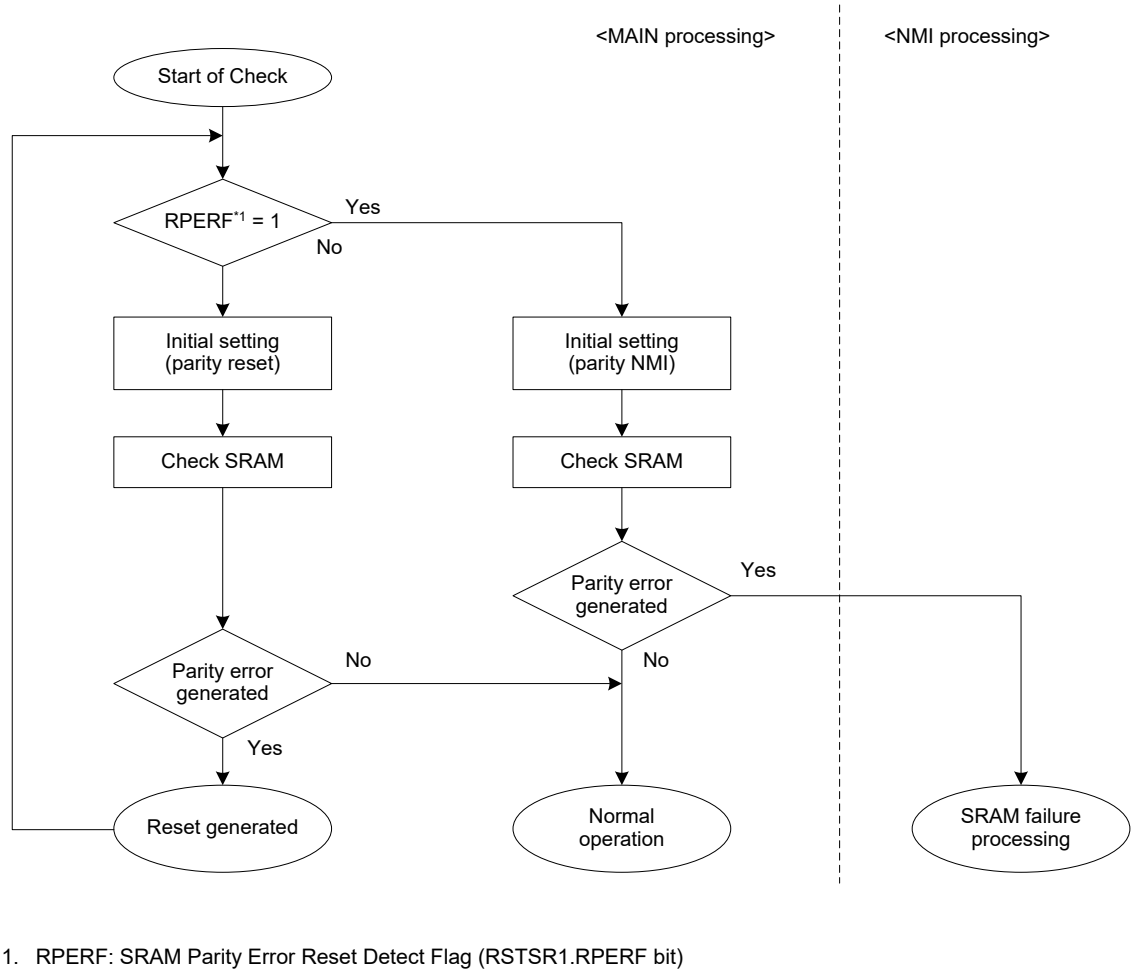
34.3.4 Parity Calculation Function

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 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 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.

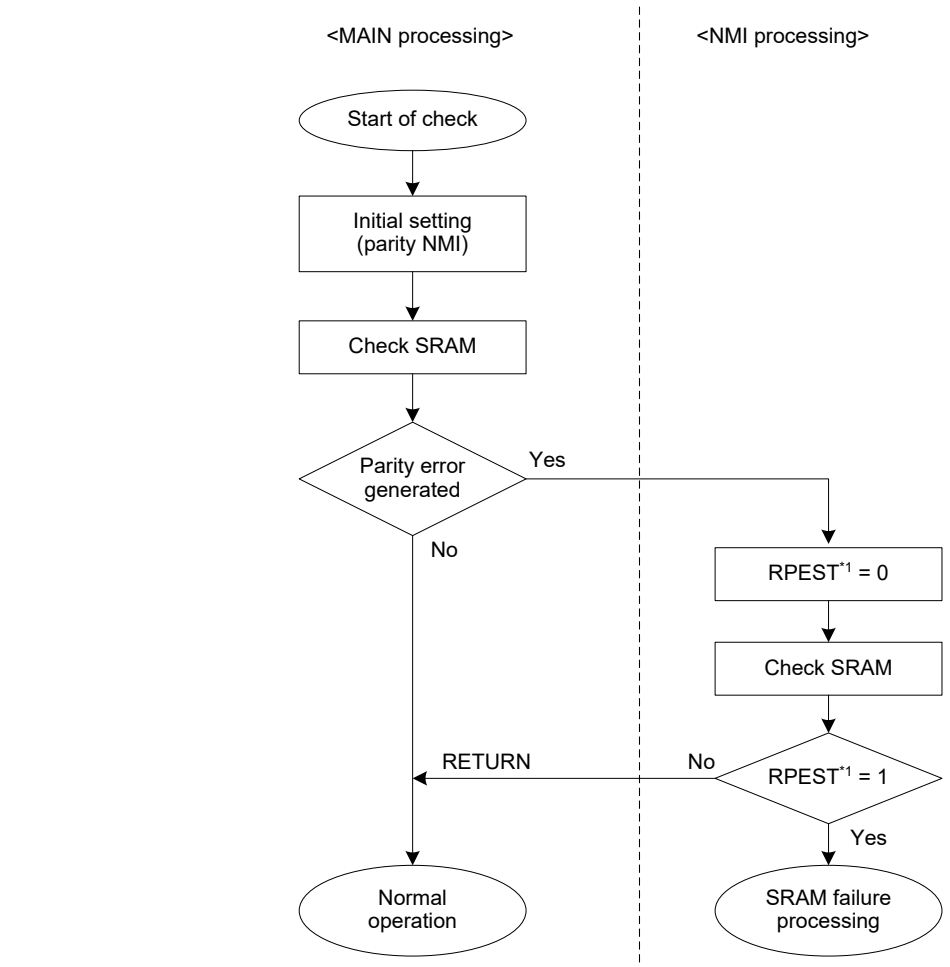
Parity errors can be occasionally caused by noise. To confirm whether the cause of the parity error is noise or corruption, follow the parity check flows shown in Figure 34.2 and Figure 34.3.

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: SRAM Parity Error Reset Detect Flag (RSTSR1.RPERF bit)

Figure 34.2 Flow of SRAM parity check when SRAM parity reset is enabled



Note 1. RPEST: SRAM Parity Error Interrupt Status Flag (NMISR.RPEST bit)

Figure 34.3 Flow of SRAM parity check when SRAM parity interrupt is enabled

34.3.5 SRAM Error Sources

An SRAM error is either an ECC error or a parity error. ECC error or parity error can generate either a non-maskable interrupt or a reset, as selected with the OAD bit in the PARIOAD register. DTC activation is not supported for SRAM ECC error and SRAM parity errors.

Table 34.2 SRAM error sources

SRAM error source	DTC activation
ECC error (SRAM0 area with ECC)	Not possible
Parity error (SRAM0 area without ECC)	Not possible

34.3.6 Access Cycle

Table 34.3 SRAM0 (ECC area 0x2000_0000 to 0x2000_0FFF)

ECC On/Off	Read (cycles)		Write (cycles)	
	Word access	Halfword/Byte access	Word access	Halfword/Byte access
ECC Off ECCMOD[1] = 0	2		2	
ECC On ECCMOD[1] = 1	2		2	4

Table 34.4 SRAM0 (parity area 0x2000_4000 to 0x2000_6FFF)

Read (cycles)		Write (cycles)	
Word access	Halfword/Byte access	Word access	Halfword/Byte access
2		2	

34.3.7 Low-Power Function

Power consumption can be further reduced in Software Standby mode as the supply voltage for SRAM0 can be off, except for the 4 KB in the head area of SRAM0 (0x2000_4000 to 0x2000_4FFF) of SRAM0 (parity area). For details on Software Standby mode, see section 10, Low Power Modes.

34.4 Usage Notes

34.4.1 Instruction Fetch from the SRAM Area

When using SRAM0 to operate a program, initialize the SRAM area so that the CPU can correctly prefetch data. If the CPU prefetches data from an SRAM area that is not initialized, ECC error or a parity error might occur. Initialize the additional 2-byte area from the end address of a program with a 4-byte boundary. Renesas recommends using the NOP instruction for data initialization.

34.4.2 SRAM Store Buffer

For fast access between SRAM and CPU, a store buffer is used. When a load instruction is executed from the same address after a store instruction to SRAM, the load instruction might read data from the buffer instead of data on the SRAM. To read data on the SRAM correctly, use either of the following procedures:

- After writing to the SRAM (address = A), use the C.NOP instruction twice, then read the SRAM (address = A)
- After writing to the SRAM (address = A), read data from area other than SRAM, then read the SRAM (address = A)
- After writing to the SRAM (address = A), writing to the SRAM (address ≠ A), then read the SRAM (address = A).

35. Flash Memory

35.1 Overview

The MCU provides up to 128-KB code flash memory and 4-KB data flash memory. The Flash Control Block (FCB) controls the programming commands. This product uses SuperFlash® technology licensed from Silicon Storage Technology, Inc.

Table 35.1 lists the specifications of the code flash memory and data flash memory, and Figure 35.1 shows a block diagram of the related modules. Figure 35.2 shows the configuration of the code flash memory, and Figure 35.3 shows the configuration of the data flash memory.

Table 35.1 Code flash memory and data flash memory specifications (1 of 2)

Parameter	Code flash memory	Data flash memory
Memory capacity	• 128-KB of user area • Configuration setting area (see section 6, Option-Setting Memory)	4-KB of data area
Read cycle	• ICLK frequency ≤ 48 MHz MEMWAIT = 1 with wait A read operation takes 3 cycles • ICLK frequency ≤ 32 MHz MEMWAIT = 0 without wait A read operation takes 2 cycles	• ICLK frequency ≤ 48 MHz FLDWAIT1 = 1 with 2 waits A read operation takes 4 cycles • ICLK frequency ≤ 32 MHz FLDWAIT1 = 0 with 1 wait A read operation takes 3 cycles
Value after erasure	0xFF	0xFF
Programming/erasing method	• Programming and erasure of code and data flash memory through the FCB commands specified in the registers • Programming by dedicated flash-memory programmer through a serial interface (serial programming) • Programming of flash memory by user program (self-programming)*1.	
Protection function	Protection against illicit tampering with or reading of data in flash memory. Protection against erroneous overwriting of flash memory: • User ID • Flash Read Protection On-chip debugger connect protection: • A connect to on-chip debugger is protected by OCDDIS bit	
Background operation (BGO)	Code flash memory can be read during data flash memory programming	
Units of programming and erasure	• 64-bit units for programming in user area • 2-KB units for erasure in user area • 8-bit units for programming in data area • 1-KB units for erasure in data area	
Software commands	The serial programming mode or self-programming mode has the following commands and programs: • Blank check • Block erase • Chip erase • Program • Access window information program • Startup area information and protection program • User ID information program Checksum can be also executed in serial programming mode. • Suspend of the block erase command can be also executed during the self-programming mode	
Start-up program protection	The user area (first 32 KB address) is capable of swapping by 16 KB unit.	
Other functions	Interrupts accepted during self-programming	
	Option-setting memory can be set in the initial MCU settings	

Table 35.1 Code flash memory and data flash memory specifications (2 of 2)

Parameter	Code flash memory	Data flash memory
On-board programming	Programming in serial programming mode (UART (SAU) boot mode): ● Asynchronous serial interface (SAU_2) used ● Transfer rate adjusted automatically. Programming in on-chip debug mode: ● cJTAG interface used ● Dedicated hardware not required. Programming by a routine for code and data flash memory programming within the user program: ● Allows code and data flash memory programming without resetting the system.	

Note 1. HOCO should be stably oscillated. See section 35.12. Self-Programming.

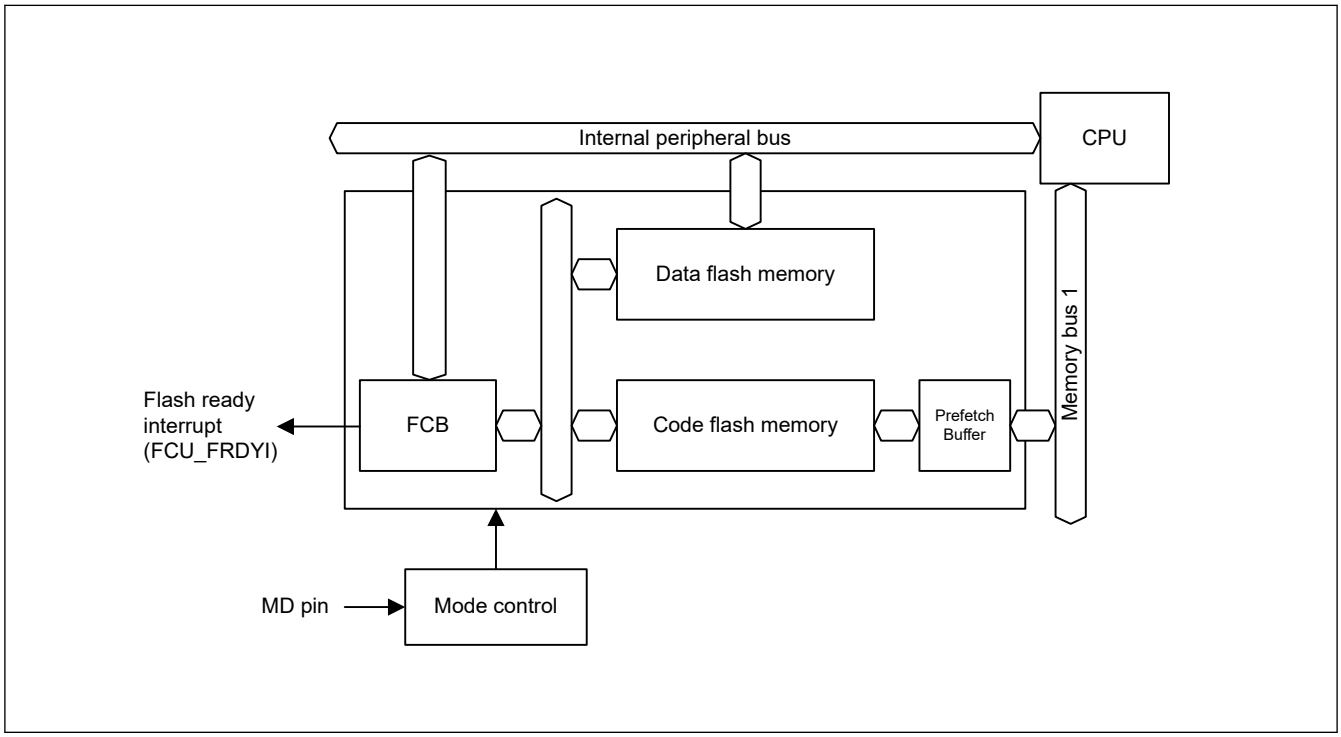


Figure 35.1 Flash memory-related modules block diagram

35.2 Memory Structure

Figure 35.2 shows the mapping of the code flash memory, and Table 35.2 shows the read and programming and erasure (P/E) addresses of the code flash memory. The user area of the code flash memory is divided into 2-KB blocks that serve as the units of erasure. The user area is available for storing the user program.

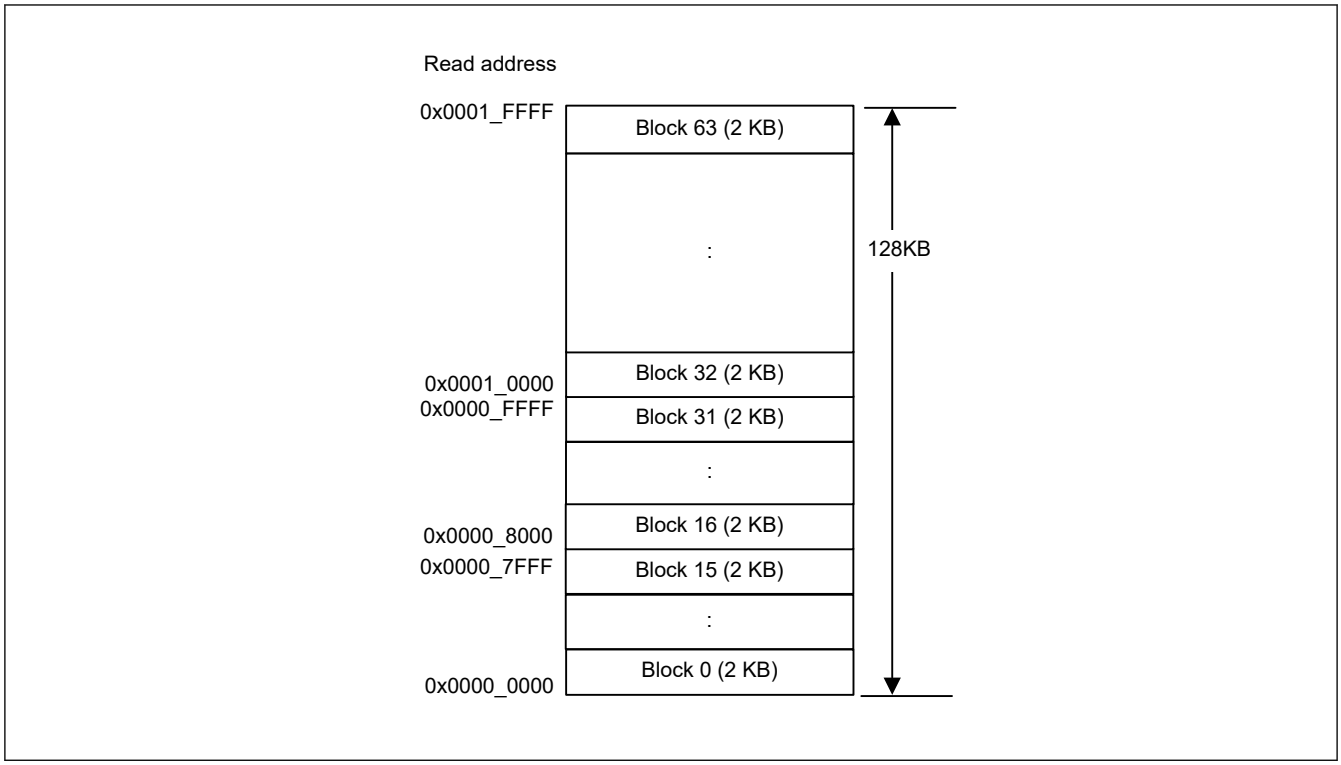


Figure 35.2 Mapping of the code flash memory

Table 35.2 Read and P/E addresses of the code flash memory

Size of code flash memory	Read address	P/E address	Number of blocks
128 KB	0x0000_0000 to 0x0001_FFFF	0x0000_0000 to 0x0001_FFFF	0 to 63

Figure 35.3 shows the mapping of the data flash memory, and Table 35.3 shows the read and programming and erasure (P/E) addresses of the data flash memory. The data area of the data flash memory is divided into 1-KB blocks, with each being a unit for erasure.

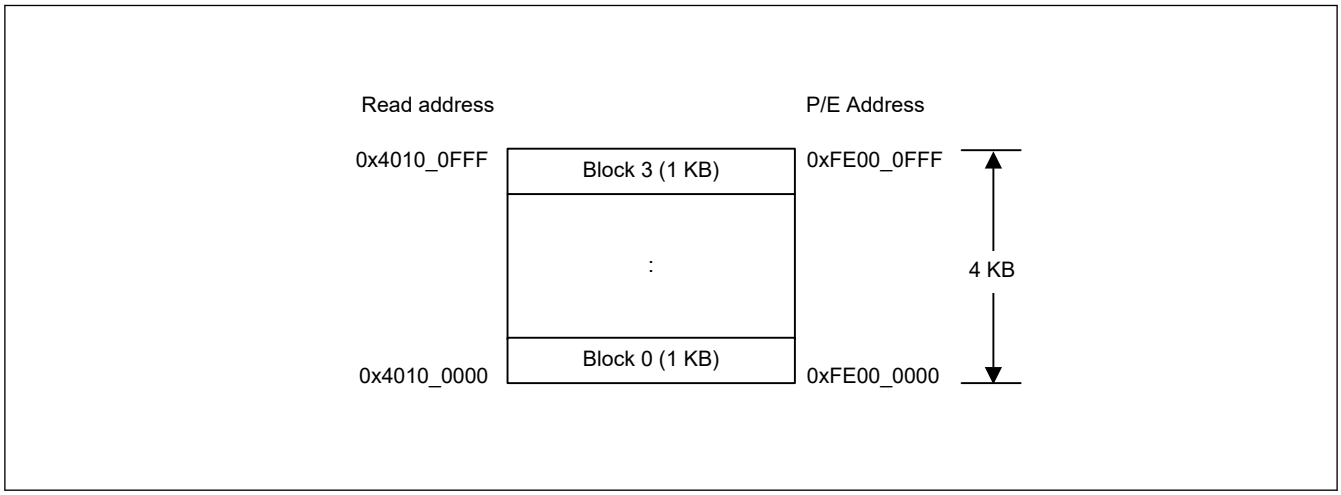


Figure 35.3 Mapping of the data flash memory

Table 35.3 Read and P/E addresses of the data flash memory

Size of data flash memory	Read address	P/E address	Number of blocks
4-KB	0x4010_0000 to 0x4010_0FFF	0xFE00_0000 to 0xFE00_0FFF	0 to 3

35.3 Register Descriptions

35.3.1 DFLCTL : Data Flash Control Register

Base address: FLCN = 0x407E_C000

Offset address: 0x0090

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

Bit	Symbol	Function	R/W
0	DFLEN	Data Flash Access Enable*1 0: Access to the data flash is disabled 1: Access to the data flash is enabled	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. It is necessary that DFLCTL.DFLEN bit is set to 1 before issuing the startup area information and protection program, access window information program, and User ID program command.

The DFLCTL register enables or disables accessing (reading, programming, and erasing) of the data flash. After setting the DFLCTL.DFLEN bit, Data Flash STOP recovery time (t_{DSTOP}) is necessary before reading the data flash or entering the data flash P/E mode.

35.3.2 PFBER : Prefetch Buffer Enable Register

Base address: FLCN = 0x407E_C000

Offset address: 0x3FC8

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

Bit	Symbol	Function	R/W
0	PFBE	Prefetch Buffer Enable bit 0: Prefetch buffer is disabled 1: Prefetch buffer is enabled	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

35.3.3 FENTRYR : Flash P/E Mode Entry Register

Base address: FLCN = 0x407E_C000

Offset address: 0x3FB0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	FEKEY[7:0]								FENTRYD	—	—	—	—	—	—	FENTRY0
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	FENTRY0	Code Flash P/E Mode Entry 0 0: The code flash is the read mode 1: The code flash is the P/E mode.	R/W
6:1	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
7	FENTRYD	Data Flash P/E Mode Entry 0: The data flash is the read mode 1: The data flash is the P/E mode.	R/W
15:8	FEKEY[7:0]	Key Code	W

To program the code flash or the data flash, either the FENTRY0 or FENTRYD bit must be set to 1 to enter the P/E mode. Clearing the FENTRY0 bit or FENTRYD bit allows the code flash or data flash to be in read mode, but it is necessary to confirm the value of this bit before changing it. See section 35.13.1. Sequencer Modes.

FENTRY0 bit (Code Flash P/E Mode Entry 0)

[Setting condition]

- Set 0xAA01 to the FENTRYR register when it is 0x0000.

[Clearing conditions]

- Data is written by byte access
- A value other than 0xAA is set to the FEKEY[7:0] bits and written to the FENTRYR register
- Set 0xAA00 to the FENTRYR register
- Data is written to the FENTRYR register while the register has a value other than 0x0000.

FENTRYD bit (Data Flash P/E Mode Entry)

[Setting condition]

- Set 0xAA80 to the FENTRYR register when the register is 0x0000.

[Clearing conditions]

- Data is written by byte access.
- A value other than 0xAA is set to the FEKEY[7:0] bits and data is written to the FENTRYR register.
- Set 0xAA00 to the FENTRYR register.
- Data is written to the FENTRYR register while the FENTRYR register has a value other than 0x0000.

FEKEY[7:0] bits (Key Code)

The FEKEY[7:0] bits protect from unauthorized setting of FENTRY0 bit or FENTRYD bit.

Setting 0xAA to FEKEY[7:0] allows setting the FENTRY0 bit or the FENTRYD bit. The FEKEY[7:0] bits are read as 0x00.

35.3.4 FPR : Protection Unlock Register

Base address: FLCN = 0x407E_C000

Offset address: 0x0180

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

Bit	Symbol	Function	R/W
7:0	FPR[7:0]	Protection Unlock This register is used to protect the FPMCR register from being rewritten inadvertently when the CPU runs out of control.	R/W

FPR[7:0] bits (Protection Unlock)

Writing to the FPMCR register is allowed only when the following procedure is used to access the register.

Procedure to unlock protection:

1. Write 0xA5 to the FPR register.
2. Write a set value to the FPMCR register
3. Write the inverted set value to the FPMCR register.
4. Write a set value to the FPMCR register again.

When a procedure other than the above is used to write data, the FPSR.PERR flag is set to 1.

35.3.5 FPSR : Protection Unlock Status Register

Base address: FLCN = 0x407E_C000

Offset address: 0x0184

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

Bit	Symbol	Function	R/W
0	PERR	Protect Error Flag 0: No error 1: An error occurs	R
7:1	—	These bits are read as 0.	R

PERR bit (Protect Error Flag)

When the FPMCR register is not accessed as described in the procedure to unlock protection, data is not written to the register and this flag is set to 1.

[Setting condition]

- The FPMCR register is not accessed as described in the procedure to unlock protection described in section 35.3.4. FPR : Protection Unlock Register.

[Clearing conditions]

- The FPMCR register is accessed according to the procedure to unlock protection described in section 35.3.4. FPR : Protection Unlock Register.

35.3.6 FPMCR : Flash P/E Mode Control Register

Base address: FLCN = 0x407E_C000

Offset address: 0x0100

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	FMS1	RPDIS	—	FMS0	—
Value after reset:	0	0	0	0	1	0	0	0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
1	FMS0	Flash Operating Mode Select 0 0: FMS1 = 0: Read mode FMS1 = 1: Data flash P/E mode. 1: FMS1 = 0: Code flash P/E mode FMS1 = 1: Setting prohibited.	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
3	RPDIS	Code Flash P/E Disable 0: Programming of the code flash is enabled 1: Programming of the code flash is disabled	R/W
4	FMS1	Flash Operating Mode Select 1 See the description of the FMS0 bit.	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

The FPMCR register sets the operating mode of the flash memory and is protected from unauthorized setting.

See Figure 35.16 and Figure 35.18 for this register write control method.

See section 35.3.4. FPR : Protection Unlock Register for the procedure to unlock the protection.

FMS0 bit, FMS1 bits (Flash Operating Mode Select 0, Flash Operating Mode Select 1)

These bits set the operating mode of the flash memory.

[How to enter the code flash from the read mode to the code flash P/E mode]

Set FMS1 bit = 0, FMS0 bit = 1, and RPDIS bit = 0. Wait for the mode setup time t_{MS} (see section 37, Electrical Characteristics).

[How to enter the data flash from the read mode to the data flash P/E mode]

Set FMS1 bit = 1, FMS0 bit = 0, and RPDIS bit = 0.

[How to enter the code flash from the code flash P/E mode to the read mode]

Set FMS1 bit = 0, FMS0 bit = 0, and RPDIS bit = 1.

Wait for the read mode transition time (see section 37, Electrical Characteristics).

RPDIS bit (Code Flash P/E Disable)

RPDIS bit protects the code flash from unauthorized programming. Setting RPDIS bit to 0 allows the code flash to program.

35.3.7 FISR : Flash Initial Setting Register

Base address: FLCN = 0x407E_C000

Offset address: 0x01D8

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

Bit	Symbol	Function	R/W
5:0	PCKA[5:0]	Flash-IF Clock Notification	R/W
7:6	SAS[1:0]	Startup Area Select 1 0: The startup area is switched to the default area temporarily 1 1: The startup area is switched to the alternate area temporarily. Others: The startup area is selected according to the settings of the extra area.	R/W

Note: Set or clear this register only in P/E mode. Additionally the SAS[1:0] bits are allowed to set or clear when the FSPR/BTPR is 1. The FSPR/BTPR bit is the protection flag of the access window/setting boot flag and is stored in the extra area.

PCKA[5:0] bits (Flash-IF Clock Notification)

These bits are used to set the frequency of the FlashIF clock (ICLK). The hardware sequencer for the flash programming executes the commands according to the PCKA[5:0] bits. For this reason, it is necessary to set the frequency to the PCKA [5:0] bits before execution of the programming and not to change during the programming.

Note: A wrong frequency setting may cause the flash macro to be damaged.

The following information describes how to set the PCKA[5:0] bits when the frequency is not an integral number, for example 31.5 MHz.

[When the frequency is higher than 4 MHz]

Set a rounded-up value for a non-integer frequency.

For example, set 32 MHz (PCKA = 011111b) when the frequency is 31.5 MHz.

[When the frequency is 4 MHz or lower]

Do not use a non-integer frequency. Use the frequency of 1, 2, 3, or 4 MHz.

Table 35.4 Frequency settings

Flash-IF clock frequency [MHz]	PCKA[5:0]	Flash-IF clock frequency [MHz]	PCKA[5:0]	Flash-IF clock frequency [MHz]	PCKA[5:0]
48	100111b	32	011111b	24	010111b
20	010011b	19	010010b	18	010001b
17	010000b	16	001111b	15	001110b
14	001101b	13	001100b	12	001011b
11	001010b	10	001001b	9	001000b
8	000111b	7	000110b	6	000101b
5	000100b	4	000011b	3	000010b
2	000001b	1	000000b	—	—

SAS[1:0] bits (Startup Area Select)

The SAS[1:0] bits select the startup area. To change the startup area, the following three methods can be used:

- When selecting the startup area according to the startup area settings of the extra area with the SAS[1:0] bits set to 00b or 01b, the startup area is selected according to the startup area settings of the extra area. The settings are enabled after a reset is released.
- When switching the startup area to the default area temporarily with 10b written to the SAS[1:0] bits, the startup area is switched to the default area immediately after data is written to the register, regardless of the startup area settings of the extra area. When a reset is generated after this, the area is selected according to the startup area settings of the extra area.
- When switching the startup area to the alternative area temporarily with 11b written to the SAS[1:0] bits, the startup area is switched to the alternative area, regardless of the startup area settings of the extra area. When a reset is generated after this, the area is selected according to the startup area settings of the extra area.

35.3.8 FRESETR : Flash Reset Register

Base address: FLCN = 0x407E_C000

Offset address: 0x0124

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

Bit	Symbol	Function	R/W
0	FRESET	Software reset of the registers 0: The registers related to the flash programming are not reset 1: The registers related to the flash programming are reset.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

FRESET bit (Software reset of the registers)

When this bit is set to 1, the FASR, FSARH, FSARL, FEARH, FEARL, FWBH0/1, FWBL0/1, FCR, and FEXCR registers are reset. Setting this bit to 0 allows the corresponding registers to be released from the reset state. Software commands are not allowed to execute while the FRESET bit is 1.

35.3.9 FASR : Flash Area Select Register

Base address: FLCN = 0x407E_C000

Offset address: 0x0104

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

Bit	Symbol	Function	R/W
0	EXS	Extra Area Select 0: User area or data area 1: Extra area	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: Set or clear this register only in P/E mode.

EXS bit (Extra Area Select)

Set the EXS bit to 1 when programming the extra area using the FEXCR register. Set this bit to 0 when not programming the extra area.

35.3.10 FCR : Flash Control Register

Base address: FLCN = 0x407E_C000

Offset address: 0x0114

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

Bit	Symbol	Function	R/W
3:0	CMD[3:0]	Software Command Setting 0x1: Program 0x3: Blank check (code flash) 0x4: Block erase 0x6: Chip erase 0xB: Blank check (data flash) Others: Setting prohibited*1.	R/W
5:4	—	These bits are read as 0. The write value should be 0.	R/W
6	STOP	Forced Processing Stop When this bit is set to 1, the processing being executed can be forcibly stopped.	R/W
7	OPST	Processing Start 0: Processing stops 1: Processing starts	R/W

Note: Set or clear this register only in P/E mode. Additionally it is not allowed to be reset by the FRESETR register while the software command is being executed.

Note 1. This does not include writing 0x00 to the FCR register when the FSTATR1.FRDY bit is 1.

CMD[3:0] bits (Software Command Setting)

The following information describes the function of each software command.

[Program]

Writes data of the FWBH0/1 and FWBL0/1 registers to the flash macro to the address pointed by the FSARH and FSARL registers.

[Blank check]

Verifies whether the flash macro is the blank state (not to be programmed) from the start address pointed by the FSARH and FSARL registers to the end address pointed by the FEARH and FEARL registers. The blank check command is allowed to execute within the region of flash macro.

Note: The blank check result cannot guarantee that the flash memory is erased.

[Block erase]

Erases block of the flash memory.

Set the start address of the target erasure block in the FSARH and FSARL registers, and set the end address of the target erasure block in the FEARH and FEARL registers. If a setting other than the specified is made, erasure may not be executed correctly. The block erase command is allowed to execute within the region of flash macro.

[Chip erase]

Erases all blocks of the flash macro

Set the start address of the target erasure block in the FSARH and FSARL registers, and set the end address of the target erasure block in the FEARH and FEARL registers. If a setting other than the specified is made, erasure may not be executed correctly.

STOP bit (Forced Processing Stop)

The STOP bit stops the execution of the erase command or the blank check command.

After setting 1 to the STOP bit, it is necessary to wait until the FSTATR1.FRDY bit becomes 1 (processing completed) before setting the OPST bit to 0.

OPST bit (Processing Start)

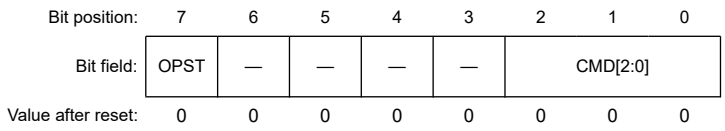
The OPST bit starts the command set for the CMD[2:0] bits. Setting the OPST bit to 0 terminates the execution of the command after the FRDY bit of the FSTATR1 register becomes 1, and is required to confirm that the FRDY bit is 0.

Note:
- Commands cannot be executed when the ID authorization for the flash programmer has failed.
- The program and the block erase command cannot be executed when the address of each command points to an area that is protected by the access window.

35.3.11 FEXCR : Flash Extra Area Control Register

Base address: FLCN = 0x407E_C000

Offset address: 0x01DC



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

Bit	Symbol	Function	R/W
2:0	CMD[2:0]	Software Command Setting 0 0 1: Protection setting 0 1 0: Access window information program 0 1 1: User ID1 program 1 0 0: User ID2 program 1 0 1: User ID3 program 1 1 0: User ID4 program Others: Setting prohibited*1	R/W
6:3	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
7	OPST	Processing Start 0: Processing stops 1: Processing starts	R/W

Note: Set or clear this register only in P/E mode. Additionally it is not allowed to be reset by the FRESETR register while the software command is being executed.
Note 1. This does not include writing 0x00 to the FEXCR register when the FSTATR1.EXRDY bit is 1.

The FEXCR register programs the extra area. Before execution of each command, it is necessary to set data to the FWBL0 and FWBH0 registers.

When programming using the FEXCR register, the programming area is erased automatically before execution, therefore it is not necessary to erase beforehand.

CMD[2:0] bits (Software Command Setting)

The CMD[2:0] bits select the software command from the:

- Protection setting
- Access window information program
- User ID program.

[Protection setting]

This command set data to the FWBL0/FWBH0 registers. This command is allowed to set the protection and product information. P/N code0 and P/N code1 are used for product information.

Bits [15:0] of the FWBH0 register are 0 or 1 (P/N code1 set product information).

Bit [14] of the FWBL0 register is 0 (the access window and user ID information cannot be updated because the access window and user ID information program command cannot be executed).

Bit [13] of the FWBL0 register is 0 (the on-chip debugger cannot be connected).

Bits [7:0] of the FWBL0 register are 0 or 1 (P/N code0 set product information).

Table 35.5 describes extra bit mapping for the startup area selection and protection setting.

Table 35.5 Mapping for the extra bit of the startup area selection and protection setting (address (P/E) : 0x0000_0008)

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
P/N code1[15:0]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	FAPR *1	OCDDIS*1*2	—	—	—	BTPR *1*2	BTFLG	P/N code0[7:0]							

Note 1. Once 0 is set as data for these bits, it cannot be changed to 1 by protection setting.
Note 2. This bit can be changed to 1 from 0 by ALeRASE sequence.

[Access window information program]

This command sets the access window used for area protection. The program command and block erase command of the protected area cannot be executed. The chip erase command cannot be executed when the access window is set (the start block address of the access window is not equal to the end block address). It is necessary to set the start block address of the access window to the FWBL0 register bits [10:0] and the next block address of the end block address of the access window to the FWBH0 register bits [10:0] before the execution of the access window information program command. When the start address and the end address are set to the same value, all areas of the code flash can be accessed. When the start address is larger than the end block address, all areas of the code flash cannot be accessed.

The FWBL0[10] bit for the start block address must be set to 0 when the access window is set (the end block address of the access window is larger than the start block address).

The following information describes mapping for the extra bit of the access window information program.

Table 35.6 Mapping for the extra bit of the access window information program (address (P/E) : 0x0000_0010)

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
—	—	—	—	—	FAWE[10:0]										
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	FAWS[10:0]										

[User ID1-4 program]

These commands set the User ID[127:0] bits.

Table 35.7 User ID settings

Command	User ID	FWBH0	FWBL0
User ID1 program	User ID [31:0]	User ID [31:16]	User ID [15:0]
User ID2 program	User ID [63:32]	User ID [63:48]	User ID [47:32]
User ID3 program	User ID [95:64]	User ID [95:80]	User ID [79:64]
User ID4 program	User ID [127:96]	User ID [127:112]	User ID [111:96]

The following information describes mapping for the extra bit of User ID1-4 program.

Table 35.8 Mapping for the extra bit of User ID1-4 program (address (P/E) : 0x0000_0018)

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
User ID[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
User ID[15:0]															

Table 35.9 Mapping for the extra bit of User ID1-4 program (address (P/E) : 0x0000_0020)

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
User ID[63:48]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
User ID[47:32]															

Table 35.10 Mapping for the extra bit of User ID1-4 program (address (P/E) : 0x0000_0028)

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
User ID[95:80]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
User ID[79:64]															

Table 35.11 Mapping for the extra bit of User ID1-4 program (address (P/E) : 0x0000_0030)

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
User ID[127:112]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
User ID[111:96]															

OPST bit (Processing Start)

The OPST bit starts the command set for the CMD[2:0] bits. Setting the OPST bit to 0 terminates the execution of the command after the EXRDY bit of the FSTATR1 register becomes 1, and is necessary to confirm that the EXRDY bit is 0.

35.3.12 FSARH : Flash Processing Start Address Register H

Base address: FLCN = 0x407E_C000

Offset address: 0x0110

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	FSARH[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	FSARH[15:0]	Flash Processing Start Address H Flash Processing Start Address upper 16 bits See FSARL for details.	R/W

Note: Set or clear this register only in P/E mode. The write value should be 0 for b8 to b5, and those bits are read as 0.

35.3.13 FSARL : Flash Processing Start Address Register L

Base address: FLCN = 0x407E_C000

Offset address: 0x0108

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	FSARL[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	FSARL[15:0]	Flash Processing Start Address L Flash processing start address lower 16 bits	R/W

Note: Set or clear this register only in P/E mode.

The FSARH and FSARL registers set the start address of the software command. When the FSARH and FSARL registers are read while executing a software command set by the FEXCR register, an undefined value is read. After execution of the program command, the sequencer of the software command increments data automatically. The auto increment function of the program command discards the setting of the next address to the FSARH and FSARL registers when the next address is a consecutive address. The increment unit is as follows:

Code flash: +0x8

Data flash: +0x1

See Figure 35.2 and Figure 35.3 for details on the addresses of the flash memory.

35.3.14 FEARH : Flash Processing End Address Register H

Base address: FLCN = 0x407E_C000

Offset address: 0x0120

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	FEARH[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	FEARH[15:0]	Flash Processing End Address H Flash processing end address upper 16 bits See FEARL for details.	R/W

Note: Set or clear this register only in P/E mode. The write value should be 0 for b8 to b5, and those bits are read as 0.

35.3.15 FEARL : Flash Processing End Address Register L

Base address: FLCN = 0x407E_C000

Offset address: 0x0118

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	FEARL[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	FEARL[15:0]	Flash Processing End Address L Flash processing end address lower 16 bits	R/W

Note: Set or clear this register only in P/E mode.

The FEARH and FEARL registers set the end address of the blank check, the block erase, and the chip erase. When the FEARH and FEARL registers are read while executing a software command set by the FEXCR register, an undefined value is read.

See Figure 35.2 and Figure 35.3 for details on the addresses of the flash memory.

35.3.16 FWBL0 : Flash Write Buffer Register L0

Base address: FLCN = 0x407E_C000

Offset address: 0x0130

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	WDATA[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	WDATA[15:0]	Flash Write Buffer L0 Flash write buffer data lower 16 bits See FWBH0 for details.	R/W

Note: Set or clear this register only in P/E mode.

35.3.17 FWBH0 : Flash Write Buffer Register H0

Base address: FLCN = 0x407E_C000

Offset address: 0x0138

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	WDATA[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	WDATA[15:0]	Flash Write Buffer H0 Flash write buffer data upper 16 bits	R/W

Note: Set or clear this register only in P/E mode.

The FWBH0 and FWBL0 registers set program data of the program command, the startup selection and protection setting command, the access window information program command, and the User ID program command. The following table describes how to set data according to each command.

Register	What is set to the register
FWBH0 FWBL0	• Bits [31:0] of the programming data of the program command for the code flash • Bits [7:0] of the programming data of the program command for the data flash • Bits [31:0] of the programming data of the startup selection and protection setting command, the access window information program command, and the User ID program command.

35.3.18 FWBL1 : Flash Write Buffer Register L1

Base address: FLCN = 0x407E_C000

Offset address: 0x0140

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	WDATA[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	WDATA[15:0]	Flash Write Buffer L1 Bits [47:32]	R/W

Note: Set or clear this register only in P/E mode.

35.3.19 FWBH1 : Flash Write Buffer Register H1

Base address: FLCN = 0x407E_C000

Offset address: 0x0144

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	WDATA[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	WDATA[15:0]	Flash Write Buffer L1 Bits [63:48]	R/W

Note: Set or clear this register only in P/E mode.

The FWBL0, FWBH0, FWBL1 and FWBH1 registers are to set program data of the program command, the protection setting command, the access window information program command, and the user ID information. The following table describes how to set data according to each command.

Table 35.12

Register	What is set to the register
FWBH1 FWBL1	Bits 63-32 of the programming data for the code flash
FWBH0 FWBL0	• Bits 31-0 of the programming data of the program command for the code flash • Bits 31-0 of the programming data of the protection setting command, the access window information program command, and the user ID information

35.3.20 FSTATR00 : Flash Status Register 0

Base address: FLCN = 0x407E_C000

Offset address: 0x0128

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	EILGLERR	ILGLERR	BCERR0	—	PRGERR0	ERERR0
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	ERERR0	Erase Error Flag 0 0: Erasure terminates normally 1: An error occurs during erasure	R
1	PRGERR0	Program Error Flag 0 0: Programming terminates normally 1: An error occurs during programming	R
2	—	This bit is read as 0.	R
3	BCERR0	Blank Check Error Flag 0 0: Blank checking terminates normally 1: An error occurs during blank checking	R
4	ILGLERR	Illegal Command Error Flag 0: No illegal software command or illegal access is detected 1: An illegal command or illegal access is detected	R
5	EILGLERR	Extra Area Illegal Command Error Flag 0: No illegal command or illegal access to the extra area is detected 1: An illegal command or illegal access to the extra area is detected	R
7:6	—	These bits are read as 0 or 1.	R
15:8	—	These bits are read as 0.	R

FSTATR00 is a status register used to confirm the execution result of a software command. Each error flag is set to 0 when the next software command is executed.

ERERR0 flag (Erase Error Flag 0)

The value of the ERERR0 bit is undefined when the FCR.STOP bit is set to 1 (processing is forcibly stopped) during erasure.

PRGERR0 flag (Program Error Flag 0)

The PRGERR0 bit is set when the program command of the FCR register or each command of the FEXCR register is abnormally terminated.

BCERR0 flag (Blank Check Error Flag 0)

The value of the BCERR0 bit is undefined when the FCR.STOP bit is set to 1 (processing is forcibly stopped) during blank check.

ILGLERR flag (Illegal Command Error Flag)

The ILGLERR flag indicates the execution of the software command of the FCR register with unexpected condition.

[Setting conditions]

- Programming/erasure commands are executed to an area protected by the access window range
- The chip erase command is executed when the access window is set (the start block address of the access window is not equal to the end one)
- The blank check, the block erase, and the chip erase commands are executed when the start address set to the FSARH and FSARL registers is larger than the end address set to the FEARH and FEARL registers
- The program, the block erase, the chip erase, and the blank check commands are executed when the FASR.EXS bit is 1

- The program, the block erase and the chip erase commands are executed to the startup area when the BTPR bit is 0
- The data flash address is set to the FSARH and FSARL registers and a software command is executed in the code flash P/E mode
- The code flash address is set to the FSARH and FSARL registers and a software command is executed in the data flash P/E mode
- The code flash and the data flash are set to P/E mode simultaneously and a software command is executed.

[Clearing condition]

- The next software command is executed.

EILGLERR flag (Extra Area Illegal Command Error Flag)

The EILGLERR flag indicates the execution of the software command of the FEXCR register with unexpected condition.

[Setting conditions]

- The software commands of the FEXCR register is executed when the EXS bit of the FASR register is 0
- The access window or user ID information program command is executed when the FAPR bit is 0

[Clearing condition]

- The next software command is executed.

35.3.21 FSTATR1 : Flash Status Register 1

Base address: FLCN = 0x407E_C000

Offset address: 0x012C

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

Bit	Symbol	Function	R/W
0	—	This bit is read as 0.	R
1	—	This bit is read as 0.	R
2	—	This bit is read as 1.	R
5:3	—	These bits are read as 0.	R
6	FRDY	Flash Ready Flag 0: The software command of the FCR register is not terminated. 1: The software command of the FCR register is terminated.	R
7	EXRDY	Extra Area Ready Flag 0: The software command of the FEXCR register is not terminated. 1: The software command of the FEXCR register is terminated.	R

FSTATR1 is a status register used to confirm the execution result of a software command. Each flag is set to 0 when the next software command is executed.

35.3.22 FEAMH : Flash Error Address Monitor Register H

Base address: FLCN = 0x407E_C000

Offset address: 0x01E8

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	FEAMH[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	FEAMH[15:0]	Flash Error Address Monitor Register H Flash error address monitor upper 16 bits See FEAML for details.	R/W

35.3.23 FEAML : Flash Error Address Monitor Register L

Base address: FLCN = 0x407E_C000

Offset address: 0x01E0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	FEAML[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	FEAML[15:0]	Flash Error Address Monitor Register L Flash error address monitor lower 16 bits	R/W

The error address is withdrawn from the FEAMH and FEAML registers after a software command execution. See Figure 35.2 and Figure 35.3 for details on the addresses of the flash memory.

35.3.24 FSECMR : Flash Protection Flag Monitor Register

Base address: FLCN = 0x407E_C000

Offset address: 0x1C0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	FAPR	OCDDIS	—	—	—	BTPR	BTFLG	—	—	—	—	—	—	—	—
Value after reset:	0	Value set by the user*1		1	1	1	1	1	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0.	R
8	BTFLG	Startup Area Select Protection Flag 0: Startup area is the alternate block (block 1) 1: Startup area is the alternate block (block 0)	R
9	BTPR	Startup Area Select Protection Flag 0: Startup area select setting is locked 1: Startup area select setting is not locked	R
12:10	—	These bits are read as 1.	R
13	OCDDIS	On-Chip Debugger Connection Disable Flag 0: On-chip debugger connection disabled 1: On-chip debugger connection enabled	R
14	FAPR	Access Window Protection Flag 0: Access window setting is locked 1: Access window setting is not locked	R
15	—	This bit is read as 0.	R

Note 1. The rest value depends on the state of the extra area.

The FSECMR register monitors the extra area setting. Data of this register is updated at the reset sequence or at the execution of the software command of the FEXCR register.

35.3.25 FAWSMR : Flash Access Window Start Address Monitor Register

Base address: FLCN = 0x407E_C000

Offset address: 0x01C8

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

Bit	Symbol	Function	R/W
10:0	FAWS[10:0]	Access Window Start Address This register is used to confirm the set value of the access window start address used for area protection	R
15:11	—	These bits are read as 0.	R

Note 1. The value of the blank product is 1. It is set to the same value set in bits [10:0] in the FWBH0 register after the access window information program command is executed.

35.3.26 FAWEMR : Flash Access Window End Address Monitor Register

Base address: FLCN = 0x407E_C000

Offset address: 0x01D0

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

Bit	Symbol	Function	R/W
10:0	FAWE[10:0]	Access Window End Address This register is used to confirm the set value of the access window end address used for area protection	R
15:11	—	These bits are read as 0.	R

Note 1. The value of the blank product is 1. It is set to the same value set in bits [10:0] in the FWBL0 register after the access window information program command is executed.

35.4 Instruction Prefetch from Flash Memory

Flash memory provides an instruction prefetch function to accelerate code execution. The prefetch function can be used by enabling the prefetch buffer. To enable the prefetch buffer, set the PFBER.PFBE bit to 1.

Note: When Flash memory is in the program or erase operation, the PFBER.PFBE bit should be set to 0 beforehand in user programming program of internal SRAM.

35.5 Operating Modes Associated with the Flash Memory

Figure 35.4 shows a diagram of the mode transitions associated with the flash memory. For information on setting up the modes, see section 3, Operating Modes.

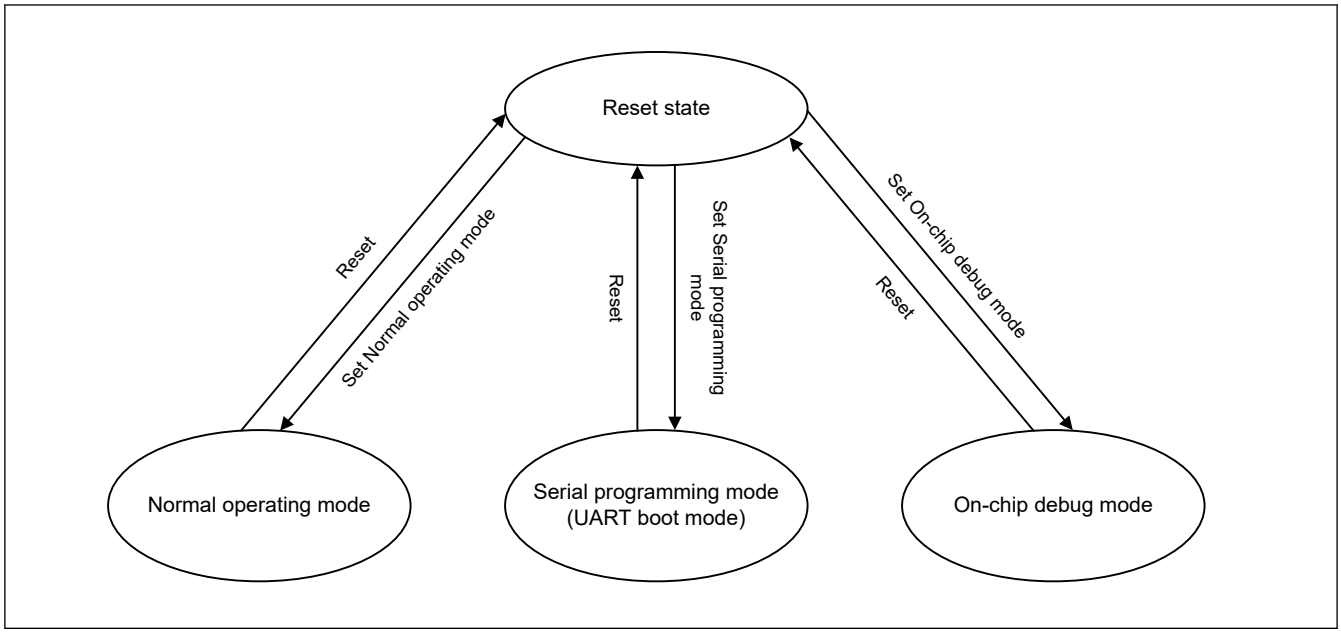


Figure 35.4 Mode transitions associated with flash memory

The flash memory areas where programming and erasure are permitted and where the boot program executes at a reset, differ with the mode. Table 35.13 shows the differences between the modes.

Table 35.13 Difference between modes

Parameter	Normal operating mode	Serial programming mode (UART boot mode)	On-chip debug mode
Programmable and erasable areas	• Code flash memory • Data flash memory.	• Code flash memory • Data flash memory.	• Code flash memory • Data flash memory.
Erasure in block units	Possible	Possible	Possible
Boot program at a reset	User area program	Embedded program for serial programming	Depends on debug command

35.5.1 ID Code Protection

The ID code protection function prohibits programming and on-chip debugging. When ID code protection is enabled, the device validates or invalidates the ID code sent from the host by comparing it with the ID code stored in the flash memory. Programming and on-chip debugging are enabled only when the two match.

The ID code in flash memory consists of four 32-bit words. ID code bits [127] and [126] determine whether ID code protection is enabled and the authentication method to use with the host. Table 35.14 shows how the ID code determines the authentication method.

Table 35.14 Specifications for ID code protection

Operating mode on boot up	ID code	State of protection	Operations on connection with the programmer or on-chip debugger
Serial programming mode (UART boot mode) On-chip debug mode	0xFF, …, 0xFF (all bytes 0xFF)	Protection disabled	The ID code is not checked, the ID code always matches, and the connection to the serial programmer or on-chip debugger*1 is permitted.
	Bit [127] = 1, bit [126] = 1, and at least one of all 16 bytes is not 0xFF	Protection enabled	Matching ID code indicates that authentication is complete and connection to the serial programmer or the on-chip debugger is permitted. Mismatching ID code indicates transition to the ID code protection wait state. When the ID code sent from the serial programmer or the on-chip debugger is ALeRASE in ASCII code (0x414C_6552_4153_45FF_FFFF_FFFF_FFFF_FFFF), the content of the user flash (code and data) area and configuration area are erased. However, forced erasure is not executed when the FAPR bit is 0.
	Bit [127] = 1 and bit [126] = 0	Protection enabled	Matching ID code indicates that authentication is complete and connection to the serial programmer or the on-chip debugger is permitted. Mismatching ID code indicates transition to the ID code protection wait state.
	Bit [127] = 0	Protection enabled	The ID code is not checked, the ID code is always mismatching, the connection to the serial programmer or the on-chip debugger is prohibited. When the ID code sent from the on-chip debugger is ALeRASE in ASCII code (0x414C_6552_4153_45FF_FFFF_FFFF_FFFF_FFFF), the content of the user flash (code and data) area and configuration area are erased. However, forced erasure is not executed when the FAPR bit is 0.

Note 1. Never send the ID code from on-chip debugger. Or send ID code 0xFF, …, 0xFF (all bytes 0xFF) from on-chip debugger.

35.6 Overview of Functions

By using a dedicated flash-memory programmer to program the on-chip flash memory through a serial interface (serial programming mode) or through cJTAG interface (on-chip debug mode), the device can be programmed before or after it is mounted on the target system. Additionally, protection functions to prohibit overwriting of the user program prevent tampering by third parties.

Programming by the user program (self-programming) is available for applications that might require updating after system manufacturing or shipment. Protection features for safely overwriting the flash memory area are also provided. Additionally, interrupt processing during self-programming is supported so that programming can proceed while processing external communications and other functions. Table 35.15 lists the programming methods and the associated operating modes.

Table 35.15 Programming methods (1 of 2)

Programming method	Functional overview	Operating mode
Serial programming	A dedicated flash-memory programmer connected through the UART interface can program the on-board flash memory after the device is mounted on the target system.	Serial programming mode
	A dedicated flash-memory programmer connected through the UART interface and a dedicated programming adapter board allow off-board programming of the flash memory, before it is mounted on the target system.	

Table 35.15 Programming methods (2 of 2)

Programming method	Functional overview	Operating mode
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 code flash memory while the data flash memory is programming. As a result, a program resident in code flash memory can program data flash memory.	Normal operating mode
cJTAG programming	A dedicated flash-memory programmer or an on-chip debugger connected through cJTAG can program the on-board flash memory after the device is mounted on the target system.	On-chip debug mode
	A dedicated flash-memory programmer or an on-chip debugger connected through cJTAG and a dedicated programming adapter board allow off-board programming of the flash memory, before it is mounted on the target system.	

Table 35.16 lists the functions of the on-chip flash memory. Use serial programmer commands for serial programming. For self-programming, use the programming commands to read the on-chip flash memory or run the user program.

Table 35.16 Basic functions

Function	Functional overview	Availability	
		Serial programming	Self-programming/ cJTAG programming
Blank check	Checks a specified block to ensure that writing to it has not already proceeded.	Not supported	Supported
Block erasure	Erases the memory contents in the specified block	Supported	Supported
Programming	Writes to the specified address	Supported	Supported
Read	Reads data programmed in the flash memory	Supported	Not supported (read by user program is possible)
ID code protection	Compares the ID code sent by the host with the code stored in the code flash memory. If the two match, the FCB enters the wait state for programming and erasure commands from the host.	Supported	Not supported (ID authentication is not performed)
Protection configuration	Configures the protection function for serial programming	Supported with conditions (only allows switching from enabled to disabled)	Supported with conditions (only allows switching from enabled to disabled)
Protection configuration	Configures the access window for flash area protection in the code flash memory	Supported	Supported

The on-chip flash memory supports the ID code check function. Authentication of ID code check is a protection function for use with serial programming and with cJTAG programming. Table 35.17 lists the protection functions supported by the on-chip flash memory, and Table 35.18 lists the available operations and protection settings.

Table 35.17 Protection functions

Function	Description
ID authentication	The result of ID authentication can be used to control the connection of a serial programmer for serial programming

Table 35.18 Available operations and protection settings

Function	All protection settings and erasure, programming, and read operations		Constraints on the protection setting configuration
	Serial programming and on-chip debug mode	Self-programming mode	
ID authentication	When the ID codes do not match: ● Block erasure commands: not supported ● Programming commands: not supported ● Read commands: not supported ● Protection configuration commands: not supported. When the ID codes match: ● Block erasure commands: supported ● Programming commands: supported ● Read commands: supported ● Protection configuration commands: supported.	● Blank check: supported ● Block erasure: supported ● Programming: supported ● Protection configuration: supported	ID authentication is not performed

35.6.1 Configuration Area Bit Map

The bits used for ID authentication, startup area select, access window protection, and protection configuration functions are mapped in Figure 35.5. The boot program must use these bits as hexadecimal data.

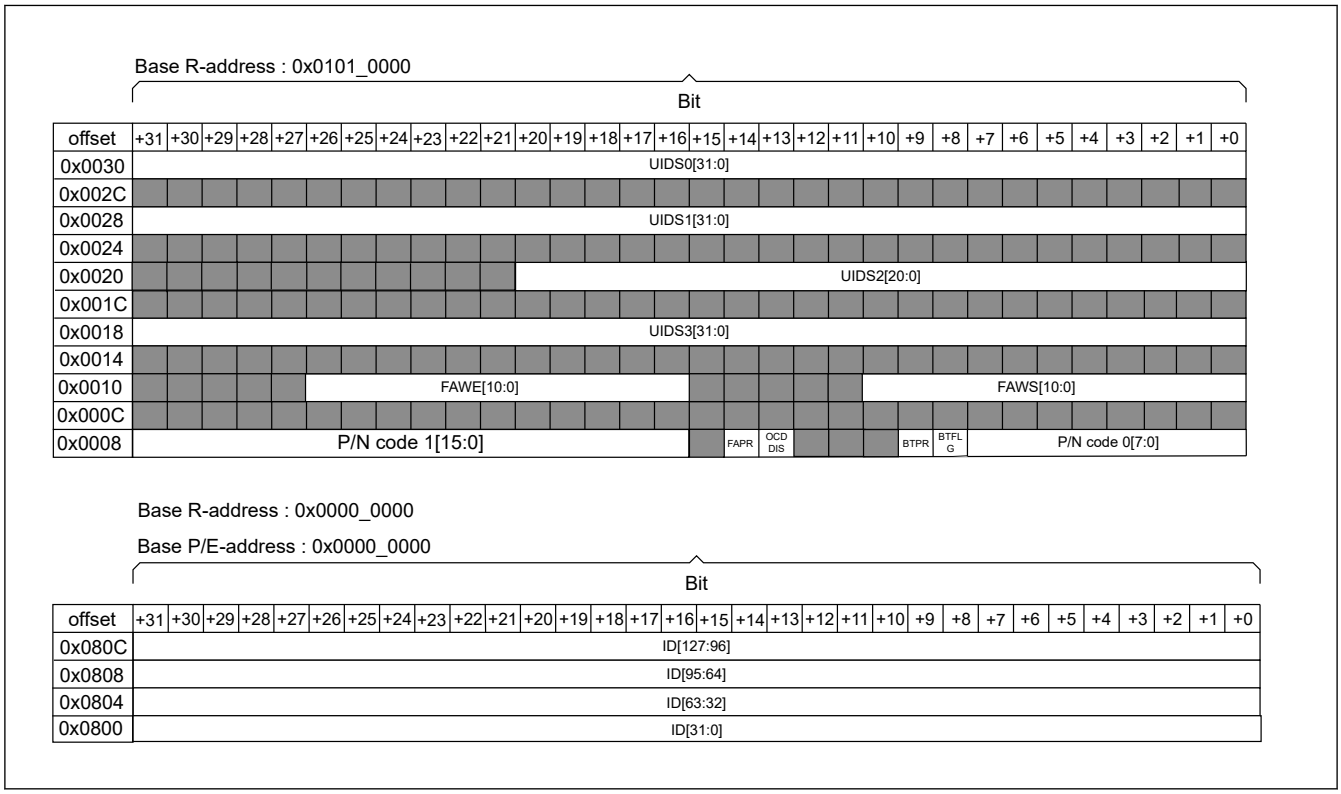
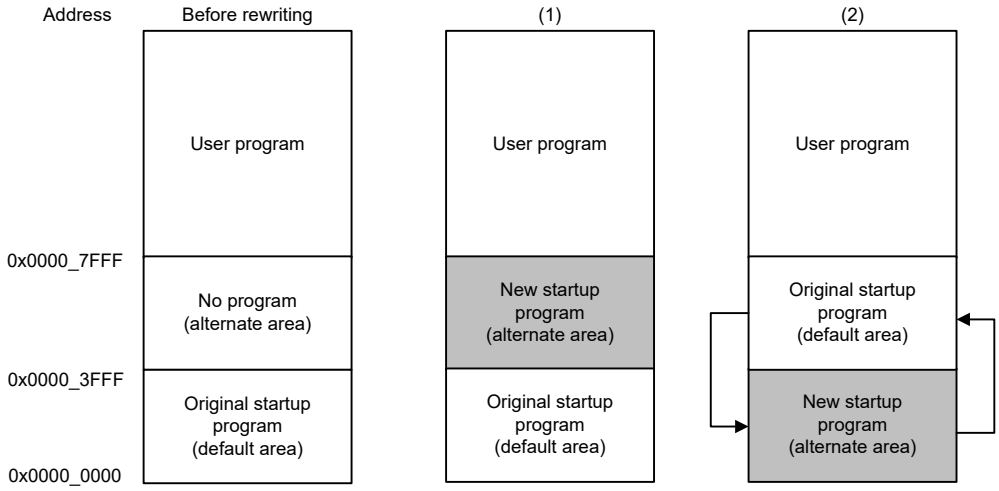


Figure 35.5 Configuration area bit map

35.6.2 Startup Area Select

The startup area select function allows the boot program to be safely updated. The startup area is 16 KB of space located in the user area. The FCB controls the address of the startup area based on the Startup Area Select Flag (BTFLG) that is located in the configuration area which names as SECS register. The startup area can be locked by the BTPR bit.

Figure 35.6 shows an overview of the startup program protection. In this figure, the default area indicates the 16-KB region from the start address and the alternate area indicates the next 16-KB region.



(1) Program a new startup program in the alternate area. If the alternate area fails to be rewritten, the new startup program can be rewritten again after starting up using the default area, because the original startup program is in the default area.
(2) After the alternate area is successfully rewritten, the default area and the alternate area are switched using the self-programming library. After that, the program in the alternate area starts after a reset.

Figure 35.6 Overview of startup program protection

35.6.3 Protection by Access Window

Issuing the program or block erase command to a flash memory area outside of the access window results in the command-locked state. The access window is only valid in the user area of the code flash memory. The access window provides protection in self-programming, serial programming, and on-chip debug modes. Issuing the read command to a flash memory area outside of the access window is protected in serial programming mode. Figure 35.7 shows an overview of flash area protection.

The access window is specified in both the FAWS [10:0] and FAWE [10:0] bits. See section 6.2.4. AWS : Access Window Setting Register. Setting of the FAWE[10:0] and FAWS[10:0] bits in various conditions is described as follows:

- FAWE [10:0] = FAWS [10:0]: The P/E/Read command can execute anywhere in the user area of the code flash memory
- FAWE [10:0] > FAWS [10:0]: The P/E/Read command can only execute in the window from the block pointed to by the FAWS bits to one block lower than the block pointed to by the FAWE[10:0] bits
- FAWE [10:0] < FAWS [10:0]: The P/E/Read command cannot execute anywhere in the user area of the code flash memory.

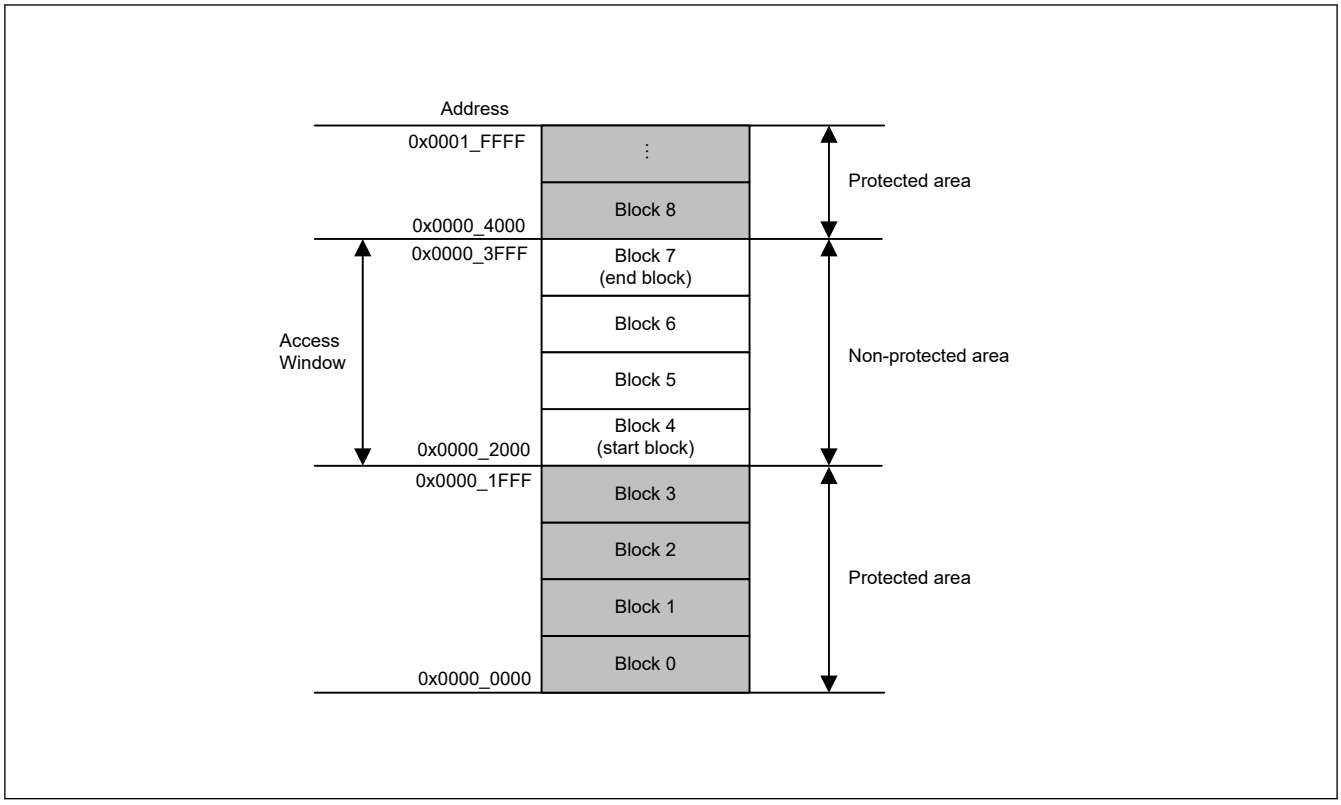


Figure 35.7 Flash area protection overview

35.7 Programming Commands

The FCB controls the programming commands.

35.8 Suspend Operation

The forced stop command forces the blank check command, the block erase command, or the chip erase command to stop. When a forced stop is executed, the stopped address values are stored in the registers. The command can restart from the stopped address after a reset to the registers for command execution by copying the saved addresses.

If a forced stop is executed during chip erase command, execute the chip erase command again and then restart.

35.9 Protection

The types of protection provided include:

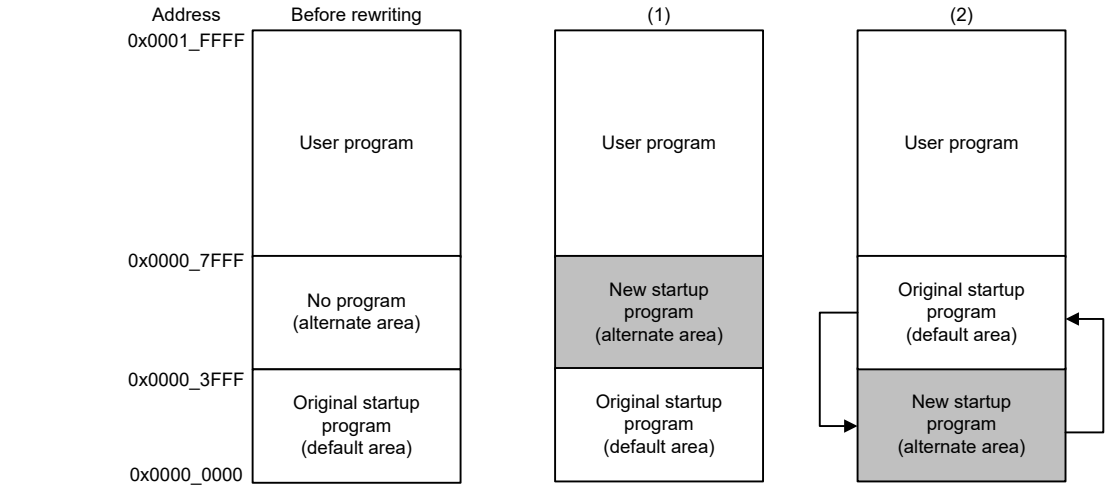
- Software protection
- Error protection
- Boot program protection
- User ID read protection

35.9.1 Startup Program Protection

When programming of the startup area is interrupted by temporary blackout, the startup program may not be successfully programmed and the user program may not start properly.

This problem can be avoided by programming the startup program without erasing the existing startup program using the startup program protection.

Figure 35.8 shows an overview of the Startup Program Protection. In this figure, the default area indicates the 16-KB region from the start address and the alternate area indicates the next 16-KB region.



Note: (1) Program a new startup program in the alternate area. If the alternate area fails to be rewritten, the new startup program can be rewritten again after starting up using the default area because the original startup program is in the default area.
(2) After the alternate area is successfully rewritten, the default area and the alternate area are switched using the self-programming library. After that, the program in the alternate area starts after a reset.

Figure 35.8 Overview of the startup program protection

35.9.2 Area Protection

Area protection enables rewriting for only selected blocks (access window) in the user area and disables programming for the other blocks. Data flash is not protected by the access window.

Select the start block and end block to set the access window. The access window is changeable and valid in programming mode (boot mode, self-programming mode, and OCD mode).

Figure 35.9 shows an overview of area protection.

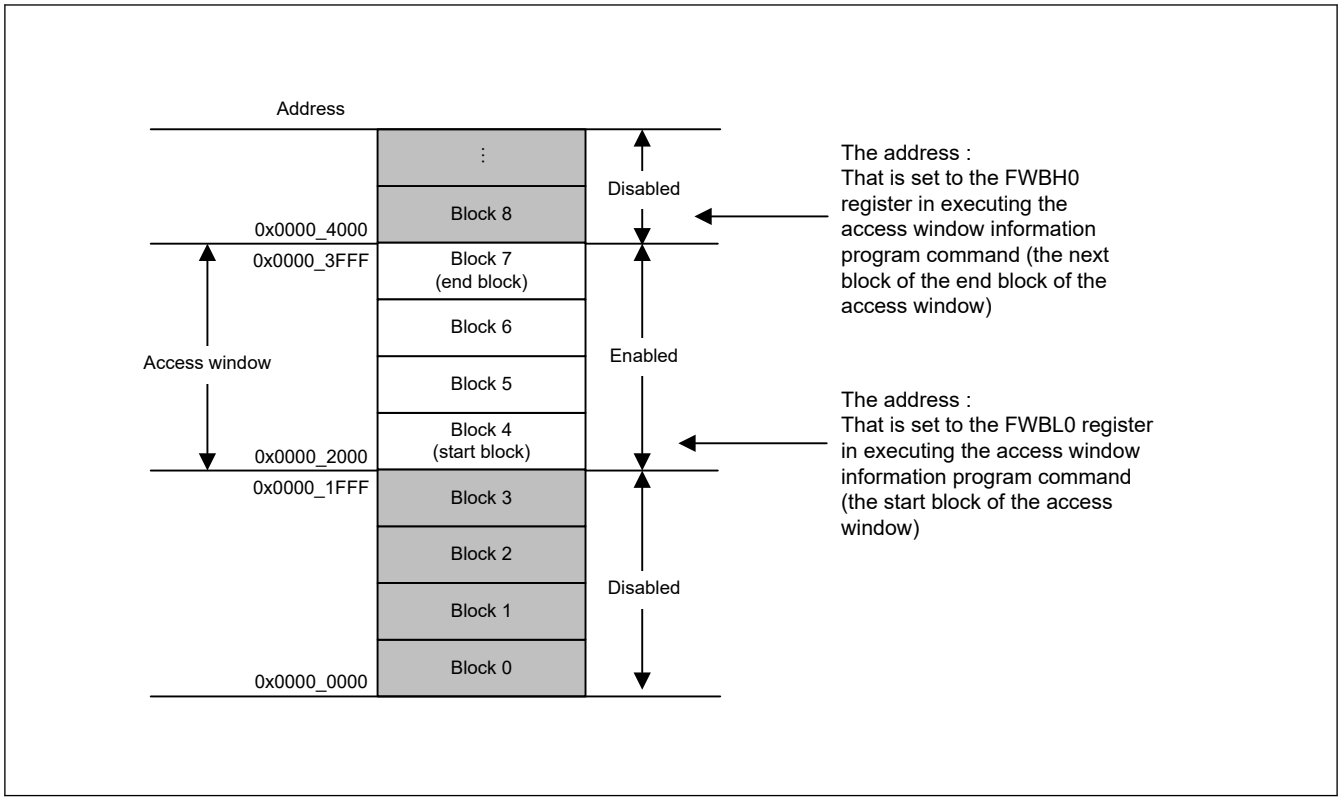


Figure 35.9 Area Protection Overview

35.9.3 User ID Read Protection

The user can limit the individuals for which the software can run using the user ID function. This function is available by entering a desired 64-bit value in the extra area. The user ID can be set by the user ID information program command. The user ID can only be read from the register when the reading condition is satisfied. Figure 35.10 shows a block diagram of the user ID function.

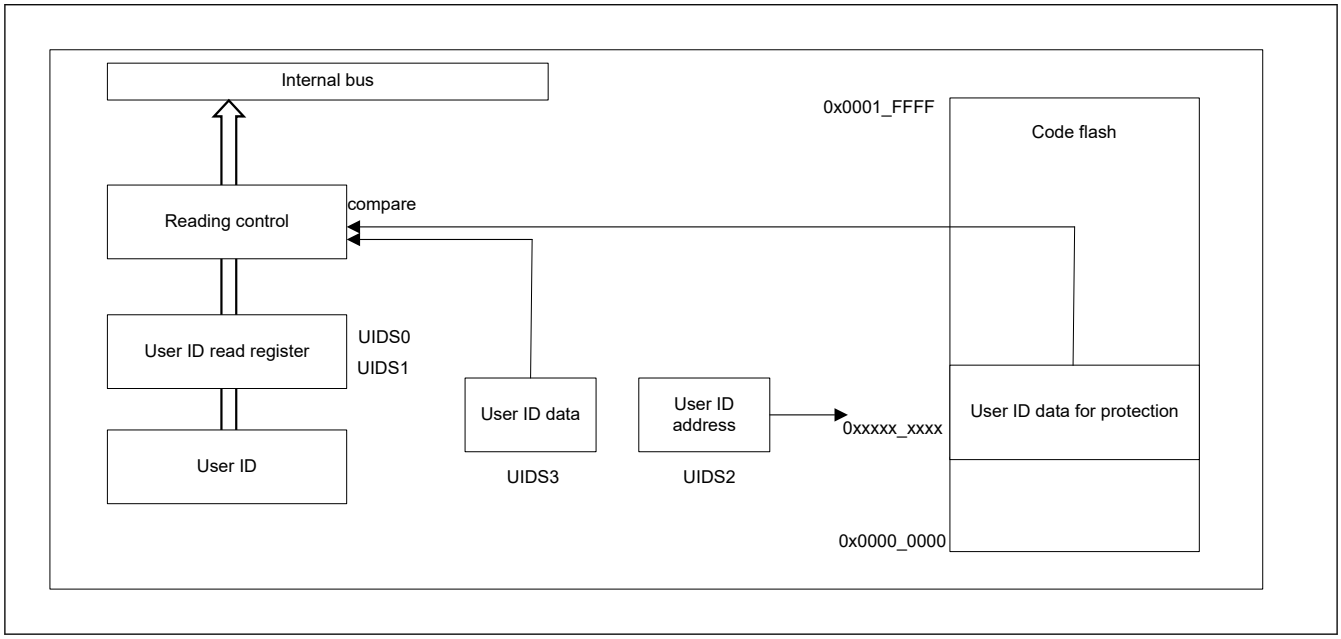


Figure 35.10 Block diagram of the user ID function

The user ID is stored in the user ID read registers by the reset sequence. After that, the user ID data (UIDS3) and user ID data for protection are compared. If they match, reading from the user ID read registers (UIDS0 and UIDS1) is enabled. If

not, reading from the user ID read registers is disabled, and the value is always 0x0000_0000. For more information, see section 6.2.6. UIDSn : User ID Setting Register n (n = 0 to 3).

35.10 Serial Programming Mode

The serial programming mode includes:

- Boot mode with SAU_2

Table 35.19 lists the I/O pins of the flash memory-related modules.

Table 35.19 I/O pins of flash memory-related modules

Pin name	I/O	Applicable modes	Function
MD	Input	UART boot mode (serial programming mode)	Selection of operating mode
P303/RXD0	Input	UART boot mode	For host communication, to receive data through the UART
P302/TXD	Output		For host communication, to transmit data through the UART

35.10.1 UART (SAU) Boot Mode

In boot mode, the host sends control commands and data for programming, and the code flash memory and data flash memory areas are programmed or erased accordingly. An on-chip UART handles transfers between the host and the MCU in asynchronous mode. Tools for transmission of control commands and the data for programming must be prepared in the host.

When the MCU is activated in boot mode, the embedded program for serial programming is executed. This program automatically adjusts the bit rate of the UART and controls programming and erasure by receiving control commands from the host.

Figure 35.11 shows the system configuration for operations in boot mode.

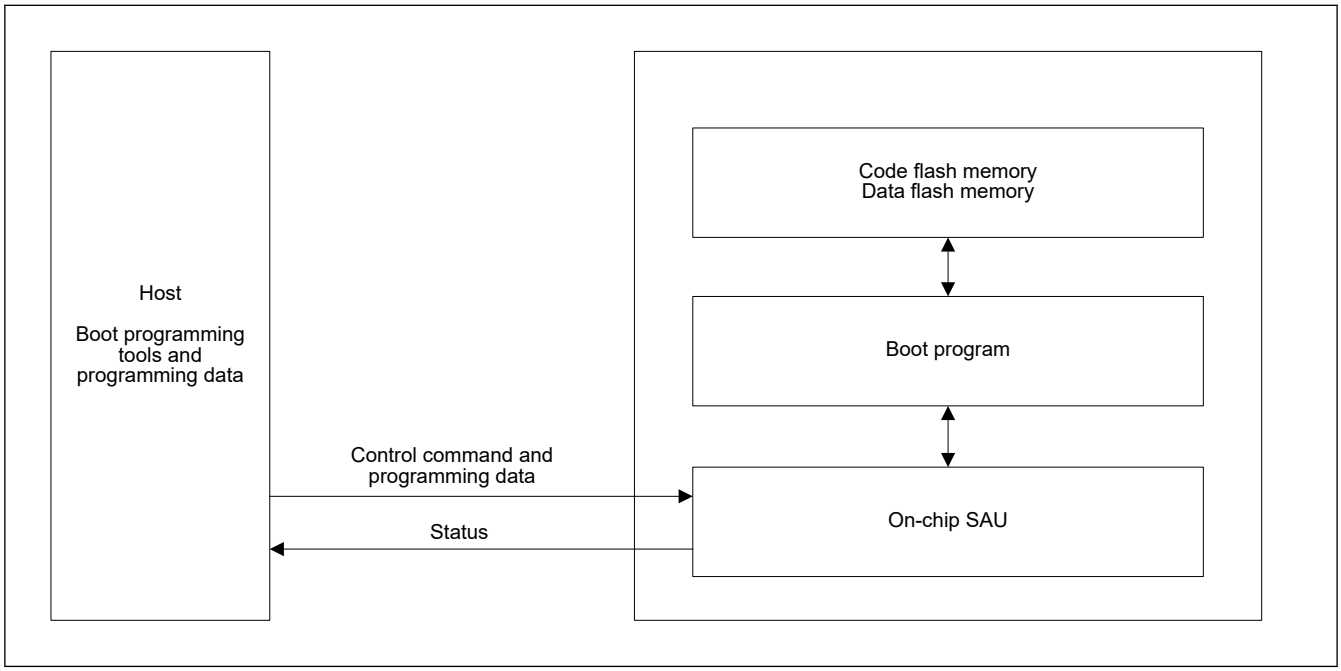


Figure 35.11 System configuration in UART (SAU) boot mode

35.11 Using a Serial Programmer

A dedicated flash memory programmer can be used to program the flash memory in serial programming mode.

35.11.1 Serial Programming

The MCU is mounted on the system board for serial programming. A connector to the board allows programming by the flash memory programmer.

Figure 35.12 shows the environment recommended by Renesas for programming the flash memory of the MCU with data.

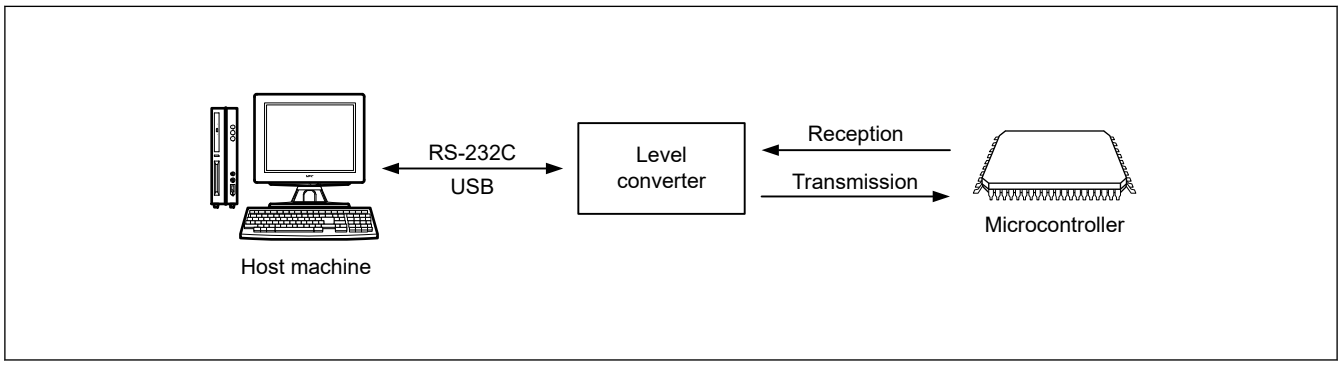


Figure 35.12 Environment for writing programs to the flash memory

35.12 Self-Programming

35.12.1 Overview

The MCU supports programming of the flash memory by the user program. The programming commands can be used with user programs for writing to the code and data flash memory. This enables updates to the user programs and overwriting of constant data fields.

The background operation facility makes it possible to execute a program from the code flash memory to program the data flash memory under the conditions shown in Figure 35.13. This program can also be copied in advance to and executed from the internal SRAM. When executing from the internal SRAM, this program can also program the code flash memory area.

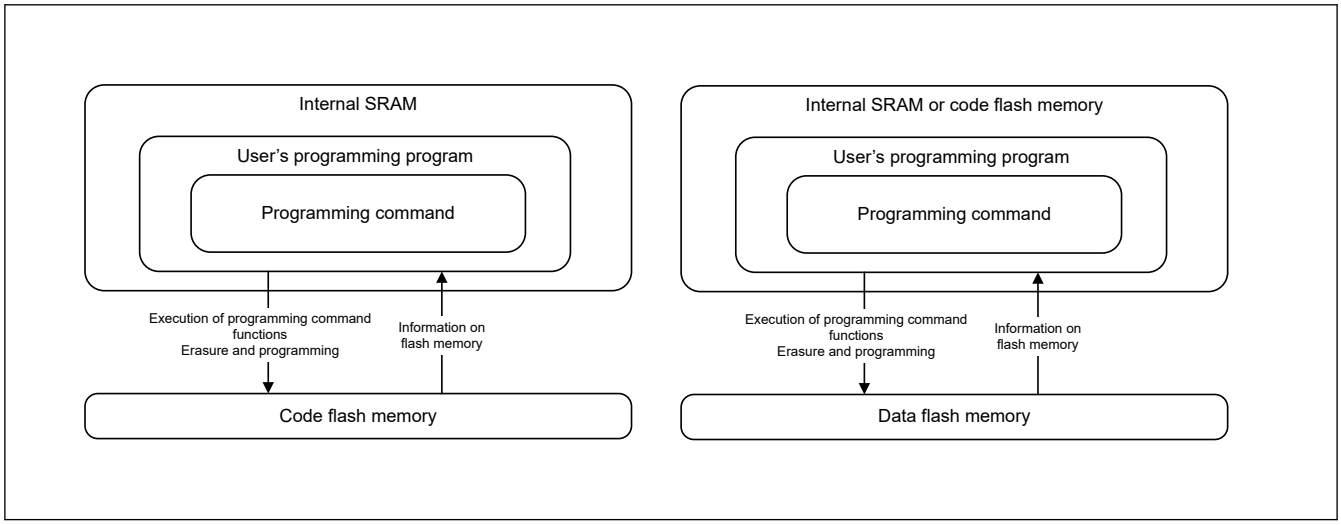


Figure 35.13 Schematic view of self-programming

35.12.2 Background Operation

Background operation can be used when a combination of the flash memory for writing and reading is as listed in Table 35.20.

Table 35.20 Conditions under which background operation is available

Product	Writable range	Readable range
All products	Data flash memory	Code flash memory

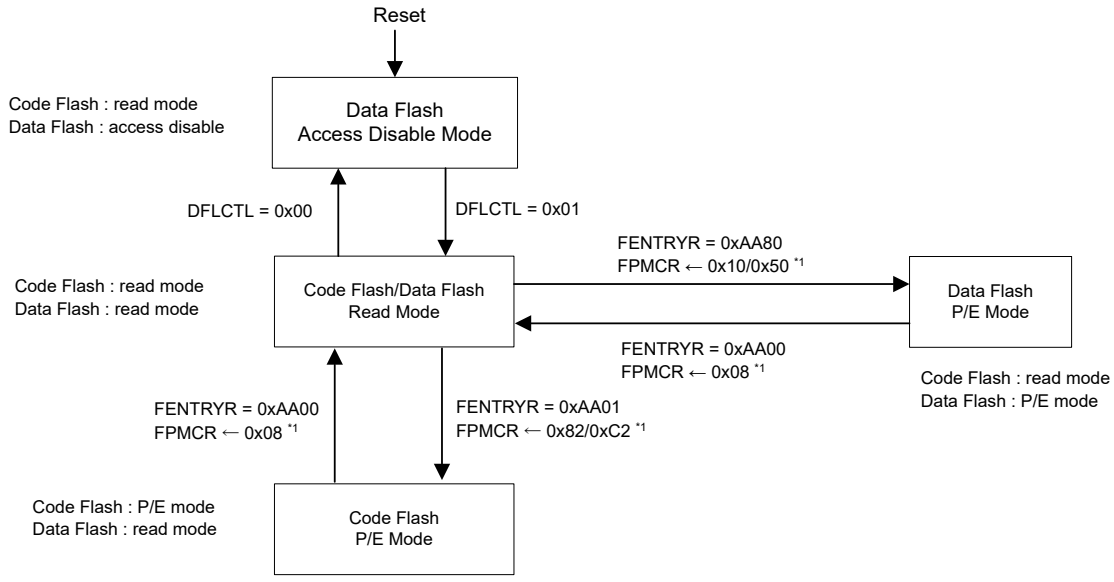
35.13 Programming and Erasure

The code flash and data flash can be programmed and erased by changing the mode of the dedicated sequencer for programming and erasure, and by issuing commands for programming and erasure.

The mode transitions and commands required to program or erase the code flash and data flash are described in the sections that follow. The descriptions apply in common to boot mode and single-chip mode.

35.13.1 Sequencer Modes

The sequencer has four modes and transitions between modes occur by writing to the FENTRYR or DFLCTL register, or by issuing commands to set the FPMCR register. Figure 35.14 shows mode transitions of the flash memory.



Note 1. Refer to each flow in details.

Figure 35.14 Mode transitions of the flash memory

35.13.1.1 Data Flash Access Disable Mode

Data flash access disable mode is to disable access to the data flash. Issuing a reset causes this mode. The data flash transitions to read mode by setting the DFLCTL.DFLEN bit to 1.

35.13.1.2 Read Mode

Read mode is used for high-speed reading of the code flash and data flash.

(1) Code Flash and Data Flash Read Mode

This mode is used for reading the code flash and data flash. The sequencer enters this mode when the FENTRYR.FENTRY0 bit is set to 0 while the FENTRYR.FENTRYD bit set to 0.

35.13.1.3 P/E Modes

(1) Code Flash P/E Mode

The code flash P/E mode is used for programming and erasure of the code flash. The sequencer enters this mode when the FENTRYR.FENTRYD bit is set to 0 while the FENTRYR.FENTRY0 bit set to 1. In this mode, it is not possible to access the data flash.

(2) Data Flash P/E Mode

The data flash P/E mode is used for programming and erasure of the data flash. High-speed reading from the code flash is possible. The sequencer enters this mode when the FENTRYR.FENTRY0 bit is set to 0 while the FENTRYR.FENTRYD bit is set to 1.

35.13.2 Software Commands

Software commands consist of commands for programming and erasure, and commands for programming startup program area information and access window information. Table 35.21 lists the software commands for use with the flash memory.

Table 35.21 Software Commands

Command	Function
Program	Code flash programming (8 bytes) Data flash programming (1 byte)
Block erase	Code flash/data flash erasure
Chip erase	Code flash/data flash erasure
Blank check	Check whether the specified area is blank. Confirm that data is not programmed in the area. This command does not guarantee whether the area remains erased.
Startup area information and protection program	Set the FAPR, OCDDIS, BTPR, or BTFLG to the extra area
Access window information program	Set the access window used for area protection to the extra area
User ID program	Set the User ID to the extra area

35.13.3 Software Command Usage

The following sections describe the usage of each software command.

(1) Switching from Data Flash Access Disable Mode to Read Mode

It is necessary to enter the code flash/data flash read mode from the data flash access disable mode. Figure 35.15 shows the procedure for entering the code flash/data flash read mode from the data flash access disable mode.

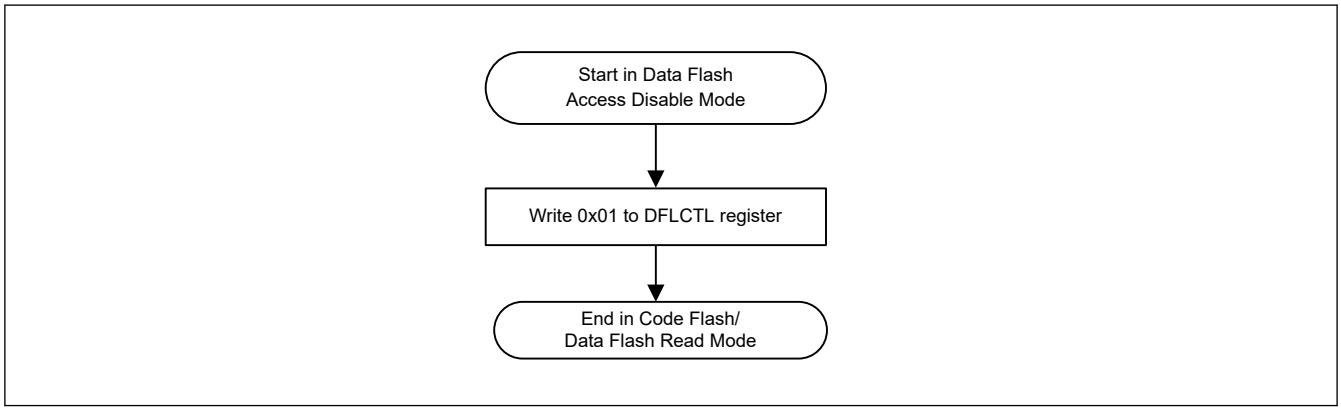
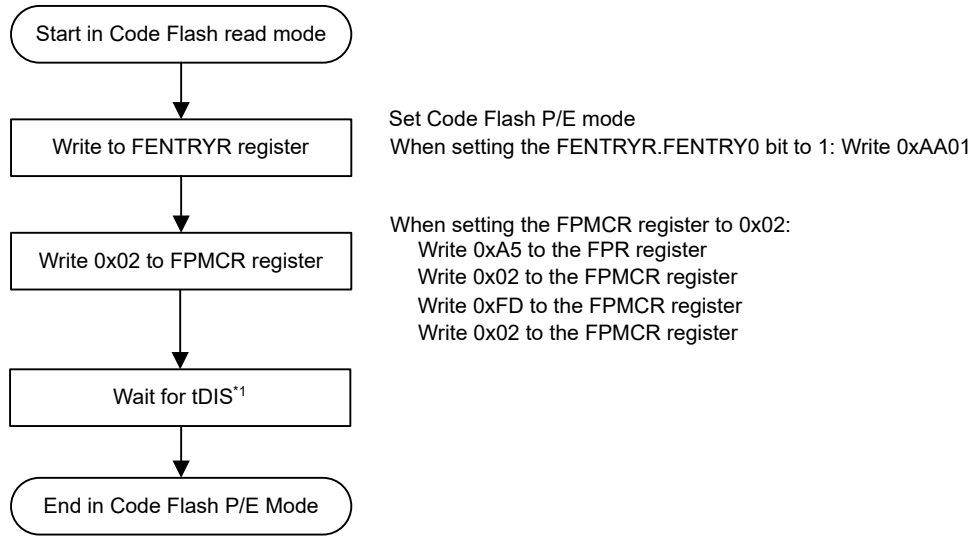


Figure 35.15 Mode transitions to read mode from data flash access disable mode

(2) Switching to Code Flash P/E Mode

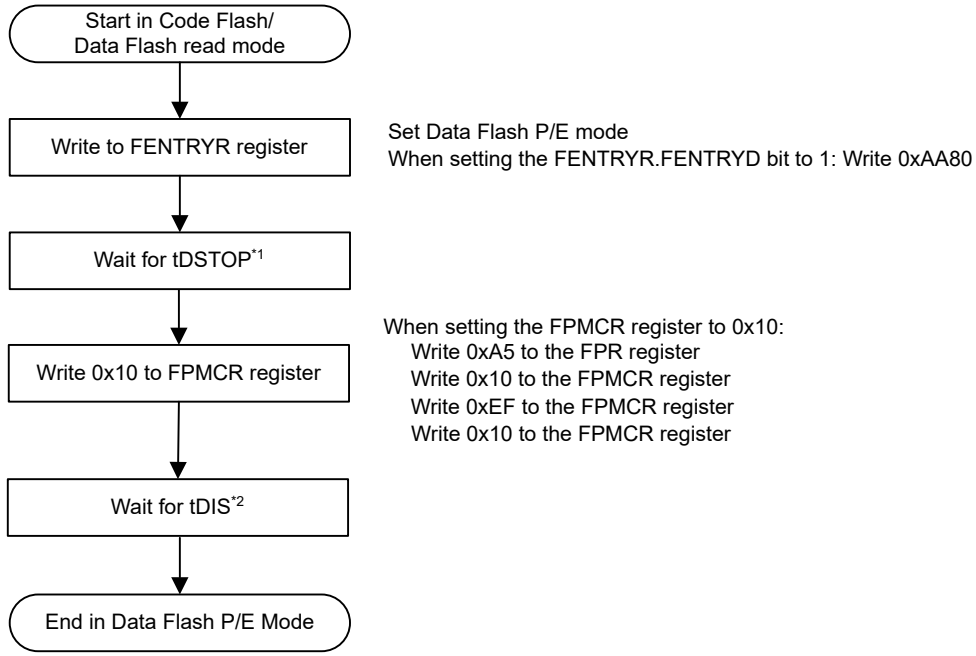
It is necessary to enter the code flash P/E mode by setting the FENTRY0 bit of the FENTRYR register before executing the software command for the code flash. Figure 35.16 shows the procedure for entering code flash P/E Mode.



Note 1. t_{DIS}: Flash memory mode transition wait time 1 (See section 37, Electrical Characteristics.)

Figure 35.16 Procedure for changing from read mode to code flash P/E mode

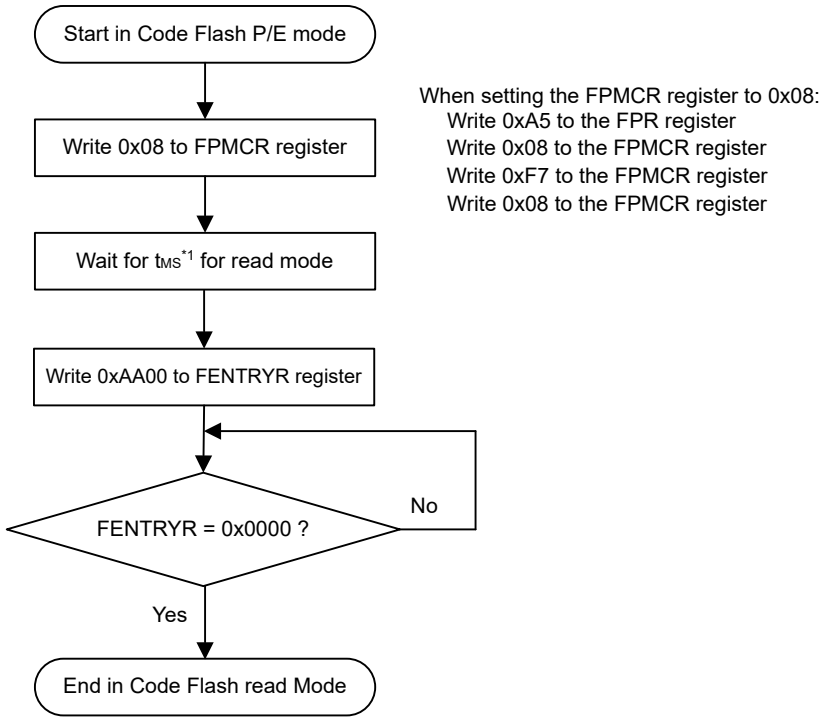
It is necessary to enter the data flash P/E mode by setting the FENTRYD bit of the FENTRYR register before executing the software command for the data flash. Figure 35.17 shows the procedure for entering to the data flash P/E Mode.



Note 1. t_{DSTOP}: STOP recovery time (See section 37, Electrical Characteristics.)
Note 2. t_{DIS}: Flash memory mode transition wait time 1 (See section 37, Electrical Characteristics.)

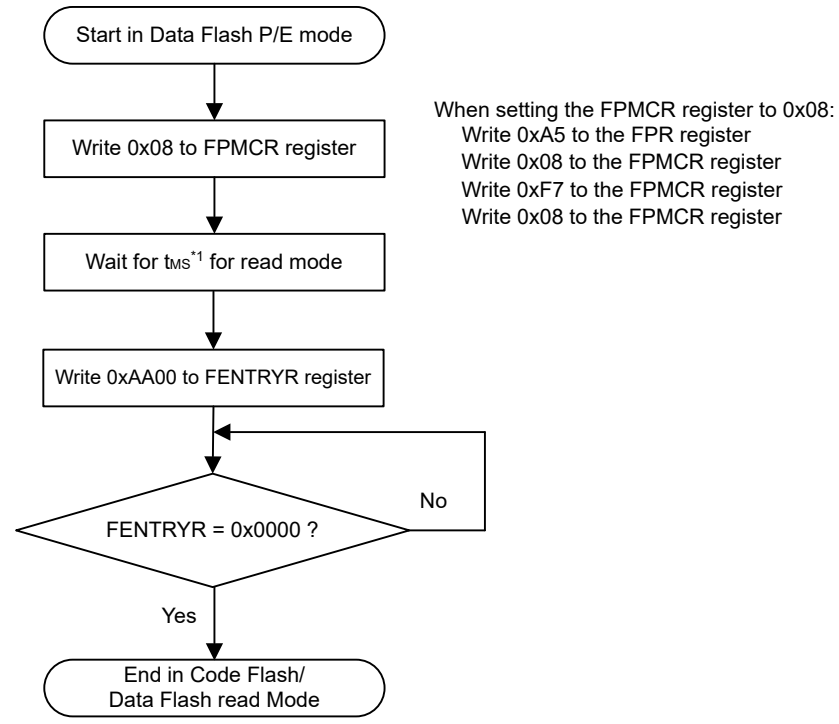
Figure 35.17 Procedure for changing from read mode to data flash P/E mode

(3) Switching the Code Flash or Data Flash P/E Mode to Read Mode



Note 1. t_{MS}: Flash memory mode transition wait time 2 (See section 37, Electrical Characteristics.)

Figure 35.18 Procedure for changing from code flash P/E mode to read mode



Note 1. t_{MS}: Flash memory mode transition wait time 2 (See section 37, Electrical Characteristics.)

Figure 35.19 Procedure for changing from data flash P/E mode to read mode

(4) Flowchart for programming the code flash or the data flash

The following figures describe the flow for programming the code flash or the data flash.

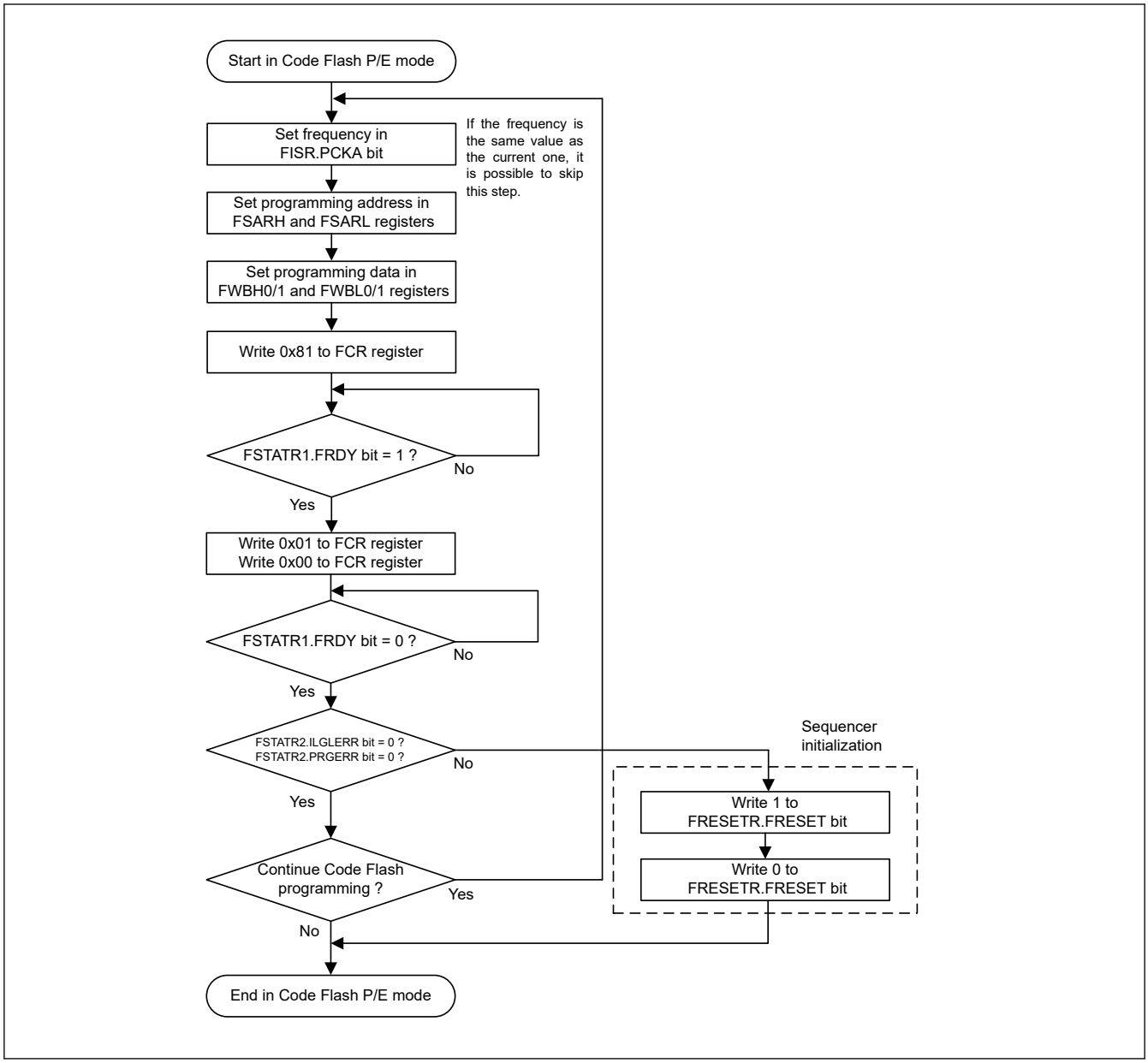


Figure 35.20 Flowchart for programming of the code flash

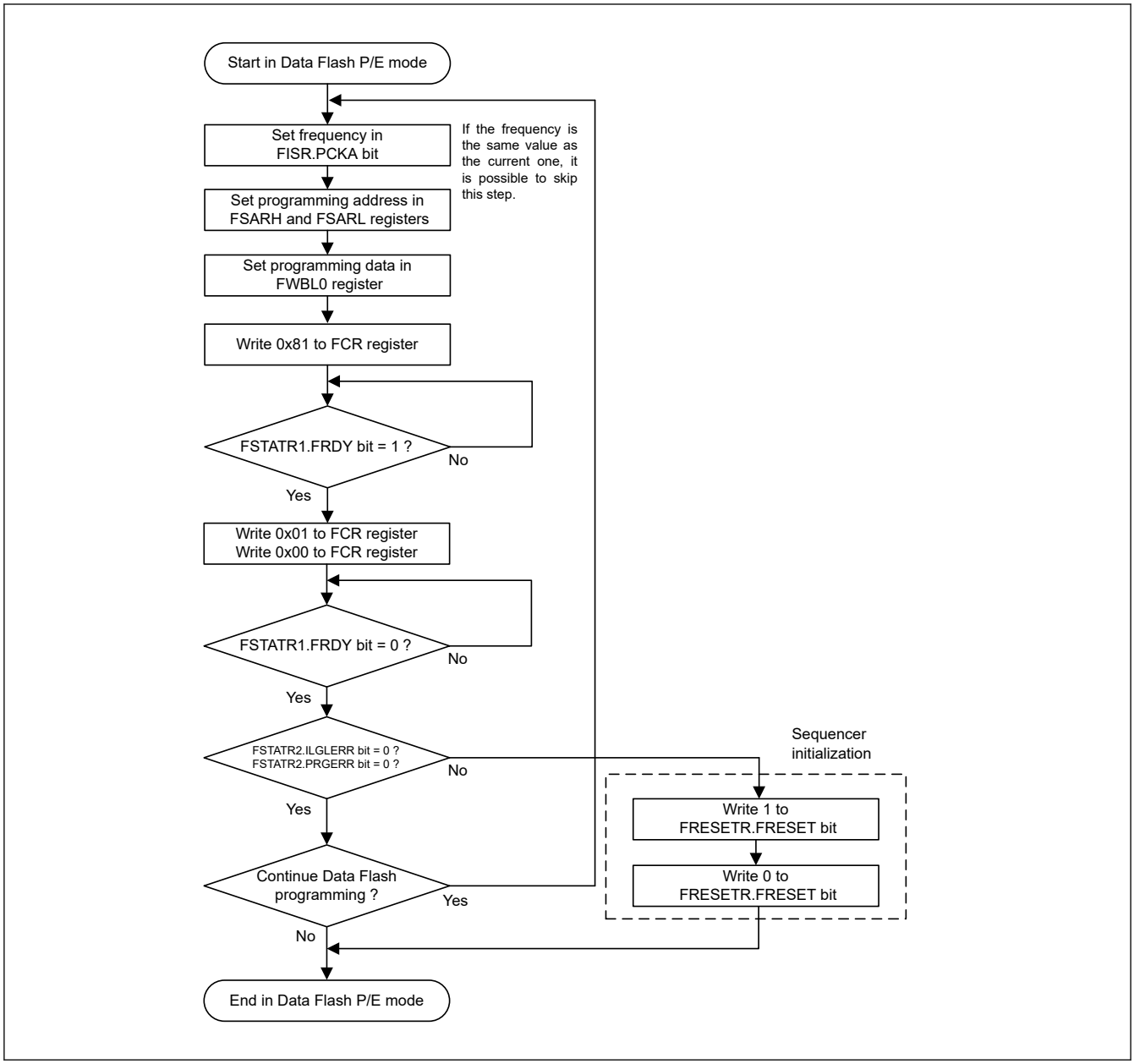


Figure 35.21 Flowchart for programming of the data flash

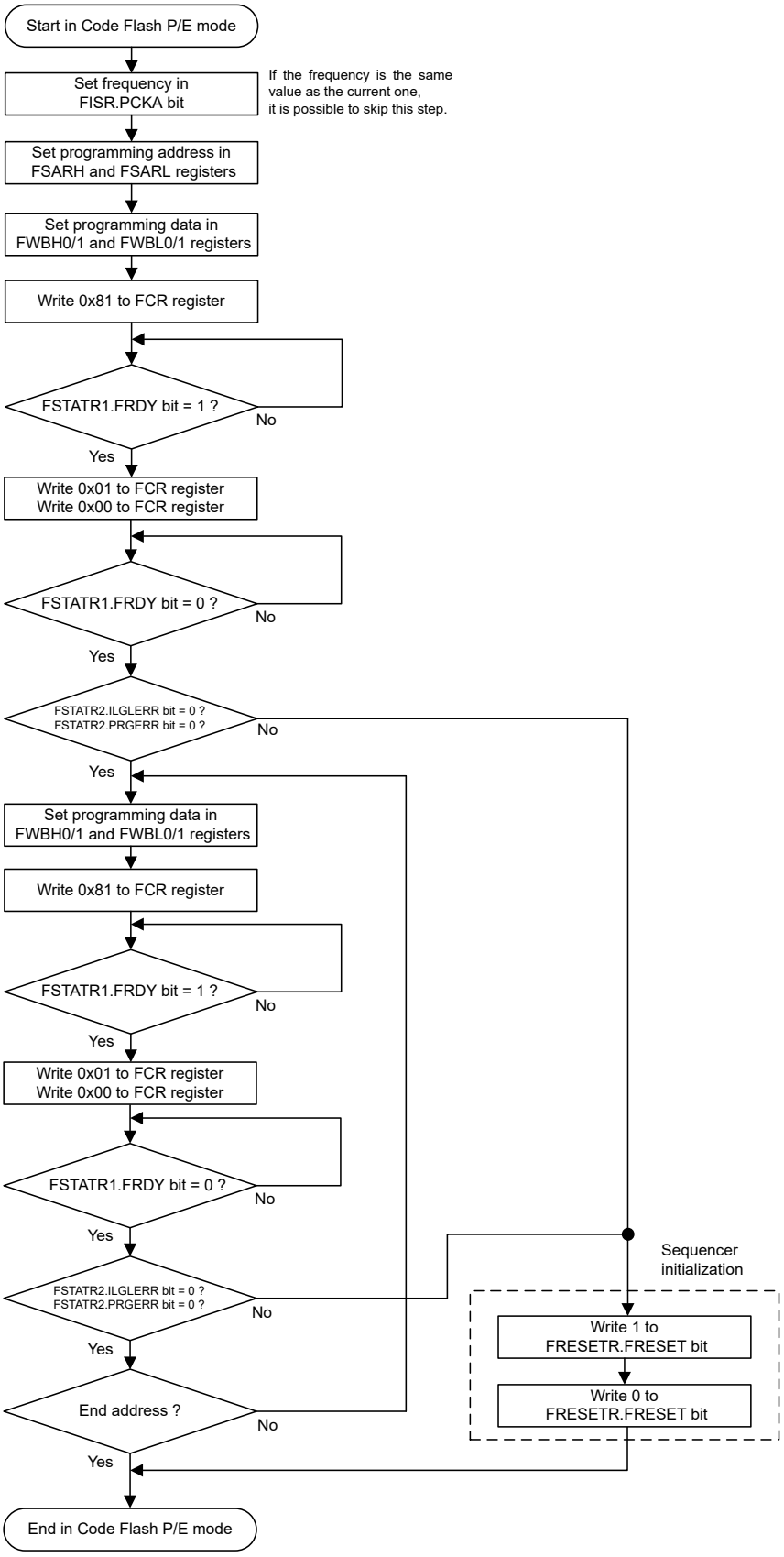


Figure 35.22 Flowchart for consecutive programming of the code flash

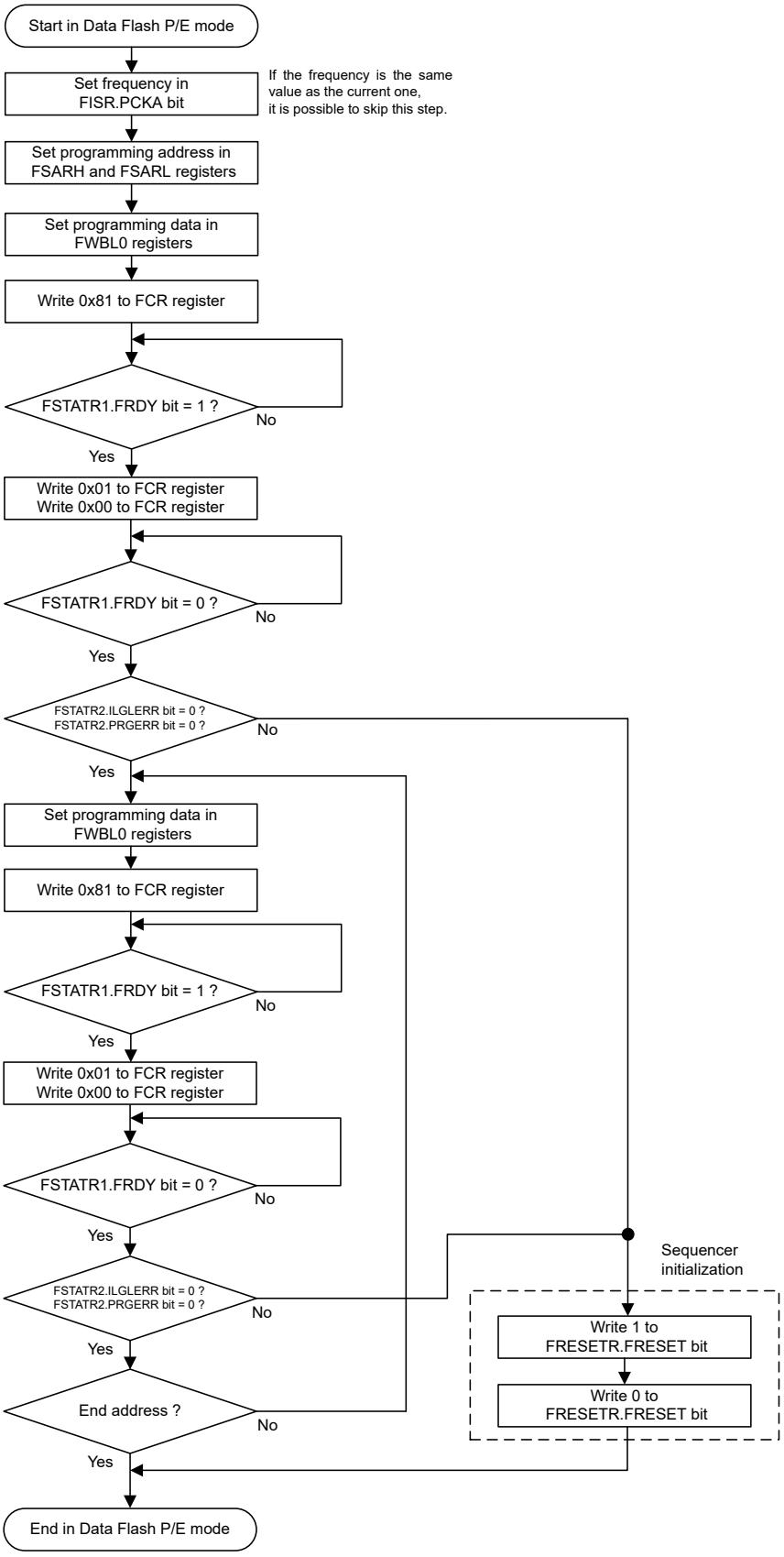


Figure 35.23 Flowchart for consecutive programming of the data flash

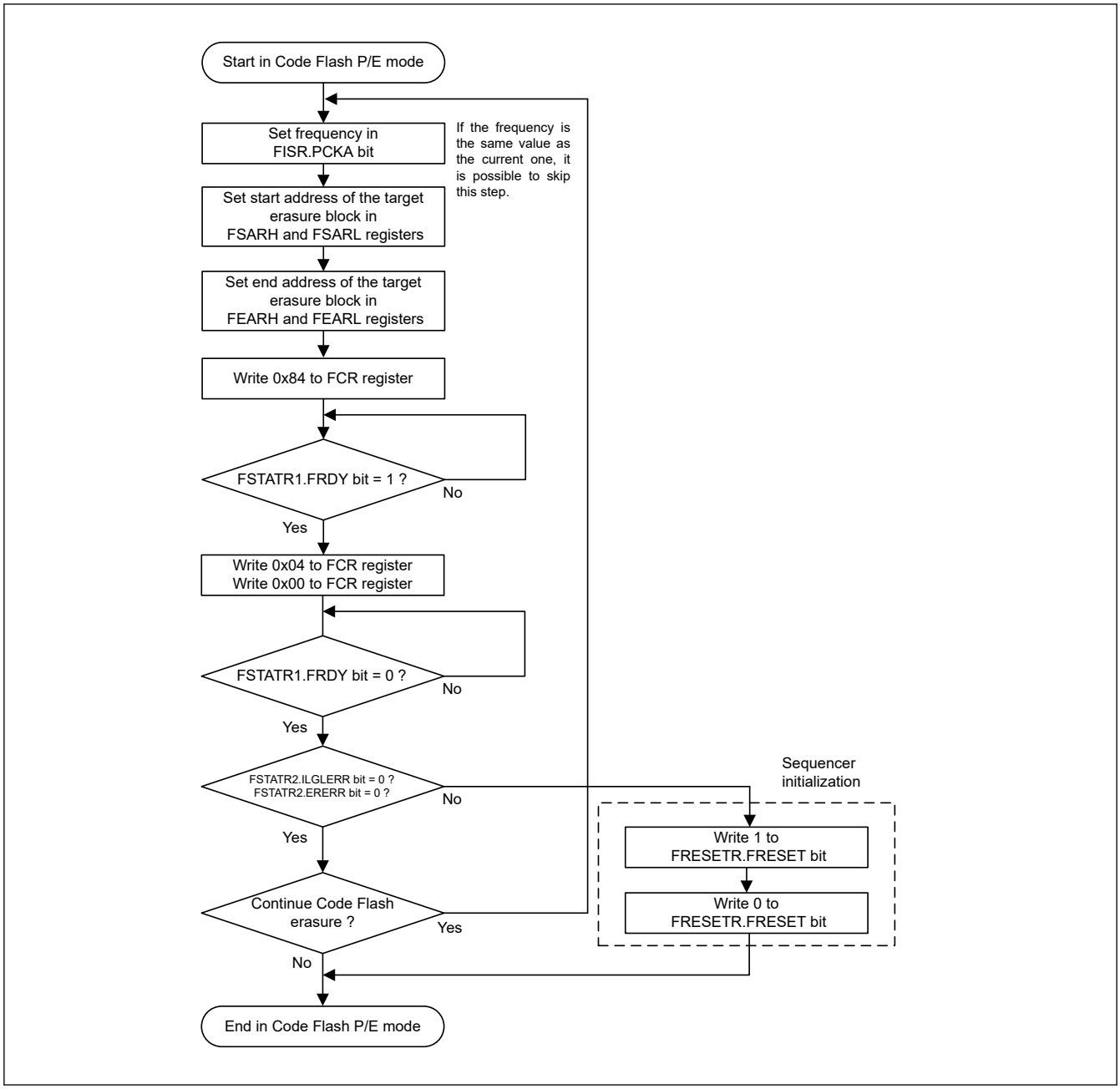


Figure 35.24 Flowchart for the code flash block erase procedure

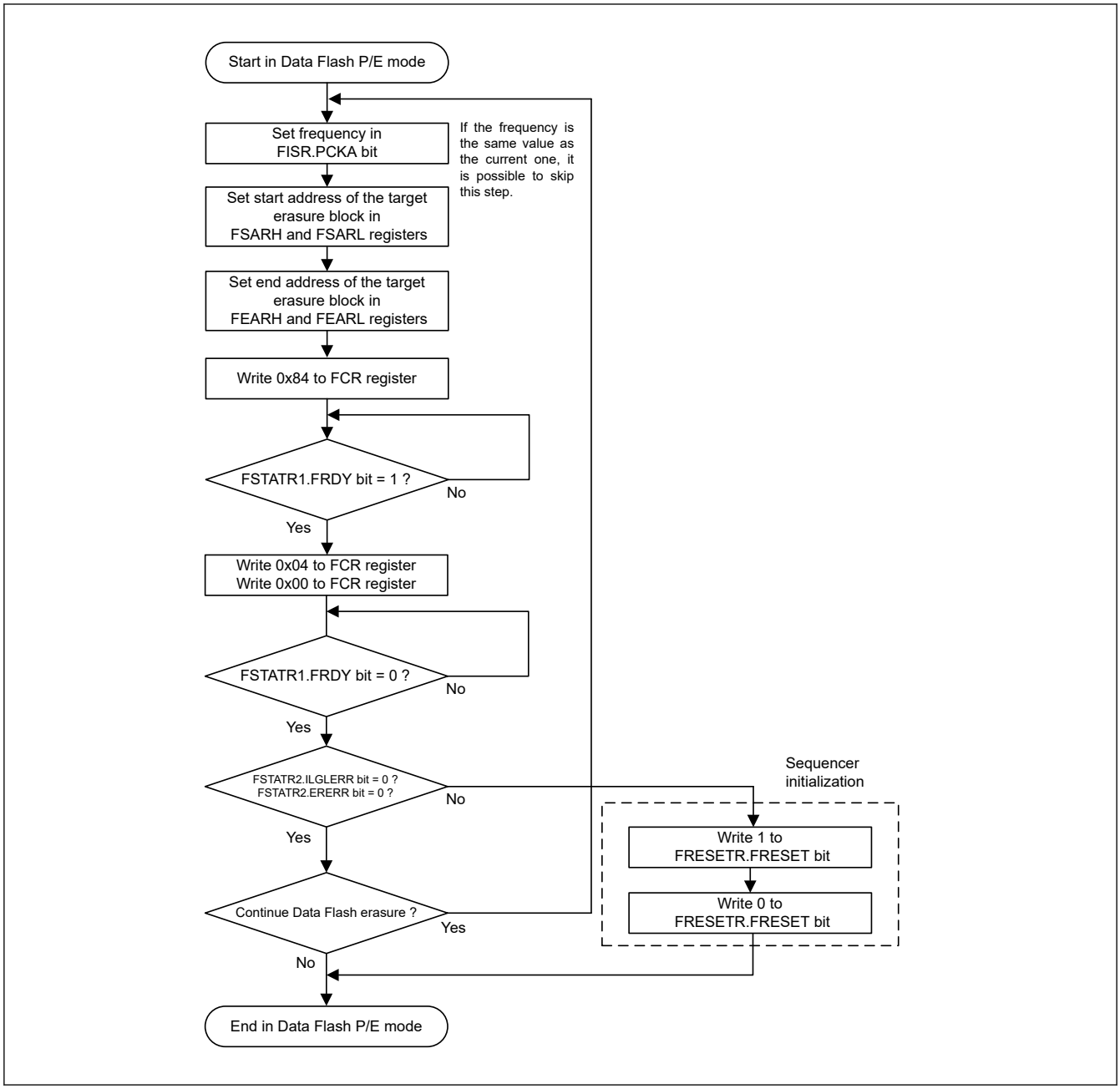


Figure 35.25 Flowchart for the data flash block erase procedure

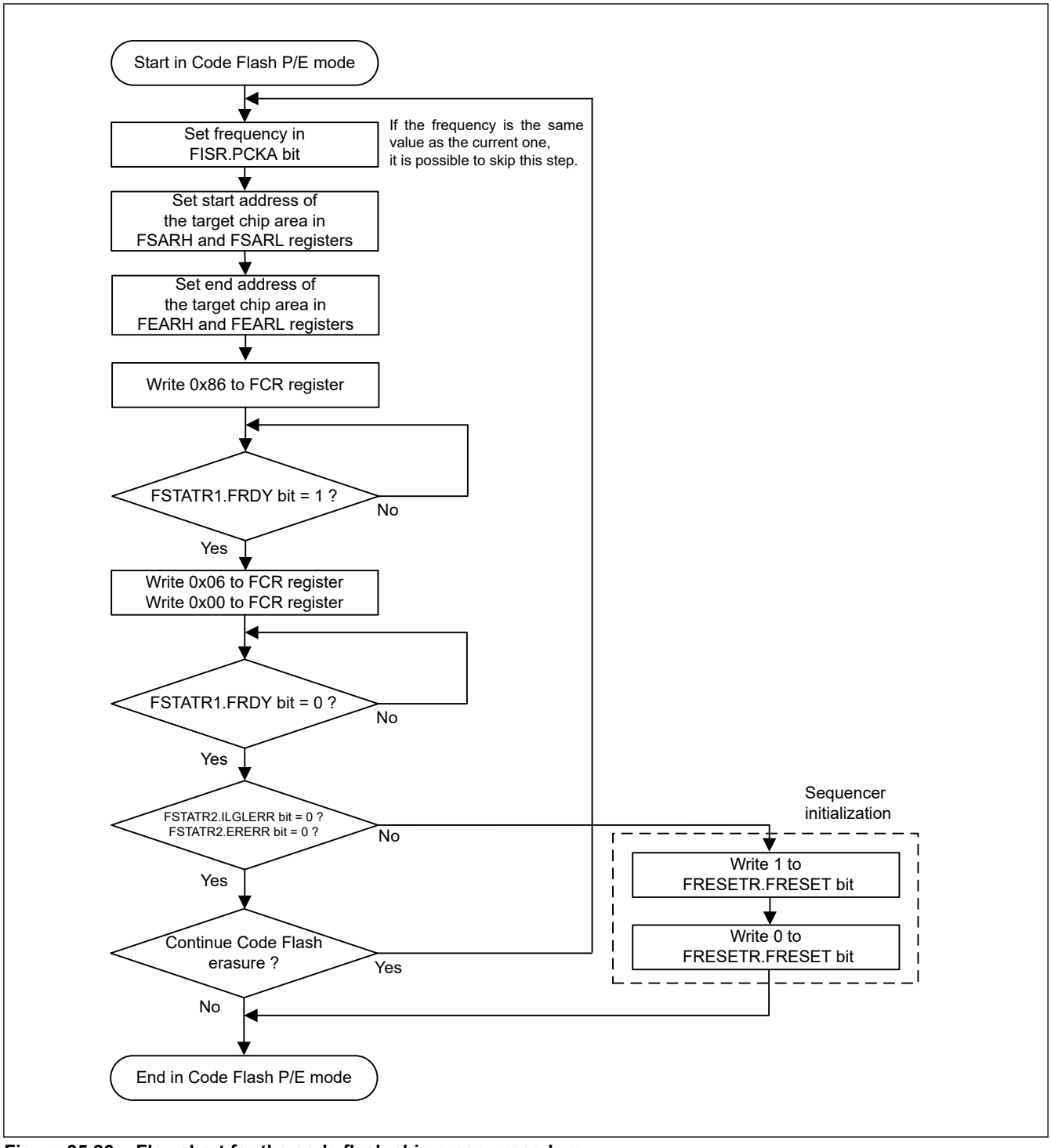


Figure 35.26 Flowchart for the code flash chip erase procedure

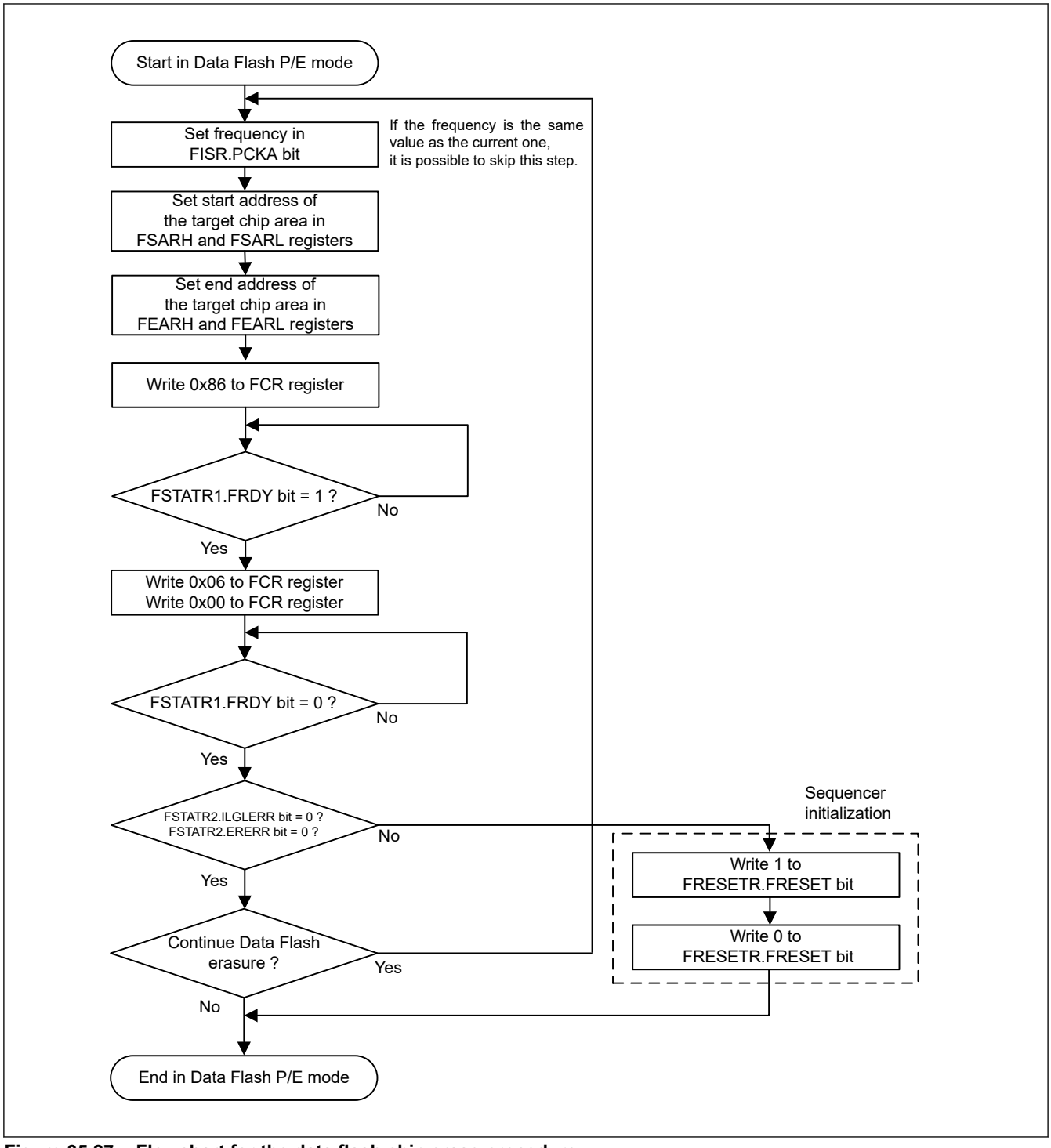


Figure 35.27 Flowchart for the data flash chip erase procedure

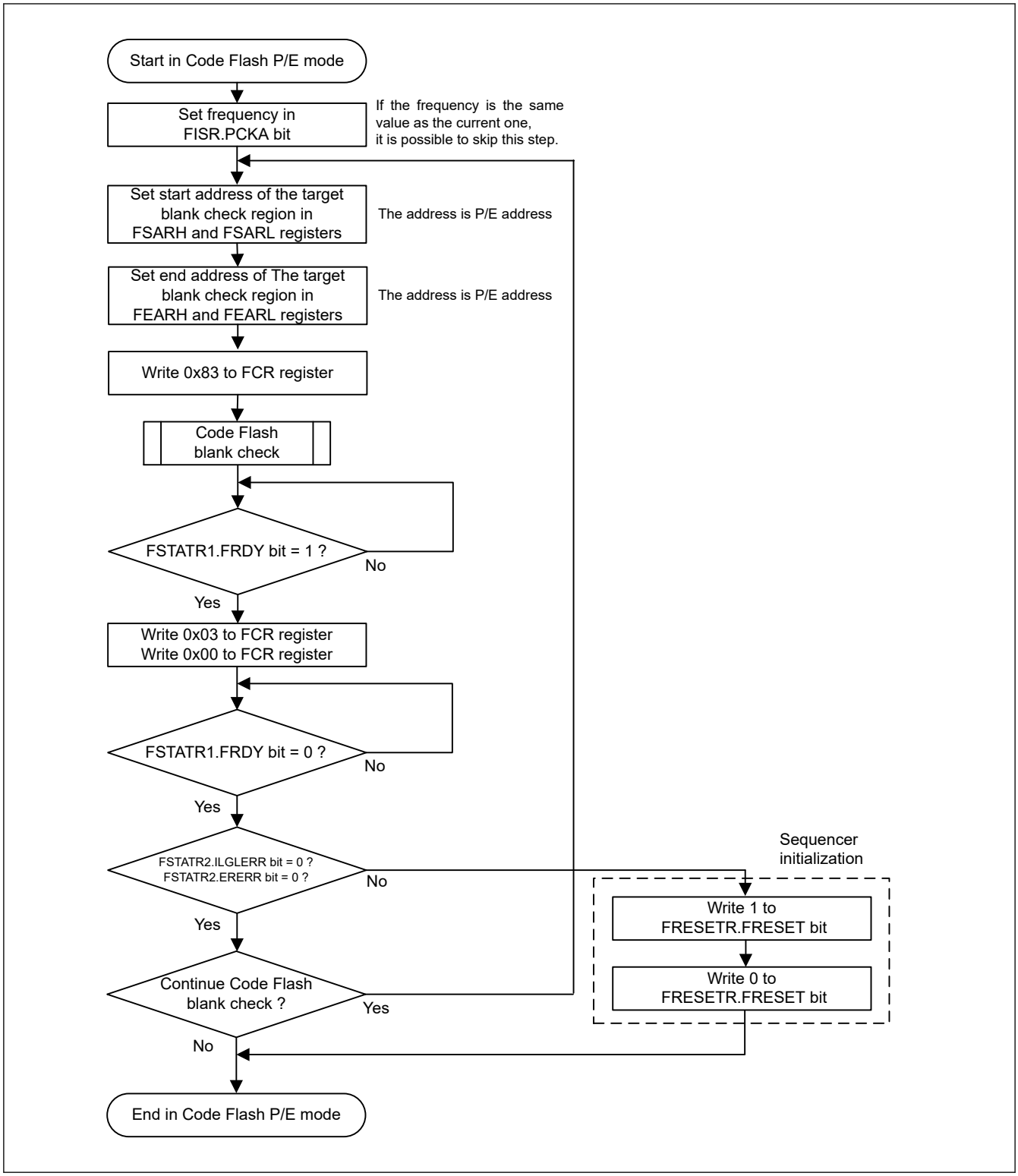


Figure 35.28 Flowchart for the code flash blank check procedure

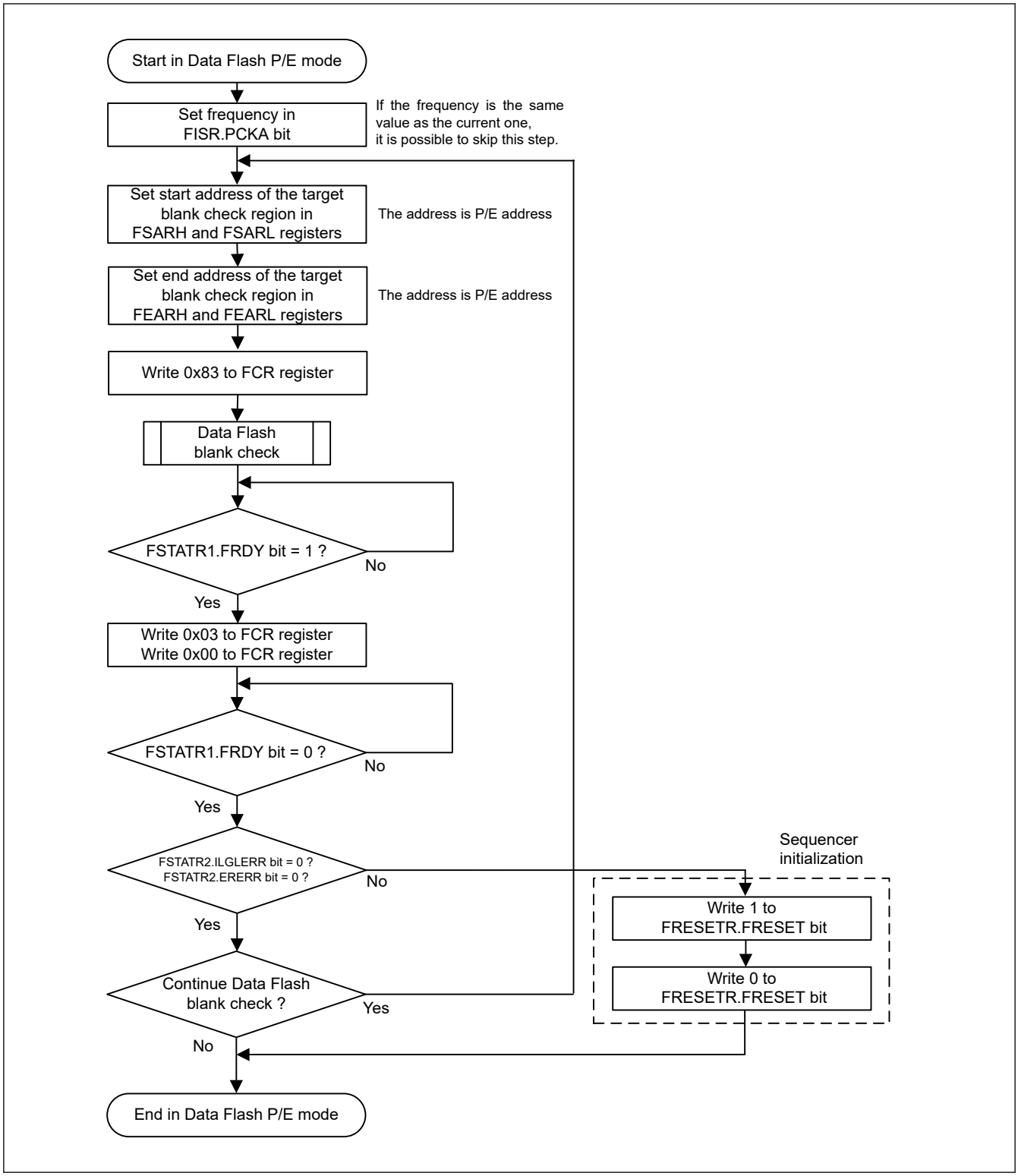
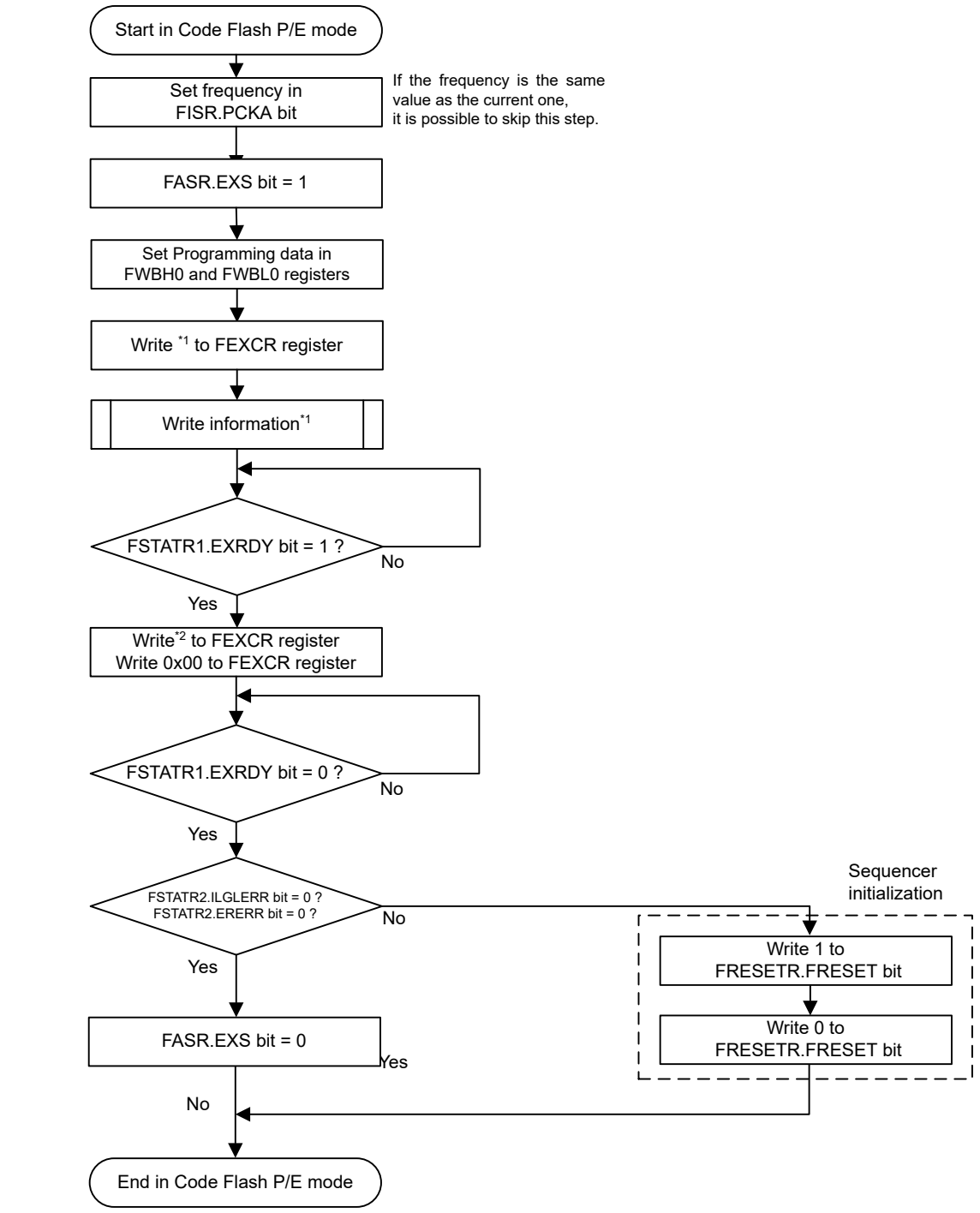


Figure 35.29 Flowchart for the data flash blank check procedure

(5) Startup Area Information and FSPR Program/Access Window Information Program/User ID Program

Figure 35.30 is a simple flowchart of the procedure for the access window information program/User ID program.



Note: Order of the FAPR bit setting by protection setting program
Note 1. Write data :
0x81 : Protection setting
0x82 : start-up area information and FSPR bit access window information
0x83 : User ID1 information
0x84 : User ID2 information
0x85 : User ID3 information
0x86 : User ID4 information
Note 2. 0x0y (y = 1 to 6)

Figure 35.30 Simple flowchart for the procedure for Startup Area Information and FSPR Program/Access Window Information Program/User ID Information Program/Protection Setting Program

The order of the FAPR bit setting by the protection setting program sets the FAPR bit after programing of the access window information and the user ID information. If the FAPR bit is set before programing of the access window information and the user ID information, the programming for the access window information and the user ID information cannot perform because of the protection function by the FAPR. When programming using the hex file, programming is the ascending order of the address. In this case, the FAPR bit is written in before the access window information and the user ID information. Therefore, divide the hex file for FAPR into another file, and use it after the access window information and the user ID information setting.

(6) Forced Stop by Software Command

Figure 35.31 shows a simple flowchart for the forced stop procedure to stop the blank check command, the block erase, or the chip erase command forcibly. When the forced stop command is executed, FEAMH/FEAML registers store the stopped address value. For the blank check command, the blank check can restart from the stopped address by copying the value of FEAMH/FEAML registers to FSARH/FSARL registers, respectively.

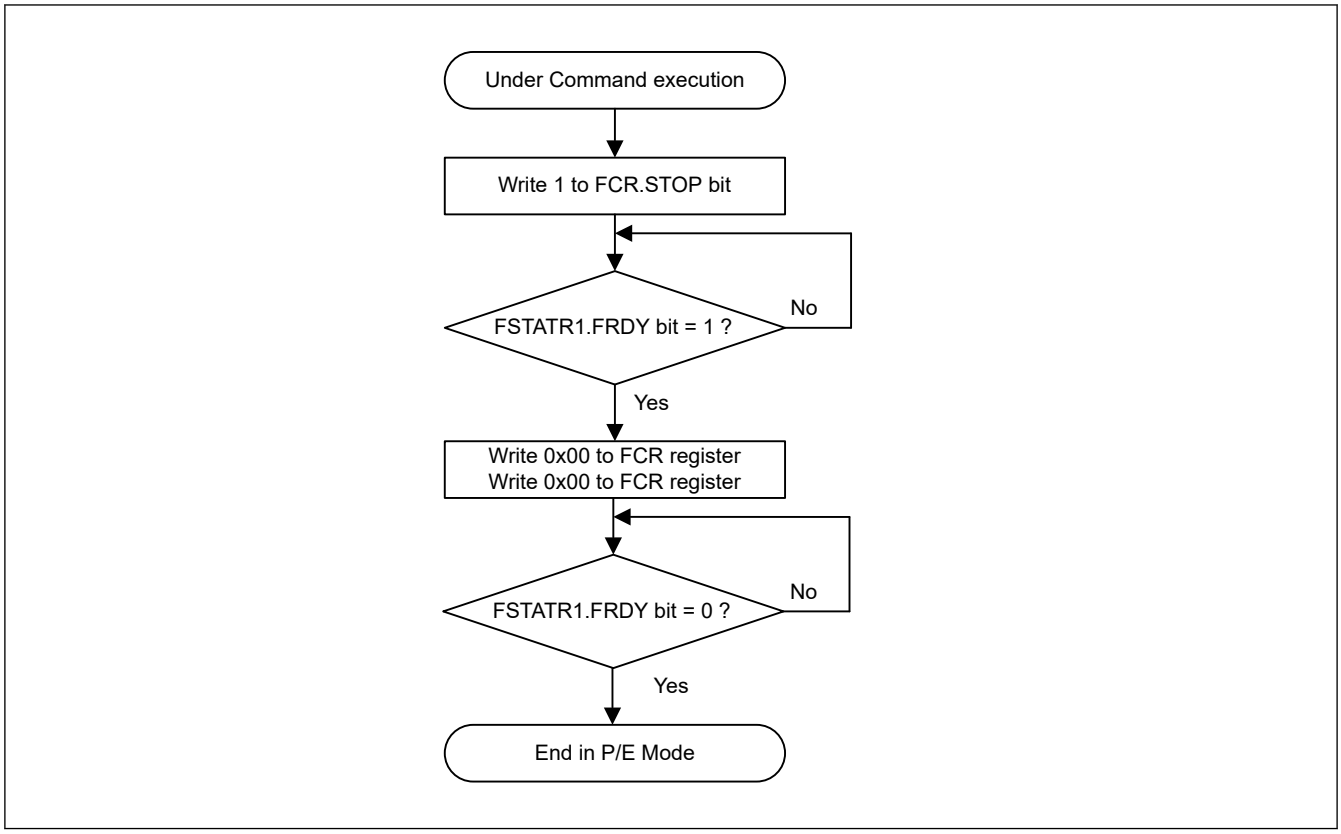


Figure 35.31 Simple flowchart for the forced stop procedure

35.14 Reading the Flash Memory

35.14.1 Reading the Code Flash Memory

No special settings are required to read the code flash memory in Normal mode. Data can be read by accessing the addresses in the code flash memory. When reading code flash memory that is erased but not yet reprogrammed, such as code flash memory in the non-programmed state, all bits are read as 1s.

35.14.2 Reading the Data Flash Memory

No special settings are required to read the data flash memory in Normal mode except when issuing a reset that causes the data flash access disable mode to disable reading. In this case, the application must transfer back to the data flash read mode. When reading data flash memory that is erased but not yet reprogrammed, such as data flash in the non-programmed state, all bits are read as 1s.

35.15 Usage Notes

35.15.1 Erase Suspended Area

Data in areas where an erase operation is suspended is undefined. To avoid malfunctions caused by reading undefined data, do not execute commands and read data in the area where erase operation is suspended.

35.15.2 Constraints on Additional Writes

Other than the configuration area, no other area can be written to twice. After a write to a flash memory area is complete, erase the area before attempting to overwrite data in that area. The configuration area can be overwritten.

35.15.3 Reset during Programming and Erasure

If inputting a reset from the RES pin, release the reset after a reset input time of at least t_{RESW}. See section 37.3.3. Reset Timing within the range of the operating voltage defined in the electrical characteristics.

The IWDT reset and software reset do not require a t_{RESW} input time.

35.15.4 Location of Interrupt Vectors during Programming and Erasure

When an interrupt occurs during a programming and erasure operation, the vector can be fetched from the code flash memory as default setting. To avoid fetching the vector from the code flash memory, set the destination for fetching interrupt vectors to an area other than the code flash memory with the interrupt table.

35.15.5 Programming and Erasure in Subosc-Speed Operating Mode

Do not program or erase the flash memory when subosc-speed operating mode is selected in the SOPCCR register for low-power consumption functions.

35.15.6 Abnormal Termination during Programming and Erasure

When the voltage exceeds the range of the operating voltage during a programming and erasure operation, or when a programming or erasure operation did not complete successfully because of a reset or prohibited actions as described in section 35.15.7. Actions Prohibited during Programming and Erasure, erase the area again.

35.15.7 Actions Prohibited during Programming and Erasure

To prevent damage to the flash memory, comply with the following instructions during programming and erasure:

- Do not use an MCU power supply that is outside the operating voltage range
- Do not update the OPCCR.OPCM[1:0] bits value
- Do not update the SOPCCR.SOPCM bit value
- Do not change the division ratio of the system clock (ICLK)
- Do not place the MCU in Software Standby mode
- Do not access the data flash memory during a program or erase operation to the code flash memory
- Do not change the DFLCTL.DFLEN bit value during a program/erase operation to the data flash.

35.15.8 Flash-IF clock (ICLK) during Program/Erase

For programming/erasure by self-programming, it is necessary to specify an integer frequency by setting the Flash Initial Setting Register (FISR).

Note: When the frequency (ICLK) is 4 to 48 MHz, a rounded-up value should be set for a non-integer frequency such as 12.5 MHz (for example, 12.5 MHz should be set rounded up to 13 MHz).

36. Internal Voltage Regulator

36.1 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 domains.

36.2 Operation

Table 36.1 lists the LDO mode pin settings, and Figure 36.1 shows the LDO mode settings. In LDO mode, the internal voltage is generated from VCC.

Table 36.1 LDO mode pin

Pins	Setting descriptions
VCC	• Connect the pin to the system power supply. • Connect the pin to VSS through a 0.1-µF multilayer ceramic capacitor. Place the capacitor close to the pin.
VCL	Connect the pin to VSS through a 4.7-µF multilayer ceramic capacitor. Place the capacitor close to the pin.

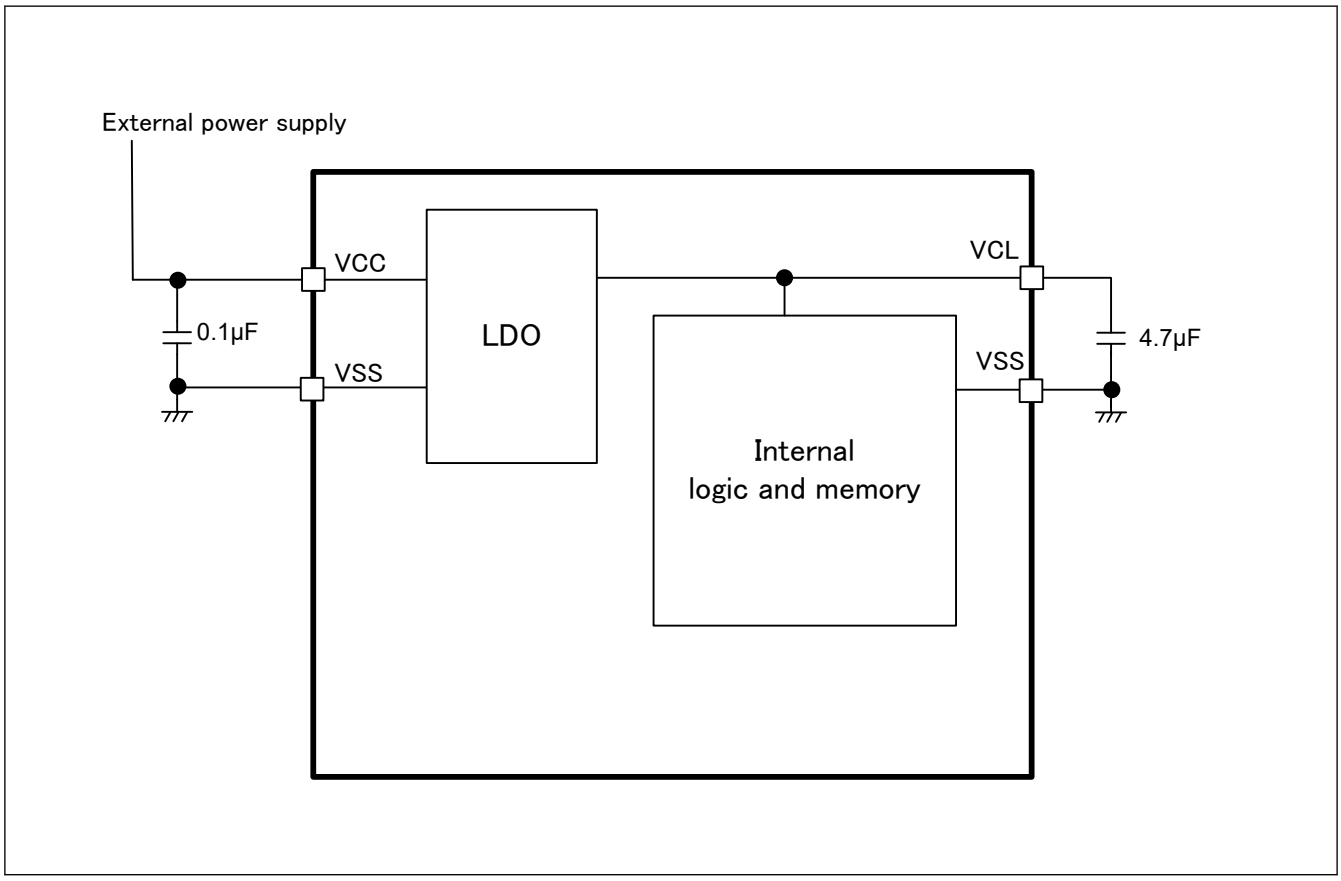


Figure 36.1 LDO mode settings

37. Electrical Characteristics

Unless otherwise specified, the electrical characteristics of the MCU are defined under the following conditions:

VCC = 1.6 to 5.5 V

VSS = 0 V, Ta = T_{opr}

Figure 37.1 shows the timing conditions.

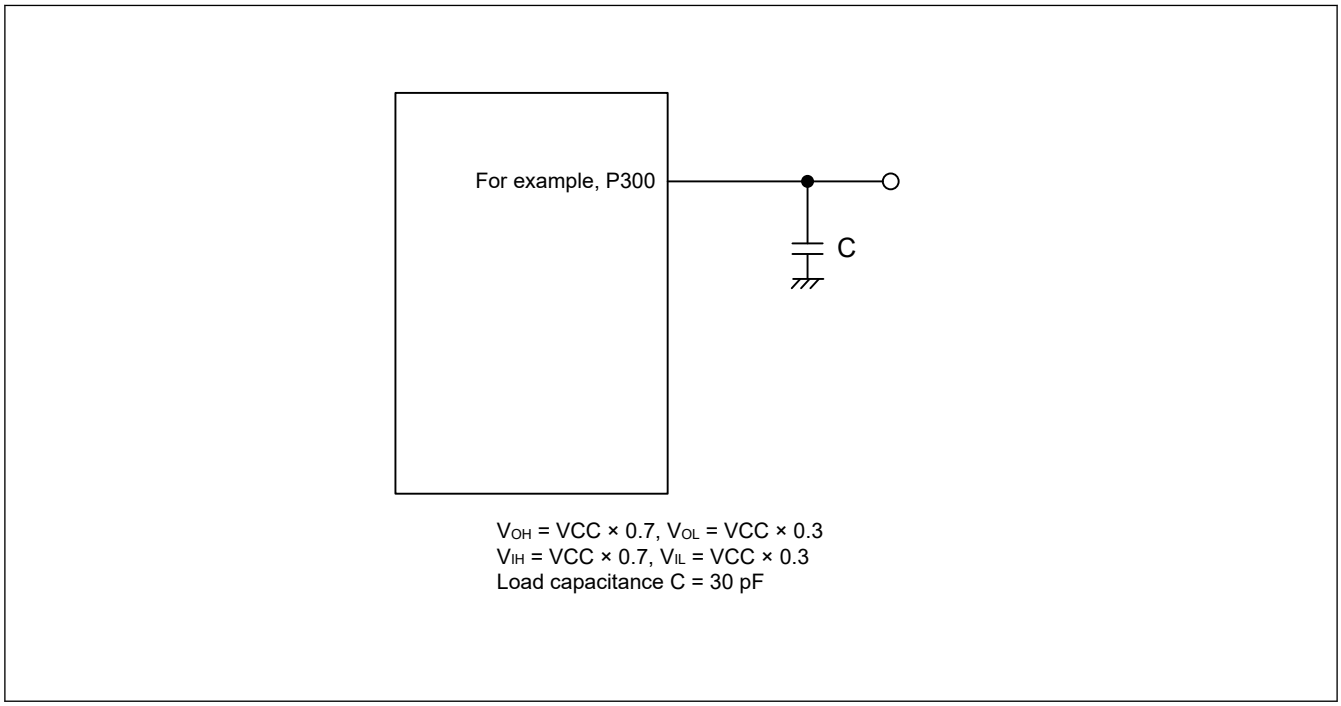


Figure 37.1 Input or output timing measurement conditions

The measurement conditions of the timing specifications for each peripheral are recommended for the best peripheral operation. However, make sure to adjust driving abilities for each pin to meet the conditions of your system.

Each function pin used for the same function must select the same drive ability. If the I/O drive ability of each function pin is mixed, the AC characteristics of each function are not guaranteed.

37.1 Absolute Maximum Ratings

Table 37.1 Absolute maximum ratings

Parameter	Symbol	Value	Unit
Power supply voltage	VCC	-0.5 to +6.5	V
Input voltage	V_{in}	-0.3 to VCC + 0.3	V
Analog input voltage	V_{AN}	-0.3 to VCC + 0.3	V
Operating temperature*1	T_{opr}	-40 to +125	°C
Storage temperature	T_{stg}	-55 to +140	°C

Note 1. See section 37.2.1. Tj/Ta Definition.

Caution: Permanent damage to the MCU may result if absolute maximum ratings are exceeded.

To preclude any malfunctions due to noise interference, insert capacitors with high frequency characteristics between the VCC and VSS pins, and between the AVREFP and AVREFM pins when AVREFP is selected as the high potential reference voltage for the ADC12. Place capacitors of the following value as close as possible to every power supply pin and use the shortest and heaviest possible traces:

- **VCC and VSS: about 0.1 µF**
- **AVREFP and AVREFM: about 0.1 µF**

Also, connect capacitors as stabilization capacitance.

Connect the VCL pin to a VSS pin by a 4.7 µF capacitor. Each capacitor must be placed close to the pin.

Table 37.2 Recommended operating conditions

Parameter	Symbol	Min	Typ	Max	Unit
Power supply voltages	VCC	1.6	—	5.5	V
	VSS	—	0	—	V

37.2 DC Characteristics

37.2.1 Tj/Ta Definition

Table 37.3 DC characteristics

Conditions: Products with operating temperature (Ta) -40 to +105°C

Parameter	Symbol	Typ	Max	Unit	Test conditions
Permissible junction temperature	Tj	—	140	°C	High-speed mode Middle-speed mode Low-speed mode Subosc-speed mode

Note: Make sure that Tj = T_a + θja × total power consumption (W), where total power consumption = (VCC - V_{OH}) × ΣI_{OH} + V_{OL} × ΣI_{OL} + I_{CC}max × VCC.

37.2.2 I/O V_{IH}, V_{IL}

Table 37.4 I/O V_{IH}, V_{IL}

Conditions: VCC = 1.6 to 5.5 V

Parameter		Symbol	Min	Typ	Max	Unit	Test Conditions
Schmitt trigger input voltage	5V-tolerant ports (P010, P011, P101, P102, P103)	V_{IH}	VCC × 0.7	—	5.8	V	—
		V_{IL}	—	—	VCC × 0.3		
	RES, NMI Other peripheral input pins	V_{IH}	VCC × 0.8	—	—		—
		V_{IL}	—	—	VCC × 0.2		—

37.2.3 I/O I_{OH}, I_{OL}

Table 37.5 I/O I_{OH}, I_{OL} (1 of 2)

Conditions: VCC = 1.6 to 5.5 V

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Permissible output current (average value per pin)	ANI0-5 ports (P002 to P003, P006 to P007, P400 to P401)	I_{OH}	—	—	-4.0	mA	—
		I_{OL}	—	—	8.0	mA	
	5V-tolerant ports (P010 to P011, P101 to P103)	I_{OH}	—	—	-4.0	mA	—
		I_{OL}	—	—	8.0	mA	
	Other output pins*1	I_{OH}	—	—	-4.0	mA	—
		I_{OL}	—	—	20.0	mA	

Table 37.5 I/O I_{OH}, I_{OL} (2 of 2)

Conditions: VCC = 1.6 to 5.5 V

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Permissible output current (max value total pins)*1	Total of ANI0-5 ports (P002 to P003, P006 to P007, P400 to P401)	$\Sigma I_{OH\ (max)}$	—	—	-24.0	mA	VCC = 2.7 to 5.5 V
			—	—	-6.0	mA	VCC = 1.8 to 2.7 V
			—	—	-3.0	mA	VCC = 1.6 to 1.8 V
		$\Sigma I_{OL\ (max)}$	—	—	48.0	mA	VCC = 2.7 to 5.5V
			—	—	3.6	mA	VCC = 1.8 to 2.7 V
			—	—	1.8	mA	VCC = 1.6 to 1.8 V
	5V-tolerant ports (P010 to P011, P101 to P103)	$\Sigma I_{OH\ (max)}$	—	—	-20.0	mA	VCC = 2.7 to 5.5V
			—	—	-5.0	mA	VCC = 1.8 to 2.7 V
			—	—	-2.0	mA	VCC = 1.6 to 1.8 V
		$\Sigma I_{OL\ (max)}$	—	—	40.0	mA	VCC = 2.7 to 5.5 V
			—	—	3.0	mA	VCC = 1.8 to 2.7 V
			—	—	1.5	mA	VCC = 1.6 to 1.8 V
	Total of other output ports	$\Sigma I_{OH\ (max)}$	—	—	-30.0	mA	VCC = 2.7 to 5.5 V
			—	—	-12.0	mA	VCC = 1.8 to 2.7 V
			—	—	-6.0	mA	VCC = 1.6 to 1.8 V
		$\Sigma I_{OL\ (max)}$	—	—	50.0	mA	VCC = 2.7 to 5.5 V
			—	—	9.0	mA	VCC = 1.8 to 2.7 V
			—	—	4.5	mA	VCC = 1.6 to 1.8 V
	Total of all output pin	$\Sigma I_{OH\ (max)}$	—	—	-50.0	mA	—
		$\Sigma I_{OL\ (max)}$	—	—	95.0	mA	

Note 1. Specification under conditions where the duty factor ≤ 70%.
The output current value that has changed to the duty factor > 70% the duty ratio can be calculated with the following expression (when changing the duty factor from 70% to n%).
Total output current of pins = (I_{OH} × 0.7)/(n × 0.01)
<Example> Where n = 80% and I_{OH} = −30.0 mA
Total output current of pins = (−30.0 × 0.7)/(80 × 0.01) ≅ −26.2 mA
However, the current that is allowed to flow into one pin does not vary depending on the duty factor.

Caution: To protect the reliability of the MCU, the output current values should not exceed the values in Table 37.5.

37.2.4 I/O V_{OH}, V_{OL}, and Other Characteristics

Table 37.6 I/O V_{OH}, V_{OL} (1)

Conditions: VCC = 4.0 to 5.5 V

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Output voltage	Output pins*1	V_{OH}	VCC - 0.8	—	—	V	I_{OH} = -4.0 mA
	P002 to P003, P006 to P007, P400 to P401	V_{OL}	—		0.8		I_{OL} = 8.0 mA
	P010 to P011, P101 to P103	V_{OL}	—	—	0.8		I_{OL} = 8.0 mA
	Other output pins*1	V_{OL}	—	—	1.2		I_{OL} = 20.0 mA

Note 1. Except for Ports P200, which are input ports, and XT1 and XT2, which are SOSC ports.

Table 37.7 I/O V_{OH}, V_{OL} (2)

Conditions: VCC = 2.7 to 4.0 V

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Output voltage	Output pins*1	V_{OH}	VCC - 0.8	—	—	V	I_{OH} = -4.0 mA
	Output pins*1	V_{OL}	—	—	0.8		I_{OL} = 8.0 mA

Note 1. Except for Ports P200, which are input ports, and XT1 and XT2, which are SOSC ports.

Table 37.8 I/O V_{OH}, V_{OL} (3)

Conditions: VCC = 1.6 to 2.7 V

Parameter		Symbol	Min	Typ	Max	Unit	Test Conditions
Output voltage	Output pins*1	V_{OH}	VCC - 0.5	—	—	V	IOH = -1.0 mA VCC = 1.8 to 2.7 V
			VCC - 0.5	—	—		IOH = -0.5 mA VCC = 1.6 to 1.8 V
	Output pins*1	V_{OL}	—	—	0.4		IOL = 0.6 mA VCC = 1.8 to 2.7 V
			—	—	0.4		IOL = 0.3 mA VCC = 1.6 to 1.8 V

Note 1. Except for Ports P200, which are input ports, and XT1 and XT2, which are SOSC ports.

Table 37.9 I/O other characteristics

Conditions: VCC = 1.6 to 5.5 V

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Constant low-level current output*1	P100, P302, P303	$CCDI_{OL}$	1.15	2	2.87	mA	PmnPFS.DSCR = b00, VCC = 4.0 V to 5.5 V
			0.97	1.7	2.59	mA	PmnPFS.DSCR = b00, VCC = 2.7 V to 4.0 V
			2.95	5	6.97	mA	PmnPFS.DSCR = b01, VCC = 4.0 V to 5.5 V
			2.64	4.2	6.38	mA	PmnPFS.DSCR = b01, VCC = 3.0 V to 4.0 V
			5.97	10	13.48	mA	PmnPFS.DSCR = b1x, VCC = 4.0 V to 5.5 V
			5.6	8.5	12.38	mA	PmnPFS.DSCR = b1x, VCC = 3.3 V to 4.0 V
Input leakage current	RES, P200, XT1, XT2	$\lvert I_{in} \rvert$	—	—	1.0	µA	V_{in} = 0 V V_{in} = VCC
Three-state leakage current (off state)	5V-tolerant ports (P010 to P011, P101 to P103)	$\lvert I_{TSI} \rvert$	—	—	1.0	µA	V_{in} = 0 V V_{in} = 5.8 V
	Other ports (except for P200, XT1, XT2 and 5V-tolerant ports)		—	—	1.0	µA	V_{in} = 0 V V_{in} = VCC
Input pull-up resistor	All ports (except for P200, XT1, XT2)	R_U	10	20	100	kΩ	V_{in} = 0 V
Input capacitance	P200	C_{in}	—	—	30	pF	V_{in} = 0 V f = 1 MHz T_a = 25°C
	Other input pins		—	—	15		

Note 1. The listed currents apply when the output current control function is enabled.

37.2.5 Operating and Standby Current

Table 37.10 Operating and standby current (1) (1 of 2)

Conditions *1 *2: VCC = 1.6 to 5.5 V

Parameter					Symbol	Typ*11	Max	Unit	Test Conditions
Supply current*3	High-speed mode*4	Normal mode	All peripheral clocks disabled, CoreMark code executing from flash*7	ICLK = 48 MHz	I_{CC}	7.80	—	mA	*9 *12
				ICLK = 32 MHz		6.45	—		*9
				ICLK = 16 MHz		4.00	—		
				ICLK = 8 MHz		2.70	—		
			All peripheral clocks enabled, code executing from flash*7	ICLK = 48 MHz		—	17.4		*12
		Sleep mode	All peripheral clocks disabled*7	ICLK = 48 MHz		1.80	—		*9
				ICLK = 32 MHz		1.40	—		
				ICLK = 16 MHz		1.00	—		
				ICLK = 8 MHz		0.80	—		
			All peripheral clocks enabled*7	ICLK = 48 MHz		3.70	—		*10
				ICLK = 32 MHz		2.60	—		
				ICLK = 16 MHz		1.65	—		
				ICLK = 8 MHz		1.10	—		
		Increase during BGO operation*8				1.95	—		—
	Middle-speed mode*4	Normal mode	All peripheral clocks disabled, CoreMark code executing from flash*7	ICLK = 24 MHz	I_{CC}	4.80	—	mA	*9
				ICLK = 4 MHz		1.35	—		
			All peripheral clocks enabled, code executing from flash*7	ICLK = 24 MHz		—	10.1		*10
		Sleep mode	All peripheral clocks disabled*7	ICLK = 24 MHz		1.20	—		*9
				ICLK = 4 MHz		0.70	—		
			All peripheral clocks enabled*7	ICLK = 24 MHz		2.20	—		*10
		Increase during BGO operation*8				2.05	—		—
	Low-speed mode*5	Normal mode	All peripheral clocks disabled, CoreMark code executing from flash*7	ICLK = 1 MHz	I_{CC}	0.35	—	mA	*9
			All peripheral clocks enabled, code executing from flash*7	ICLK = 1 MHz		—	2.8		*10
		Sleep mode	All peripheral clocks disabled*7	ICLK = 1 MHz		0.20	—		*9
			All peripheral clocks enabled*7	ICLK = 1 MHz		0.25	—		*10

Table 37.10 Operating and standby current (1) (2 of 2)

Conditions *1 *2: VCC = 1.6 to 5.5 V

Parameter					Symbol	Typ*11	Max	Unit	Test Conditions
Supply current*3	Subosc-speed mode*6	Normal mode	All peripheral clocks enabled, code executing from flash*7	ICLK = 32.768 kHz	I_{CC}	—	1.6	mA	*10
		Sleep mode	All peripheral clocks disabled*7	ICLK = 32.768 kHz		2.30	—	µA	*10
			All peripheral clocks enabled*7	ICLK = 32.768 kHz		3.65	—		*10

Note 1. Conditions for high-speed mode are VCC = 1.8 to 5.5 V.
Note 2. Conditions for middle-speed mode are VCC = 1.8 to 5.5 V when ICLK = 24 MHz.
Note 3. Supply current is the total current flowing into VCC, including analog power supply current. Supply current values apply when internal pull-up MOSs are in the off state and these values do not include output charge/discharge current from any of the pins.
Note 4. The clock source is HOCO.
Note 5. The clock source is MOCO.
Note 6. The clock source is the sub-clock oscillator.
Note 7. This does not include BGO operation.
Note 8. This is the increase for programming or erasure of the flash memory for data storage during program execution.
Note 9. PCLKB is set to be divided by 64.
Note 10. PCLKB is the same frequency as that of ICLK.
Note 11. VCC = 3.3 V.
Note 12. The prefetch is operating.

Table 37.11 Operating and standby current (2)

Conditions: VCC = 1.6 to 5.5 V

Parameter					Symbol	Typ*3	Max	Unit	Test conditions
Supply current *1	Software Standby mode*2	Peripheral modules stop	All SRAM (0x2000_0000 to 0x2000_0FFF and 0x2000_4000 to 0x2000_6FFF) is on	T_a = 25°C	I_{CC}	0.30	1.8	µA	—
				T_a = 55°C		0.45	5.1		
				T_a = 85°C		1.15	20		
				T_a = 105°C		2.75	48		
				T_a = 125°C		6.95	112		
			8KB SRAM (0x2000_0000 to 0x2000_0FFF and 0x2000_4000 to 0x2000_4FFF) is on	T_a = 25°C		0.30	1.8		
				T_a = 55°C		0.45	4.8		
				T_a = 85°C		1.15	19		
				T_a = 105°C		2.75	47		
				T_a = 125°C		6.95	108		
		Increment for RTC operation with low-speed on-chip*4				0.65	—		—
		Increment for RTC operation in normal operation mode with sub-clock oscillator*4				0.23	—		SOMCR.SODRV[1:0] are 11b (Low power mode 3) RTCC0.RTC128EN is 0 (RTC operation in normal operation mode)
						0.97	—		SOMCR.SODRV[1:0] are 00b (normal mode) RTCC0.RTC128EN is 0 (RTC operation in normal operation mode)
		Increment for RTC operation in low-consumption clock mode with sub-clock oscillator*4				0.22	—		SOMCR.SODRV[1:0] are 11b (Low power mode 3) RTCC0.RTC128EN is 1 (RTC operation in low-consumption clock mode)
						0.95	—		SOMCR.SODRV[1:0] are 00b (normal mode) RTCC0.RTC128EN is 1 (RTC operation in low-consumption clock mode)

Note 1. Supply current is the total current flowing into VCC, including analog power supply current. Supply current values apply when internal pull-up MOSs are in the off state and these values do not include output charge/discharge current from any of the pins.
Note 2. The IWDT and LVD are not operating.
Note 3. VCC = 3.3 V.
Note 4. Includes the low-speed on-chip oscillator or sub-oscillation circuit current.

Table 37.12 Operating and standby current (3)

Conditions *1, *2: VCC = 1.6 to 5.5 V

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Analog power supply current	During 12-bit A/D conversion (at high-speed conversion)	I_{VCCADC}	—	—	1.44	mA	—
	During 12-bit A/D conversion (at low-power conversion)		—	—	0.78	mA	—
	CMP enabled (at high-speed mode, per channel)	I_{VCCCMP}	—	6.0	—	µA	—
	CMP enabled (at low-speed mode, per channel)		—	2.0	—	µA	—
	DAC8 enabled (per channel)*1	I_{VCCDAC}	—	—	0.5	mA	—
Reference power supply current	During 12-bit A/D conversion	I_{REFH}	—	—	0.15	mA	—
Temperature Sensor (TSN) operating current*2		I_{TSN}	—	0.13	—	mA	—
12-bit A/D converter internal reference voltage current*2		I_{ADREF}	—	0.13	—	mA	—
Output current control operating current	CCDE register is not 0x00	I_{CCDA}	—	120*3	—	µA	—
	Per single output current control port*4	I_{CCDP}	—	30	—		Setting of the low-level output current: Hi-Z
			—	200	—		Setting of the low-level output current: 2 to 15 mA

Note 1. Conditions for DAC8 use are VCC = 2.7 to 5.5 V
Note 2. Conditions for TSN and internal reference voltage use are VCC = 1.8 to 5.5 V
Note 3. This current is added to the supply current when the output voltage control port is set at CCDIOL typical current CCTRM.IADJ setting if VCC = 4 V
Note 4. This current does not include the current flowing into the I/O port pins

37.2.6 VCC Rise and Fall Gradient and Ripple Frequency

Table 37.13 Rise and fall gradient characteristics

Conditions: VCC = 0 to 5.5 V

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Power-on VCC rising gradient	Voltage monitor 0 reset disabled at startup	SrVCC	0.02	—	2	ms/V	—
	Voltage monitor 0 reset enabled at startup*1				—		

Note 1. When OFS1.LVDAS = 0.

Table 37.14 Rising and falling gradient and ripple frequency characteristics

Conditions: VCC = 1.6 to 5.5 V
The ripple voltage must meet the allowable ripple frequency $f_{r(VCC)}$ within the range between the VCC upper limit (5.5 V) and lower limit (1.6 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 37.2 $V_{r\ (VCC)} \leq VCC \times 0.2$
		—	—	1	MHz	Figure 37.2 $V_{r\ (VCC)} \leq VCC \times 0.08$
		—	—	10	MHz	Figure 37.2 $V_{r\ (VCC)} \leq VCC \times 0.06$
Allowable voltage change rising and falling gradient	dt/dVCC	1.0	—	—	ms/V	When VCC change exceeds VCC ± 10%

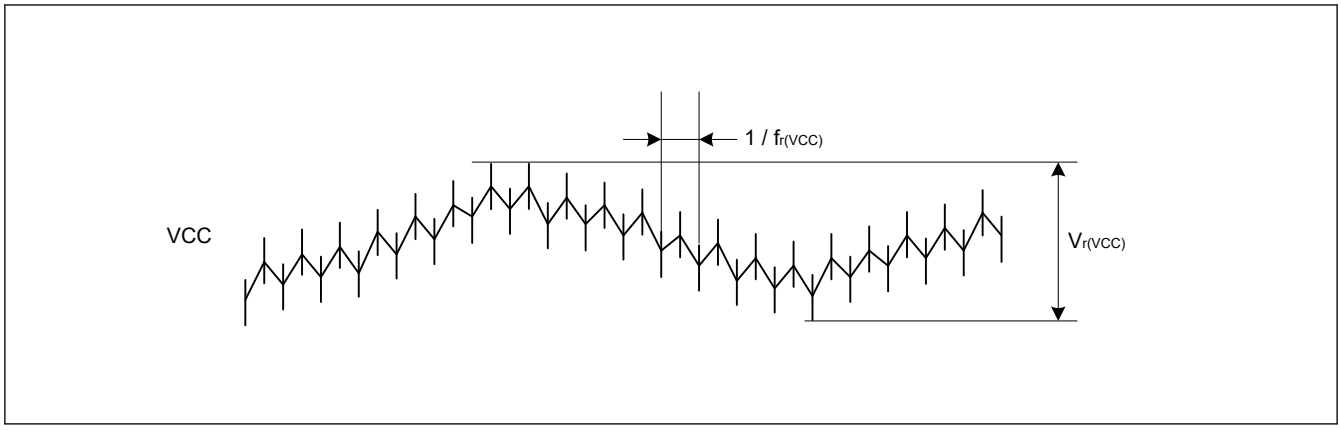


Figure 37.2 Ripple waveform

37.3 AC Characteristics

37.3.1 Frequency

Table 37.15 Operation frequency in high-speed mode

Conditions: VCC = 1.8 to 5.5 V

Parameter			Symbol	Min	Typ	Max*4	Unit
Operation frequency	System clock (ICLK)*1 *2 *3	1.8 to 5.5 V	f	0.032768	—	48	MHz
	Peripheral module clock (PCLKB) *3	1.8 to 5.5 V		—	—	48	

Note 1. The lower-limit frequency of ICLK is 1 MHz while programming or erasing the flash memory. When using ICLK for programming or erasing the flash memory at below 4 MHz, the frequency can be set to 1 MHz, 2 MHz, or 3 MHz. A non-integer frequency such as 1.5 MHz cannot be set.
Note 2. The frequency accuracy of ICLK must be ± 1.5% during programming or erasing the flash memory. Confirm the frequency accuracy of the clock source.
Note 3. See section 8, Clock Generation Circuit for the relationship of frequencies between ICLK and PCLKB.
Note 4. The maximum value of operation frequency does not include internal oscillator errors. For details on the range for guaranteed operation, see Table 37.19.

Table 37.16 Operation frequency in middle-speed mode

Conditions: VCC = 1.6 to 5.5 V

Parameter			Symbol	Min	Typ	Max*4	Unit
Operation frequency	System clock (ICLK)*1 *2 *3	1.8 to 5.5 V	f	0.032768	—	24	MHz
		1.6 to 1.8 V		0.032768	—	4	
	Peripheral module clock (PCLKB)*3	1.8 to 5.5 V		—	—	24	
		1.6 to 1.8 V		—	—	4	

Note 1. The lower-limit frequency of ICLK is 1 MHz while programming or erasing the flash memory. When using ICLK for programming or erasing the flash memory at below 4 MHz, the frequency can be set to 1 MHz, 2 MHz, or 3 MHz. A non-integer frequency such as 1.5 MHz cannot be set.
Note 2. The frequency accuracy of ICLK must be ± 1.5% while programming or erasing the flash memory. Confirm the frequency accuracy of the clock source.
Note 3. See section 8, Clock Generation Circuit for the relationship of frequencies between ICLK and PCLKB.
Note 4. The maximum value of operation frequency does not include internal oscillator errors. For details on the range for guaranteed operation, see Table 37.19.

Table 37.17 Operation frequency in low-speed mode

Conditions: VCC = 1.6 to 5.5 V

Parameter			Symbol	Min	Typ	Max*4	Unit
Operation frequency	System clock (ICLK)*1 *2 *3	1.6 to 5.5 V	f	0.032768	—	1	MHz
	Peripheral module clock (PCLKB) *3	1.6 to 5.5 V		—	—	1	

Note 1. The lower-limit frequency of ICLK is 1 MHz while programming or erasing the flash memory.

Note 2. The frequency accuracy of ICLK must be ± 1.5% while programming or erasing the flash memory. Confirm the frequency accuracy of the clock source.
Note 3. See section 8, Clock Generation Circuit for the relationship of frequencies between ICLK and PCLKB.
Note 4. The maximum value of operation frequency does not include internal oscillator errors. For details on the range for guaranteed operation, see Table 37.19.

Table 37.18 Operation frequency in Subosc-speed mode

Parameter			Symbol	Min	Typ	Max	Unit
Operation frequency	System clock (ICLK)*1*2	1.6 to 5.5 V	f	27.8528	32.768	37.6832	kHz
	Peripheral module clock (PCLKB)*2	1.6 to 5.5 V		—	—	37.6832	

Note 1. Programming and erasing the flash memory is not possible.
Note 2. See section 8, Clock Generation Circuit for the relationship of frequencies between ICLK and PCLKB.

37.3.2 Clock Timing

Table 37.19 Clock timing

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
EXTAL external clock input cycle time	t_{Xcyc}	50	—	—	ns	Figure 37.3
EXTAL external clock input high pulse width	t_{XH}	20	—	—	ns	—
EXTAL external clock input low pulse width	t_{XL}	20	—	—	ns	—
EXTAL external clock rising time	t_{Xr}	—	—	5	ns	—
EXTAL external clock falling time	t_{Xf}	—	—	5	ns	—
EXTAL external clock input wait time*1	t_{EXWT}	0.3	—	—	μs	—
EXTAL external clock input frequency	f_{EXTAL}	—	—	20	MHz	1.8 ≤ VCC ≤ 5.5
		—	—	4	MHz	1.6 ≤ VCC < 1.8
LOCO clock oscillation frequency	f_{LOCO}	27.8528	32.768	37.6832	kHz	—
LOCO clock oscillation stabilization time	t_{LOCO}	—	—	100	μs	Figure 37.4
IWDT-dedicated clock oscillation frequency	f_{ILOCO}	12.75	15	17.25	kHz	—
MOCO clock oscillation frequency	f_{MOCO}	6.8	8	9.2	MHz	—
MOCO clock oscillation stabilization time	t_{MOCO}	—	—	1	μs	—
HOCO clock oscillation frequency*5	f_{HOCO24}	23.64	24	24.36	MHz	Ta = -40 to 125°C 1.6 ≤ VCC ≤ 5.5
	f_{HOCO32}	31.52	32	32.48		Ta = -40 to 125°C 1.6 ≤ VCC ≤ 5.5
	f_{HOCO48}	47.28	48	48.72		Ta = -40 to 125°C 1.6 ≤ VCC ≤ 5.5
HOCO clock oscillation stabilization time*3 *4	t_{HOCO24} t_{HOCO32} t_{HOCO48}	—	6.7	7.7	μs	Figure 37.5
Sub-clock oscillator oscillation frequency	f_{SUB}	—	32.768	—	kHz	—
Sub-clock oscillation stabilization time*2	t_{SUBOSC}	—	0.5	—	s	Figure 37.6

Note 1. Time until the clock can be used after the external clock input stop bit (MOSCCR.MOSTP) is set to 0 (operating) when the external clock is stable.
Note 2. After changing the setting of the SOSCCR.SOSTP bit to start sub-clock oscillator operation, only start using the sub-clock oscillator after the sub-clock oscillation stabilization wait time elapsed. Use the oscillator wait time value recommended by the oscillator manufacturer.
Note 3. This is a characteristic when the HOCOCR.HCSTP bit is set to 0 (oscillation) in the MOCO stop state. When the HOCOCR.HCSTP bit is set to 0 (oscillation) during MOCO oscillation, this specification is shortened by 1 μs.
Note 4. Check OSCSF.HOCOSF to confirm whether stabilization time has elapsed.
Note 5. Accuracy at production test.

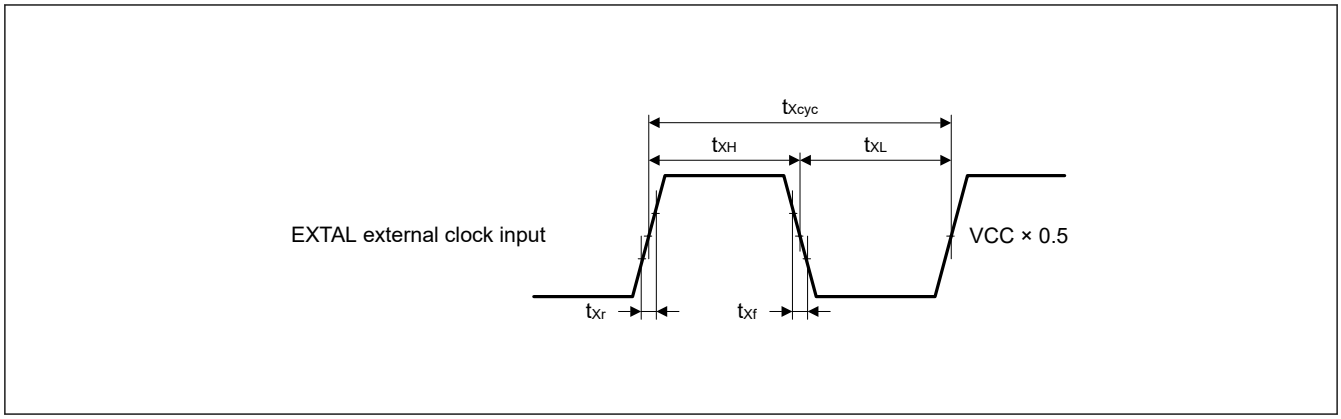


Figure 37.3 EXTAL external clock input timing

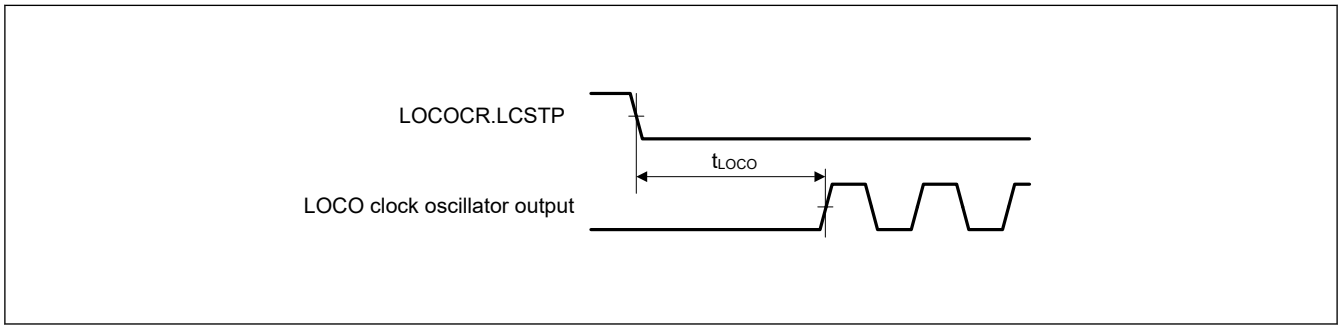


Figure 37.4 LOCO clock oscillation start timing

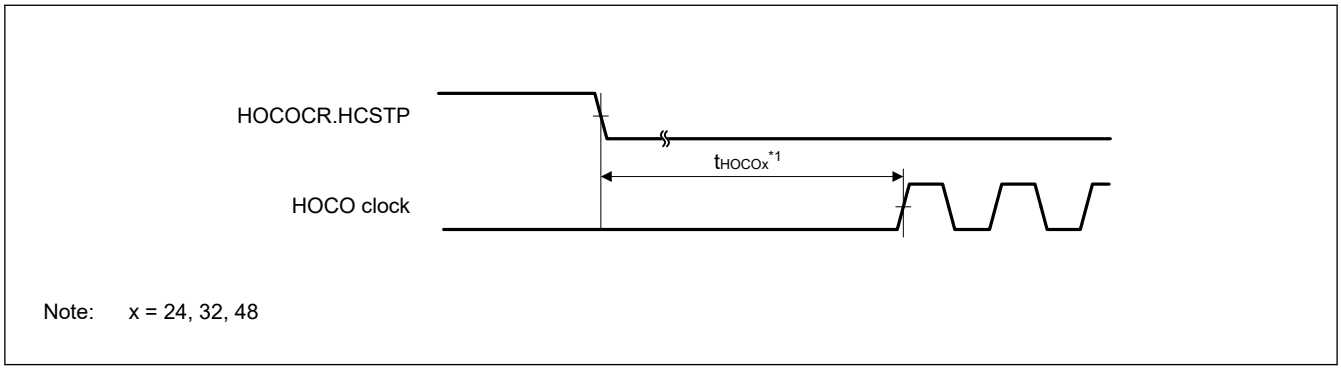


Figure 37.5 HOCO clock oscillation start timing (started by setting the HOCOCR.HCSTP bit)

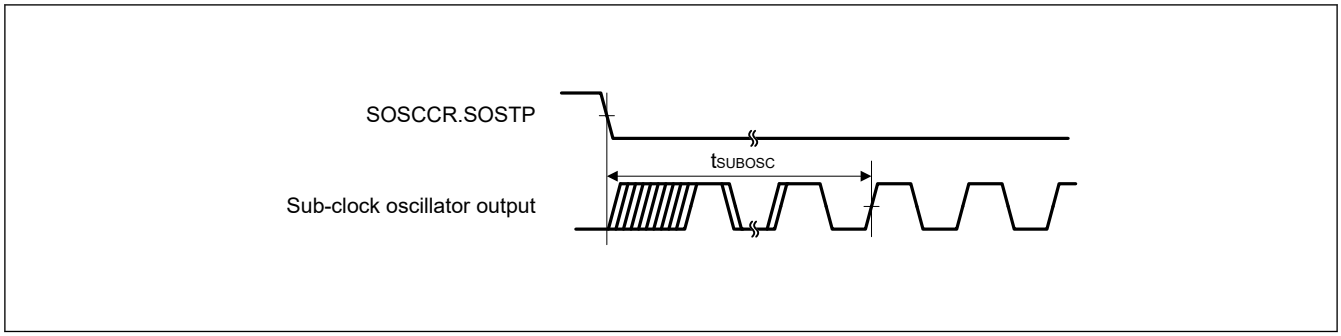


Figure 37.6 Sub-clock oscillation start timing

37.3.3 Reset Timing

Table 37.20 Reset timing

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
RES pulse width	At power-on	t_{RESWP}	10	—	—	ms	Figure 37.7
	Not at power-on	t_{RESW}	30	—	—	μs	Figure 37.8
Wait time after RES cancellation (at power-on)	LVD0 enabled*1	t_{RESWT}	—	0.9	—	ms	Figure 37.7
	LVD0 disabled*2		—	0.2	—		
Wait time after RES cancellation (during powered-on state)	LVD0 enabled*1	t_{RESWT2}	—	0.9	—	ms	Figure 37.8
	LVD0 disabled*2		—	0.2	—		
Wait time after internal reset cancellation (Watchdog timer reset, SRAM parity error reset, SRAM ECC error reset, bus error reset, debug reset, software reset)	LVD0 enabled*1	t_{RESWT3}	—	0.9	—	ms	Figure 37.9
	LVD0 disabled*2		—	0.2	—		

Note 1. When OFS1.LVDAS = 0.
Note 2. When OFS1.LVDAS = 1.

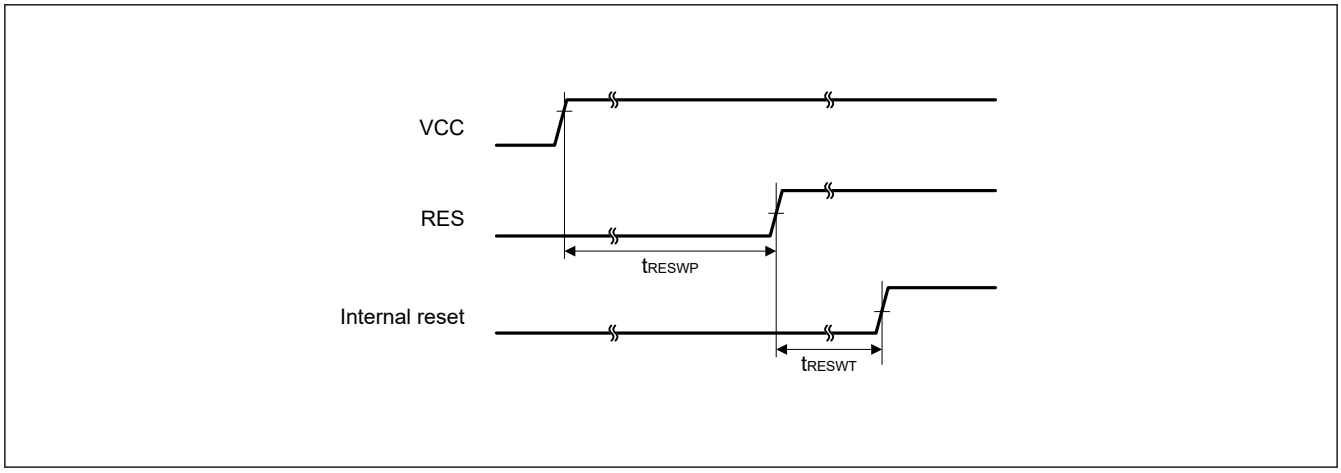


Figure 37.7 Reset input timing at power-on

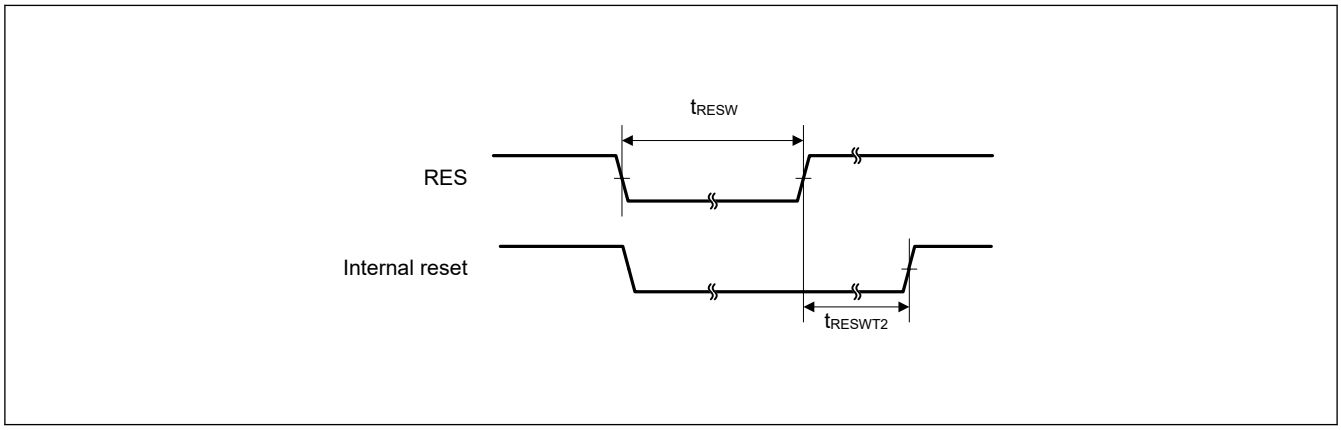


Figure 37.8 Reset input timing (1)

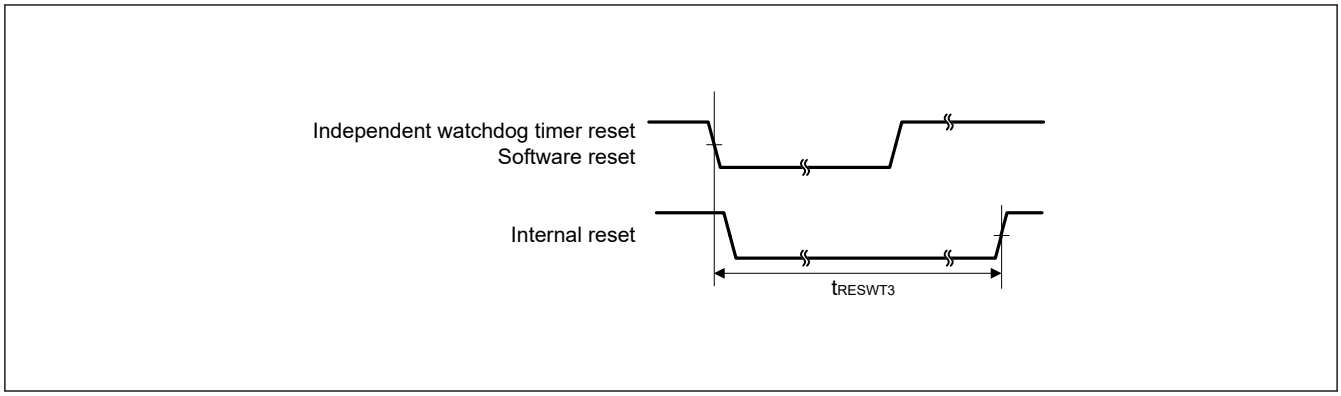


Figure 37.9 Reset input timing (2)

37.3.4 Wakeup Time

Table 37.21 Timing of recovery from low power modes (1)

Parameter				Symbol	Min	Typ	Max	Unit	Test conditions
Recovery time from Software Standby mode*1	High-speed mode	External clock input	System clock source is external clock input (20 MHz)	t_{SBYEX}	—	2.4	3.1	µs	Figure 37.10
		System clock source is HOCO (HOCO clock is 32 MHz)*2		t_{SBYHO}	—	7.4	9.1	µs	
		System clock source is HOCO (HOCO clock is 48 MHz)*3				7.2	8.9	µs	
		System clock source is MOCO (8 MHz)		t_{SBYMO}	—	4	5	µs	

Note 1. The division ratio of ICLK and PCLKx is the minimum division ratio within the allowable frequency range. The recovery time is determined by the system clock source.
Note 2. The system clock is 32 MHz.
Note 3. The system clock is 48 MHz.

Table 37.22 Timing of recovery from low power modes (2)

Parameter				Symbol	Min	Typ	Max	Unit	Test conditions
Recovery time from Software Standby mode*1	Middle-speed mode	External clock input	System clock source is external clock input (20 MHz) VCC = 1.8 V to 5.5 V	t_{SBYEX}	—	2.4	3.1	µs	Figure 37.10
			System clock source is external clock input (20 MHz) VCC = 1.6 V to 1.8 V		—	11.7	13		
		System clock source is HOCO*2	VCC = 1.8 V to 5.5 V	t_{SBYHO}	—	7.7	9.4	µs	
			VCC = 1.6 V to 1.8 V		—	15.7	17.9		
		System clock source is MOCO (8 MHz)	VCC = 1.8 V to 5.5 V	t_{SBYMO}	—	4	5	µs	
			VCC = 1.6 V to 1.8 V		—	7.2	9		

Note 1. The division ratio of ICLK and PCLKx is the minimum division ratio within the allowable frequency range. The recovery time is determined by the system clock source.
Note 2. The system clock is 24 MHz.

Table 37.23 Timing of recovery from low power modes (3)

Parameter				Symbol	Min	Typ	Max	Unit	Test conditions
Recovery time from Software Standby mode*1	Low-speed mode	External clock input	System clock source is external clock input (1 MHz)	t_{SBYEX}	—	25	40	µs	Figure 37.10
		System clock source is MOCO (1 MHz)		t_{SBYMO}	—	26	40	µs	

Note 1. The division ratio of ICLK and PCLKx is the minimum division ratio within the allowable frequency range. The recovery time is determined by the system clock source.

Table 37.24 Timing of recovery from low power modes (4)

Parameter			Symbol	Min	Typ	Max	Unit	Test conditions
Recovery time from Software Standby mode*1	Subosc-speed mode	System clock source is sub-clock oscillator (32.768 kHz)	t_{SBYSC}	—	0.85	1	ms	Figure 37.10
		System clock source is LOCO (32.768 kHz)	t_{SBYLO}	—	0.85	1.2	ms	

Note 1. The sub-clock oscillator or LOCO itself continues oscillating in Software Standby mode during Subosc-speed mode.

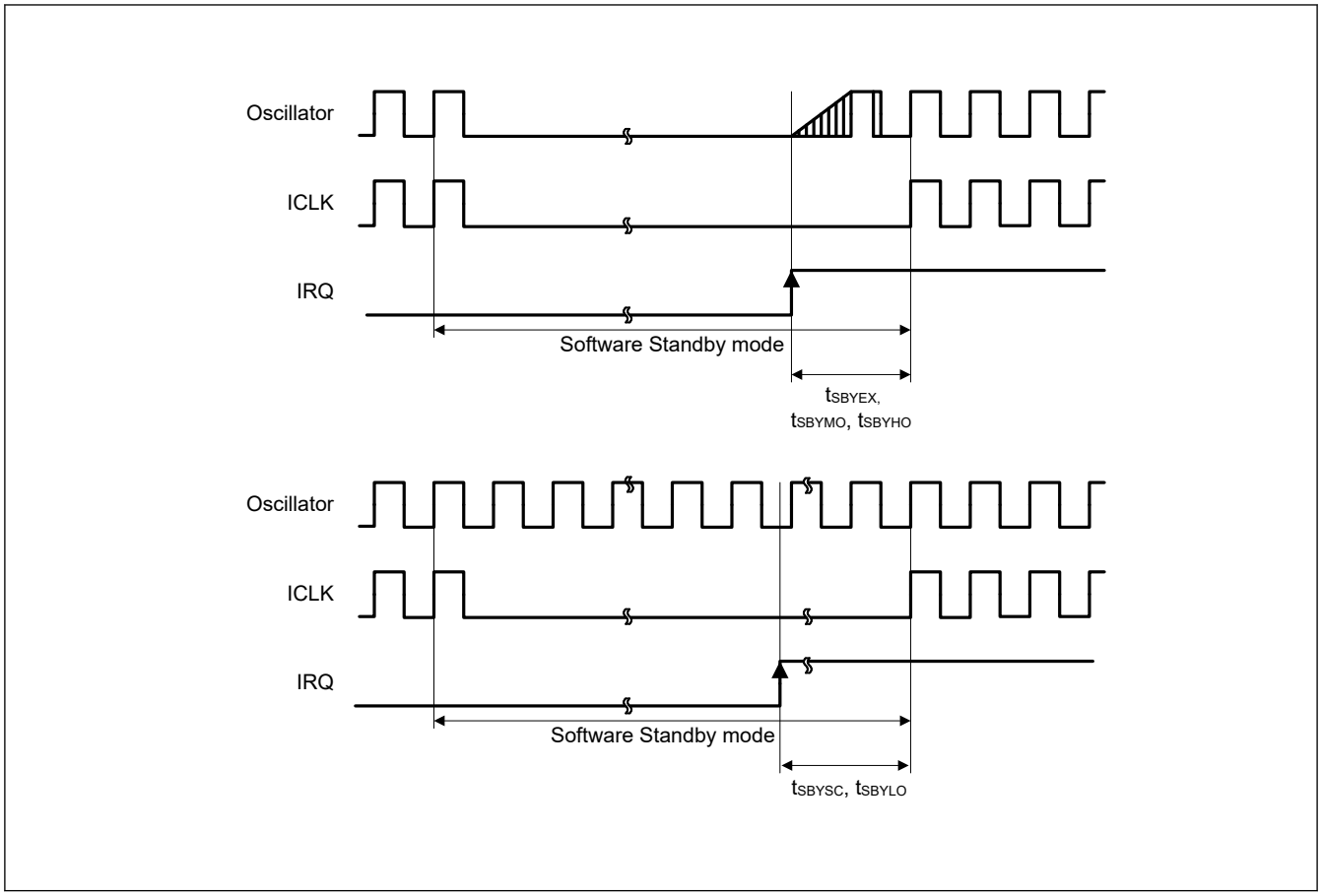
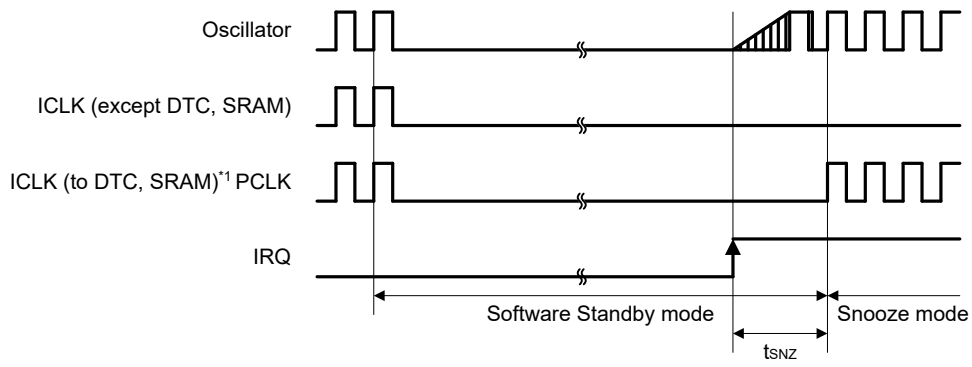


Figure 37.10 Software Standby mode cancellation timing

Table 37.25 Timing of recovery from low power modes (5)

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Recovery time from Software Standby mode to Snooze mode	High-speed mode System clock source is HOCO	t_{SNZ}	—	6.6	8.1	µs	Figure 37.11
	Middle-speed mode System clock source is HOCO (24 MHz) VCC = 1.8 V to 5.5 V	t_{SNZ}	—	6.7	8.2	µs	
	Middle-speed mode System clock source is HOCO (24 MHz) VCC = 1.6 V to 1.8 V	t_{SNZ}	—	10.8	12.9	µs	
	Low-speed mode System clock source is MOCO (1 MHz)	t_{SNZ}	—	9.2	16	µs	



Note 1. When SNZCR.SNZDTCEN bit is set to 1, ICLK is supplied to DTC and SRAM.

Figure 37.11 Recovery time from Software Standby mode to Snooze mode

37.3.5 NMI and IRQ Noise Filter

Table 37.26 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 being switched it is needed to add 4 clock cycle of 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 (i = 0 to 7).

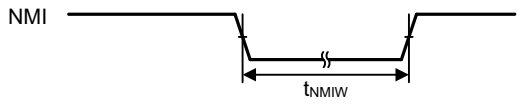


Figure 37.12 NMI interrupt input timing

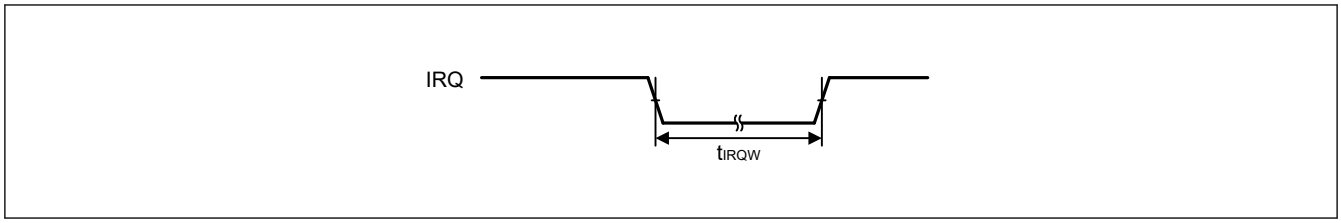


Figure 37.13 IRQ interrupt input timing

37.3.6 I/O Ports, KINT and ADC12 Trigger Timing

Table 37.27 I/O Ports, KINT and ADC12 trigger timing

Parameter			Symbol	Min	Max	Unit*1	Test conditions
I/O Ports	Input data pulse width	1.6 V ≤ VCC ≤ 5.5 V	t_{PRW}	2	—	t_{Pcyc}	Figure 37.14
KINT	KRn (n = 00 to 05) pulse width		t_{KR}	250	—	ns	Figure 37.15

Note: If the clock source is being switched, add 4 clock cycles to the switched source.
Note 1. t_{Pcyc}: PCLKB cycle

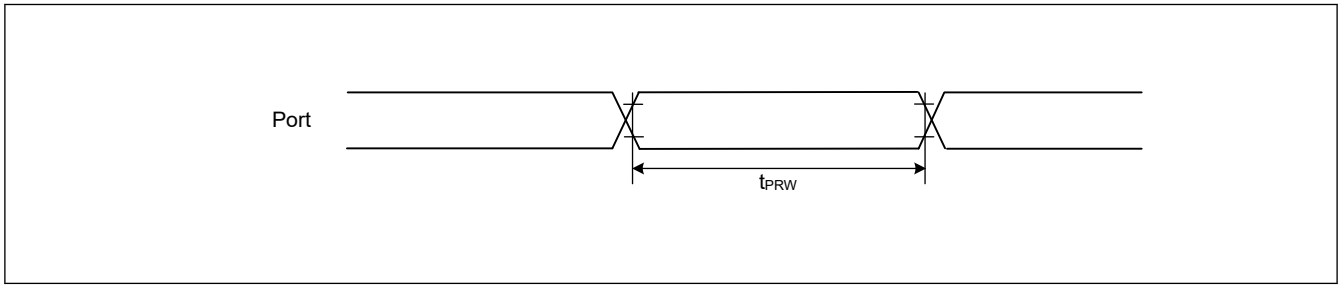


Figure 37.14 I/O ports input timing

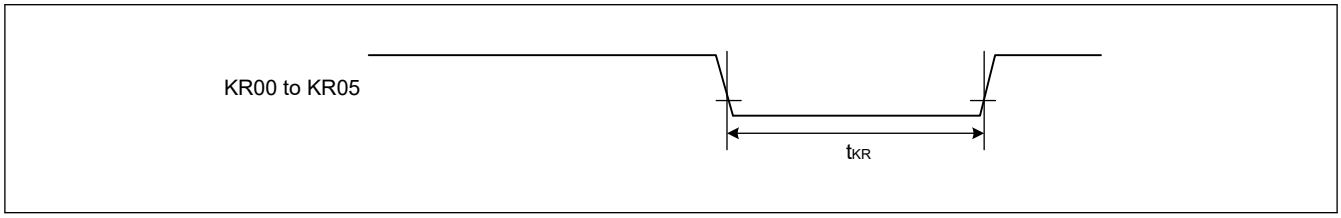


Figure 37.15 KINT input timing

37.3.7 TAU Timing

Table 37.28 TAU timing

Conditions: Ta = -40 to +125°C, VCC = 1.6 to 5.5 V

Parameter			Symbol	Min	Typ	Max	Unit	Test conditions
TI00 to TI07 input high-level width			t_{TIH}	$1/f_{MCK}$ + 10	—	—	ns	Figure 37.16
TI00 to TI07 input low-level width			t_{TIL}		—	—		
TO00 to TO07 output frequency	High-speed mode	2.7 V ≤ VCC ≤ 5.5 V	f_{TO}	—	—	24	MHz	
		2.4 V ≤ VCC ≤ 2.7 V		—	—	12		
		1.8 V ≤ VCC ≤ 2.4 V		—	—	6		
	Middle-speed mode	2.7 V ≤ VCC ≤ 5.5 V		—	—	24		
		2.4 V ≤ VCC ≤ 2.7 V		—	—	12		
		1.8 V ≤ VCC ≤ 2.4 V		—	—	6		
		1.6 V ≤ VCC ≤ 1.8 V		—	—	2		
	Low-speed mode	1.6 V ≤ VCC ≤ 5.5 V		—	—	1		

Note: f_{MCK}: Timer array unit operating clock frequency.

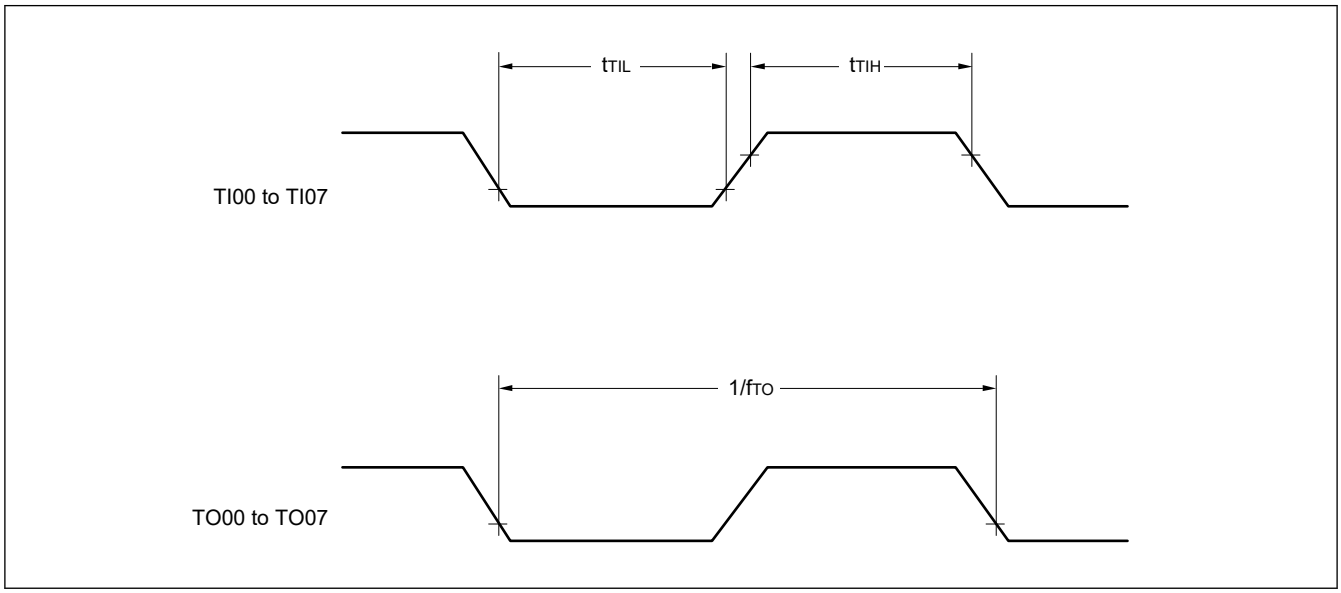


Figure 37.16 TAU input/output timing

37.3.8 CAC Timing

Table 37.29 CAC timing

Conditions: VCC = 1.6 to 5.5 V

Parameter			Symbol	Min	Typ	Max	Unit	Test conditions
CAC	CACREF input pulse width	t_{Pcyc}*1 ≤ t_{CAC}*2	t_{CACREF}	4.5 × t_{CAC} + 3 × t_{Pcyc}	—	—	ns	—
		t_{Pcyc}*1 > t_{CAC}*2		5 × t_{CAC} + 6.5 × t_{Pcyc}	—	—	ns	

Note 1. t_{Pcyc}: PCLKB cycle.
Note 2. t_{CAC}: CAC count clock source cycle.

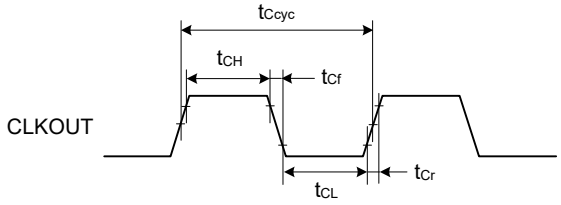
37.3.9 CLKOUT Timing

Table 37.30 CLKOUT timing

Parameter			Symbol	Min	Max	Unit	Test conditions
CLKOUT	CLKOUT pin output cycle*1	2.7 V ≤ VCC ≤ 5.5 V	t_{Ccyc}	62.5	—	ns	Figure 37.17
		1.8 V ≤ VCC < 2.7 V		125	—		
		1.6 V ≤ VCC < 1.8 V		250	—		
	CLKOUT pin high pulse width*2	2.7 V ≤ VCC ≤ 5.5 V	t_{CH}	15	—	ns	
		1.8 V ≤ VCC < 2.7 V		30	—		
		1.6 V ≤ VCC < 1.8 V		150	—		
	CLKOUT pin low pulse width*2	2.7 V ≤ VCC ≤ 5.5 V	t_{CL}	15	—	ns	
		1.8 V ≤ VCC < 2.7 V		30	—		
		1.6 V ≤ VCC < 1.8 V		150	—		
	CLKOUT pin output rise time	2.7 V ≤ VCC ≤ 5.5 V	t_{Cr}	—	12	ns	
		1.8 V ≤ VCC < 2.7 V		—	25		
		1.6 V ≤ VCC < 1.8 V		—	50		
	CLKOUT pin output fall time	2.7 V ≤ VCC ≤ 5.5 V	t_{Cf}	—	12	ns	
		1.8 V ≤ VCC < 2.7 V		—	25		
		1.6 V ≤ VCC < 1.8 V		—	50		

Note 1. When the EXTAL external clock input is used with division by 1 (the CKOCR.CKOSEL[2:0] bits are 011b and the CKOCR.CKODIV[2:0] bits are 000b) to output from CLKOUT, specifications in Table 37.30 should be satisfied with 45% to 55% of input duty cycle.

Note 2. When MOCO is selected as the clock output source (the CKOCR.CKOSEL[2:0] bits are 001b), set the clock output division ratio to be divided by 2 (the CKOCR.CKODIV[2:0] bits are 001b).



Test conditions: V_{OH} = VCC × 0.7, V_{OL} = VCC × 0.3, I_{OH} = -1.0 mA, I_{OL} = 1.0 mA, C = 30 pF

Figure 37.17 CLKOUT output timing

37.3.10 Serial Array Unit (SAU)

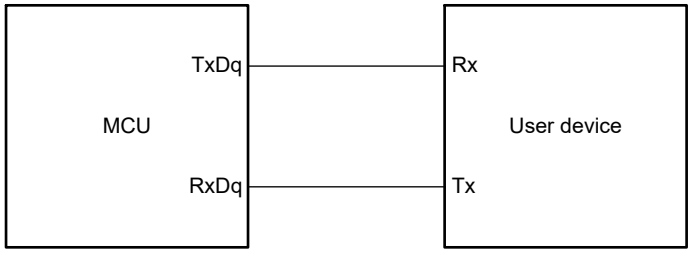
Table 37.31 UART communication

Conditions: T_a = -40 to +125°C, VCC = 1.6 to 5.5 V, VSS = 0 V

Parameter			Symbol	High-speed mode		Middle-speed mode		Low-speed mode		Unit	Test conditions
				Min.	Max.	Min.	Max.	Min.	Max.		
Transfer rate*1	1.6 V ≤ VCC ≤ 5.5 V		—	—	f_{MCK}/6	—	f_{MCK}/6	—	f_{MCK}/6	bps	Figure 37.18 Figure 37.19
		Theoretical value of the maximum transfer rate f_{MCK} = PCLKB*2		—	5.3	—	4	—	0.16	Mbps	

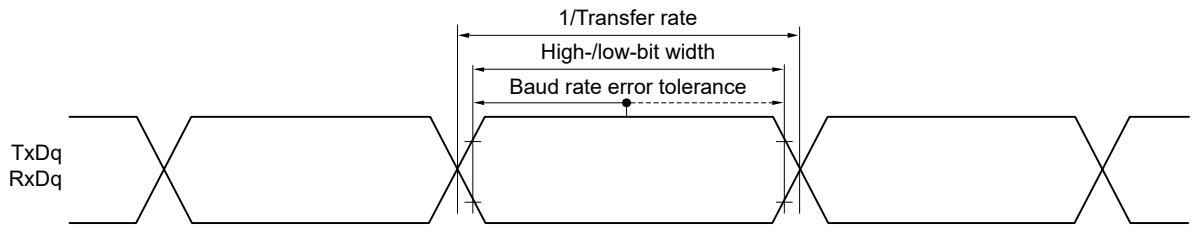
Note: Select the CMOS output for the TxDq pin by using NCODR bit in Port gh Pin Function Select Register (PghPFS).
Note:
- q: UART number (q = 0 to 2), gh: Port number (g = 0 to 4, h = 00 to 15)
- f_{MCK}: Serial array unit operation clock frequency

Note 1. The transfer rate in the Snooze mode is within the range from 4800 to 9600 bps.
Note 2. The maximum operating frequencies of PCLKB are as follows:
High-speed mode: 32 MHz (1.8 V ≤ VCC ≤ 5.5 V)
Middle-speed mode: 24 MHz (1.8 V ≤ VCC ≤ 5.5 V), 4 MHz (1.6 V ≤ VCC ≤ 5.5 V)
Low-speed mode: 1 MHz (1.6 V ≤ VCC ≤ 5.5 V)



Note: q: UART number (q = 0 to 2)

Figure 37.18 Connection in the UART communications



Note: q: UART number (q = 0 to 2)

Figure 37.19 Bit width in the UART communications

Table 37.32 Simplified SPI communication in master mode (only for SPI00)

Conditions: T_a = -40 to +125°C, VCC = 2.7 to 5.5 V, VSS = 0 V

Parameter			Symbol	High-speed mode		Middle-speed mode		Low-speed mode		Unit	Test conditions
				Min.	Max.	Min.	Max.	Min.	Max.		
SCKp cycle time	$t_{KCY1} \geq 2/PCLKB$	4.0 V ≤ VCC ≤ 5.5 V	t_{KCY1}	62.5	—	83.3	—	1000	—	ns	Figure 37.21 Figure 37.22
		2.7 V ≤ VCC ≤ 5.5 V		83.3	—	125	—	1000	—	ns	
SCKp high-/low-level width	4.0 V ≤ VCC ≤ 5.5 V		t_{KH1}, t_{KL1}	$t_{KCY1}/2$ - 7	—	$t_{KCY1}/2$ - 10	—	$t_{KCY1}/2$ - 50	—	ns	
	2.7 V ≤ VCC ≤ 5.5 V			$t_{KCY}1/2$ - 10	—	$t_{KCY1}/2$ - 15	—	$t_{KCY1}/2$ - 50	—	ns	
SIp setup time (to SCKp↑)*1	4.0 V ≤ VCC ≤ 5.5 V		t_{SIK1}	23	—	33	—	110	—	ns	
	2.7 V ≤ VCC ≤ 5.5 V			33	—	50	—	110	—	ns	
SIp hold time (from SCKp↑)*1	2.7 V ≤ VCC ≤ 5.5 V		t_{KSI1}	10	—	10	—	10	—	ns	
Delay time from SCKp↓ to SOp output*2	C = 20 pF*3		t_{KSO1}	—	10	—	10	—	10	ns	

Note: Select the CMOS output for the SOp pin and SCKp pin by using NCODR bit in Port gh Pin Function Select Register (PghPFS).
Note: p: SPI number (p = 00), m: Unit number (m = 0), n: Channel number (n = 0), gh: Port number (g = 0 to 4, h = 00 to 15).
Note 1. The setting applies when SCRmn.DCP[1:0] = 00b or 11b. The setting for the SIp setup time changes to SCKp↓ and that for the SIp hold time changes from SCKp↓ when SCRmn.DCP[1:0] = 01b or 10b.
Note 2. This setting applies when SCRmn.DCP[1:0] = 00b or 11b. The setting for the delay time to SOp output changes from SCKp↑ when SCRmn.DCP[1:0] = 01b or 10b.
Note 3. C is the load capacitance of the SCKp and SOp output lines.

Table 37.33 Simplified SPI communication in master mode (except for SPI00)

Conditions: T_a = -40 to +125°C, VCC = 1.6 to 5.5 V, VSS = 0 V

Parameter			Symbol	High-speed mode*1		Middle-speed mode		Low-speed mode		Unit	Test conditions
				Min.	Max.	Min.	Max.	Min.	Max.		
SCKp cycle time	t_{KCY1} ≥ 4/ PCLKB	2.7 V ≤ VCC ≤ 5.5 V	t_{KCY1}	125	—	166	—	2000	—	ns	Figure 37.21 Figure 37.22
		2.4 V ≤ VCC ≤ 5.5 V		250	—	250	—	2000	—	ns	
		1.8 V ≤ VCC ≤ 5.5 V		500	—	500	—	2000	—	ns	
		1.6 V ≤ VCC ≤ 5.5 V		—	—	1000	—	2000	—	ns	
SCKp high-/low-level width	4.0 V ≤ VCC ≤ 5.5 V		t_{KH1}, t_{KL1}	t_{KCY1}/2 - 12	—	t_{KCY1}/2 - 21	—	t_{KCY1}/2 - 50	—	ns	
	2.7 V ≤ VCC ≤ 5.5 V			t_{KCY1}/2 - 18	—	t_{KCY1}/2 - 25	—	t_{KCY1}/2 - 50	—	ns	
	2.4 V ≤ VCC ≤ 5.5 V			t_{KCY1}/2 - 38	—	t_{KCY1}/2 - 38	—	t_{KCY1}/2 - 50	—	ns	
	1.8 V ≤ VCC ≤ 5.5 V			t_{KCY1}/2 - 50	—	t_{KCY1}/2 - 50	—	t_{KCY1}/2 - 50	—	ns	
	1.6 V ≤ VCC ≤ 5.5 V			—	—	t_{KCY1}/2 - 100	—	t_{KCY1}/2 - 100	—	ns	
SIp setup time (to SCKp↑)*2	4.0 V ≤ VCC ≤ 5.5 V		t_{SIK1}	44	—	54	—	110	—	ns	
	2.7 V ≤ VCC ≤ 5.5 V			44	—	54	—	110	—	ns	
	2.4 V ≤ VCC ≤ 5.5 V			75	—	75	—	110	—	ns	
	1.8 V ≤ VCC ≤ 5.5 V			110	—	110	—	110	—	ns	
	1.6 V ≤ VCC ≤ 5.5 V			—	—	220	—	220	—	ns	
SIp hold time (from SCKp↑)*2	1.6 V ≤ VCC ≤ 5.5 V		t_{KSI1}	19	—	19	—	19	—	ns	
Delay time from SCKp↓ to SOp output*3	1.6 V ≤ VCC ≤ 5.5 V C = 30 pF*4		t_{KSO1}	—	25	—	25	—	25	ns	

Note: Select the CMOS output for the SOp pin and SCKp pin by using NCODR bit in Port gh Pin Function Select Register (PghPFS).
Note: p: SPI number (p = 00, 01, 10, 11, 20, 21), m: Unit number, n: Channel number (mn = 00 to 03, 10 to 11), gh: Port number (g = 0 to 4, h = 00 to 15).
Note 1. Operating voltages in high-speed mode are 1.8 V ≤ VCC ≤ 5.5 V.
Note 2. This setting applies when SCRmn.DCP[1:0] = 00b or 11b. The setting for the SIp setup time changes to SCKp↓ and that for the SIp hold time changes from SCKp↓ when SCRmn.DCP[1:0] = 01b or 10b.
Note 3. This setting applies when SCRmn.DCP[1:0] = 00b or 11b. The setting for the delay time to SOp output changes from SCKp↑ when SCRmn.DCP[1:0] = 01b or 10b.
Note 4. C is the load capacitance of the SCKp and SOp output lines.

Table 37.34 Simplified SPI communication in slave mode

Conditions: T_a = -40 to +125°C, VCC = 1.6 to 5.5 V, VSS = 0 V

Parameter			Symbol	High-speed mode*1		Middle-speed mode		Low-speed mode		Unit	Test conditions
				Min.	Max.	Min.	Max.	Min.	Max.		
SCKp cycle time*2	4.0 V ≤ VCC ≤ 5.5 V	20 MHz < f_{MCK}	t_{KCY2}	8/f_{MCK}	—	8/f_{MCK}	—	—	—	ns	Figure 37.21 Figure 37.22
		f_{MCK} ≤ 20 MHz		6/f_{MCK}	—	6/f_{MCK}	—	6/f_{MCK}	—	ns	
	2.7 V ≤ VCC ≤ 5.5 V	16 MHz < f_{MCK}		8/f_{MCK}	—	8/f_{MCK}	—	—	—	ns	
		f_{MCK} ≤ 16 MHz		6/f_{MCK}	—	6/f_{MCK}	—	6/f_{MCK}	—	ns	
	2.4 V ≤ VCC ≤ 5.5 V			6/f_{MCK} + 500	—	6/f_{MCK} + 500	—	6/f_{MCK} + 500	—	ns	
	1.8 V ≤ VCC ≤ 5.5 V			6/f_{MCK} + 750	—	6/f_{MCK} + 750	—	6/f_{MCK} + 750	—	ns	
	1.6 V ≤ VCC ≤ 5.5 V			—	—	6/f_{MCK} + 1500	—	6/f_{MCK} + 1500	—	ns	
SCKp high-/low-level width	4.0 V ≤ VCC ≤ 5.5 V		t_{KH2}, t_{KL2}	t_{KCY2}/2 - 7	—	t_{KCY2}/2 - 7	—	t_{KCY2}/2 - 7	—	ns	
	2.7 V ≤ VCC ≤ 5.5 V			t_{KCY2}/2 - 8	—	t_{KCY2}/2 - 8	—	t_{KCY2}/2 - 8	—	ns	
	1.8 V ≤ VCC ≤ 5.5 V			t_{KCY2}/2 - 18	—	t_{KCY2}/2 - 18	—	t_{KCY2}/2 - 18	—	ns	
	1.6 V ≤ VCC ≤ 5.5 V			—	—	t_{KCY2}/2 - 66	—	t_{KCY2}/2 - 66	—	ns	
SIp setup time (to SCKp↑)*3	2.7 V ≤ VCC ≤ 5.5 V		t_{SIK2}	1/f_{MCK} + 20	—	1/f_{MCK} + 30	—	1/f_{MCK} + 30	—	ns	
	1.8 V ≤ VCC ≤ 5.5 V			1/f_{MCK} + 30	—	1/f_{MCK} + 30	—	1/f_{MCK} + 30	—	ns	
	1.6 V ≤ VCC ≤ 5.5 V			—	—	1/f_{MCK} + 40	—	1/f_{MCK} + 40	—	ns	
SIp hold time (from SCKp↑)*3	1.8 V ≤ VCC ≤ 5.5 V		t_{KSI2}	1/f_{MCK} + 31	—	1/f_{MCK} + 31	—	1/f_{MCK} + 31	—	ns	
	1.6 V ≤ VCC ≤ 5.5 V			—	—	1/f_{MCK} + 250	—	1/f_{MCK} + 250	—	ns	
Delay time from SCKp↓ to SOp output*4	C = 30 pF*5	2.7 V ≤ VCC ≤ 5.5 V	t_{KSO2}	—	2/f_{MCK} + 44	—	2/f_{MCK} + 110	—	2/f_{MCK} + 110	ns	
		2.4 V ≤ VCC ≤ 5.5 V		—	2/f_{MCK} + 75	—	2/f_{MCK} + 110	—	2/f_{MCK} + 110	ns	
		1.8 V ≤ VCC ≤ 5.5 V		—	2/f_{MCK} + 110	—	2/f_{MCK} + 110	—	2/f_{MCK} + 110	ns	
		1.6 V ≤ VCC ≤ 5.5 V		—	—	—	2/f_{MCK} + 220	—	2/f_{MCK} + 220	ns	

Note: Select the CMOS output for the SOp pin by using NCODR bit in Port gh Pin Function Select Register (PghPFS).

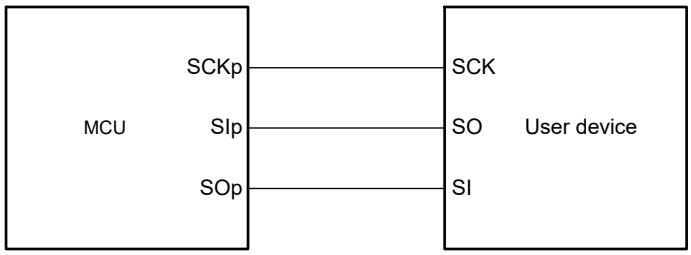
Note:
- p: SPI number (p = 00, 01, 10, 11, 20, 21), m: Unit number, n: Channel number (mn = 00 to 03, 10 to 11), gh: Port number (g = 0 to 4, h = 00 to 15)
- f_{MCK}: Serial array unit operation clock frequency

Note 1. Operating voltages in high-speed mode are 1.8 V ≤ VCC ≤ 5.5 V.

Note 2. Transfer rate in the Snooze mode is 0.5 Mbps at the maximum.

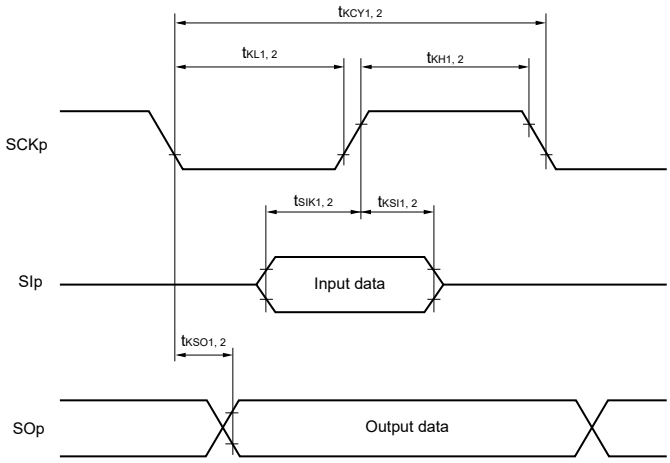
Note 3. This setting applies when SCRmn.DCP[1:0] = 00b or 11b. The setting for the SIp setup time changes to SCKp↓ and that for the SIp hold time changes from SCKp↓ when SCRmn.DCP[1:0] = 01b or 10b.

Note 4. This setting applies when SCRmn.DCP[1:0] = 00b or 11b. The setting for the delay time to SOp output changes from SCKp↑ when SCRmn.DCP[1:0] = 01b or 10b.
Note 5. C is the load capacitance of the SOp output line.



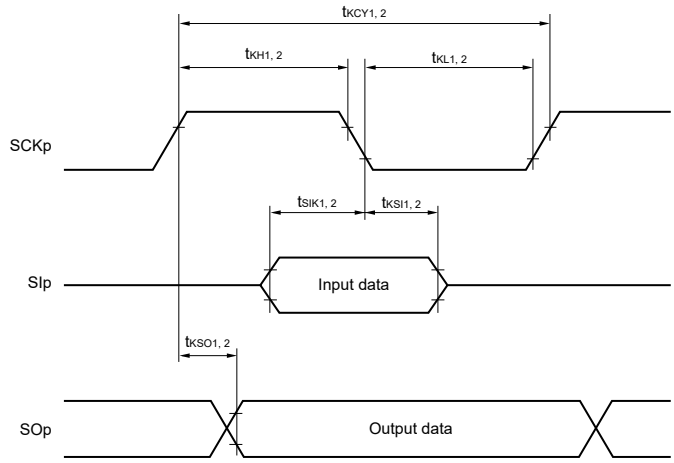
Note: p: SPI number (p = 00, 01, 10, 11, 20, 21)

Figure 37.20 Connection in the simplified SPI communications



Note:
- p: SPI number (p = 00, 01, 10, 11, 20, 21)
- m: Unit number, n: Channel number (mn = 00 to 03, 10 to 11)

Figure 37.21 Timing of serial transfer in the simplified SPI communications when SCRmn.DCP[1:0] = 00b or 11b



Note:
- p: SPI number (p = 00, 01, 10, 11, 20, 21)
- m: Unit number, n: Channel number (mn = 00 to 03, 10 to 11)

Figure 37.22 Timing of serial transfer in the simplified SPI communications when SCRmn.DCP[1:0] = 01b or 10b

Table 37.35 Simplified I²C communication

Conditions: T_a = -40 to +125°C, VCC = 1.6 to 5.5 V, VSS = 0 V

Parameter		Symbol	High-speed mode*1		Middle-speed mode		Low-speed mode		Unit	Test conditions
			Min.	Max.	Min.	Max.	Min.	Max.		
SCLr clock frequency	2.7 V ≤ VCC ≤ 5.5 V, Cb = 50 pF, Rb = 2.7 kΩ	f_{SCL}	—	1000*2	—	1000*2	—	400*2	kHz	Figure 37.23 Figure 37.24
	1.8 V ≤ VCC ≤ 5.5 V, Cb = 100 pF, Rb = 3 kΩ		—	400*2	—	400*2	—	400*2	kHz	
	1.8 V ≤ VCC < 2.7 V, Cb = 100 pF, Rb = 5 kΩ		—	300*2	—	300*2	—	300*2	kHz	
	1.6 V ≤ VCC < 1.8 V, Cb = 100 pF, Rb = 5 kΩ		—	—	—	250*2	—	250*2	kHz	
Hold time when SCLr is low	2.7 V ≤ VCC ≤ 5.5 V, Cb = 50 pF, Rb = 2.7 kΩ	t_{LOW}	475	—	475	—	1150	—	ns	
	1.8 V ≤ VCC ≤ 5.5 V, Cb = 100 pF, Rb = 3 kΩ		1150	—	1150	—	1150	—	ns	
	1.8 V ≤ VCC < 2.7 V, Cb = 100 pF, Rb = 5 kΩ		1550	—	1550	—	1550	—	ns	
	1.6 V ≤ VCC < 1.8 V, Cb = 100 pF, Rb = 5 kΩ		—	—	1850	—	1850	—	ns	
Hold time when SCLr is high	2.7 V ≤ VCC ≤ 5.5 V, Cb = 50 pF, Rb = 2.7 kΩ	t_{HIGH}	475	—	475	—	1150	—	ns	
	1.8 V ≤ VCC ≤ 5.5 V, Cb = 100 pF, Rb = 3 kΩ		1150	—	1150	—	1150	—	ns	
	1.8 V ≤ VCC < 2.7 V, Cb = 100 pF, Rb = 5 kΩ		1550	—	1550	—	1550	—	ns	
	1.6 V ≤ VCC < 1.8 V, Cb = 100 pF, Rb = 5 kΩ		—	—	1850	—	1850	—	ns	
Data setup time (reception)	2.7 V ≤ VCC ≤ 5.5 V, Cb = 50 pF, Rb = 2.7 kΩ	$t_{SU:DAT}$	$1/f_{MCK}$ + 85*3	—	$1/f_{MCK}$ +85*3	—	$1/f_{MCK}$ +145*3	—	ns	
	1.8 V ≤ VCC ≤ 5.5 V, Cb = 100 pF, Rb = 3 kΩ		$1/f_{MCK}$ + 145*3	—	$1/f_{MCK}$ + 145*3	—	$1/f_{MCK}$ +145*3	—	ns	
	1.8 V ≤ VCC < 2.7 V, Cb = 100 pF, Rb = 5 kΩ		$1/f_{MCK}$ + 230*3	—	$1/f_{MCK}$ + 230*3	—	$1/f_{MCK}$ + 230*3	—	ns	
	1.6 V ≤ VCC < 1.8 V, Cb = 100 pF, Rb = 5 kΩ		—	—	$1/f_{MCK}$ + 290*3	—	$1/f_{MCK}$ + 290*3	—	ns	
Data hold time (transmission)	2.7 V ≤ VCC ≤ 5.5 V, Cb = 50 pF, Rb = 2.7 kΩ	$t_{HD:DAT}$	0	305	0	305	0	305	ns	
	1.8 V ≤ VCC ≤ 5.5 V, Cb = 100 pF, Rb = 3 kΩ		0	355	0	355	0	355	ns	
	1.8 V ≤ VCC < 2.7 V, Cb = 100 pF, Rb = 5 kΩ		0	405	0	405	0	405	ns	
	1.6 V ≤ VCC < 1.8 V, Cb = 100 pF, Rb = 5 kΩ		—	—	0	405	0	405	ns	

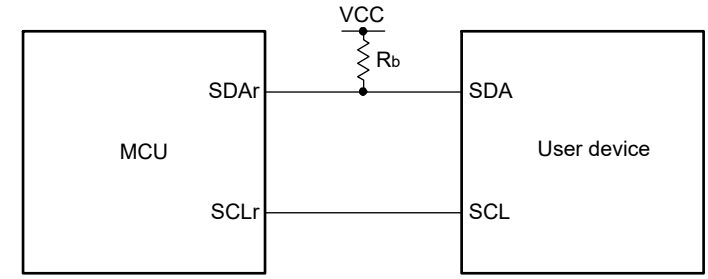
Note: Select the NMOS open-drain output for the SDAr pin and the CMOS output for the SCLr pin by using NCODR bit in Port gh Pin Function Select Register (PghPFS).

Note:
- r: IIC number (r = 00, 01, 10, 11, 20, 21), gh: Port number (g = 0 to 4, h = 00 to 15)
- f_{MCK}: Serial array unit operation clock frequency
- Rb[Ω]: Communication line (SDAr) pull-up resistance, Cb[F]: Communication line (SDAr, SCLr) load capacitance

Note 1. Operating voltages in high-speed mode are 1.8 V ≤ VCC ≤ 5.5 V.

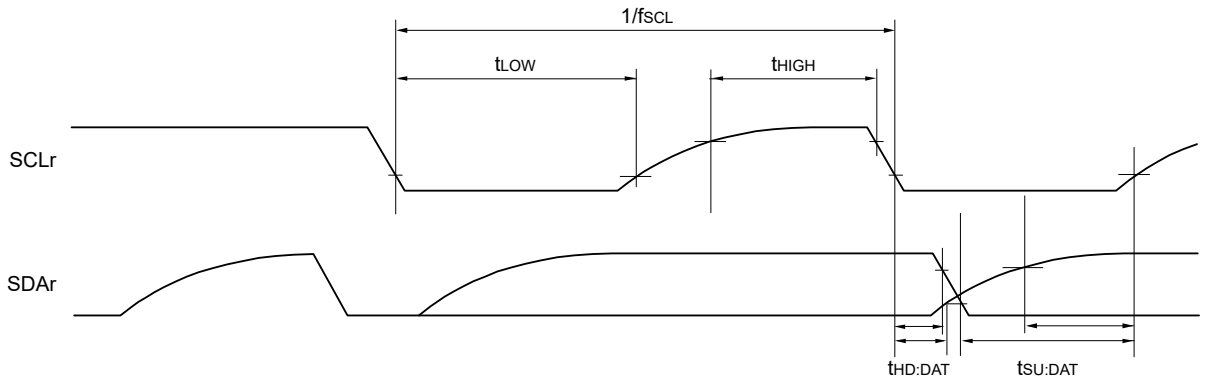
Note 2. The listed times must be no greater than $f_{MCK}/4$.

Note 3. Set f_{MCK} so that it will not exceed the hold time when SCLr is low or high.



Note: r: IIC number (r = 00, 01, 10, 11, 20, 21)

Figure 37.23 Connection in the simplified I²C communications



Note: r: IIC number (r = 00, 01, 10, 11, 20, 21)

Figure 37.24 Timing of serial transfer in the simplified I²C communications

37.3.11 Serial Interface UARTA (UARTA)

Table 37.36 UARTA communications

Conditions: T_a = -40 to +125°C, VCC = 1.6 to 5.5 V, VSS = 0 V

Parameter	Symbol	Min.	Typ.	Max.	Unit	Test conditions
Transfer rate	—	200	—	153600	bps	—

Note: Select the CMOS output for the TxDAn pin by using NCODR bit in Port gh Pin Function Select Register (PghPFS).
Note: n: Unit number (n = 0, 1), gh: Port number (g = 0 to 4, h = 00 to 15).

37.3.12 I^2C Bus Interface (IICA)

Table 37.37 I^2C standard mode

Conditions: T_a = -40 to +125°C, VCC = 1.6 to 5.5 V, VSS = 0 V

Parameter		Symbol	Min.	Typ.	Max.	Unit	Test conditions
SCLAn clock frequency	Standard mode: PCLKB ≥ 1 MHz	f_{SCL}	0	—	100	kHz	Figure 37.25
Setup time of restart condition	—	$t_{SU:STA}$	4.7	—	—	µs	
Hold time*1	—	$t_{HD:STA}$	4	—	—	µs	
Hold time when SCLAn is low	—	t_{LOW}	4.7	—	—	µs	
Hold time when SCLAn is high	—	t_{HIGH}	4	—	—	µs	
Data setup time (reception)	—	$t_{SU:DAT}$	250	—	—	ns	
Data hold time (transmission)*2	—	$t_{HD:DAT}$	0	—	3.45	µs	
Setup time of stop condition	—	$t_{SU:STO}$	4	—	—	µs	
Bus-free time	—	t_{BUF}	4.7	—	—	µs	

Note: n = 0, 1
Note: The maximum value of communication line capacitance (Cb) and communication line pull-up resistor (Rb) are as follows.
Cb = 400 pF, Rb = 2.7 kΩ
Note 1. The first clock pulse is generated after this period when the start or restart condition is detected.
Note 2. The maximum value of $t_{HD:DAT}$ applies to normal transfer. The clock stretching is inserted on reception of an acknowledgment (ACK) signal.

Table 37.38 I^2C fast mode

Conditions: T_a = -40 to +125°C, VCC = 1.8 to 5.5 V, VSS = 0 V

Parameter		Symbol	Min.	Typ.	Max.	Unit	Test conditions
SCLAn clock frequency	Fast mode: PCLKB ≥ 3.5 MHz	f_{SCL}	0	—	400	kHz	Figure 37.25
Setup time of restart condition	—	$t_{SU:STA}$	0.6	—	—	µs	
Hold time*1	—	$t_{HD:STA}$	0.6	—	—	µs	
Hold time when SCLAn is low	—	t_{LOW}	1.3	—	—	µs	
Hold time when SCLAn is high	—	t_{HIGH}	0.6	—	—	µs	
Data setup time (reception)	—	$t_{SU:DAT}$	100	—	—	ns	
Data hold time (transmission)*2	—	$t_{HD:DAT}$	0	—	0.9	µs	
Setup time of stop condition	—	$t_{SU:STO}$	0.6	—	—	µs	
Bus-free time	—	t_{BUF}	1.3	—	—	µs	

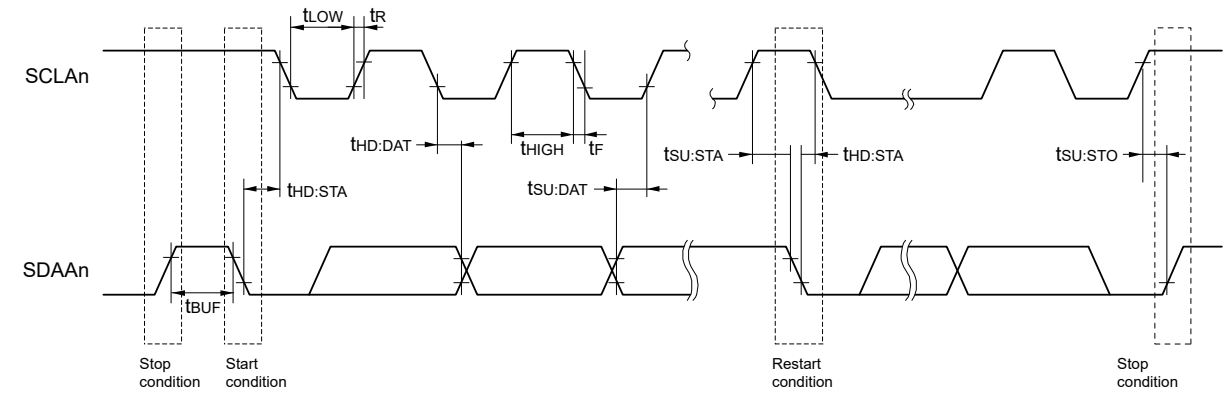
Note: n = 0, 1
Note: The maximum value of communication line capacitance (Cb) and communication line pull-up resistor (Rb) are as follows.
Cb = 320 pF, Rb = 1.1 kΩ
Note 1. The first clock pulse is generated after this period when the start or restart condition is detected.
Note 2. The maximum value of $t_{HD:DAT}$ applies to normal transfer. The clock stretching is inserted on reception of an acknowledgment (ACK) signal.

Table 37.39 I²C fast mode plus

Conditions: T_a = -40 to +125°C, VCC = 2.7 to 5.5 V, VSS = 0 V

Parameter		Symbol	Min.	Typ.	Max.	Unit	Test conditions
SCLAn clock frequency	Fast mode plus: PCLKB ≥ 10 MHz	f_{SCL}	0	—	1000	kHz	Figure 37.25
Setup time of restart condition	—	$t_{SU:STA}$	0.26	—	—	µs	
Hold time*1	—	$t_{HD:STA}$	0.26	—	—	µs	
Hold time when SCLAn is low	—	t_{LOW}	0.5	—	—	µs	
Hold time when SCLAn is high	—	t_{HIGH}	0.26	—	—	µs	
Data setup time (reception)	—	$t_{SU:DAT}$	50	—	—	ns	
Data hold time (transmission)*2	—	$t_{HD:DAT}$	0	—	0.45	µs	
Setup time of stop condition	—	$t_{SU:STO}$	0.26	—	—	µs	
Bus-free time	—	t_{BUF}	0.5	—	—	µs	

Note: n = 0, 1
Note: The maximum value of communication line capacitance (Cb) and communication line pull-up resistor (Rb) are as follows.
Cb = 120 pF, Rb = 1.1 kΩ
Note 1. The first clock pulse is generated after this period when the start or restart condition is detected.
Note 2. The maximum value of $t_{HD:DAT}$ applies to normal transfer. The clock stretching is inserted on reception of an acknowledgment (ACK) signal.



Note: n = 0, 1

Figure 37.25 I²C serial transfer timing

37.4 ADC12 Characteristics

Table 37.40 A/D conversion characteristics (1) in normal modes 1 and 2 (1 of 2)

Conditions: VCC = AVREFP = 4.5 to 5.5 V, VSS = AVREFM = 0 V
Reference voltage (+) = AVREFP, Reference voltage (-) = AVREFM
Target pins for conversion: ANI2 to ANI5, internal reference voltage, and temperature sensor output voltage

Parameter	Min	Typ	Max	Unit	Test conditions
Resolution	—	—	12	Bit	—
Conversion clock (fAD)	1	—	48	MHz	—
Conversion time*4	1.33	—	—	µs	—
Offset error*1 *2 *3	—	—	±7.0	LSB	—
Full-scale error*1 *2 *3	—	—	±7.0	LSB	—
Absolute accuracy*1 *2 *3	—	—	±7.5	LSB	—

Table 37.40 A/D conversion characteristics (1) in normal modes 1 and 2 (2 of 2)

Conditions: VCC = AVREFP = 4.5 to 5.5 V, VSS = AVREFM = 0 V
Reference voltage (+) = AVREFP, Reference voltage (-) = AVREFM
Target pins for conversion: ANI2 to ANI5, internal reference voltage, and temperature sensor output voltage

Parameter	Min	Typ	Max	Unit	Test conditions
DNL differential nonlinearity error*1	—	±1.0	—	LSB	—
INL integral nonlinearity error*1 *3 *3	—	—	±3.0	LSB	—
Analog input voltage range	0	—	AVREFP	V	—

Note: These specification values apply during A/D conversion operation, while CPU is in Sleep mode and peripheral modules other than A/D in module standby.
If CPU is running or peripheral modules other than A/D are running during A/D conversion, values might not fall within the indicated ranges.
Note 1. This value does not include the quantization error (± 1/2 LSB).
Note 2. When pins ANI16 to ANI19 are selected as the target pins for conversion, the maximum values are as follows:
Absolute accuracy: Add ± 3 LSB to the maximum value.
Offset/full-scale error: Add ± 2 LSB to the maximum value.
Note 3. When AVREFP < VCC, the maximum values are as follows:
Absolute accuracy/Offset error/full-scale error: Add (± 0.75 LSB × (VCC voltage (V) - AVREFP voltage (V)) to the maximum value.
INL integral nonlinearity error: Add (± 0.2 LSB × (VDD voltage (V) - AVREFP voltage (V)) to the maximum value.
Note 4. When the internal reference voltage or the temperature sensor output voltage is selected as the target for conversion, the sampling time must be at least 5 µs. Accordingly, use normal mode 2 and fAD = 32 MHz or less with the longer sampling time.

Table 37.41 A/D conversion characteristics (2) in normal modes 1 and 2

Conditions: VCC = AVREFP = 2.7 to 5.5 V, VSS = AVREFM = 0 V
Reference voltage (+) = AVREFP, Reference voltage (-) = AVREFM
Target pins for conversions: ANI2 to ANI5, internal reference voltage, and temperature sensor output voltage

Parameter	Min	Typ	Max	Unit	Test conditions
Resolution	—	—	12	Bit	—
Conversion clock (fAD)	1	—	48	MHz	—
Conversion time*4	1.33	—	—	µs	—
Offset error*1 *2 *3	—	—	±8.5	LSB	—
Full-scale error *1 *2 *3	—	—	±8.5	LSB	—
Absolute accuracy*1 *2 *3	—	—	±9.0	LSB	—
DNL differential nonlinearity error*1	—	±1.0	—	LSB	—
INL integral nonlinearity error*1 *3	—	—	±3.0	LSB	—
Analog input voltage range	0	—	AVREFP	V	—

Note: These specification values apply during A/D conversion operation, while CPU is in Sleep mode and peripheral modules other than A/D in module standby.
If CPU is running or peripheral modules other than A/D are running during A/D conversion, values might not fall within the indicated ranges.
Note 1. This value does not include the quantization error (± 1/2 LSB).
Note 2. When pins ANI16 to ANI19 are selected as the target pins for conversion, the maximum values are as follows:
Absolute accuracy: Add ± 3 LSB to the maximum value.
Offset/full-scale error: Add ± 2 LSB to the maximum value.
Note 3. When AVREFP < VCC, the maximum values are as follows:
Absolute accuracy/Offset error/full-scale error: Add (± 0.75 LSB × (VCC voltage (V) - AVREFP voltage (V)) to the maximum value.
INL integral nonlinearity error: Add (± 0.2 LSB × (VDD voltage (V) - AVREFP voltage (V)) to the maximum value.
Note 4. When the internal reference voltage or the temperature sensor output voltage is selected as the target for conversion, the sampling time must be at least 5 µs. Accordingly, use normal mode 2 and fAD = 32 MHz or less with the longer sampling time.

Table 37.42 A/D conversion characteristics (3) in normal modes 1 and 2 (1 of 2)

Conditions: VCC = AVREFP = 2.4 to 5.5 V, VSS = AVREFM = 0 V
Reference voltage (+) = AVREFP, Reference voltage (-) = AVREFM
Target pins for conversions: ANI2 to ANI5, internal reference voltage, and temperature sensor output voltage

Parameter	Min	Typ	Max	Unit	Test conditions
Resolution	—	—	12	Bit	—
Conversion clock (PCLKB)	1	—	32	MHz	—

Table 37.42 A/D conversion characteristics (3) in normal modes 1 and 2 (2 of 2)

Conditions: VCC = AVREFP = 2.4 to 5.5 V, VSS = AVREFM = 0 V
Reference voltage (+) = AVREFP, Reference voltage (-) = AVREFM
Target pins for conversions: ANI2 to ANI5, internal reference voltage, and temperature sensor output voltage

Parameter	Min	Typ	Max	Unit	Test conditions
Conversion time*4	2.0	—	—	µs	—
Offset error*1 *2 *3	—	—	±9.0	LSB	—
Full-scale error *1 *2 *3	—	—	±9.0	LSB	—
Absolute accuracy*1 *2 *3	—	—	±9.5	LSB	—
DNL differential nonlinearity error*1	—	±1.0	—	LSB	—
INL integral nonlinearity error*1 *3	—	—	±3.0	LSB	—
Analog input voltage range	0	—	AVREFP	V	—

Note: These specification values apply during A/D conversion operation, while CPU is in Sleep mode and peripheral modules other than A/D in module standby.
If CPU is running or peripheral modules other than A/D are running during A/D conversion, values might not fall within the indicated ranges.
Note 1. This value does not include the quantization error (± 1/2 LSB).
Note 2. When pins ANI16 to ANI19 are selected as the target pins for conversion, the maximum values are as follows:
Absolute accuracy: Add ± 3 LSB to the maximum value.
Offset/full-scale error: Add ± 2 LSB to the maximum value.
Note 3. When AVREFP < VCC, the maximum values are as follows:
Absolute accuracy/Offset error/full-scale error: Add (± 0.75 LSB × (VCC voltage (V) - AVREFP voltage (V)) to the maximum value.
INL integral nonlinearity error: Add (± 0.2 LSB × (VDD voltage (V) - AVREFP voltage (V)) to the maximum value.
Note 4. When the internal reference voltage or the temperature sensor output voltage is selected as the target for conversion, the sampling time must be at least 5 µs. Accordingly, use normal mode 2 and fAD = 32 MHz or less with the longer sampling time.

Table 37.43 A/D conversion characteristics (1) in Low-voltage modes 1 and 2

Conditions: VCC = AVREFP = 2.7 to 5.5 V, VSS = AVREFM = 0 V
Reference voltage (+) = AVREFP, Reference voltage (-) = AVREFM
Target pins for conversion: ANI2 to ANI5, internal reference voltage*4, and temperature sensor output voltage*4

Parameter	Min	Typ	Max	Unit	Test conditions
Resolution	—	—	12	Bit	—
Conversion clock (fAD)	1	—	24	MHz	—
Conversion time*5	3.33	—	—	µs	—
Offset error*1 *2 *3	—	—	±8.5	LSB	—
Full-scale error *1 *2 *3	—	—	±8.5	LSB	—
Absolute accuracy*1 *2 *3	—	—	±9.0	LSB	—
DNL differential nonlinearity error*1	—	±1.5	—	LSB	—
INL integral nonlinearity error*1 *3	—	—	±4.0	LSB	—
Analog input voltage range	0	—	AVREFP	V	—

Note: These specification values apply during A/D conversion operation, while CPU is in Sleep mode and peripheral modules other than A/D in module standby.
If CPU is running or peripheral modules other than A/D are running during A/D conversion, values might not fall within the indicated ranges.
Note 1. This value does not include the quantization error (± 1/2 LSB).
Note 2. When pins ANI16 to ANI19 are selected as the target pins for conversion, the maximum values are as follows:
Absolute accuracy: Add ± 3 LSB to the maximum value.
Offset/full-scale error: Add ± 2 LSB to the maximum value.
Note 3. When AVREFP < VCC, the maximum values are as follows:
Absolute accuracy/Offset error/full-scale error: Add (± 0.75 LSB × (VCC voltage (V) - AVREFP voltage (V)) to the maximum value.
INL integral nonlinearity error: Add (± 0.2 LSB × (VDD voltage (V) - AVREFP voltage (V)) to the maximum value.
Note 4. If the internal reference voltage or temperature sensor output voltage is to be A/D converted, VCC must be at least 1.8 V.
Note 5. When the internal reference voltage or the temperature sensor output voltage is selected as the target for conversion, the sampling time must be at least 5 µs. Accordingly, use a low-voltage mode 2 and fAD = 16 MHz or less with the longer sampling time.

Table 37.44 A/D conversion characteristics (2) in Low-voltage modes 1 and 2

Conditions: VCC = AVREFP = 2.4 to 5.5 V, VSS = AVREFM = 0 V
Reference voltage (+) = AVREFP, Reference voltage (-) = AVREFM
Target pins for conversions: ANI2 to ANI5, internal reference voltage*4, and temperature sensor output voltage*4

Parameter	Min	Typ	Max	Unit	Test conditions
Resolution	—	—	12	Bit	—
Conversion clock (fAD)	1	—	16	MHz	—
Conversion time*5	5.0	—	—	µs	—
Offset error*1 *2 *3	—	—	±9.0	LSB	—
Full-scale error*1 *2 *3	—	—	±9.0	LSB	—
Absolute accuracy*1 *2 *3	—	—	±9.5	LSB	—
DNL differential nonlinearity error*1	—	±1.5	—	LSB	—
INL integral nonlinearity error*1 *3	—	—	±4.0	LSB	—
Analog input voltage range	0	—	AVREFP	V	—

Note: These specification values apply during A/D conversion operation, while CPU is in Sleep mode and peripheral modules other than A/D in module standby.
If CPU is running or peripheral modules other than A/D are running during A/D conversion, values might not fall within the indicated ranges.
Note 1. This value does not include the quantization error (± 1/2 LSB).
Note 2. When pins ANI16 to ANI19 are selected as the target pins for conversion, the maximum values are as follows:
Absolute accuracy: Add ± 3 LSB to the maximum value.
Offset/full-scale error: Add ± 2 LSB to the maximum value.
Note 3. When AVREFP < VCC, the maximum values are as follows:
Absolute accuracy/Offset error/full-scale error: Add (± 0.75 LSB × (VCC voltage (V) - AVREFP voltage (V)) to the maximum value.
INL integral nonlinearity error: Add (± 0.2 LSB × (VDD voltage (V) - AVREFP voltage (V)) to the maximum value.
Note 4. If the internal reference voltage or temperature sensor output voltage is to be A/D converted, VCC must be at least 1.8 V.
Note 5. When the internal reference voltage or the temperature sensor output voltage is selected as the target for conversion, the sampling time must be at least 5 µs. Accordingly, use low-voltage mode 2 and fAD = 16 MHz or less with the longer sampling time.

Table 37.45 A/D conversion characteristics (3) in Low-voltage modes 1 and 2

Conditions: VCC = AVREFP = 1.8 to 5.5 V, VSS = AVREFM = 0 V
Reference voltage (+) = AVREFP, Reference voltage (-) = AVREFM
Target pins for conversion: ANI2 to ANI5, internal reference voltage*4, and temperature sensor output voltage*4

Parameter	Min	Typ	Max	Unit	Test conditions
Resolution	—	—	12	Bit	—
Conversion clock (fAD)	1	—	8	MHz	—
Conversion time*5	10.0	—	—	µs	—
Offset error*1 *2 *3	—	—	± 13.0	LSB	—
Full-scale error*1 *2 *3	—	—	± 13.0	LSB	—
Absolute accuracy*1 *2 *3	—	—	± 13.5	LSB	—
DNL differential nonlinearity error*1	—	± 2.0	—	LSB	—
INL integral nonlinearity error*1 *3	—	—	± 4.5	LSB	—
Analog input voltage range	0	—	AVREFP	V	—

Note: These specification values apply during A/D conversion operation, while CPU is in Sleep mode and peripheral modules other than A/D in module standby.
If CPU is running or peripheral modules other than A/D are running during A/D conversion, values might not fall within the indicated ranges.
Note 1. This value does not include the quantization error (± 1/2 LSB).
Note 2. When pins ANI16 to ANI19 are selected as the target pins for conversion, the maximum values are as follows:
Absolute accuracy: Add ± 3 LSB to the maximum value.
Offset/full-scale error: Add ± 2 LSB to the maximum value.
Note 3. When AVREFP < VCC, the maximum values are as follows:
Absolute accuracy/Offset error/full-scale error: Add (± 0.75 LSB × (VCC voltage (V) - AVREFP voltage (V)) to the maximum value.
INL integral nonlinearity error: Add (± 0.2 LSB × (VDD voltage (V) - AVREFP voltage (V)) to the maximum value.
Note 4. If the internal reference voltage or temperature sensor output voltage is to be A/D converted, VCC must be at least 1.8 V.

Note 5. When the internal reference voltage or the temperature sensor output voltage is selected as the target for conversion, the sampling time must be at least 5 µs. Accordingly, use low-voltage mode 2 and fAD = 16 MHz or less with the longer sampling time.

Table 37.46 A/D conversion characteristics (4) in Low-voltage modes 1 and 2

Conditions: VCC = AVREFP = 1.6 to 5.5 V, VSS = AVREFM = 0 V
Reference voltage (+) = AVREFP, Reference voltage (-) = AVREFM
Target pins for conversion: ANI2 to ANI5, internal reference voltage*4, and temperature sensor output voltage*4

Parameter	Min	Typ	Max	Unit	Test conditions
Resolution	—	—	12	Bit	—
Conversion clock (PCLKB)	1	—	4	MHz	—
Conversion time*5	20.0	—	—	µs	—
Offset error*1 *2 *3 *4	—	—	±13.5	LSB	—
Full-scale error*1 *2 *3 *4	—	—	±13.5	LSB	—
Absolute accuracy*1 *2 *3 *4	—	—	±14.0	LSB	—
DNL differential nonlinearity error*1	—	±2.0	—	LSB	—
INL integral nonlinearity error*1 *3 *4	—	—	±4.5	LSB	—
Analog input voltage range	0	—	AVREFP	V	—

Note: These specification values apply during A/D conversion operation, while CPU is in Sleep mode and peripheral modules other than A/D in module standby.
If CPU is running or peripheral modules other than A/D are running during A/D conversion, values might not fall within the indicated ranges.
Note 1. This value does not include the quantization error (± 1/2 LSB).
Note 2. When pins ANI16 to ANI19 are selected as the target pins for conversion, the maximum values are as follows:
Absolute accuracy: Add ± 3 LSB to the maximum value.
Offset/full-scale error: Add ± 2 LSB to the maximum value.
Note 3. When AVREFP < VCC, the maximum values are as follows:
Absolute accuracy/Offset error/full-scale error: Add (± 0.75 LSB × (VCC voltage (V) - AVREFP voltage (V)) to the maximum value.
INL integral nonlinearity error: Add (± 0.2 LSB × (VDD voltage (V) - AVREFP voltage (V)) to the maximum value.
Note 4. If the internal reference voltage or temperature sensor output voltage is to be A/D converted, VCC must be at least 1.8 V.
Note 5. When the internal reference voltage or the temperature sensor output voltage is selected as the target for conversion, the sampling time must be at least 5 µs. Accordingly, use low-voltage mode 2 and fAD = 16 MHz or less with the longer sampling time.

Table 37.47 A/D conversion characteristics in Low-voltage modes 1 and 2 when the internal reference voltage is selected as reference voltage (+)

Conditions: VCC = 1.8 to 5.5 V, VSS = AVREFM = 0 V
Reference voltage (+) = internal reference voltage, Reference voltage (-) = AVREFM

Parameter	Min	Typ	Max	Unit	Test conditions
Resolution	—	—	8	Bit	—
Conversion clock (fAD)	1	—	2	MHz	—
Offset error*1	—	—	2	LSB	—
DNL differential nonlinearity error*1	—	1	—	LSB	—
INL integral nonlinearity error*1	—	—	2	LSB	—
Analog input voltage range	0	—	VBGR	V	

Note: These specification values apply during A/D conversion operation, while CPU is in Sleep mode and peripheral modules other than A/D in module standby.
If CPU is running or peripheral modules other than A/D are running during A/D conversion, values might not fall within the indicated ranges.
Note 1. This value does not include the quantization error (± 1/2 LSB).

Table 37.48 12-bit A/D converter channel classification (1 of 2)

Classification	Channel	Conditions	Remarks
High-precision channel	ANI0 to ANI5	VCC = 1.6 to 5.5 V	Pins ANI0 to ANI5 cannot be used as general I/O, TS transmission, when the A/D converter is in use.
Normal-precision channel	ANI16 to ANI19		—

Table 37.48 12-bit A/D converter channel classification (2 of 2)

Classification	Channel	Conditions	Remarks
Internal reference voltage input channel	Internal reference voltage	VCC = 1.8 to 5.5 V	—
Temperature sensor input channel	Temperature sensor output		—

Table 37.49 A/D internal reference voltage characteristics

Conditions: VCC = 1.8 to 5.5 V, VSS = 0 V

Parameter	Min	Typ	Max	Unit	Test conditions
Internal reference voltage input channel*1	1.40	1.47	1.54	V	—
Sampling time*2	5.0	—	—	µs	—

Note 1. The 12-bit A/D internal reference voltage indicates the voltage when the internal reference voltage is input to the 12-bit A/D converter.
Note 2. When the internal reference voltage is converted.

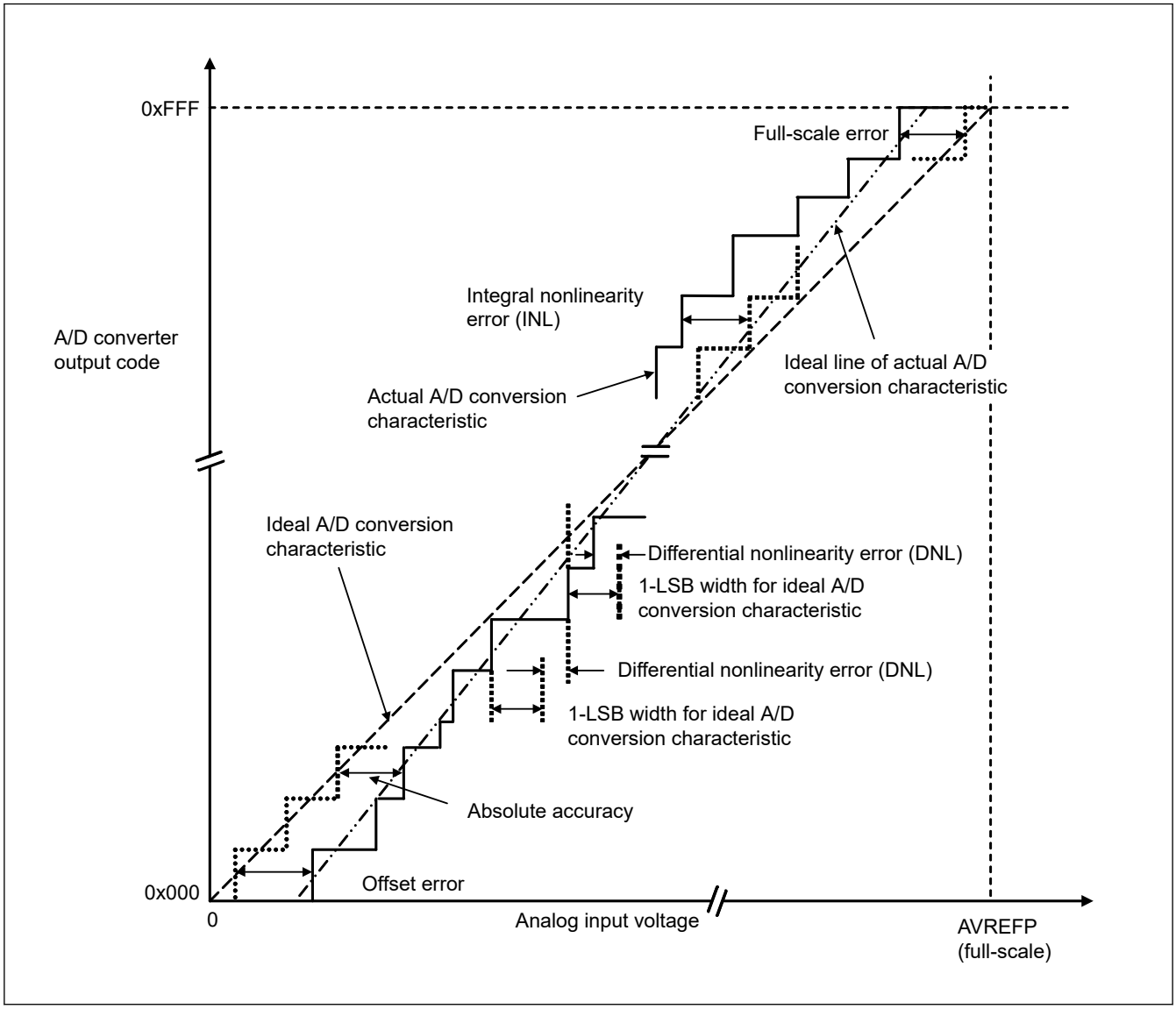


Figure 37.26 Example of 12-bit A/D converter characteristic terms

Absolute accuracy

Absolute accuracy is the difference between output code based on the theoretical A/D conversion characteristics, and the actual A/D conversion result. When measuring absolute accuracy, the voltage at the midpoint of the width of the analog input voltage (1-LSB width), which can meet the expectation of outputting an equal code based on the theoretical A/D conversion characteristics, is used as the analog input voltage. For example, if 12-bit resolution is used and the reference voltage AVREFP = 3.072 V, then 1-LSB width becomes 0.75 mV, and 0 mV, 0.75 mV, and 1.5 mV are used as the analog input voltages. If analog input voltage is 6 mV, an absolute accuracy of ±5 LSB means that the actual A/D conversion result is in the range of 0x003 to 0x00D, though an output code of 0x008 can be expected from the theoretical A/D conversion characteristics.

Integral nonlinearity error (INL)

Integral nonlinearity error is the maximum deviation between the ideal line when the measured offset and full-scale errors are zeroed, and the actual output code.

Differential nonlinearity error (DNL)

Differential nonlinearity error is the difference between 1-LSB width based on the ideal A/D conversion characteristics and the width of the actual output code.

Offset error

Offset error is the difference between the transition point of the ideal first output code and the actual first output code.

Full-scale error

Full-scale error is the difference between the transition point of the ideal last output code and the actual last output code.

37.5 CMP Characteristics

Table 37.50 CMP characteristics

Conditions: VCC = 1.6 to 5.5 V, VSS = 0 V

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Input voltage range		IVREF	0	—	VCC − 1.4	V	Input to the IVREF0 and IVREF1 pins C0LVL = 0, C1LVL = 0
			1.4	—	VCC		Input to the IVREF0 and IVREF1 pins C0LVL = 1, C1LVL = 1
		IVCMP	−0.3	—	VCC + 0.3		Input to the IVCMP0 and IVCMP1 pins
Output delay	High-speed mode	—	—	—	1.5	µs	VCC = 3.0 V Input slew rate > 1 V/us
	Low-speed mode		—	3.0	—		
Offset voltage	High-speed mode	—	—	—	50	mV	—
	Low-speed mode		—	—	40		
Operation stabilization wait time		t_{CMP}	30	—	—	µs	—
Internal reference voltage*1		—	1.34	1.44	1.54	V	—

Note 1. The internal reference voltage can be selected as CMP reference voltage only when 1.8 V ≤ VCC ≤ 5.5 V.

37.6 DAC8 Characteristics

Table 37.51 D/A conversion characteristics

Conditions: VCC = 2.7 to 5.5 V, VSS = 0 V

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
Resolution	—	—	—	8	bit	—
Conversion time	t_{DCONV}	—	—	3.0	µs	—
Absolute accuracy	—	—	—	±3.0	LSB	—
Resistive load	—	4	—	—	MΩ	—
Capacitive load*1	—	—	—	35	pF	—
Output resistance	—	—	9.0	—	kΩ	—

Note 1. Including IO input capacitance of 15 pF.

37.7 TSN Characteristics

Table 37.52 TSN characteristics

Conditions: VCC = 1.8 to 5.5 V, VSS = 0 V

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
Temperature slope	—	—	−3.3	—	mV/°C	—
Output voltage (at 25°C)	—	—	1.05	—	V	VCC = 3.3 V
Sampling time	—	5.0	—	—	µs	—

37.8 POR and LVD Characteristics

Table 37.53 Power-on reset circuit and voltage detection circuit characteristics (1) (1 of 2)

Parameter			Symbol	Min	Typ	Max	Unit	Test Conditions
Voltage detection level*1	Power-on reset (POR)	When power supply rise	V_{POR}	1.47	1.51	1.55	V	Figure 37.27
		When power supply fall	V_{PDR}	1.46	1.50	1.54		Figure 37.28
	Voltage detection circuit (LVD0)*2	When power supply rise	V_{det0_0}	3.74	3.91	4.06	V	Figure 37.29 At falling edge VCC
		When power supply fall		3.68	3.85	4.00		
		When power supply rise	V_{det0_1}	2.73	2.9	3.01		
		When power supply fall		2.68	2.85	2.96		
		When power supply rise	V_{det0_2}	2.44	2.59	2.70		
		When power supply fall		2.38	2.53	2.64		
		When power supply rise	V_{det0_3}	1.83	1.95	2.07		
		When power supply fall		1.78	1.90	2.02		
		When power supply rise	V_{det0_4}	1.66	1.75	1.88		
		When power supply fall		1.60	1.69	1.82		

Table 37.53 Power-on reset circuit and voltage detection circuit characteristics (1) (2 of 2)

Parameter			Symbol	Min	Typ	Max	Unit	Test Conditions
Voltage detection level*1	Voltage detection circuit (LVD1)*3	When power supply rise	V_{det1_0}	4.23	4.39	4.55	V	Figure 37.30 At falling edge VCC
		When power supply fall		4.13	4.29	4.45		
		When power supply rise	V_{det1_1}	4.07	4.25	4.39		
		When power supply fall		3.98	4.16	4.30		
		When power supply rise	V_{det1_2}	3.97	4.14	4.29		
		When power supply fall		3.86	4.03	4.18		
		When power supply rise	V_{det1_3}	3.74	3.92	4.06		
		When power supply fall		3.68	3.86	4.00		
		When power supply rise	V_{det1_4}	3.05	3.17	3.29		
		When power supply fall		2.98	3.10	3.22		
		When power supply rise	V_{det1_5}	2.95	3.06	3.17		
		When power supply fall		2.89	3.00	3.11		
		When power supply rise	V_{det1_6}	2.86	2.97	3.08		
		When power supply fall		2.79	2.90	3.01		
		When power supply rise	V_{det1_7}	2.74	2.85	2.96		
		When power supply fall		2.68	2.79	2.90		
Voltage detection level*1	Voltage detection circuit (LVD1)*3	When power supply rise	V_{det1_8}	2.63	2.75	2.85	V	Figure 37.30 At falling edge VCC
		When power supply fall		2.58	2.68	2.78		
		When power supply rise	V_{det1_9}	2.54	2.64	2.75		
		When power supply fall		2.48	2.58	2.68		
		When power supply rise	V_{det1_A}	2.43	2.53	2.63		
		When power supply fall		2.38	2.48	2.58		
		When power supply rise	V_{det1_B}	2.16	2.26	2.36		
		When power supply fall		2.10	2.20	2.30		
		When power supply rise	V_{det1_C}	1.88	2	2.09		
		When power supply fall		1.84	1.96	2.05		
		When power supply rise	V_{det1_D}	1.78	1.9	1.99		
		When power supply fall		1.74	1.86	1.95		
		When power supply rise	V_{det1_E}	1.67	1.79	1.88		
		When power supply fall		1.63	1.75	1.84		
		When power supply rise	V_{det1_F}	1.65	1.7	1.78		
		When power supply fall		1.60	1.65	1.73		
Voltage detection level*1	Voltage detection circuit (LVD2)*4	When power supply rise	V_{det2_0}	4.20	4.40	4.57	V	Figure 37.31 At falling edge VCC
		When power supply fall		4.11	4.31	4.48		
		When power supply rise	V_{det2_1}	4.05	4.25	4.42		
		When power supply fall		3.97	4.17	4.34		
		When power supply rise	V_{det2_2}	3.91	4.11	4.28		
		When power supply fall		3.83	4.03	4.20		
		When power supply rise	V_{det2_3}	3.71	3.91	4.08		
		When power supply fall		3.64	3.84	4.01		

Note 1. These characteristics apply when noise is not superimposed on the power supply. When a setting causes this voltage detection level to overlap with that of the voltage detection circuit, it cannot be specified whether LVD1 or LVD2 is used for voltage detection.
Note 2. # in the symbol $V_{det0\_\#}$ denotes the value of the OFS1.VDSEL0[2:0] bits.

Note 3. # in the symbol $V_{det1\_\#}$ denotes the value of the LVDLVLR.LVD1LVL[4:0] bits.
Note 4. # in the symbol $V_{det2\_\#}$ denotes the value of the LVDLVLR.LVD2LVL[2:0] bits.

Table 37.54 Power-on reset circuit and voltage detection circuit characteristics (2)

Parameter		Symbol	Min	Typ	Max	Unit	Test Conditions
Wait time after power-on reset cancellation	LVD0: enable	t_{POR}	—	4.3	—	ms	—
	LVD0: disable	t_{POR}	—	3.7	—	ms	—
Wait time after voltage monitor 0, 1, 2 reset cancellation	LVD0: enable*1	$t_{LVD0,1,2}$	—	1.4	—	ms	—
	LVD0: disable*2	$t_{LVD1,2}$	—	0.7	—	ms	—
Power-on reset response delay time*3		t_{det}	—	—	500	µs	Figure 37.27, Figure 37.28
LVD0 response delay time*3		t_{det}	—	—	500	µs	Figure 37.29
LVD1 response delay time*3		t_{det}	—	—	350	µs	Figure 37.30
LVD2 response delay time*3		t_{det}	—	—	600	µs	Figure 37.31
Minimum VCC down time		t_{VOFF}	500	—	—	µs	Figure 37.27, VCC = 1.0 V or above
Power-on reset enable time		$t_{W\ (POR)}$	1	—	—	ms	Figure 37.28, VCC = below 1.0 V
LVD1 operation stabilization time (after LVD1 is enabled)		$T_{d\ (E-A)}$	—	—	300	µs	Figure 37.30
LVD2 operation stabilization time (after LVD2 is enabled)		$T_{d\ (E-A)}$	—	—	1200	µs	Figure 37.31
Hysteresis width (POR)		V_{PORH}	—	10	—	mV	—
Hysteresis width (LVD0, LVD1 and LVD2)		V_{LVH}	—	60	—	mV	LVD0 selected
			—	110	—		V_{det1_0} to V_{det1_2} selected
			—	70	—		V_{det1_3} to V_{det1_9} selected
			—	60	—		V_{det1_A} to V_{det1_B} selected
			—	50	—		V_{det1_C} to V_{det1_F} selected
			—	90	—		LVD2 selected

Note 1. When OFS1.LVDAS = 0.
Note 2. When OFS1.LVDAS = 1.
Note 3. 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 the POR/LVD.

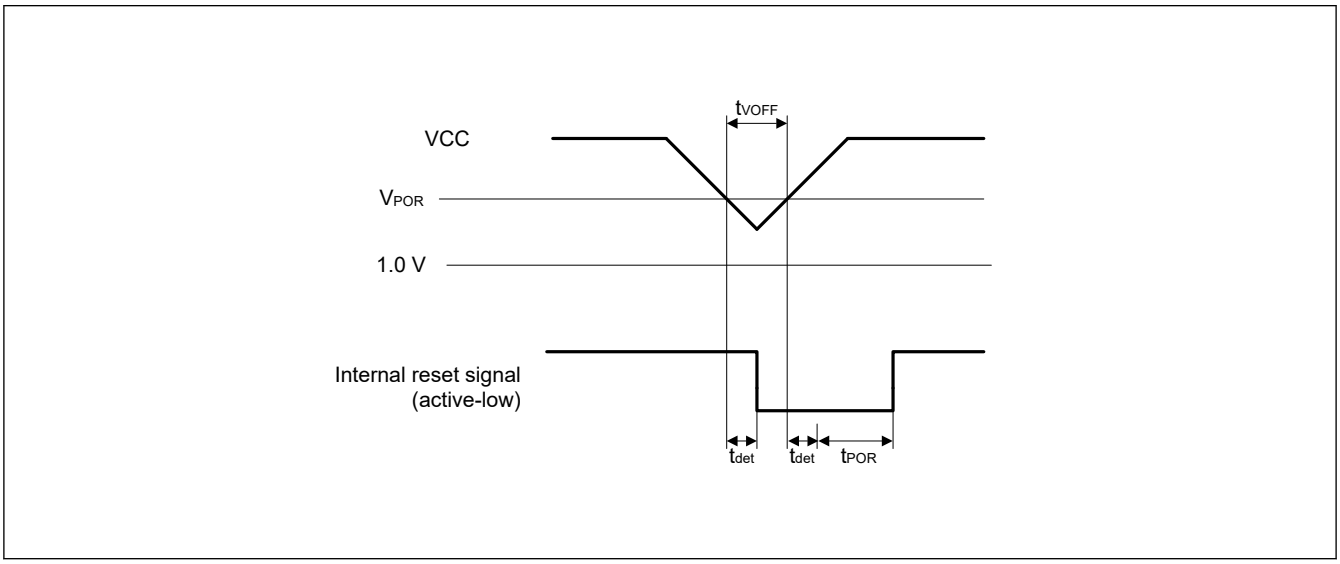
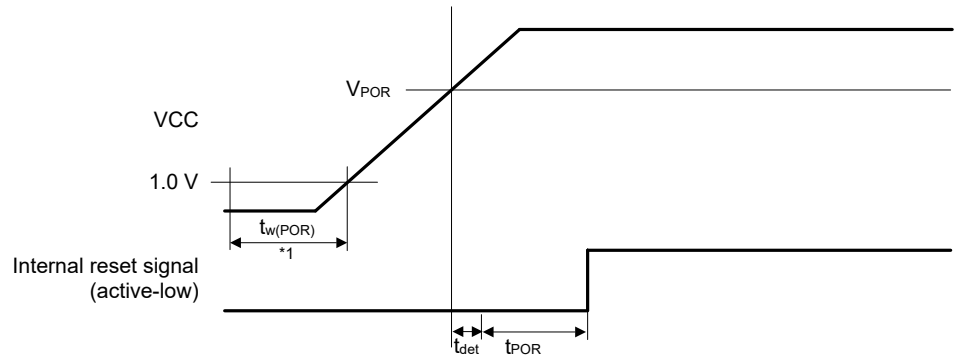


Figure 37.27 Voltage detection reset timing



Note 1. $t_{w(POR)}$ is the time required for a power-on reset to be enabled while the external power VCC is being held below the valid voltage (1.0 V).
When VCC turns on, maintain $t_{w(POR)}$ for 1.0 ms or more.

Figure 37.28 Power-on reset timing

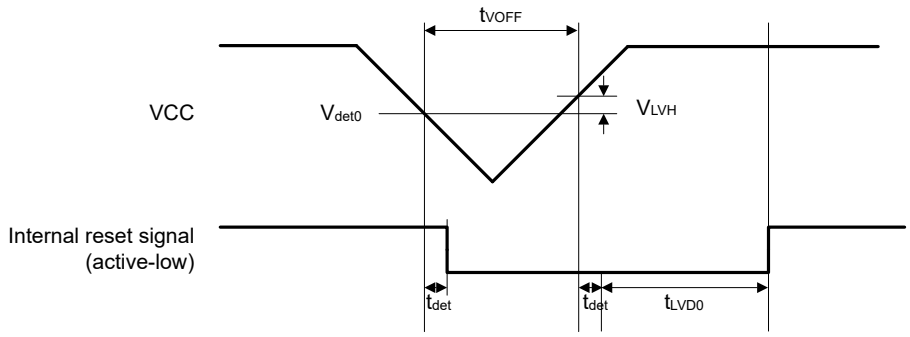


Figure 37.29 Voltage detection circuit timing (V_{det0})

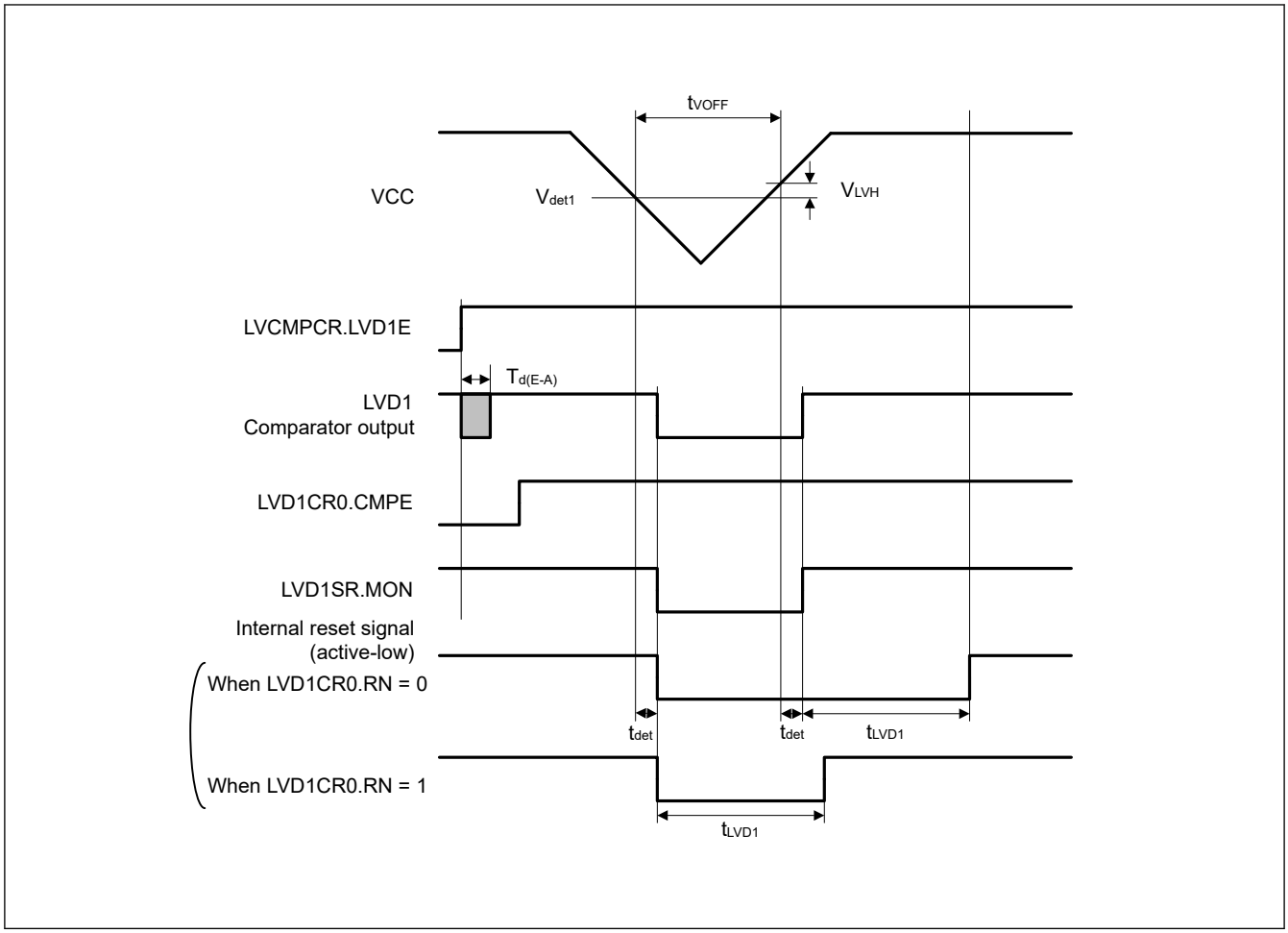


Figure 37.30 Voltage detection circuit timing (V_{det1})

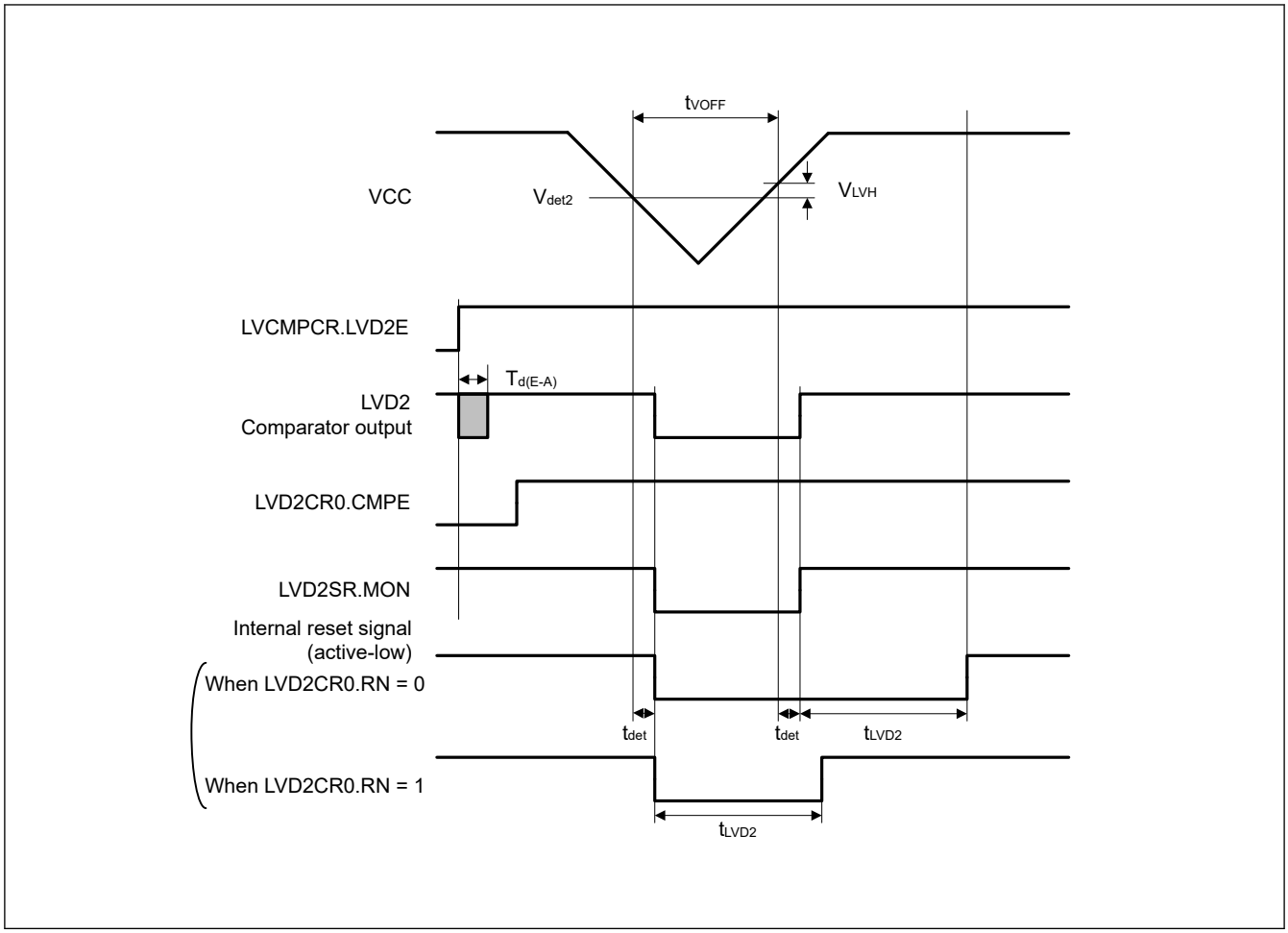


Figure 37.31 Voltage detection circuit timing (V_{det2})

37.9 Flash Memory Characteristics

37.9.1 Code Flash Memory Characteristics

Table 37.55 Code flash characteristics (1)

Parameter		Symbol	Min	Typ	Max	Unit	Conditions
Reprogramming/erasure cycle*1		N_{PEC}	10000	—	—	Times	—
Data hold time	After 10000 times N_{PEC}	t_{DRP}	20*2 *3	—	—	Year	T_a = 105°C
			10	—	—		T_a = 125°C

Note 1. The reprogram/erase cycle is the number of erasures for each block. When the reprogram/erase cycle is n times (n = 1,0000), erasing can be performed n times for each block. For instance, when 8-byte programming is performed 256 times for different addresses in 2-KB blocks, and then the entire block is erased, the reprogram/erase cycle is counted as one. However, programming the same address for several times as one erasure is not enabled (overwriting is prohibited).
Note 2. Characteristic when using the flash memory programmer and the self-programming library provided by Renesas Electronics.
Note 3. This result is target spec, may changed after reliability testing.

Table 37.56 Code flash characteristics (2) (1 of 2)

High-speed mode
Conditions: VCC = 1.8 to 5.5 V, T_a = −40°C to 125°C

Parameter		Symbol	ICLK = 1 MHz			ICLK = 48 MHz			Unit
			Min	Typ	Max	Min	Typ	Max	
Programming time	8-byte	t_{P4}	—	97	843	—	47	446	µs
Erasure time	2-KB	t_{E2K}	—	8.7	282	—	5.7	221	ms

Table 37.56 Code flash characteristics (2) (2 of 2)

High-speed mode
Conditions: VCC = 1.8 to 5.5 V, T_a = −40°C to 125°C

Parameter		Symbol	ICLK = 1 MHz			ICLK = 48 MHz			Unit
			Min	Typ	Max	Min	Typ	Max	
Blank check time	8-byte	t_{BC4}	—	—	45	—	—	8.7	µs
	2-KB	t_{BC2K}	—	—	3239	—	—	235	µs
Erase suspended time		t_{SED}	—	—	22.8	—	—	11.0	µs
Access window information program Start-up area selection and security setting time		t_{AWSSAS}	—	16.3	509	—	11.8	444	ms
OCD/serial programmer ID setting time		t_{OSIS}	—	65.1	2036	—	46.9	1773.9	µs
Flash memory mode transition wait time 1		t_{DIS}	2	—	—	2	—	—	
Flash memory mode transition wait time 2		t_{MS}	15	—	—	15	—	—	µs

Note: Does not include the time until each operation of the flash memory is started after instructions are executed by software.
Note: The lower-limit frequency of ICLK is 1 MHz during programming or erasing the flash memory. When using ICLK at below 4 MHz, the frequency can be set to 1 MHz, 2 MHz, or 3 MHz. A non-integer frequency such as 1.5 MHz cannot be set.
Note: The frequency accuracy of ICLK must be ± 1.5% during programming or erasing the flash memory. Confirm the frequency accuracy of the clock source.

Table 37.57 Code flash characteristics (3)

Middle-speed mode
Conditions: VCC = 1.6 to 5.5 V, T_a = −40°C to 125°C

Parameter		Symbol	ICLK = 1 MHz			ICLK = 24 MHz			Unit
			Min	Typ	Max	Min	Typ	Max	
Programming time	8-byte	t_{P4}	—	97	843	—	48	450	µs
Erasure time	2-KB	t_{E2K}	—	8.7	282	—	5.7	220	ms
Blank check time	8- byte	t_{BC4}	—	—	45	—	—	9.1	µs
	2-KB	t_{BC2K}	—	—	3239	—	—	236	µs
Erase suspended time		t_{SED}	—	—	22.8	—	—	11.2	µs
Access window information program Start-up area selection and security setting time		t_{AWSSAS}	—	16.3	509	—	11.4	442	ms
OCD/serial programmer ID setting time*1		t_{OSIS}	—	84.7	2280	—	45.3	1690	ms
Flash memory mode transition wait time 1		t_{DIS}	2	—	—	2	—	—	µs
Flash memory mode transition wait time 2		t_{MS}	15	—	—	15	—	—	µs

Note: Does not include the time until each operation of the flash memory is started after instructions are executed by software.
Note: The lower-limit frequency of ICLK is 1 MHz during programming or erasing the flash memory. When using ICLK at below 4 MHz, the frequency can be set to 1 MHz, 2 MHz, or 3 MHz. A non-integer frequency such as 1.5 MHz cannot be set.
Note: The frequency accuracy of ICLK must be ± 1.5% during programming or erasing the flash memory. Confirm the frequency accuracy of the clock source.
Note 1. When 1.8 V ≤ VCC = AVCC0 ≤ 5.5 V

Table 37.58 Code flash characteristics (4)

Low-speed mode
Conditions: VCC = 1.6 to 5.5 V, T_a = −40°C to 125°C

Parameter		Symbol	ICLK = 1 MHz			Unit
			Min	Typ	Max	
Programming time	8-byte	t_{P4}	—	97	843	µs
Erasure time	2-KB	t_{E2K}	—	8.7	282	ms
Blank check time	8-byte	t_{BC4}	—	—	45	µs
	2-KB	t_{BC2K}	—	—	3239	µs
Erase suspended time		t_{SED}	—	—	22.8	µs
Access window information program Start-up area selection and security setting time		t_{AWSSAS}	—	16.3	509	ms
OCD/serial programmer ID setting time		t_{OSIS}	—	65.1	2036	ms
Flash memory mode transition wait time 1		t_{DIS}	2	—	—	µs
Flash memory mode transition wait time 2		t_{MS}	15	—	—	µs

Note: Does not include the time until each operation of the flash memory is started after instructions are executed by software.
Note: The lower-limit frequency of ICLK is 1 MHz during programming or erasing the flash memory.
Note: The frequency accuracy of ICLK must be ± 1.5% during programming or erasing the flash memory. Confirm the frequency accuracy of the clock source.

37.9.2 Data Flash Memory Characteristics

Table 37.59 Data flash characteristics (1)

Parameter		Symbol	Min	Typ	Max	Unit	Conditions
Reprogramming/erasure cycle*1		N_{DPEC}	100000	1000000	—	Times	—
Data hold time	After 10000 times of NDPEC	t_{DDRP}	20*2 *3	—	—	Year	T_a = 105°C
			10	—	—	Year	T_a = 125°C
	After 100000 times of NDPEC		5*2 *3	—	—	Year	
	After 1000000 times of NDPEC		—	1*2 *3	—	Year	Ta = 25°C

Note 1. The reprogram/erase cycle is the number of erasure for each block. When the reprogram/erase cycle is n times (n = 100,000), erasing can be performed n times for each block. For instance, when 1-byte programming is performed 1,024 times for different addresses in 1-KB blocks, and then the entire block is erased, the reprogram/erase cycle is counted as one. However, programming the same address for several times as one erasure is not enabled (overwriting is prohibited).
Note 2. Characteristics when using the flash memory programmer and the self-programming library provided by Renesas Electronics.
Note 3. These results are target spec, and may changed after reliability testing.

Table 37.60 Data flash characteristics (2)

High-speed mode
Conditions: VCC = 1.8 to 5.5 V, T_a = −40°C to 125°C

Parameter		Symbol	ICLK = 1 MHz			ICLK = 48 MHz			Unit
			Min	Typ	Max	Min	Typ	Max	
Programming time	1-byte	t_{DP1}	—	84	708	—	36	336	µs
Erasure time	1-KB	t_{DE1K}	—	8.6	281	—	6.3	234	ms
Blank check time	1-byte	t_{DBC1}	—	—	14.8	—	—	8.7	µs
	1-KB	t_{DBC1K}	—	—	1602	—	—	450	µs
Suspended time during erasing		t_{DSED}	—	—	22.8	—	—	11.0	µs
Data flash STOP recovery time		t_{DSTOP}	250	—	—	250	—	—	µs

Note: Does not include the time until each operation of the flash memory is started after instructions are executed by software.
Note: The lower-limit frequency of ICLK is 1 MHz during programming or erasing the flash memory. When using ICLK at below 4 MHz, the frequency can be set to 1 MHz, 2 MHz, or 3 MHz. A non-integer frequency such as 1.5 MHz cannot be set.
Note: The frequency accuracy of ICLK must be ± 1.0% during programming or erasing the flash memory. Confirm the frequency accuracy of the clock source.

Table 37.61 Data flash characteristics (3)

Middle-speed mode
Conditions: VCC = 1.6 to 5.5 V, T_a = −40℃ to 125℃

Parameter		Symbol	ICLK = 1 MHz			ICLK = 24 MHz*1			Unit
			Min	Typ	Max	Min	Typ	Max	
Programming time	1-byte	t_{DP1}	—	84	708	—	40	365	µs
Erasure time	1-KB	t_{DE1K}	—	8.6	281	—	7	249	ms
Blank check time	1- byte	t_{DBC1}	—	—	14.8	—	—	11.2	µs
	1-KB	t_{DBC1K}	—	—	1602	—	—	806	µs
Suspended time during erasing		t_{DSED}	—	—	22.8	—	—	11.2	µs
Data flash STOP recovery time		t_{DSTOP}	250	—	—	250	—	—	µs

Note: Does not include the time until each operation of the flash memory is started after instructions are executed by software.
Note: The lower-limit frequency of ICLK is 1 MHz during programming or erasing the flash memory. When using ICLK at below 4 MHz, the frequency can be set to 1 MHz, 2 MHz, or 3 MHz. A non-integer frequency such as 1.5 MHz cannot be set.
Note: The frequency accuracy of ICLK must be ± 1.0% during programming or erasing the flash memory. Confirm the frequency accuracy of the clock source.
Note 1. When 1.8 V ≤ VCC ≤ 5.5 V

Table 37.62 Data flash characteristics (4)

Low-speed mode
Conditions: VCC = 1.6 to 5.5 V, T_a = −40℃ to 125℃

Parameter		Symbol	ICLK = 1 MHz			Unit
			Min	Typ	Max	
Programming time	1-byte	t_{DP1}	—	84	708	µs
Erasure time	1-KB	t_{DE1K}	—	8.6	281	ms
Blank check time	1-byte	t_{DBC1}	—	—	14.8	µs
	1-KB	t_{DBC1K}	—	—	1602	µs
Suspended time during erasing		t_{DSED}	—	—	22.8	µs
Data flash STOP recovery time		t_{DSTOP}	250	—	—	µs

Note: Does not include the time until each operation of the flash memory is started after instructions are executed by software.
Note: The lower-limit frequency of ICLK is 1 MHz during programming or erasing the flash memory.
Note: The frequency accuracy of ICLK must be ± 1.0% during programming or erasing the flash memory. Confirm the frequency accuracy of the clock source.

37.10 Compact JTAG (cJTAG)

Table 37.63 cJTAG Characteristics

Conditions: VCC = 2.7 to 5.5 V

No.	Parameter	Symbol	Min	Max	Unit
1	TCKC clock cycle time	$t_{CTCKcyc}$	160	—	ns
1a	TCKC clock high pulse width	t_{CTCKH}	70	—	ns
1b	TCKC clock low pulse width	t_{CTCKL}	70	—	ns
2	TMSC setup time	t_{CTMSS}	14	—	ns
3	TMSC hold time	t_{CTMSH}	2	—	ns
4	Delay time, TCKC to TMSC valid/disable	$t_{d(CTCKL-CTMS)}$	5	60	ns

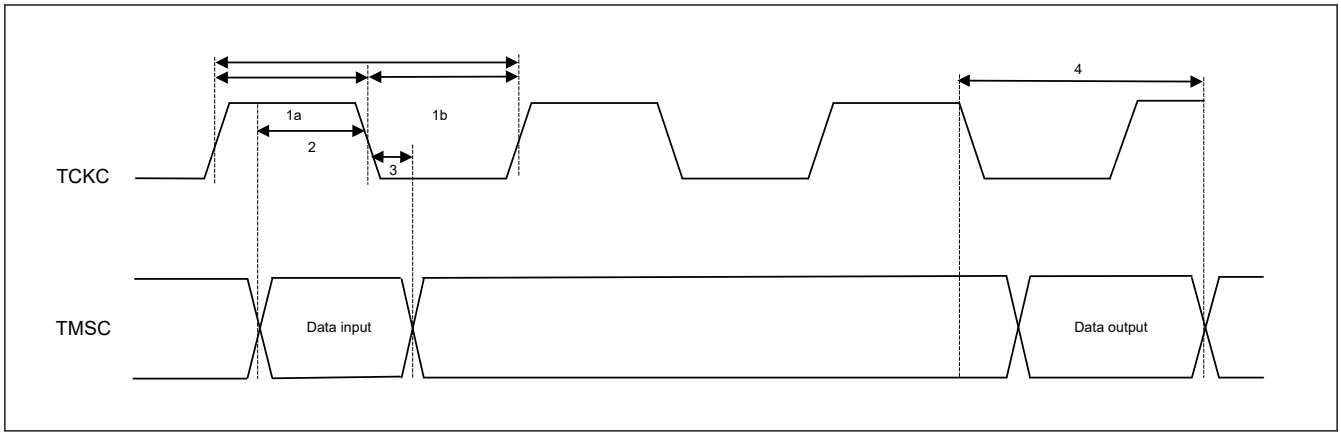


Figure 37.32 cJTAG timing

Appendix 1. Port States in Each Processing Mode

Table 1.1 Port states in each processing mode

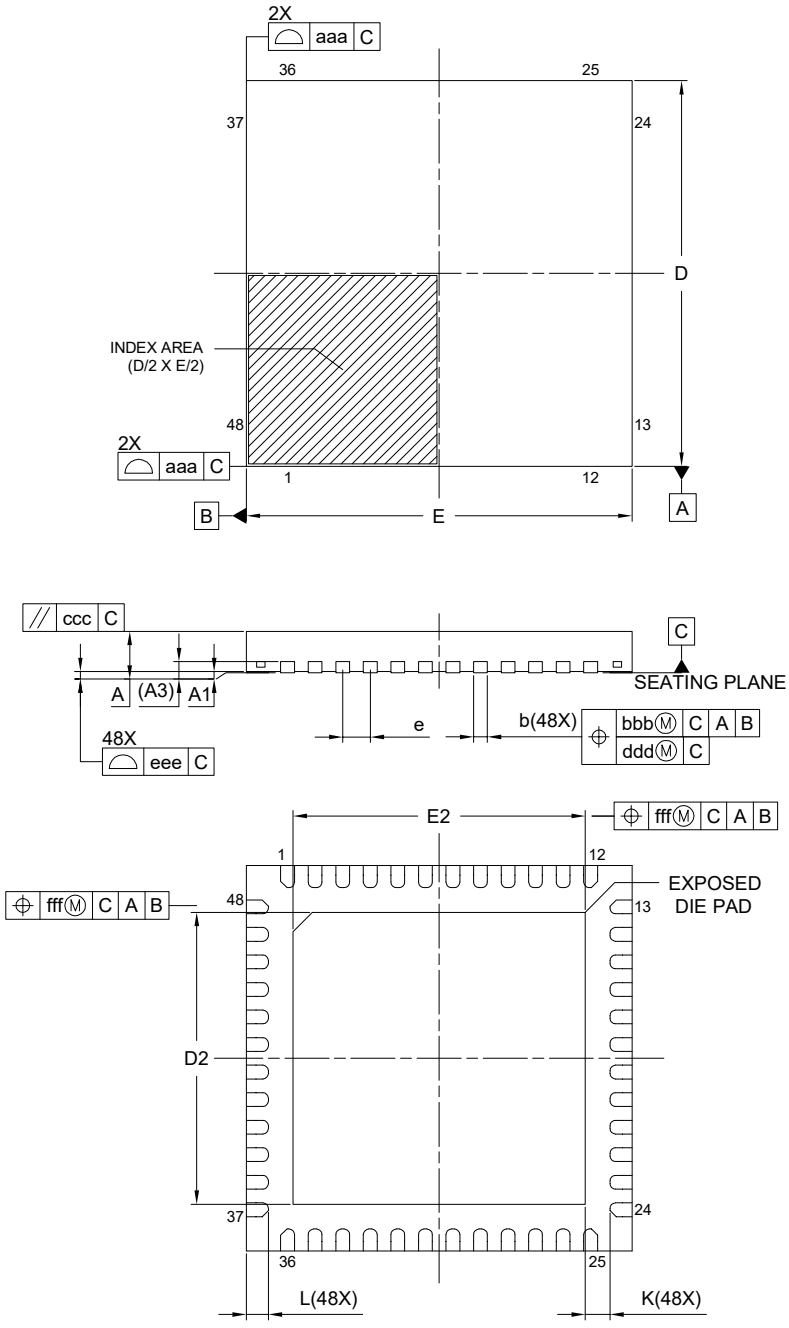
Function	Pin function	Reset	Software Standby mode
Mode	MD	Pull-up	Keep-O
cJTAG	TMSC/TCKC	Pull-up	Keep-O
IRQ	IRQn	Hi-Z	Keep-O*1 *2
	NMI	Hi-Z	Hi-Z*3
SOSC	XT1, XT2	Hi-Z	[Sub-clock Oscillator selected] Sub-clock Oscillator is operating [Other than the above] Hi-Z
KINT	KR0n	Hi-Z	Keep-O*1 *2
SAU	[UART mode] RXD0, RXD2 [SPI mode] SCK00, SCK20	Hi-Z	Keep-O*2
IICA	SCLAn/SDAAn	Hi-Z	Keep-O*1
UARTA	TxDAn/RxDAn/CLKAn	Hi-Z	Keep-O*1
REMC	RIN0	Hi-Z	Keep-O*2
RTC	RTC1HZ	Hi-Z	[RTC selected] RTC1HZ output
CLKOUT	CLKOUT_A/B	Hi-Z	[CLKOUT selected] CLKOUT output
CMP	VCOUTn	Hi-Z	[VCOUT selected] VCOUT output
DAC8	DACOUTn	Hi-Z	[DACOUTn output (DAOE = 1)] D/A output retained
P303	—	Pull-up	Keep-O
Others	—	Hi-Z	Keep-O

Note: Hi-Z: High-impedance
Keep-O: Output pins retain their previous values. Input pins become high-impedance.
Note 1. Input is enabled if the pin is specified as the software standby canceling source while it is used as an external interrupt pin.
Note 2. Input is enabled if the pin is specified as the snooze mode request trigger in software Standby mode while it is used as an external interrupt pin.
Note 3. Input is enabled.

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.

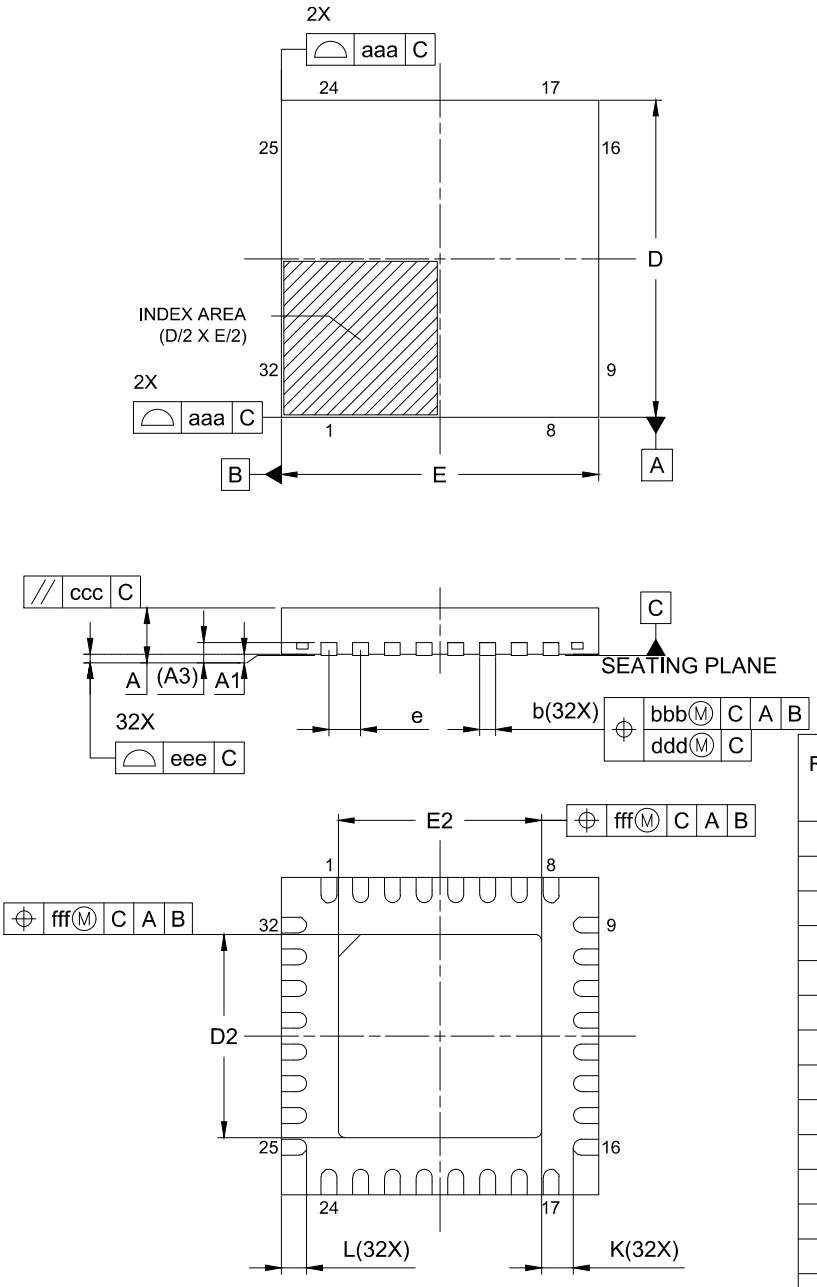
JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-HWQFN048-7x7-0.50	PWQN0048KC-A	0.13 g



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 2.1 HWQFN 48-pin

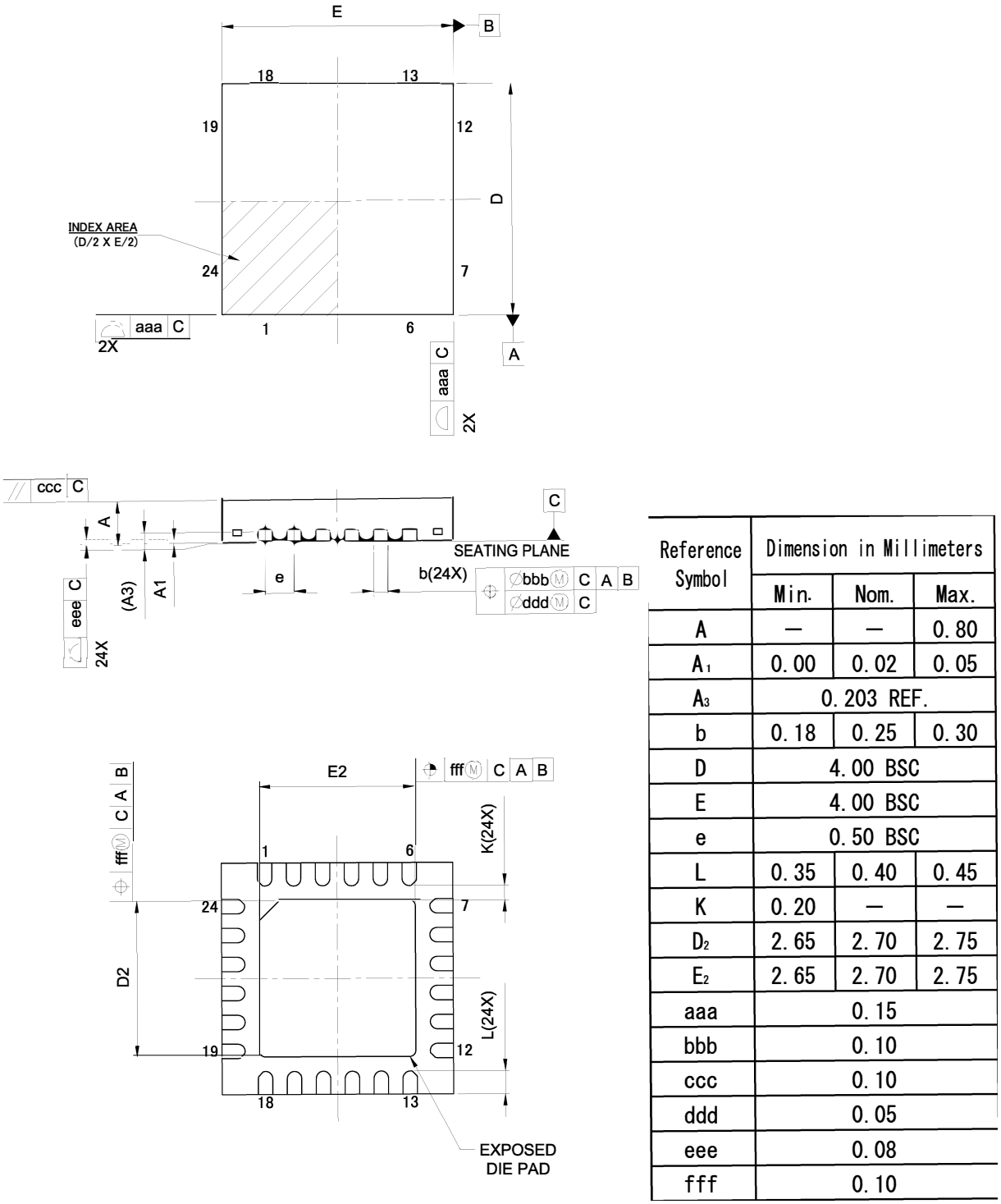
JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-HWQFN032-5x5-0.50	PWQN0032KE-A	0.06



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	0.18	0.25	0.30
D		5.00 BSC	
E		5.00 BSC	
e		0.50 BSC	
L	0.35	0.40	0.45
K	0.20	—	—
D_2	3.15	3.20	3.25
E_2	3.15	3.20	3.25
aaa		0.15	
bbb		0.10	
ccc		0.10	
ddd		0.05	
eee		0.08	
fff		0.10	

Figure 2.2 HWQFN 32-pin

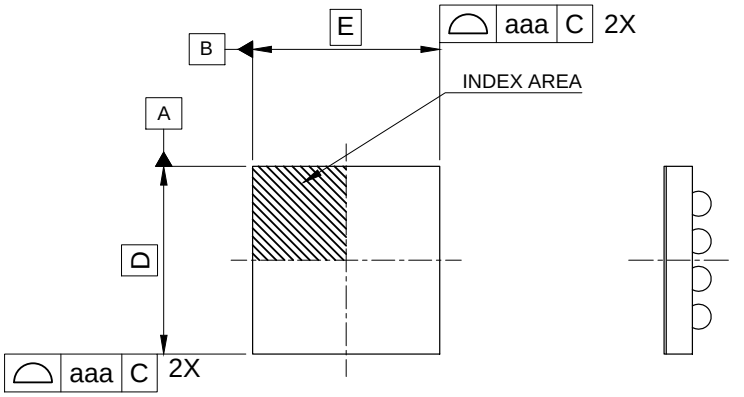
JEITA Package Code	RENESAS Code	MASS(Typ.)[g]
P-HWQFN24-4×4-0.50	PWQN0024KG-A	0.04



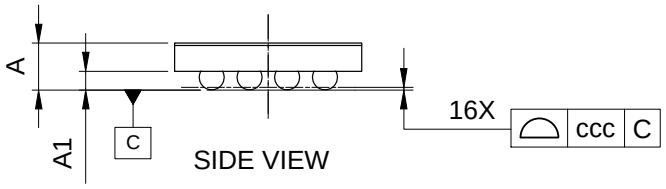
Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
A	—	—	0.80
A_1	0.00	0.02	0.05
A_3		0.203 REF.	
b	0.18	0.25	0.30
D		4.00 BSC	
E		4.00 BSC	
e		0.50 BSC	
L	0.35	0.40	0.45
K	0.20	—	—
D_2	2.65	2.70	2.75
E_2	2.65	2.70	2.75
aaa		0.15	
bbb		0.10	
ccc		0.10	
ddd		0.05	
eee		0.08	
fff		0.10	

Figure 2.3 HWQFN 24-pin

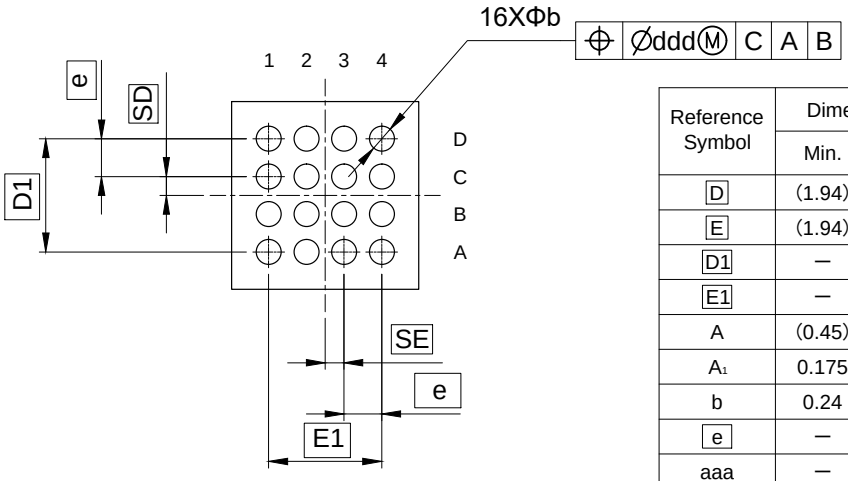
JEITA Package code	RENESAS code	MASS(TYP.)[g]
S-UFBGA16-1.99x1.99-0.40	SUBG0016LC-A	0.01



TOP VIEW



SIDE VIEW



BOTTOM VIEW

Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
D	(1.94)	1.99	(2.04)
E	(1.94)	1.99	(2.04)
D1	—	1.20	—
E1	—	1.20	—
A	(0.45)	(0.50)	0.55
A_1	0.175	0.20	0.225
b	0.24	0.265	0.29
e	—	0.40	—
aaa	—	—	0.05
ccc	—	—	0.05
ddd	—	—	0.05
SD	—	0.200	—
SE	—	0.200	—

Figure 2.4 WLCSP 16-pin

Appendix 3. I/O Registers

This appendix describes I/O register addresses, access cycles, and reset values 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 2)

Name	Description	Base address
SRAM	SRAM Control	0x4000_2000
BUS	BUS Control	0x4000_3000
DTC	Data Transfer Controller	0x4000_5400
ICU	Interrupt Controller	0x4000_6000
CPU_AUX	CPU Auxiliary Registers	0x4001_A000
CPU_DBG	Debug Function	0x4001_B000
SYSC	System Control	0x4001_E000
PORT0	Port 0 Control Registers	0x4004_0000
PORT1	Port 1 Control Registers	0x4004_0020
PORT2	Port 2 Control Registers	0x4004_0040
PORT3	Port 3 Control Registers	0x4004_0060
PORT4	Port 4 Control Registers	0x4004_0080
PFS	Pmn Pin Function Control Register	0x4004_0800
ELC	Event Link Control	0x4004_1000
WDT	Watchdog Timer	0x4004_4200
IWDT	Independent Watchdog Timer	0x4004_4400
CAC	Clock Frequency Accuracy Measurement Circuit	0x4004_4600
MSTP	Module Stop Control B, C, D	0x4004_7000
DAC8	8-bit D/A Converter	0x4005_E000
CRC	CRC Calculator	0x4007_4000
KINT	Key Interrupt Function	0x4008_0000
DOC	Data Operation Circuit	0x4008_5F00
PORGA	Product Organize Register	0x4009_1000
TRNG	True Random Number Generator	0x4009_1100
CMP	Comparator	0x4009_1200
RTC	Realtime Clock	0x4009_2000
REMC	Remote Control Signal Receiver	0x4009_2100
TML32	32-bit Interval Timer	0x4009_2200
IICA0	I^2C Bus Interface 0	0x4009_3000
IICA1	I^2C Bus Interface 1	0x4009_3100
SAU0	Serial Array Unit 0	0x4009_4000
SAU1	Serial Array Unit 1	0x4009_4100
TAU	Timer Array Unit	0x4009_5000
UARTA	Serial Interface UARTA	0x4009_6000
ADC12	12-bit A/D Converter	0x4009_C000

Table 3.1 Peripheral base address (2 of 2)

Name	Description	Base address
FLCN	Flash I/O Registers	0x407E_C000
CLIC	Core-Local Interrupt Controller	0xE200_0000
IMT	Machine Timer	0xE600_0000
DBG	Debug Module	0xE680_0000

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.

The following information applies to Table 3.2:

- 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.

Note: This applies to the number of cycles when access from the CPU does not conflict with the instruction fetching to the external memory or bus access from other bus master such as DTC.

Table 3.2 shows the register access cycles.

Table 3.2 Access cycles (1 of 2)

Peripherals	Address		Number of access cycles				Cycle unit	Related function
			ICLK = PCLK		ICLK > PCLK*1			
	From	To	Read	Write	Read	Write		
RAM, BUS, DTC, ICU, CPU_AUX, CPU_DBG	0x4000_0000	0x4001_BFFF	2				ICLK	Memory Protection Unit, SRAM, Buses, Data Transfer Controller, Interrupt Controller, CPU
SYSC*2	0x4001_E000	0x4001_EFFF	4				ICLK	Low Power Modes, Resets, Low Voltage Detection, Clock Generation Circuit, Register Write Protection
PORT, PFS, ELC	0x4004_0000	0x4004_1FFF	3*3	3	2 to 4*3	2 to 4	PCLKB	I/O Ports, Event Link Control
WDT, IWDT, CAC, MSTP, DAC8	0x4004_2000	0x4005_FFFF	3		2 to 4		PCLKB	Watchdog Timer, Independent Watchdog Timer, Clock Frequency Accuracy Measurement Circuit, Module Stop Control, Data Operation Circuit, 12-bit A/D Converter, 8-bit D/A Converter
CRC	0x4007_4000	0x4007_40FF	3		2 to 4		PCLKB	CRC Calculator
KINT	0x4008_0000	0x4008_00FF	2		2	1 to 3	PCLKB	Key Interrupt Function
DOC	0x4008_5F00	0x4008_5FFF	3		3	2 to 4	PCLKB	Data Operation Circuit

Table 3.2 Access cycles (2 of 2)

Peripherals	Address		Number of access cycles				Cycle unit	Related function
			ICLK = PCLK		ICLK > PCLK*1			
	From	To	Read	Write	Read	Write		
PORGA	0x4009_1000	0x4009_10FF	2		1 to 3		PCLKB	Product Organize Register
TRNG	0x4009_1100	0x4009_11FF	3		2 to 4		PCLKB	True Random Number Generator
CMP, RTC	0x4009_1200	0x4009_20FF	2		2	1 to 3	PCLKB	Comparator, Realtime Clock
REMC, TML32	0x4009_2100	0x4009_22FF	2		1 to 3		PCLKB	Remote Control Signal Receiver, 32-bit Interval Timer
IICA, SAU, TAU, UARTA, ADC12	0x4009_3000	0x4009_C0FF	2		1 to 3		PCLKB	I^2C Bus Interface, Serial Array Unit, Timer Array Unit, Serial Interface UARTA, 12-bit A/D Converter
FLCN	0x407E_0000	0x407F_FFFF	3				ICLK	Temperature Sensor, Flash Control
CLIC, IMT, DBG	0xE200_0000	0xE680_0FFF	2				ICLK	CPU

Note: When accessing the 16-bit register (RDRHL, TDRHL, and CDR), access is 2 cycles more than the value in Table 3.2.
Note 1. If the number of PCLK 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.
Note 2. These values indicate the minimum numbers of cycles for access by the CPU. They do not include the cycles required for changes in the source of the ICLK clock and frequency after changes to the SCKSCR and SCKDIVCR registers.
Note 3. When reading the PCNTR2, PIDR, and PmnPFS* registers, access is (setting value of the PRWCNTR register) cycles more than this value.

Revision History

Revision 1.00 — Nov 15, 2023

Initial release

Revision 1.10 — February 29, 2024

Features:
- Updated information for Connectivity and Timers.

1. Overview:
- Updated Figure 1.1 Block diagram.

2. CPU:
- Added Figure 2.1 for block diagram of RISC-V CPU core.
- Updated Table 2.20 Instruction throughput and latency.
- Updated Table 2.21 Correspondence between instruction types and instructions.

5. Resets:
- Updated the clearing conditions for symbols in RSTSR0 : Reset Status Register 0, RSTSR1 : Reset Status Register 1, and RSTSR2 : Reset Status Register 2.

7. Low Voltage Detection (LVD):
- Updated function description of RN symbol for LVD1CR0 : Voltage Monitor 1 Circuit Control Register 0 and LVD2CR0 : Voltage Monitor 2 Circuit Control Register 0.

12. Interrupt Controller Unit (ICU):
- Updated Note 1 in Figure 12.2 Interrupt path of the ICU and CPU (CLIC).

18. Timer Array Unit (TAU):
- Updated description for One-count mode in Table 18.8 Operation mode selected with OPIRQ bit.
- Removed the Note in TNFEN : TAU Noise Filter Enable Register.
- Updated 18.4.1 Count Clock (fTCLK).
- Updated Figure 18.32 States of operation following set and reset signals.
- Updated Table 18.18 Procedure for operations when the interval timer or square wave output function is to be used.
- Updated Table 18.52 Example of TOL0 settings for the master channel when the one-shot pulse output function is to be used.
- Updated Figure 18.32 States of operation following set and reset signals.

19. 32-bit Interval Timer (TML32):
- Updated description for Step 11 in Table 19.8 Interrupt sources in 8-bit, 16-bit, and 32-bit counter modes.

21. Watchdog Timer (WDT):
- Updated 21.5.1 ICU Event Link Setting Register n (IELSRn) Setting.

22. Independent Watchdog Timer (IWDT):
- Updated 22.4.3 Constraints on the ICU Event Link Setting Register n (IELSRn) Setting.

23. Serial Array Unit (SAU):
- Removed n = 0 to 3 from bit n in Table 23.65 Example of serial output register m (SOm) contents for slave transmission and reception of simplified SPI.
- Updated description in Step 9 of Table 23.109 Procedure for simplified I2C address field transmission.
- Updated description in Step 4 of Table 23.117 Procedure for simplified I2C data transmission.
- Updated description in Step 9 of Table 23.125 Procedure for data reception.

24. I2C Bus Interface (IICA):
- Updated description in 24.4.1 Start Conditions.

25. Serial Interface UARTA (UARTA):
- Updated the Note in Table 25.3 Step of communication procedure.

26. Remote Control Signal Receiver (REMC):
- Updated description for bit RBIT0 and bits RBIT[5:0] of 26.2.17 REMRBIT : Receive Bit Count Register.

29. 12-bit A/D Converter (ADC12):
- Updated Note 4 in Table 29.9 A/D conversion time in Normal mode 1 and 2.
- Updated Note 4 in Table 29.10 A/D conversion time in Low voltage mode 1 and 2.
- Updated the Note in 29.3.6 ADS : Analog Input Channel Specification Register.
- Added a Note in Figure 29.18 Example of hardware trigger no-wait mode (scan mode sequential conversion mode) operation timing.
- Updated step <5> in Table 29.15 Setting up software trigger wait mode.
- Updated step <9> in Table 29.20 Procedure for setting up Snooze mode (hardware trigger).

35. Flash Memory:
- Corrected register name of FCR to FEXCR for Note 1 in 35.3.11 FEXCR : Flash Extra Area Control Register.
- Corrected bit name of FRDY to EXRDY for Note 1 in 35.3.11 FEXCR : Flash Extra Area Control Register.

Revision 1.10 — February 29, 2024

37. Electrical Characteristics:

- Updated Table 37.4 I/O VIH, VIL.
- Updated values in the Typ column of Table 37.10 Operating and standby current (1).
- Updated the Note in Table 37.11 Operating and standby current (2).
- Updated Table 37.12 Operating and standby current (3).
- Updated Table 37.56 Code flash characteristics (2).
- Updated Table 37.57 Code flash characteristics (3).
- Updated Table 37.58 Code flash characteristics (4).
- Updated Table 37.60 Data flash characteristics (2).
- Updated Table 37.61 Data flash characteristics (3).
- Updated Table 37.62 Data flash characteristics (4).

R9A02G021 User’s Manual: Hardware

Publication Date: Rev.1.10 Feb 29, 2024
Rev.1.00 Nov 15, 2023

Published by: Renesas Electronics Corporation

R9A02G021

Renesas Electronics Corporation

R01UH1036EJ0110