User's Manual

32

RA0E2 Group

User's Manual: Hardware

32-Bit MCU
Renesas Advanced (RA) Family
Renesas RA0 Series

Renesas Electronics
www.renesas.com

Rev.1.10 Nov 2025

Notice

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

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

(Rev.5.0-1 October 2020)

Corporate Headquarters

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

Contact Information

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

Trademarks

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

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

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

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

Preface

1. About this document

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

2. Audience

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

3. Renesas Publications

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

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

4. Numbering Notation

The following numbering notation is used throughout this manual:

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

5. Typographic Notation

The following typographic notation is used throughout this manual:

Example	Description
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

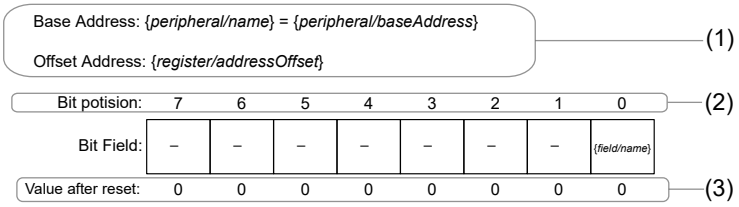
The following terms have special meanings.

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, {*peripheral/name*}, register symbol, {*register/name*}, and address assignment of this register are generally expressed. Base Address and Offset Address mean {*register/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
APB-AP	APB Access Port
BCD	Binary Coded Decimal
HMI	Human Machine Interface
I/O	Input/Output
IrDA	Infrared Data Association
LSB	Least Significant Bit
MSB	Most Significant Bit
MTB	Micro Trace Buffer
NVIC	Nested Vector Interrupt Controller
PC	Program Counter
PFS	Port Function Select
POR	Power-on reset
PWM	Pulse Width Modulation
S/H	Sample and Hold
SP	Stack Pointer
SWD	Serial Wire Debug
SW-DP	Serial Wire-Debug Port
SWJ-DP	Serial Wire JTAG Debug Port
TRNG	True Random Number Generator
UART	Universal Asynchronous Receiver/Transmitter
WFI	Wait for interrupt

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 32 MHz Arm® Cortex®-M23 core, up to 128-KB code flash memory, 16-KB SRAM, 12-bit A/D Converter, Serial interfaces and Safety features, operating temperature range up to Ta:125℃.

Features

■ Arm Cortex-M23 Core

- Armv8-M architecture
- Maximum operating frequency: 32 MHz
- Debug and Trace: DWT, FPB, CoreSight™ MTB-M23
- CoreSight Debug Port: SW-DP

■ Memory

- Up to 128-KB code flash memory
- 2-KB data flash memory (1,000,000 (TYP) program/erase cycles)
- 16-KB SRAM
- Flash read protection (FRP)
- 128-bit unique ID

■ Connectivity

- Serial Array Unit (SAU)
  - Simplified SPI × 6
  - Simplified IIC × 6
  - UART × 2
  - UART (LIN-bus supported) × 1
- Serial Interface UARTA (UARTA) × 2
- I^2C Bus interface (IICA) × 2

■ Analog

- 12-bit A/D Converter (ADC12)
- Temperature Sensor (TSN)

■ Timers

- 16-bit Timer Array Unit (TAU) × 8
- 32-bit interval timer (TML32) × 1
  - 1 channel in 32-bit counter mode
  - 2 channels in 16-bit counter mode
  - 4 channels in 8-bit counter mode

■ Safety

- SRAM parity error check
- Flash area protection
- ADC self-diagnosis function
- Cyclic Redundancy Check (CRC)
- Independent Watchdog Timer (IWDT)
- GPIO readback level detection
- Register write protection
- Illegal memory access detection

■ Security

- True Random Number Generator (TRNG)

■ System and Power Management

- Low power modes
- Realtime Clock (RTC)
- Event Link Controller (ELC)
- Data Transfer Controller (DTC)
- Power-on reset
- Low Voltage Detection (LVD) with voltage settings

■ Multiple Clock Sources

- Main clock oscillator (MOSC) (1 to 20 MHz)
- Sub-clock oscillator (SOSC) (32.768 kHz)
- High-speed on-chip oscillator (HOCO) (24/32 MHz)
- Middle-speed on-chip oscillator (MOCO) (4 MHz)
- Low-speed on-chip oscillator (LOCO) (32.768 kHz)
- Clock trimming function for HOCO/MOCO/LOCO
- Clock out support

■ Up to 60 pins for general I/O ports

- 5-V tolerance, open drain, input pull-up

■ Operating Voltage

- VCC: 1.6 to 5.5 V

■ Operating Temperature and Packages

- Ta = -40℃ to +105℃
  - 64-pin LFQFP (10 mm × 10 mm, 0.5 mm pitch)
  - 48-pin LFQFP (7 mm × 7 mm, 0.5 mm pitch)
  - 48-pin HWQFN (7 mm × 7 mm, 0.5 mm pitch)
  - 32-pin LQFP (7 mm × 7 mm, 0.8 mm pitch)
  - 32-pin HWQFN (5 mm × 5 mm, 0.5 mm pitch)
- Ta = -40℃ to +125℃
  - 64-pin LFQFP (10 mm × 10 mm, 0.5 mm pitch)
  - 48-pin LFQFP (7 mm × 7 mm, 0.5 mm pitch)
  - 48-pin HWQFN (7 mm × 7 mm, 0.5 mm pitch)
  - 32-pin LQFP (7 mm × 7 mm, 0.8 mm pitch)
  - 32-pin HWQFN (5 mm × 5 mm, 0.5 mm pitch)

1. Overview

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

The MCU in this series incorporates an energy-efficient Arm Cortex®-M23 32-bit core, that is particularly well suited for cost-sensitive and low-power applications, with the following features:

- Up to 128-KB code flash memory
- 16-KB SRAM
- Serial Interface (SAU, UARTA, IICA)
- General Purpose Timer (TAU, TML32)
- 12-bit A/D Converter (ADC12)

1.1 Function Outline

Table 1.1 Arm core

Feature	Functional description
Arm Cortex-M23 core	• Maximum operating frequency: up to 32 MHz • Arm Cortex-M23 core: – Revision: r1p0-00rel0 – Armv8-M architecture profile – Single-cycle integer multiplier – 19-cycle integer divider • SysTick timer: – Driven by SYSTICCLK (LOCO) or ICLK

Table 1.2 Memory

Feature	Functional description
Code flash memory	Maximum 128-KB of code flash memory. See section 28, Flash Memory.
Data flash memory	2-KB of data flash memory. See section 28, Flash Memory.
Option-setting memory	The option-setting memory determines the state of the MCU after a reset. See section 6, Option-Setting Memory.
SRAM	On-chip SRAM with parity bit. See section 27, SRAM.

Table 1.3 System (1 of 2)

Feature	Functional description
Operating modes	Operating mode: • Single-chip mode See section 3, Operating Modes.
Resets	The MCU provides 7 resets (RES pin reset, power-on reset, independent watchdog timer reset, voltage monitor 0/1 resets, SRAM parity error reset, software reset). See section 5, Resets.
Low Voltage Detection (LVD)	The Low Voltage Detection (LVD) module monitors the voltage level input to the VCC pin. The detection level can be selected by register settings. The LVD module consists of two separate voltage level detectors (LVD0, LVD1). LVD0 and LVD1 measure the voltage level input to the VCC pin. LVD registers allow your application to configure detection of VCC changes at various voltage thresholds. See section 7, Low Voltage Detection (LVD).

Table 1.3 System (2 of 2)

Feature	Functional description
Clocks	• Main clock oscillator (MOSC) • Sub-clock oscillator (SOSC) • High-speed on-chip oscillator (HOCO) • Middle-speed on-chip oscillator (MOCO) • Low-speed on-chip oscillator (LOCO) • Clock output / Buzzer output support See section 8, Clock Generation Circuit.
Interrupt Controller Unit (ICU)	The Interrupt Controller Unit (ICU) controls which event signals are linked to the Nested Vector Interrupt Controller (NVIC), and the Data Transfer Controller (DTC) modules. The ICU also controls non-maskable interrupts. See section 11, Interrupt Controller Unit (ICU).
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 9, 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 10, Register Write Protection.
Flash Read Protection	The MCU incorporates the flash read protection with one secure regions that include the code flash. The secure region can be protected from non-secure program accesses. A non-secure program cannot access a protected region. See section 13, Flash Read Protection (FRP).
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 the LOCO, 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 20, 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 (1 of 2)

Feature	Functional description
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 functional timer. See section 17, Timer Array Unit (TAU).
32-bit Interval Timer (TML32)	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. See section 18, 32-bit Interval Timer (TML32).

Table 1.6 Timers (2 of 2)

Feature	Functional description
Realtime Clock (RTC)	The Realtime Clock (RTC) 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 19, Realtime Clock (RTC).

Table 1.7 Communication interfaces

Feature	Functional description
Serial Array Unit (SAU)	A Serial Array Unit (SAU) has two units. Unit 0 has four channels and Unit 1 has two channels. Each channel can achieve simplified SPI, UART or simplified IIC. Only UART2 can support LIN-bus. See section 21, Serial Array Unit (SAU)).
I^2C Bus Interface (IICA)	The I^2C Bus Interface (IICA) has 2 channels. The IICA module conforms I^2C (Inter-Integrated Circuit) Bus Interface functions. See section 22, I^2C Bus Interface (IICA).
Serial Interface UARTA (UARTA)	The Serial Interface UARTA (UARTA) has 2 channels. UARTA performs an asynchronous communication. See section 23, Serial Interface UARTA (UARTA).

Table 1.8 Analog

Feature	Functional description
12-bit A/D Converter (ADC12)	A 12-bit successive approximation A/D converter is provided. Up to 15 analog input channels are selectable. Temperature sensor output and internal reference voltage are selectable for conversion. See section 25, 12-bit A/D Converter (ADC12).
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 26, 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. Two CRC-generation polynomials (CRC-CCITT, CRC-32) are available. See section 24, Cyclic Redundancy Check (CRC).

Table 1.10 Security

Feature	Functional description
True Random Number Generator (TRNG)	The True Random Number Generator (TRNG) generates 32-bit random number seeds. See section 29, True Random Number Generator (TRNG).

Table 1.11 I/O ports

Feature	Functional description
I/O ports	• I/O ports for the 64-pin LFQFP – I/O pins: 57 – Input pins: 3 – Pull-up resistors: 40 – N-ch open-drain outputs: 42 – 5-V tolerance: 4 • I/O ports for the 48-pin LFQFP/HWQFN – I/O pins: 41 – Input pins: 3 – Pull-up resistors: 26 – N-ch open-drain outputs: 28 – 5-V tolerance: 4 • I/O ports for the 32-pin LQFP/HWQFN – I/O pins: 26 – Input pins: 3 – Pull-up resistors: 16 – N-ch open-drain outputs: 15 – 5-V tolerance: 2

1.2 Block Diagram

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

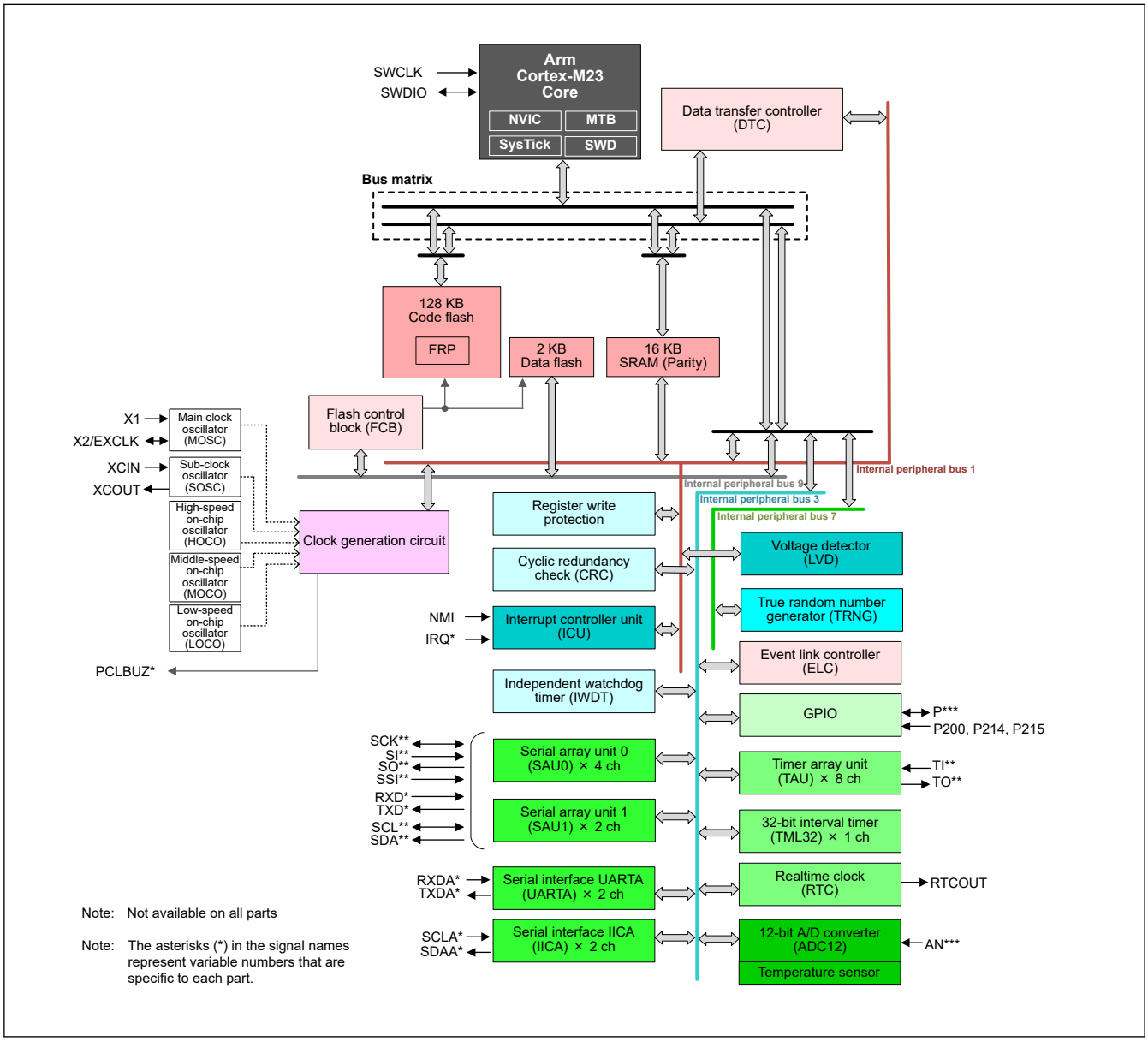
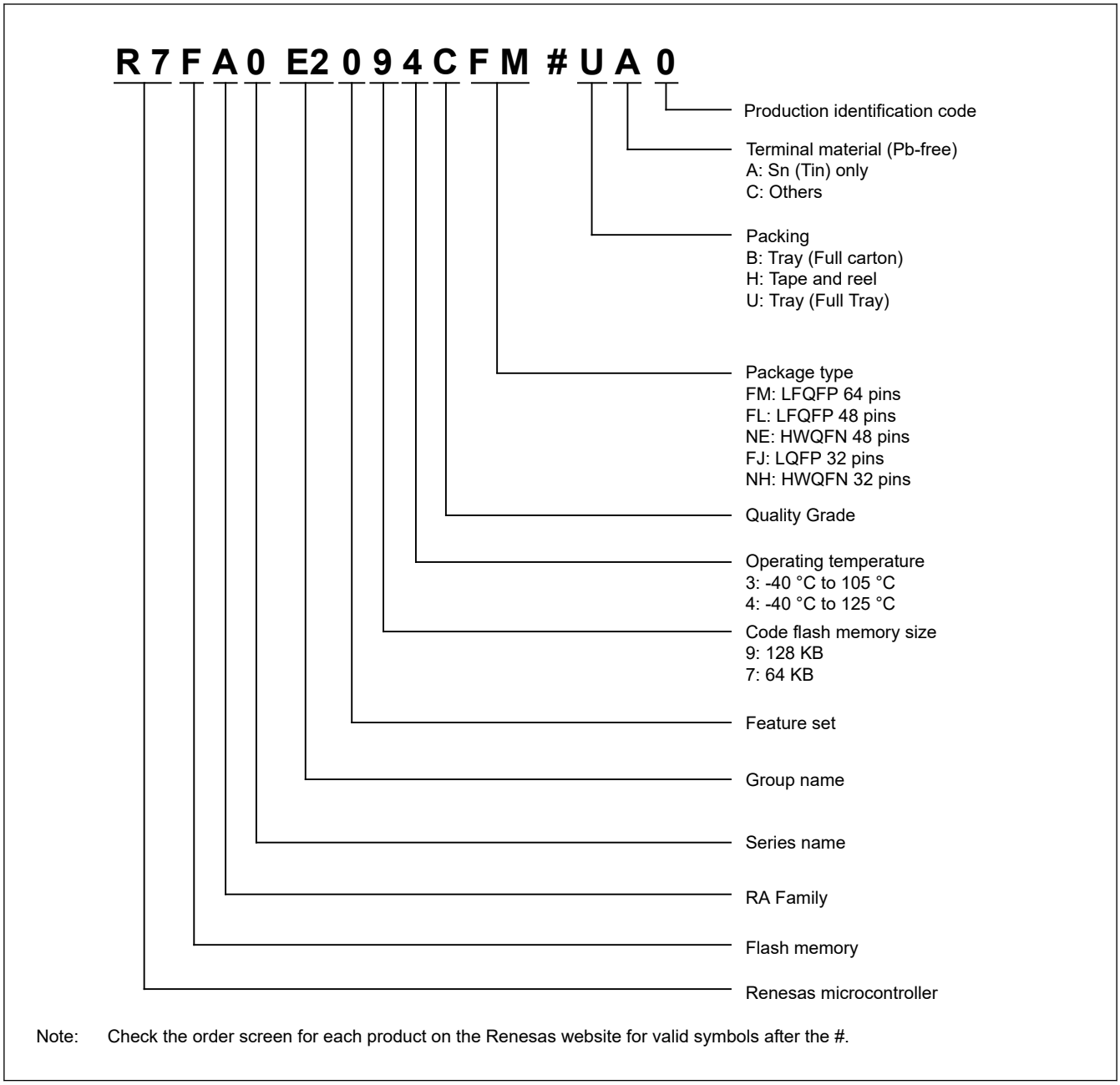


Figure 1.1 Block diagram

1.3 Part Numbering

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



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

Figure 1.2 Part numbering scheme

Table 1.12 Product list

Product part number	Package code	Code flash	Data flash	SRAM	Operating temperature
R7FA0E2094CFM	PLQP0064KB-C	128 KB	2 KB	16 KB	-40 to +125°C
R7FA0E2094CFL	PLQP0048KB-B				
R7FA0E2094CNE	PWQN0048KC-A				
R7FA0E2094CFJ	PLQP0032GB-A				
R7FA0E2094CNH	PWQN0032KE-A				
R7FA0E2093CFM	PLQP0064KB-C				-40 to +105°C
R7FA0E2093CFL	PLQP0048KB-B				
R7FA0E2093CNE	PWQN0048KC-A				
R7FA0E2093CFJ	PLQP0032GB-A				
R7FA0E2093CNH	PWQN0032KE-A				
R7FA0E2074CFM	PLQP0064KB-C	64 KB	2 KB	16 KB	-40 to +125°C
R7FA0E2074CFL	PLQP0048KB-B				
R7FA0E2074CNE	PWQN0048KC-A				
R7FA0E2074CFJ	PLQP0032GB-A				
R7FA0E2074CNH	PWQN0032KE-A				
R7FA0E2073CFM	PLQP0064KB-C				-40 to +105°C
R7FA0E2073CFL	PLQP0048KB-B				
R7FA0E2073CNE	PWQN0048KC-A				
R7FA0E2073CFJ	PLQP0032GB-A				
R7FA0E2073CNH	PWQN0032KE-A				

1.4 Function Comparison

Table 1.13 Function comparison

Parts number		R7FA0E2094CFM R7FA0E2093CFM	R7FA0E2074CFM R7FA0E2073CFM	R7FA0E2094CFL R7FA0E2093CFL R7FA0E2094CNE R7FA0E2093CNE	R7FA0E2074CFL R7FA0E2073CFL R7FA0E2074CNE R7FA0E2073CNE	R7FA0E2094CFJ R7FA0E2093CFJ R7FA0E2094CNH R7FA0E2093CNH	R7FA0E2074CFJ R7FA0E2073CFJ R7FA0E2074CNH R7FA0E2073CNH
Pin count		64		48		32	
Package		LFQFP		LFQFP/HWQFN		LQFP/HWQFN	
Code flash memory		128 KB	64 KB	128 KB	64 KB	128 KB	64 KB
Data flash memory		2 KB		2 KB		2 KB	
SRAM (Parity)		16 KB		16 KB		16 KB	
System	CPU clock	32 MHz		32 MHz		32 MHz	
	Sub-clock oscillator	Yes		Yes		Yes	
	ICU	Yes		Yes		Yes	
Event control	ELC	Yes		Yes		Yes	
DMA	DTC	Yes		Yes		Yes	
Timers	TAU	8 (PWM outputs: 7)		8 (PWM outputs: 7)		8 (PWM outputs: 7)	
	TML32	1 (32-bit counter mode), 2 (16-bit counter mode), 4 (8-bit counter mode)		1 (32-bit counter mode), 2 (16-bit counter mode), 4 (8-bit counter mode)		1 (32-bit counter mode), 2 (16-bit counter mode), 4 (8-bit counter mode)	
	RTC	Yes		Yes		Yes	
	IWDT	Yes		Yes		Yes	
Communication	SAU*1	6 (simplified SPI), 6 (simplified IIC), 2 (UART), 1 (UART supporting LIN-bus)		5 (simplified SPI), 6 (simplified IIC), 2 (UART), 1 (UART supporting LIN-bus)		3 (simplified SPI), 4 (simplified IIC), 2 (UART), 1 (UART supporting LIN-bus)	
	UARTA	2		2		2	
	IICA	2		2		2	
Analog	ADC12	15		13		10	
	TSN	Yes		Yes		Yes	
Data processing	CRC	Yes		Yes		Yes	
Security		TRNG		TRNG		TRNG	
I/O ports	I/O pins	57		41		26	
	Input pins	3		3		3	
	Pull-up resistors	40		26		16	
	N-ch open-drain outputs	42		28		15	
	5-V tolerance	4		4		2	

Note 1. SAU consists of several channels. Each channel can be assigned only one function at a time.

1.5 Pin Functions

Table 1.14 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	X2	I/O	Pins for a crystal resonator. An external clock signal can be input through the X2 pin.
	X1	Input	
	XCIN	Input	Input/output pins for the sub-clock oscillator. Connect a crystal resonator between XCOUT and XCIN.
	XCOUT	Output	
	PCLBUZ0, PCLBUZ1	Output	Clock output / Buzzer output
	EXCLK	Input	External clock input for the main clock
System control	RES	Input	Reset signal input pin. The MCU enters the reset state when this signal goes low.
On-chip debug	SWDIO	I/O	Serial wire debug data input/output pin
	SWCLK	Input	Serial wire clock pin
Interrupt	NMI	Input	Non-maskable interrupt request pin
	IRQ0 to IRQ7	Input	Maskable interrupt request pins
TAU	TI00 to TI07	Input	Pins for inputting an external counting clock/capture trigger to 16-bit timers 00 to 07
	TO00 to TO07	Output	Timer output pins for 16-bit timers 00 to 07
RTC	RTCOUT	Output	Output pin for 1-Hz clock
IICA	SCLA0, SCLA1	I/O	Input/output pins for the clock
	SDAA0, SDAA1	I/O	Input/output pins for data
SAU	SCK00, SCK01, SCK10, SCK11, SCK20, SCK21	I/O	Serial clock I/O pins for serial interfaces SPI00, SPI01, SPI10, SPI11, SPI20 and SPI21
	SI00, SI01, SI10, SI11, SI20, SI21	Input	Serial data input pins for serial interfaces SPI00, SPI01, SPI10, SPI11, SPI20 and SPI21
	SO00, SO01, SO10, SO11, SO20, SO21	Output	Serial data output pins for serial interfaces SPI00, SPI01, SPI10, SPI11, SPI20 and SPI21
	SSI00	Input	Chip select pin for serial interfaces SPI00
	SCL00, SCL01, SCL10, SCL11, SCL20, SCL21	Output	Serial clock output pins for serial interfaces IIC00, IIC01, IIC10, IIC11, IIC20 and IIC21
	SDA00, SDA01, SDA10, SDA11, SDA20, SDA21	I/O	Serial data I/O pins for serial interfaces IIC00, IIC01, IIC10, IIC11, IIC20 and IIC21
	RXD0, RXD1, RXD2	Input	Serial data input pins for serial interfaces UART0, UART1, and UART2
	TXD0, TXD1, TXD2	Output	Serial data output pins for serial interfaces UART0, UART1, and UART2
UARTA	RXDA0, RXDA1	Input	Serial data input pin for the UARTA0 and UARTA1 serial interface
	TXDA0, TXDA1	Output	Serial data output pin for the UARTA0 and UARTA1 serial interface
Analog power supply	VREFH0	Input	Analog reference voltage supply pin for the ADC12. Connect this pin to external reference voltage or VCC.
	VREFL0	Input	Analog reference ground pin for the ADC12. Connect this pin to external reference ground voltage or VSS.

Table 1.14 Pin functions (2 of 2)

Function	Signal	I/O	Description
ADC12	AN000 to AN012, AN021 to AN022	Input	Input pins for the analog signals to be processed by the A/D converter.
I/O ports	P000 to P004, P008 to P015	I/O	General-purpose input/output pins
	P100 to P115	I/O	General-purpose input/output pins
	P200	Input	General-purpose input pin
	P201, P204 to P208, P212, P213	I/O	General-purpose input/output pins
	P214, P215	Input	General-purpose input pins
	P300 to P304	I/O	General-purpose input/output pins
	P400 to P403, P407 to P411	I/O	General-purpose input/output pins
	P500 to P502	I/O	General-purpose input/output pins
	P913 to P915	I/O	General-purpose input/output pins

1.6 Pin Assignments

Figure 1.3 to Figure 1.5 show the pin assignments from the top view.

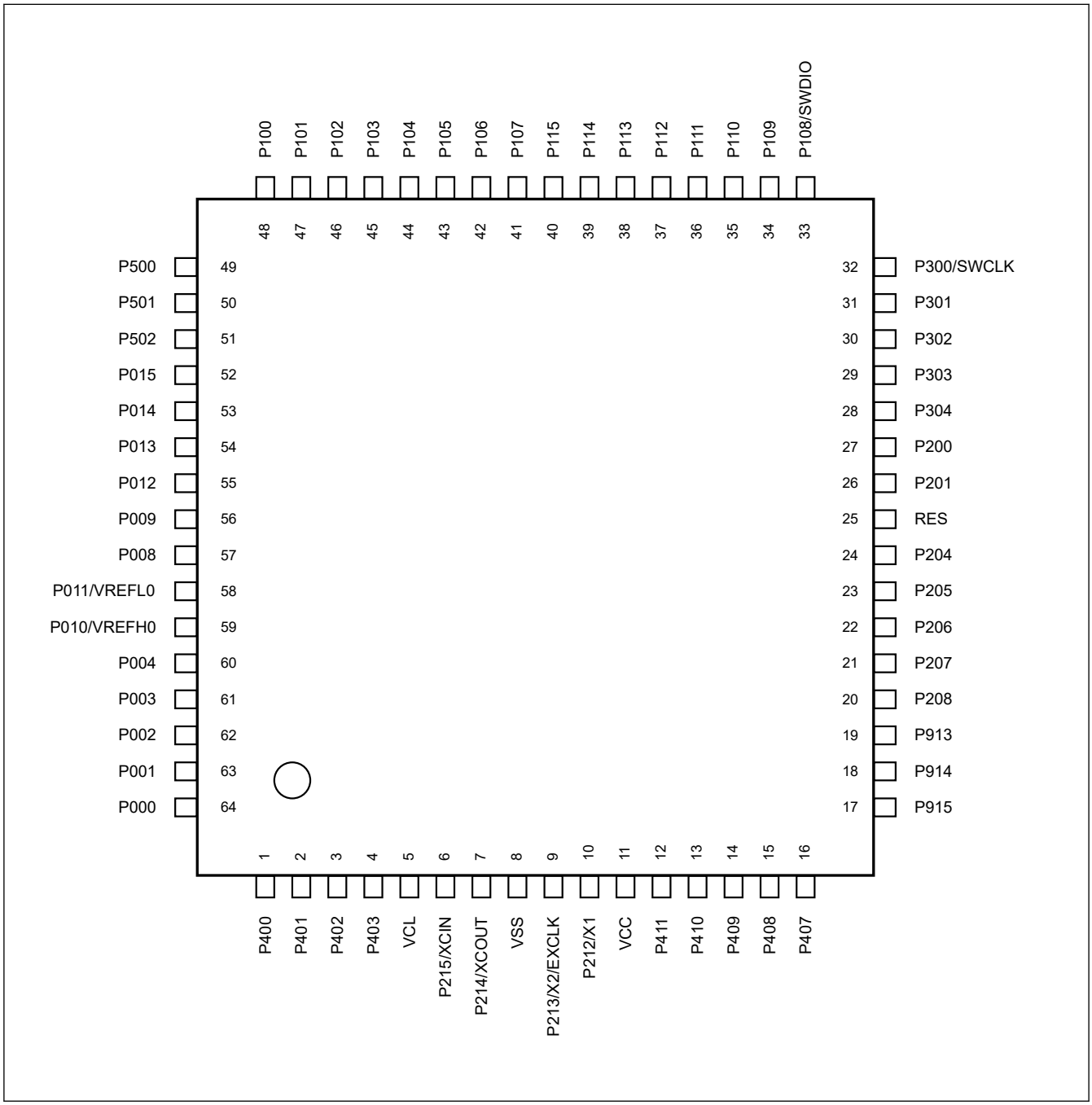


Figure 1.3 Pin assignment for LFQFP 64-pin (top view)

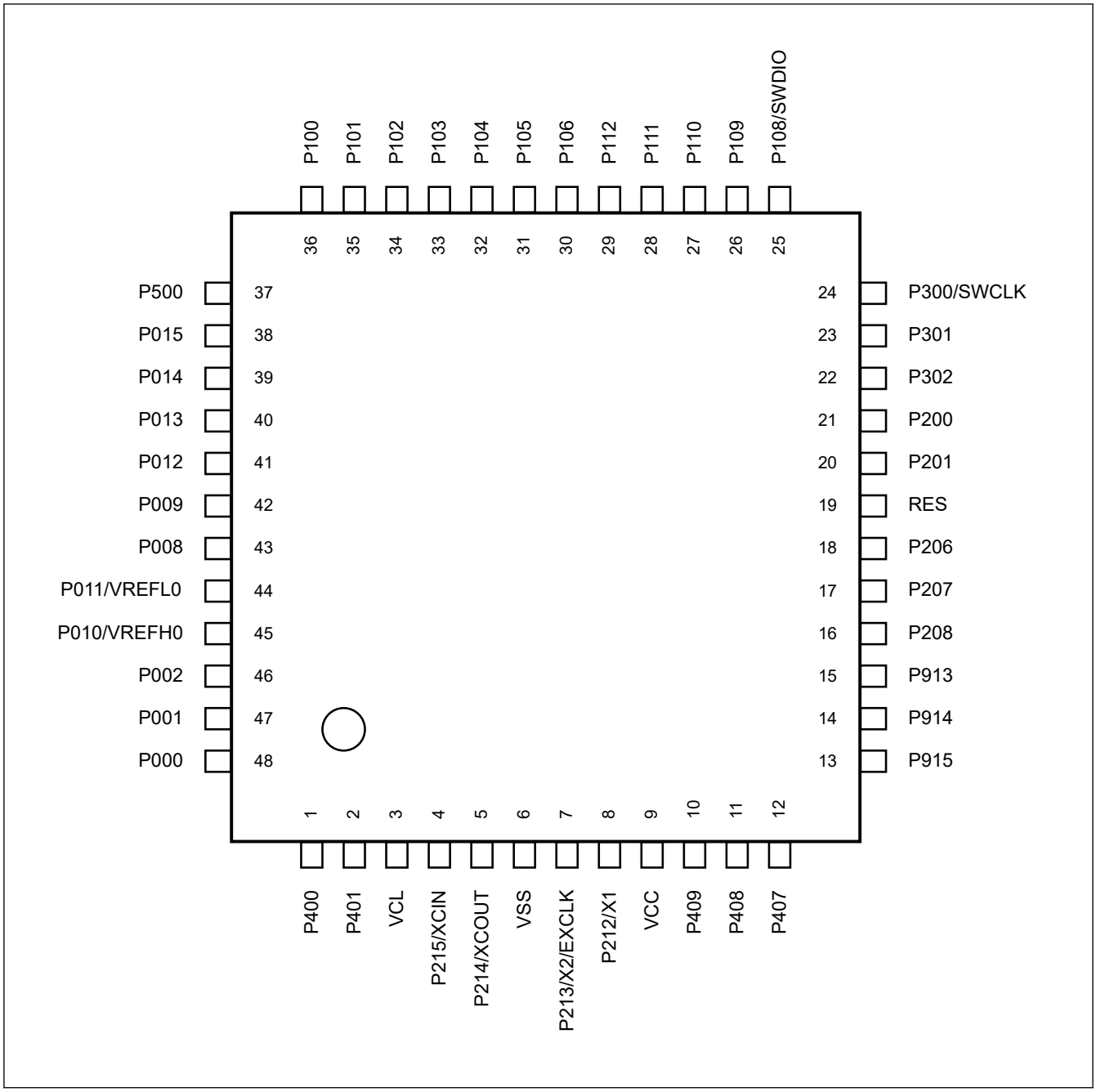


Figure 1.4 Pin assignment for LFQFP / HWQFN 48-pin (top view)

Note: For the QFN package product, solder the exposed die pad to the PCB.
The potential of the exposed die pad is recommended to design as electrically open.

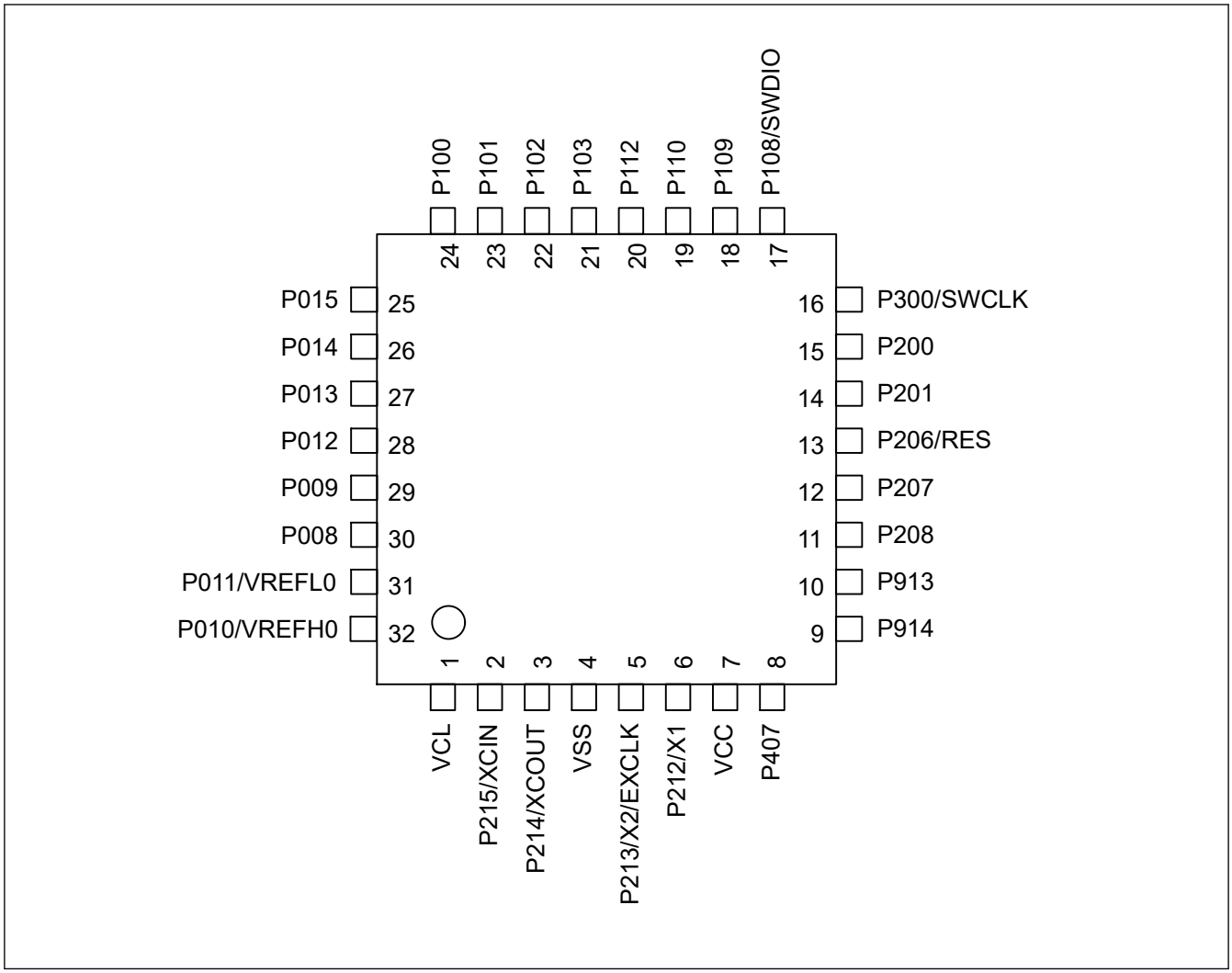


Figure 1.5 Pin assignment for LQFP / HWQFN 32-pin (top view)

Note: For the QFN package product, solder the exposed die pad to the PCB.
The potential of the exposed die pad is recommended to design as electrically open.

1.7 Pin Lists

Table 1.15 Pin list (1 of 2)

Pin number			Power, System, Clock, Debug	I/O ports	Interrupt	Timers		Communication interfaces			Analogs
64-pin	48-pin	32-pin				TAU	RTC	SAU	IICA	UARTA	ADC
1	1	—	—	P400	—	—	—	—	SCLA1_D	—	—
2	2	—	—	P401	—	—	—	—	SDAA1_D	—	—
3	—	—	—	P402	IRQ2_D	—	—	TXD2_B/ SO20_B	—	TXDA0_F	—
4	—	—	—	P403	IRQ4_E	—	—	RXD2_B/SI20_B/ SDA20_B	—	RXDA0_F	—
5	3	1	VCL	—	—	—	—	—	—	—	—
6	4	2	XCIN	P215	—	—	—	—	—	—	—
7	5	3	XCOUT	P214	—	—	—	—	—	—	—
8	6	4	VSS	—	—	—	—	—	—	—	—
9	7	5	X2/EXCLK	P213	IRQ0_B	TI00_A/TI02_B/ TO02_B	—	TXD1_A/SO11_A	SDAA0_B	TXDA0_B	—
10	8	6	X1	P212	IRQ1_B	TO00_A/TI03_C/ TO03_C	—	RXD1_A/SI11_A/ SDA11_A	SCLA0_B	RXDA0_B	—
11	9	7	VCC	—	—	—	—	—	—	—	—
12	—	—	—	P411	IRQ3_D	TI01_C/TO01_C	—	SCK11_D/ SCL11_D	SDAA0_E	TXDA1_D	—
13	—	—	—	P410	IRQ4_D	TI02_C/TO02_C	—	SCK20_B/ SCL20_B/ SSI00_D	SCLA0_E	RXDA1_D	—
14	10	—	—	P409	IRQ6_B	TI03_E/TO03_E	—	SCK11_C/ SCL11_C	—	—	—
15	11	—	—	P408	IRQ7_B	TI04_C/TO04_C	—	—	SCLA1_F	—	—
16	12	8	PCLBUZ0_C	P407	IRQ4_C	—	RTCOUT_A	SCK11_A/ SCL11_A	SDAA1_F*1	—	—
17	13	—	—	P915	—	—	—	SO01_B	—	—	—
18	14	9	—	P914	—	—	—	—	SCLA0_A	—	—
19	15	10	—	P913	—	—	—	—	SDAA0_A	—	—
20	16	11	—	P208	IRQ3_C	TI00_B	—	SCK01_B*1/ SCL01_B	SDAA1_A	TXDA0_A	—
21	17	12	—	P207	IRQ2_C	TO00_B	—	SI01_B*1/ SDA01_B	SCLA1_A	RXDA0_A	—
22	18	—	—	P206	IRQ0_C	—	—	SO01_A*3	SDAA1_E*3	TXDA1_E*3	—
23	—	—	PCLBUZ1_A	P205	IRQ5_C	—	—	SI01_A/ SDA01_A	SCLA1_E	RXDA1_E	—
24	—	—	—	P204	—	—	—	SCK01_A/ SCL01_A	—	—	—
25	19	13	RES	P206 *2	—	—	—	—	—	—	—
26	20	14	PCLBUZ0_A	P201	IRQ5_B	TI05_B/TO05_B	RTCOUT_B	SCK11_B/ SCL11_B/ SSI00_B	—	—	—
27	21	15	—	P200	IRQ0_A/NMI	—	—	—	—	—	—
28	—	—	—	P304	—	—	—	—	—	—	—
29	—	—	—	P303	—	—	—	SO21_A	—	—	—
30	22	—	—	P302	IRQ0_D	TI05_C/TO05_C	—	SCK21_A*3/ SCL21_A	SDAA1_C	TXDA1_C	—
31	23	—	—	P301	IRQ6_A	TI06_B/TO06_B	—	SI21_A*3/ SDA21_A	SCLA1_C	RXDA1_C	—
32	24	16	SWCLK	P300	—	TI04_B/TO04_B	—	—	—	—	—
33	25	17	SWDIO	P108	—	TI03_B/TO03_B	—	—	—	—	—
34	26	18	PCLBUZ1_B	P109	IRQ4_B	TI02_A/TO02_A	—	TXD2_A/ SO20_A	SDAA0_C	TXDA0_C	—

Table 1.15 Pin list (2 of 2)

Pin number			Power, System, Clock, Debug	I/O ports	Interrupt	Timers		Communication interfaces			Analogs
64-pin	48-pin	32-pin				TAU	RTC	SAU	IICA	UARTA	ADC
35	27	19	—	P110	IRQ3_B	TI01_A/TO01_A	—	RXD2_A/SI20_A/ SDA20_A	SCLA0_C	RXDA0_C	—
36	28	—	—	P111	IRQ1_C	TI07_B/TO07_B	—	—	—	—	—
37	29	20	—	P112	IRQ2_B	TI03_A/TO03_A	—	SCK20_A/ SCL20_A/ SSI00_C	—	—	—
38	—	—	—	P113	—	—	—	SO21_B	—	—	—
39	—	—	—	P114	—	—	—	SI21_B/ SDA21_B	—	—	—
40	—	—	—	P115	—	—	—	SCK21_B/ SCL21_B	—	—	—
41	—	—	—	P107	IRQ7_D	—	—	—	—	—	—
42	30	—	—	P106	IRQ0_E	—	—	SO10_A	—	TXDA1_B	—
43	31	—	—	P105	IRQ1_D	TI01_D/TO01_D/ TO00_D	—	SI10_A/ SDA10_A	—	RXDA1_B	—
44	32	—	—	P104	IRQ6_C	TI02_D/TO02_D/ TI00_D	—	SCK10_A/ SCL10_A	—	—	—
45	33	21	—	P103	IRQ5_A	TI05_A/TO05_A	—	SSI00_A	SDAA1_B	TXDA1_A	—
46	34	22	PCLBUZ0_B	P102	IRQ4_A	TI06_A/TO06_A/ TO00_C	RTCOUT_C	SCK00_A/ SCL00_A	SCLA1_B	RXDA1_A	—
47	35	23	—	P101	IRQ3_A	TI07_A/TO07_A/ TI00_C	—	TXD0_A/ SO00_A	SDAA0_D	TXDA0_D	AN021
48	36	24	—	P100	IRQ2_A	TI04_A/TO04_A/ TI01_B/TO01_B	—	RXD0_A/SI00_A/ SDA00_A	SCLA0_D	RXDA0_D	AN022
49	37	—	—	P500	—	TI03_D/TO03_D	—	SCK00_B*3/ SCL00_B*3	—	—	—
50	—	—	—	P501	—	TI04_D/TO04_D	—	TXD0_B/ SO00_B	SDAA0_F	TXDA0_E	—
51	—	—	—	P502	IRQ5_D	—	—	RXD0_B/SI00_B/ SDA00_B	SCLA0_F	RXDA0_E	—
52	38	25	—	P015	IRQ1_A	—	—	—	—	—	AN007
53	39	26	—	P014	—	—	—	—	—	—	AN006
54	40	27	—	P013	—	—	—	—	—	—	AN005
55	41	28	—	P012	—	—	—	—	—	—	AN004
56	42	29	—	P009	—	—	—	—	—	—	AN003
57	43	30	—	P008	—	—	—	—	—	—	AN002
58	44	31	VREFL0	P011	—	—	—	—	—	—	AN001
59	45	32	VREFH0	P010	—	—	—	—	—	—	AN000
60	—	—	—	P004	IRQ2_E	—	—	—	—	—	AN012
61	—	—	—	P003	—	—	—	—	—	—	AN011
62	46	—	—	P002	IRQ7_C	—	—	—	—	—	AN010
63	47	—	—	P001	IRQ7_A	—	—	—	—	—	AN009
64	48	—	—	P000	IRQ6_D	—	—	—	—	—	AN008

Note 1. Available only in 48-pin and 64-pin products.
Note 2. Available only in 32-pin products.
Note 3. Available only in 64-pin products.

Note: Some signal names have _A, _B, _C, _D, _E, or _F suffixes, but these suffixes can be ignored when assigning functionality, except for SAU and IICA. For SAU and IICA, only signals, except for SCK11, SCL11, and SSI00, bearing the same suffix can be selected. Assigning the same function to two or more pins simultaneously is prohibited.

2. CPU

The MCU is based on the Arm® Cortex®-M23 core.

2.1 Overview

2.1.1 CPU

- Arm Cortex-M23
  - Revision: r1p0-00rel0
  - Armv8-M architecture profile
  - Main Extension is not implemented
  - Single-cycle integer multiplier
  - 19-cycle integer divider
- SysTick timer
  - Driven by SYSTICCLK (LOCO) or ICLK

See reference 1. and reference 2. in section 2.8. References for details.

2.1.2 Debug

- Arm® CoreSight™ MTB-M23
  - Revision: r0p0-00rel0
  - Buffer size: 1 KB of 16-KB MTB SRAM
- Data Watchpoint Unit (DWT)
  - 2 comparators for watchpoints
- Flash Patch and Break point Unit (FPB)
  - 4 instruction comparators
- CoreSight Debug Access Port (DAP)
  - Serial Wire-Debug Port (SW-DP)
- Debug Register Module (DBGREG)
  - Reset control
  - Halt control

See reference 1. and reference 2. in section 2.8. References for details.

2.1.3 Operating Frequency

The operating frequencies for the MCU are as follows:

- CPU: maximum 32 MHz
- Serial Wire Debug (SWD) interface: maximum 12.5 MHz

2.1.4 Block Diagram

Figure 2.1 shows a block diagram of the Cortex-M23 core.

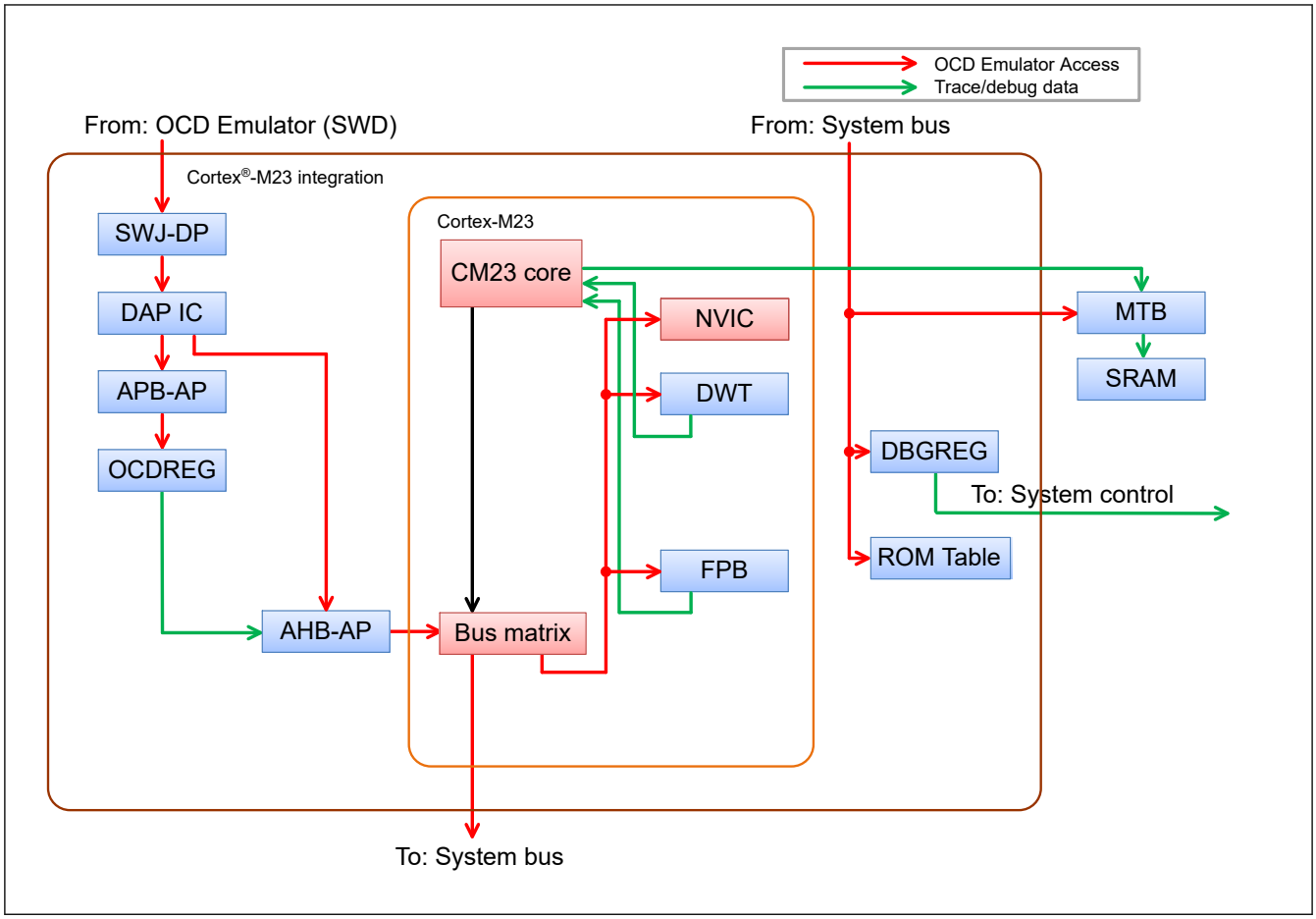


Figure 2.1 Cortex-M23 block diagram

2.2 Implementation Options

Table 2.1 shows the implementation options of the MCU.

Table 2.1 Implementation options (1 of 2)

Option	Implementation
Non-secure MPU	Not included
Secure MPU	Not included
Security extension	Not included
Single-cycle multiplier	Included
Divider	Included, 19 cycles
Number of interrupts	64*1
Number of Wakeup Interrupt Controllers (WIC)	Not included
Cross Trigger Interface (CTI)	Not included
Micro Trace Buffer (MTB)	Included
Embedded Trace Macrocell (ETM)	Not included
Multi-drop support for serial wire	Not supported
Sleep mode power saving	Sleep mode and other low power modes are supported. For more details, see section 9, Low Power Modes. Note: SCB.SCR.SLEEPDEEP is ignored.
Endianness	Little-endian
SysTick timer	Included

Table 2.1 Implementation options (2 of 2)

Option	Implementation	
SYST_CALIB register (0x4000_0147)	Bit [31] = 0 Bit [30] = 1 Bits [29:24] = 0x00 Bits [23:0] = 0x000147	Reference clock provided TENMS value is inexact Reserved TENMS: (32768 × 10 ms) - 1/32.768 kHz = 326.66 decimal = 327 with skew = 0x000147
Event input/output	Not implemented	
System reset request output	The SYSRESETREQ bit in Application Interrupt and Reset Control Register causes a CPU reset	
Auxiliary fault inputs (AUXFAULT)	Not implemented	

Note 1. Some of them are reserved. For the detail of the interrupt sources, section 11.3.1. Interrupt Vector Table

2.3 SWD Interface

Table 2.2 shows the SWD pins.

Table 2.2 SWD pins

Name	I/O	Function	What to do when not in use
SWCLK*1	Input	Serial wire clock pin	Pull-up
SWDIO*1	I/O	Serial wire data I/O pin	Pull-up

Note 1. After the reset, the built-in pull-up is enabled for that pin.

2.4 Debug Function

2.4.1 Debug Mode Definition

Table 2.3 shows the CPU debug modes and usage conditions.

Table 2.3 CPU debug mode and conditions

Conditions		Mode	
OCD connect*1	SWD authentication	Debug mode	Debug authentication*2
Not connected	—	User mode	Disabled
Connected	Failed	User mode	Disabled
Connected	Passed	OCD mode	Enabled

Note: When CSYSPWRUPREQ bit is set, OFS1.PORTSELB value is masked and fixed to the RES input.
Note: Do not use on-chip debugging with products designated as part of mass production, because using this function may cause the guaranteed number of times the flash memory is rewritten to be exceeded, and product reliability therefore cannot be guaranteed.
Note 1. OCD connect is determined by the CDBGPWRUPREQ bit and CSYSPWRUPREQ bit output in the SWJ-DP register. The bit can only be written by the OCD.
However, the level of the bit can be confirmed by reading the DBGSTR.CDBGPWRUPREQ bit.
Note 2. Debug authentication is defined by the Armv8-M Architecture. Enabled means that both invasive and non-invasive CPU debugging are permitted. Disabled means that both are not permitted.

2.4.2 Debug Mode Effects

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

2.4.2.1 Low power mode

All CoreSight debug components can retain the register settings even when the CPU enters Software Standby, or Snooze mode. However, AHB-AP cannot respond to On-Chip Debug (OCD) emulator access in these low power modes. The OCD emulator must wait for cancellation of the low power mode to access the CoreSight debug components. To request low power mode cancellation, the OCD emulator can set the DBIRQ bit in the MCUCTRL register. For details, see section 2.5.6.3. MCUCTRL : MCU Control Register.

2.4.2.2 Reset

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

Table 2.4 Reset or interrupt and mode setting

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

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

2.4.3 Trace Control (for the MTB)

The Micro Trace Buffer (MTB) has programmable registers to control the behavior of the trace features and the POSITION, MASTER, FLOW, and BASE registers. Table 2.5 shows the registers in offset order from the base address.

Table 2.5 Address of MTB registers

Address	Register	Value on reset
MTB_BASE + 0x000	MTB_POSITION	Bits [31:0] = UNKNOWN
MTB_BASE + 0x004	MTB_MASTER	Bits [31] = 0, Bits [30:10] = UNKNOWN, Bits [9:8] = 0, Bits [7] = 1, Bits [6:5] = 0, Bits [4:0] = UNKNOWN
MTB_BASE + 0x008	MTB_FLOW	Bits [31:2] = UNKNOWN, Bits [1:0] = 0
MTB_BASE + 0x00C	MTB_BASE	Bits [31:0] = 0x2000_4000

Note: MTB_BASE = 0x4001_9000

For more information on these registers, see the *ARM® CoreSight™ MTB-M23 Technical Reference Manual (ARM DDI 0564C)*.

Note: Do not attempt to access reserved or unused address locations.

The MTB for trace is limited from 0x2000_4000 to 0x2000_7FFF.

2.4.4 CoreSight™ (for MTB)

See the *ARM® CoreSight™ Architecture Specification* for more information about the registers and access types. Table 2.6 shows the registers in offset order from the base address.

Table 2.6 Address of CoreSight

Address	Register
MTB_BASE + 0xFF0 to 0xFFC	Component ID
MTB_BASE + 0xFE0 to 0xFDC	Peripheral ID
MTB_BASE + 0xFCC	Device Type Identifier
MTB_BASE + 0xFC8	Device Configuration
MTB_BASE + 0xFBC	Device Architecture
MTB_BASE + 0xFB8	Authentication Status
MTB_BASE + 0xFB4	Lock Status
MTB_BASE + 0xFB0	Lock Access

Note: MTB_BASE = 0x4001_9000

For more information on these registers, see the *ARM® CoreSight™ MTB-M23 Technical Reference Manual (ARM DDI 0564C)*.

Note: Do not attempt to access reserved or unused address locations.

2.5 Programmers Model

2.5.1 Address Spaces

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

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

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

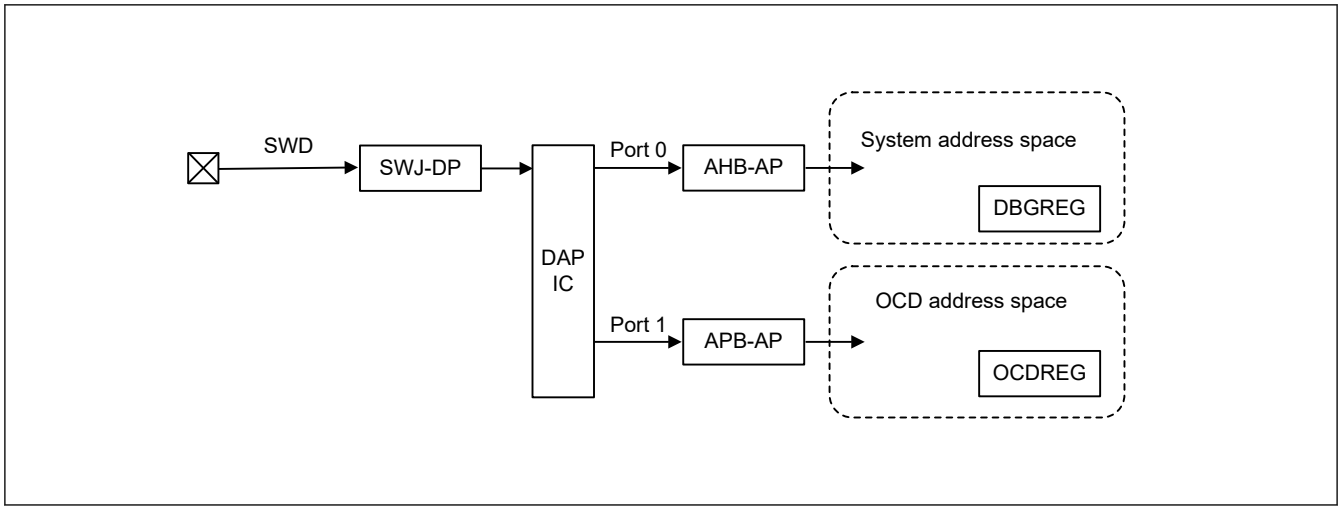


Figure 2.2 SWD authentication block diagram

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

2.5.2 Cortex-M23 Peripheral Address Map

In the system address space, the Cortex-M23 core has a Private Peripheral Bus (PPB) that can only be accessed from the CPU and OCD emulator. Table 2.7 shows the address map of the MCU.

Table 2.7 Cortex-M23 peripheral address map

Component name	Start address	End address	Note
DWT	0xE000_1000	0xE000_1FFF	See reference 2. in section 2.8. References
FPB	0xE000_2000	0xE000_2FFF	See reference 2. in section 2.8. References
SCS	0xE000_E000	0xE000_EFFF	See reference 2. in section 2.8. References

2.5.3 External Debug Address Map

In the system address space, the Cortex-M23 core has external debug components. These components can be accessed from the CPU and other bus masters through the system bus. Table 2.8 shows the address map of the Cortex-M23 external debug components.

Table 2.8 External debug address map

Component name	Start address	End address	Note
MTB (SRAM area)	0x2000_4000	0x2000_7FFF	MTB uses up to 1 KB as trace buffer See reference 6. in section 2.8. References.
MTB (SFR area)	0x4001_9000	0x4001_9FFF	See reference 6. in section 2.8. References.
ROM Table	0x4001_A000	0x4001_AFFF	See reference 6. in section 2.8. References.

2.5.4 CoreSight ROM Table

The MCU contains one CoreSight ROM Table, which lists all components implemented in the user area.

2.5.4.1 ROM entries

Table 2.9 shows the ROM entries. The OCD emulator can use the ROM entries to determine which components are implemented in a system. See reference 4. in section 2.8. References for details.

Table 2.9 ROM entries

#	Address	Access size	R/W	Value	Target module pointer
0	0x4001_A000	32 bits	R	0x9FFF4003	SCS
1	0x4001_A004	32 bits	R	0x9FFE7003	DWT
2	0x4001_A008	32 bits	R	0x9FFE8003	FPB
3	0x4001_A00C	32 bits	R	0xFFFFF003	MTB
4	0x4001_A010	32 bits	R	0x00000000	End of entries

2.5.4.2 CoreSight component registers

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

Table 2.10 shows the registers. See reference 5. in section 2.8. References for details of each register.

Table 2.10 CoreSight component registers in the CoreSight ROM Table

Name	Address	Access size	R/W	Initial value
MEMTYPE	0x4001_AFCC	32 bits	R	0x00000001
PIDR4	0x4001_AFD0	32 bits	R	0x00000004
PIDR5	0x4001_AFD4	32 bits	R	0x00000000
PIDR6	0x4001_AFD8	32 bits	R	0x00000000
PIDR7	0x4001_AFDC	32 bits	R	0x00000000
PIDR0	0x4001_AFE0	32 bits	R	0x0000006B
PIDR1	0x4001_AFE4	32 bits	R	0x00000030
PIDR2	0x4001_AFE8	32 bits	R	0x0000000A
PIDR3	0x4001_AFEC	32 bits	R	0x00000000
CIDR0	0x4001_AFF0	32 bits	R	0x0000000D
CIDR1	0x4001_AFF4	32 bits	R	0x00000010
CIDR2	0x4001_AFF8	32 bits	R	0x00000005
CIDR3	0x4001_AFFC	32 bits	R	0x000000B1

2.5.5 DBGREG Module

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

Table 2.11 shows the DBGREG registers other than the CoreSight component registers.

Table 2.11 Non-CoreSight DBGREG registers

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

2.5.5.1 DBGSTR : Debug Status Register

Base address: DBG = 0x4001_B000

Offset address: 0x0000

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

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

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

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

2.5.5.2 DBGSTOPCR : Debug Stop Control Register

Base address: DBG = 0x4001_B000

Offset address: 0x0010

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

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

Bit	Symbol	Function	R/W
1	—	This bit is read as 1. The write value should be 1.	R/W
13:2	—	These bits are read as 0. The write value should be 0.	R/W
14	DBGSTOP_TIM	Mask Bit for RTC, TAU, TML32 Reset/Interrupt in the OCD break mode. In the OCD break mode, the reset/interrupt is masked and each operation is stopped. 0: Enable RTC, TAU, TML32 reset/interrupt 1: Mask RTC, TAU, TML32 reset/interrupt	R/W
15	DBGSTOP_SIR	Mask Bit for SAU, IICA, UARTA, PORT_IRQ0-7 Reset/Interrupt in the OCD break mode. In the OCD break mode, the reset/interrupt is masked, and each operation is stopped. 0: Enable SAU, IICA, UARTA, PORT_IRQ0 to 7 reset/interrupt 1: Mask SAU, IICA, UARTA, PORT_IRQ0 to 7 reset/interrupt	R/W
16	DBGSTOP_LVD0	Mask Bit for LVD0 Reset 0: Enable LVD0 reset 1: Mask LVD0 reset	R/W
17	DBGSTOP_LVD1	Mask Bit for LVD1 Reset/Interrupt 0: Enable LVD1 reset/interrupt 1: Mask LVD1 reset/interrupt	R/W
23:18	—	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
31:25	—	These bits are read as 0. The write value should be 0.	R/W

The Debug Stop Control Register (DBGSTOPCR) controls the functional stop in OCD mode. All bits in the register are regarded as 0 when the MCU is not in OCD mode. In OCD break mode, the states of the CPU and peripheral functions may deviate. Also, when released OCD break mode and run the program again, the state of the CPU and peripheral functions may be different and the program may be executed.

2.5.5.3 DBGREG CoreSight component registers

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

Table 2.12 shows the registers. See reference 4. in section 2.8. References for details of each register.

Table 2.12 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	0x0000001A
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

2.5.6 OCDREG Module

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

Table 2.13 lists the OCDREG registers.

Table 2.13 OCDREG registers

Name		DAP port	Address	Access size	R/W
ID Authentication Code Register 0	IAUTH0	Port 1	0x8000_0000	32 bits	W
ID Authentication Code Register 1	IAUTH1	Port 1	0x8000_0100	32 bits	W
ID Authentication Code Register 2	IAUTH2	Port 1	0x8000_0200	32 bits	W
ID Authentication Code Register 3	IAUTH3	Port 1	0x8000_0300	32 bits	W
MCU Status Register	MCUSTAT	Port 1	0x8000_0400	32 bits	R
MCU Control Register	MCUCTRL	Port 1	0x8000_0410	32 bits	R/W

Note: OCDREG is located in the dedicated OCD address space. This address map is independent from the system address map. See section 2.5.2. Cortex-M23 Peripheral Address Map.

2.5.6.1 IAUTHn : ID Authentication Code Register (n = 0 to 3)

Four authentication registers are provided for writing the 128-bit key. These registers must be written in sequential order from IAUTHn (n = 0 to 3).

The initial value of the registers is all 0xFFFFFFFF. This means that SWD access is initially permitted when the ID code in the OSIS register has the initial value. See section 2.7.1. ID Code.

Base address: CPU_OCD = 0x8000_0000

Offset address: 0x0000 + 0x100 × n

Bit position:	31																												0
Bit field:	IAUTHn: AID (32 + 32 × n - 1) to (32 × n) bits																												
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

2.5.6.2 MCUSTAT : MCU Status Register

Base address: CPU_OCD = 0x8000_0000

Offset address: 0x0400

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:	—	—	—	—	—	—	—	—	—	—	—	—	—	CPUSTOPCLK	CPUSLEEP	AUTH
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	x*1	x*1	0

Bit	Symbol	Function	R/W
0	AUTH	Authentication Status 0: Authentication failed 1: Authentication succeeded	R
1	CPUSLEEP	Sleep Mode Status 0: CPU is not in Sleep mode 1: CPU in Sleep mode	R
2	CPUSTOPCLK	CPU Clock Status 0: CPU clock is not stopped. The MCU is in neither Software Standby nor Snooze mode. 1: CPU clock is stopped. The MCU is in Software Standby or Snooze mode.	R
31:3	—	These bits are read as 0.	R

Note 1. Depends on the MCU status.

2.5.6.3 MCUCTRL : MCU Control Register

Base address: CPU_OCD = 0x8000_0000

Offset address: 0x0410

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	—	—	—	—	—	—	—	EDBGRQ
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	EDBGRQ	External Debug Request Writing 1 to the bit causes a CPU halt. When the EDBGRQ bit is set to 0 or the CPU is halted, the EDBGRQ bit is cleared. 0: Debug event not requested 1: Debug event requested	R/W
7:1	—	These bits are read as 0. The write value should be 0.	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

Note: Set DBIRQ and EDBGRQ to the same value.

2.5.6.4 OCDREG CoreSight component registers

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

Table 2.14 shows the registers. See reference 4. in section 2.8. References for details of each register.

Table 2.14 OCDREG CoreSight component registers

Name	Address	Access size	R/W	Initial value
PIDR4	0x8000_0FD0	32 bits	R	0x00000004
PIDR5	0x8000_0FD4	32 bits	R	0x00000000
PIDR6	0x8000_0FD8	32 bits	R	0x00000000
PIDR7	0x8000_0FDC	32 bits	R	0x00000000
PIDR0	0x8000_0FE0	32 bits	R	0x00000004
PIDR1	0x8000_0FE4	32 bits	R	0x00000030
PIDR2	0x8000_0FE8	32 bits	R	0x0000000A
PIDR3	0x8000_0FEC	32 bits	R	0x00000000
CIDR0	0x8000_0FF0	32 bits	R	0x0000000D
CIDR1	0x8000_0FF4	32 bits	R	0x000000F0
CIDR2	0x8000_0FF8	32 bits	R	0x00000005
CIDR3	0x8000_0FFC	32 bits	R	0x000000B1

2.6 SysTick Timer

The SysTick timer provides a simple 24-bit down counter. The reference clock for the timer can be selected as the CPU clock (ICLK) or SysTick timer clock (SYSTICCLK). See section 8, Clock Generation Circuit and reference 1. in section 2.8. References for details.

2.7 OCD Emulator Connection

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

Figure 2.3 shows a block diagram of the authentication mechanism.

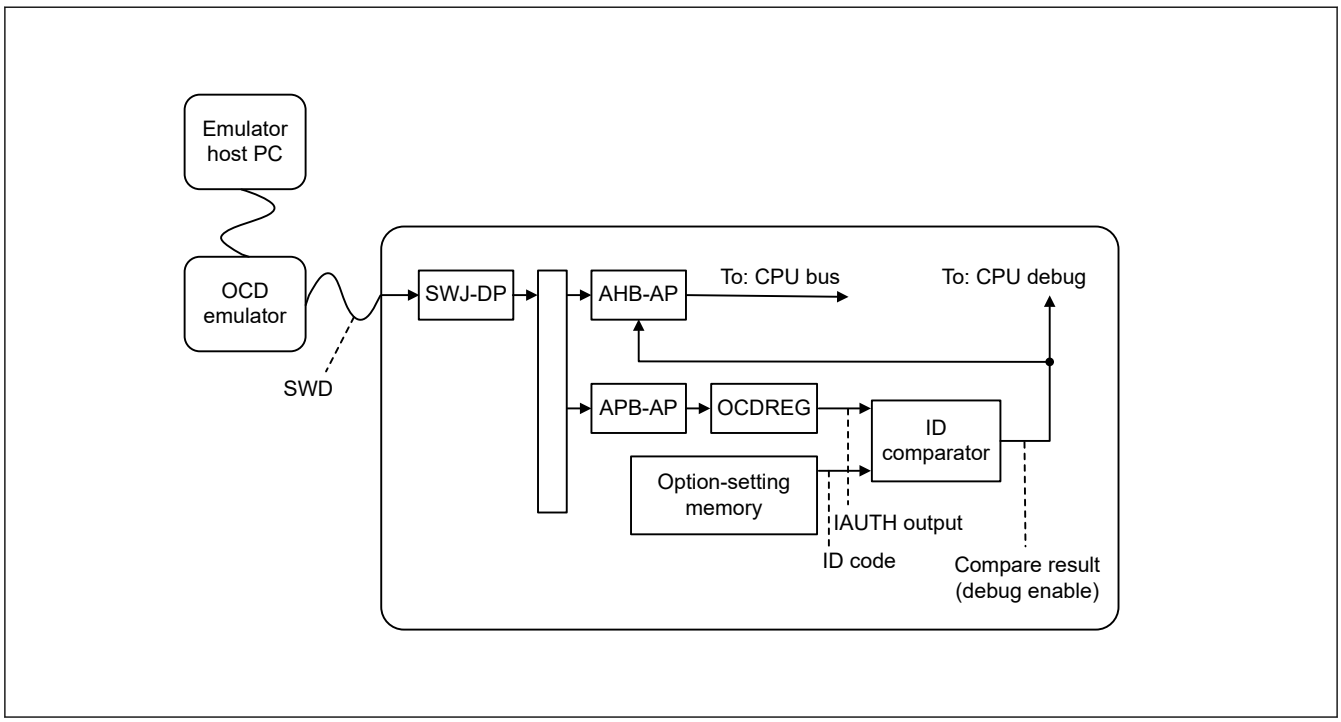


Figure 2.3 SWD authentication mechanism block diagram

An ID comparator is available in the MCU for authentication. The comparator compares the 128-bit IAUTH output from the OCDREG and the 128-bit ID code from the option-setting memory. When the two outputs are identical, the CPU debug functions and system bus access from the OCD emulator are permitted.

After the OCD emulator gets access permission, the OCD emulator must set the DBGEN bit in the System Control OCD Control Register (SYOCDCR). In addition, the OCD emulator must clear the DBGEN bit before disconnecting. See section 9.2.9. SYOCDCR : System Control OCD Control Register.

2.7.1 ID Code

The ID code is used for checking permission for debug and access to on-chip resources. If the ID code matches the 128-bit data written in the ID Authentication Code Registers 0 to 3, the SWD debugger obtains access permission. ID code is written in the OCD/Serial Programmer ID Setting Register (OSIS) in the option-setting memory. The initial value of the ID code is all 1s (0xFFFFFFFF_FFFFFFFF_FFFFFFFF_FFFFFFFF). See section 6, Option-Setting Memory for details.

2.7.2 DBGEN

After the OCD emulator gets access permission, the OCD emulator must set the DBGEN bit in the System Control OCD Control Register (SYOCDCR). In addition, the OCD emulator must clear the DBGEN bit before disconnecting it. See section 9, Low Power Modes for details.

2.7.3 Restrictions on Connecting an OCD emulator

This section describes the restrictions on emulator access.

2.7.3.1 Starting connection while in low power mode

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

2.7.3.2 Changing low power mode while in OCD mode

When the MCU is in OCD mode, the low power mode can be changed. However, system bus access from AHB-AP is prohibited in Software Standby or Snooze mode. Only SWJ-DP, APB-AP, and OCDREG can be accessed from the OCD emulator in these modes. Table 2.15 shows the restrictions.

Table 2.15 Restrictions by mode

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

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

2.7.3.3 Modify the ID code in OSIS

After modifying the ID code in the OSIS, the OCD emulator must reset the MCU by asserting the RES pin or setting the SYSRESETREQ bit of the Application Interrupt and Reset Control Register in the system control block to 1. The modified ID code is reflected after reset. For the system control block, see reference 2. listed in section 2.8. References.

The emulator must set the modified ID code in the IAUTH0 to IAUTH3 registers immediately before the MCU is placed in the reset state. When the IAUTH0 to IAUTH3 registers have been overwritten, writing to the SYSRESETREQ bit is not possible. Place the MCU in the reset state by asserting the signal on the RES pin.

2.7.3.4 Connecting sequence and SWD authentication

Because the OCD emulator is protected by the SWD authentication mechanism, the OCD might be required to input the ID code to the SWD authentication registers. The OSIS value in the option-setting memory determines whether the code is required. After negation of the RES pin, a wait time is required before comparing the OSIS value at cold start. See section 31.4.1. Reset Timing. The SWD authentication process is described in detail below.

(1) When MSB of the OSIS register is 0 (bit [127] = 0)

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

(2) When bits in the OSIS register is all 1s (initial value)

ID authentication is not required and the OCD can use the AHB-AP without authentication. For details of the settings for using the AHB-AP, see reference 4. in section 2.8. References.

1. Connect the OCD emulator to the MCU through the SWD interface.
2. Set up SWJ-DP to access the DAP bus. In the setup, the OCD emulator must assert CDBGPWRUPREQ in the SWJ-DP Control Status Register, then wait until CDBGPWRUPACK in the same register is asserted.
3. Set up the AHB-AP to access the system address space. The AHB-AP is connected to the DAP bus port 0.
4. Start accessing the CPU debug resources using the AHB-AP.

(3) When OSIS[127:126] = 10b

ID authentication is required and the OCD must write the ID code to the IAUTH registers 0 to 3 in OCDREG before using the AHB-AP.

1. Connect the OCD debugger to the MCU through the SWD interface.

2. Set up SWJ-DP to access the DAP bus. In the setup, the OCD emulator must assert CDBGPWRUPREQ in SWJ-DP Control Status Register, then wait until CDBGPWRUPACK in the same register is asserted.
3. Set up the APB-AP to access OCDREG. The APB-AP is connected to the DAP bus port 1.
4. Write the 128-bit ID code to IAUTH registers 0 to 3 in OCDREG using the APB-AP.
5. If the 128-bit ID code matches the OSIS value, the AHB-AP is authorized to issue an AHB transaction. The authorization result can be confirmed by the AUTH bit in the MCUSTAT Register or the DbgStatus bit in the AHB-AP Control Status Word Register.
   - When the DbgStatus bit is 1, the 128-bit ID code is a match with the OSIS value. AHB transfers are permitted.
   - When the DbgStatus bit is 0, the 128-bit ID code is not a match with the OSIS value. AHB transfers are not permitted.
6. Set up the AHB-AP to access the system address space. The AHB-AP is connected to the DAP bus port 0.
7. Start accessing the CPU debug resources using the AHB-AP.

(4) When OSIS[127:126] = 11b

OCD authentication is required and the OCD emulator must write the ID code to the IAUTH registers 0 to 3 in OCDREG. The connection sequence is the same when OSIS[127:126] is 10b except for “ALeRASE” capability.

When IATUH registers 0 to 3 are “ALeRASE” in ASCII code, the contents of the code flash, data flash, and configuration area are erased at once. See section 28, Flash Memory for details.

The ALeRASE sequence is as follows:

1. Connect the OCD debugger to the MCU through the SWD interface.
2. Set up SWJ-DP to access the DAP bus. In the setup, the OCD emulator must assert CDBGPWRUPREQ in the SWJ-DP Control Status Register, then wait until CDBGPWRUPACK in the same register is asserted.
3. Set up the APB-AP to access OCDREG. The APB-AP is connected to the DAP bus port 1.
4. Write the 128-bit ID code to IAUTH registers 0 to 3 in OCDREG using the APB-AP.
5. 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.

2.8 References

1. *ARM®v8-M Architecture Reference Manual* (ARM DDI 0553B.a)
2. *ARM® Cortex®-M23 Processor Technical Reference Manual* (ARM DDI 0550C)
3. *ARM® Cortex®-M23 Device Generic User Guide* (ARM DUI 1095A)
4. *ARM® CoreSight™ SoC-400 Technical Reference Manual* (ARM DDI 0480G)
5. *ARM® CoreSight™ Architecture Specification* (ARM IHI 0029E)
6. *ARM® CoreSight™ MTB-M23 Technical Reference Manual* (ARM DDI 0564C)

2.9 Usage Notes

In TBLOFF field of the Vector Table Offset Register (VTOR), TBLOFF[31:8] are valid and writing 1 to TBLOFF[7] bit is ignored. For the detail of Vector Table Offset Register, see reference 3. listed in section 2.8. References.

3. Operating Modes

3.1 Overview

The MCU starts in single-chip mode and the on-chip flash is enabled when a reset is released. 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.

3.2 Operating Modes Transitions

3.2.1 Operating Mode Transitions

Figure 3.1 shows operating mode transitions.

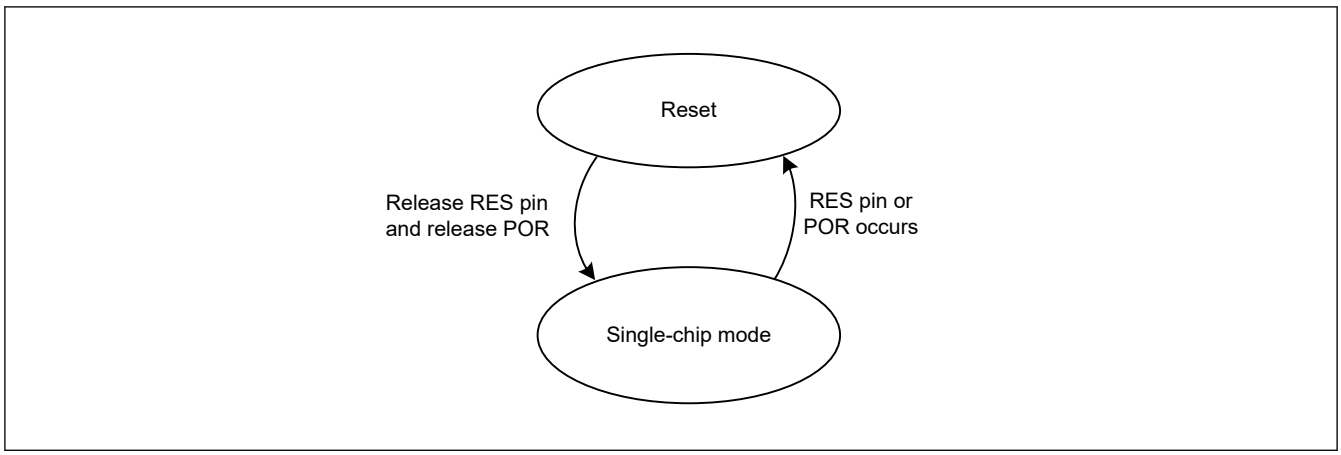
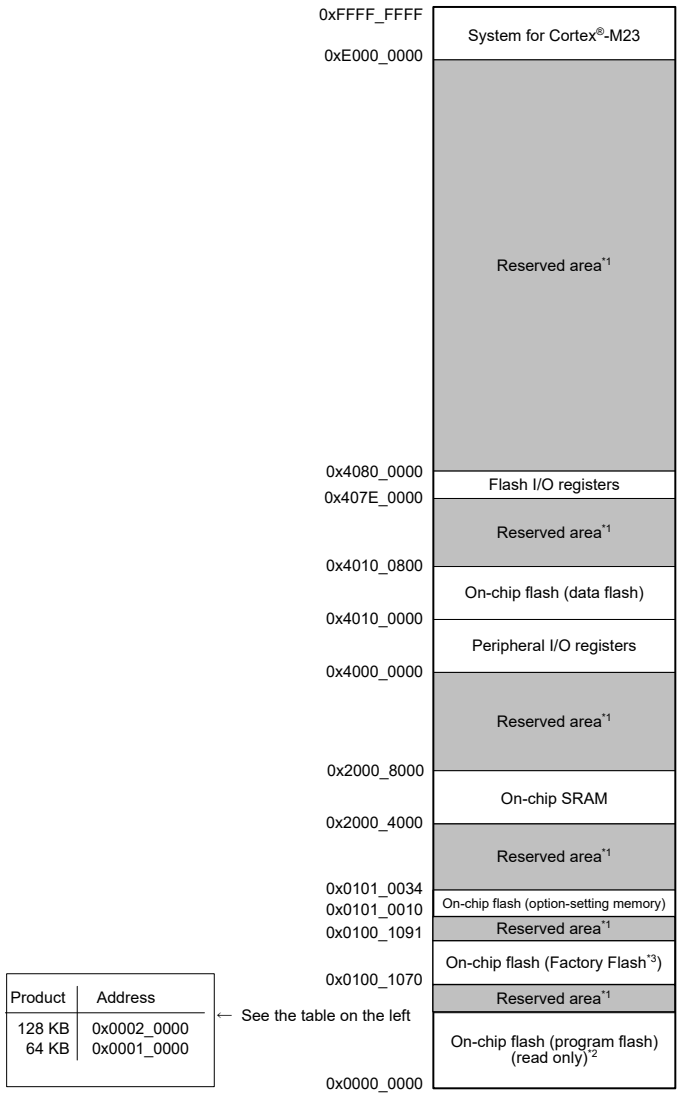


Figure 3.1 Operating mode

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 of a 128-KB/64-KB flash product.



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. Factory Flash includes UIDRn, PNRn, MCUVER registers. See section 28.3.24. UIDRn : Unique ID Registers n (n = 0 to 3), section 28.3.25. PNRn : Part Numbering Register n (n = 0 to 3), section 28.3.26. MCUVER : MCU Version Register in details.

Figure 4.1 Memory map

5. Resets

5.1 Overview

The MCU provides 7 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}) and VCC fall (voltage detection V_{PDR})
Independent watchdog timer reset	IWDT underflow or refresh error
Voltage monitor 0 reset	VCC fall (voltage detection V_{det0})*1
Voltage monitor 1 reset	VCC rise/fall (voltage detection V_{det1})*1
SRAM parity error reset	SRAM parity error detection
Software reset	Register setting (use the Arm® software reset bit AIRCR.SYSRESETREQ)

Note 1. For details on the voltages to be monitored (V_{POR}, V_{det0}, and V_{det1}), see section 7, Low Voltage Detection (LVD) and section 31, 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 2)

Flags to be initialized	Reset source			
	RES pin reset	Power-on reset	Voltage monitor 0 reset	Independent watchdog timer reset
Power-On Reset Detect Flag (PORSR.PORF)	—	✓	—	—
Voltage Monitor 0/1 Reset Detect Flag (RESF.LVIRF)	✓	✓	—	—
Independent Watchdog Timer Reset Detect Flag (RESF.IWDTRF)	✓	✓	—	—
Software Reset Detect Flag (RESF.SWRF)	✓	✓	—	—
SRAM Parity Error Reset Detect Flag (RESF.RPERF)	✓	✓	—	—

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

Flags to be initialized	Reset source		
	Voltage monitor 1 reset	Software reset	SRAM parity error reset
Power-On Reset Detect Flag (PORSR.PORF)	—	—	—
Voltage Monitor 0/1 Reset Detect Flag (RESF.LVIRF)	—	—	—
Independent Watchdog Timer Reset Detect Flag (RESF.IWDTRF)	—	—	—
Software Reset Detect Flag (RESF.SWRF)	—	—	—
SRAM Parity Error Reset Detect Flag (RESF.RPERF)	—	—	—

Note: ✓ : Initialized to 0
— : Not initialized

Table 5.3 Module-related registers initialized by each reset source (1 of 2)

Registers to be initialized		Reset source			
		RES pin reset	Power-on reset	Voltage monitor 0 reset	Independent watchdog timer reset
Registers related to the IWDT	IWDTRR, IWDTSR	✓	✓	✓	✓
Registers related to the voltage monitor function 1	LVD1CR, LVD1MKR, LVD1SR	✓	✓	✓	✓
Register related to the SOSC	SOSCCR	—	✓	—	—
	SOMRG	✓	✓	✓	✓
	CMC.SOSEL, CMC.XTSEL, CMC.SODRV[1:0]	—	✓	—	—
Register related to the RTC	RTCC0, RTCC1, SUBCUD	—	✓	—	—
	Other than above	—	—	—	—
Pin state (except XCIN/XCOUT pin)		✓	✓	✓	✓
Pin state (XCIN/XCOUT pin)		—	✓	—	—
Registers other than those shown, CPU, and internal state		✓	✓	✓	✓

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

Registers to be initialized		Reset source		
		Voltage monitor 1 reset	Software reset	SRAM parity error reset
Registers related to the IWDT	IWDTRR, IWDTSR	✓	✓	✓
Registers related to the voltage monitor function 1	LVD1CR, LVD1MKR, LVD1SR	✓	✓	✓
Register related to the SOSC	SOSCCR	—	—	—
	SOMRG	✓	✓	✓
	CMC.SOSEL, CMC.XTSEL, CMC.SODRV[1:0]	—	—	—
Register related to the RTC	RTCC0, RTCC1, SUBCUD	—	—	—
	Other than above	—	—	—
Pin state (except XCIN/XCOUT pin)		✓	✓	✓
Pin state (XCIN/XCOUT pin)		—	—	—
Registers other than those shown, CPU, and internal state		✓	✓	✓

Note: ✓ : Initialized
— : Not initialized

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 low power mode 1	Continue with the state that was selected before the reset occurred
	XCIN/XCOUT	Initialized to general-purpose input pins	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	Other
LOCO	Enable or disable	Initialized to disable. However, during IWDT operation, LOCO oscillates regardless of the value of LCSTP.	

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 RESF : Reset Status Flag Register

Base address: SYSC = 0x4001_E000

Offset address: 0x0830

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	SWRF	IWDTRF	—	RPERF	—	LVIRF
Value after reset:	0	0	x*1	x*1	0	x*1	0	x*1

Bit	Symbol	Function	R/W
0	LVIRF	Internal Reset Request by Voltage Detector (LVD0 or LVD1) 0: Internal reset request is not generated, or the RESF register is cleared. 1: Internal reset request is generated.	R
1	—	This bit is read as 0.	R
2	RPERF	Internal Reset Request by RAM Parity Error 0: Internal reset request is not generated, or the RESF register is cleared. 1: Internal reset request is generated.	R
3	—	This bit is read as 0.	R
4	IWDTRF	Internal Reset Request by Independent Watchdog Timer (IWDT) 0: Internal reset request is not generated, or the RESF register is cleared. 1: Internal reset request is generated.	R
5	SWRF	Internal Reset Request by Software Reset 0: Internal reset request is not generated, or the RESF register is cleared. 1: Internal reset request is generated.	R
7:6	—	These bits are read as 0.	R

Note 1. The value after reset varies depending on the reset source.

The RESF register indicates which reset source generated the reset request.

This register is cleared after it is read, in addition to when a reset occurs as described in Table 5.2.

5.2.2 PORSR : Power-On Reset Status Register

Base address: SYSC = 0x4001_E000

Offset address: 0x0831

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

Bit	Symbol	Function	R/W
0	PORF	Checking Occurrence of Power-on Reset 0: A value 1 has not been written, or a power-on reset has occurred. 1: No power-on reset has occurred.*1	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. If the PORF bit is set to 1, it guarantees that no power-on reset has occurred, but it does not guarantee that the RAM value is retained.

The PORSR register is used to check the occurrence of a Power-on reset.

This register is reset only by a Power-on reset; it retains the value when a reset caused by another source occurs.

PORF bit (Checking Occurrence of Power-on Reset)

The PORF bit indicates whether a power-on reset has occurred.

This bit is valid only for writing 1; writing 0 is invalid.

When checking whether or not a power-on reset has occurred, write 1 to this bit in advance.

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}, t_{RESWT2}) elapses. The CPU then starts the reset exception handling.

For details, see section 31, 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 PORSR is clear to 0. 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.

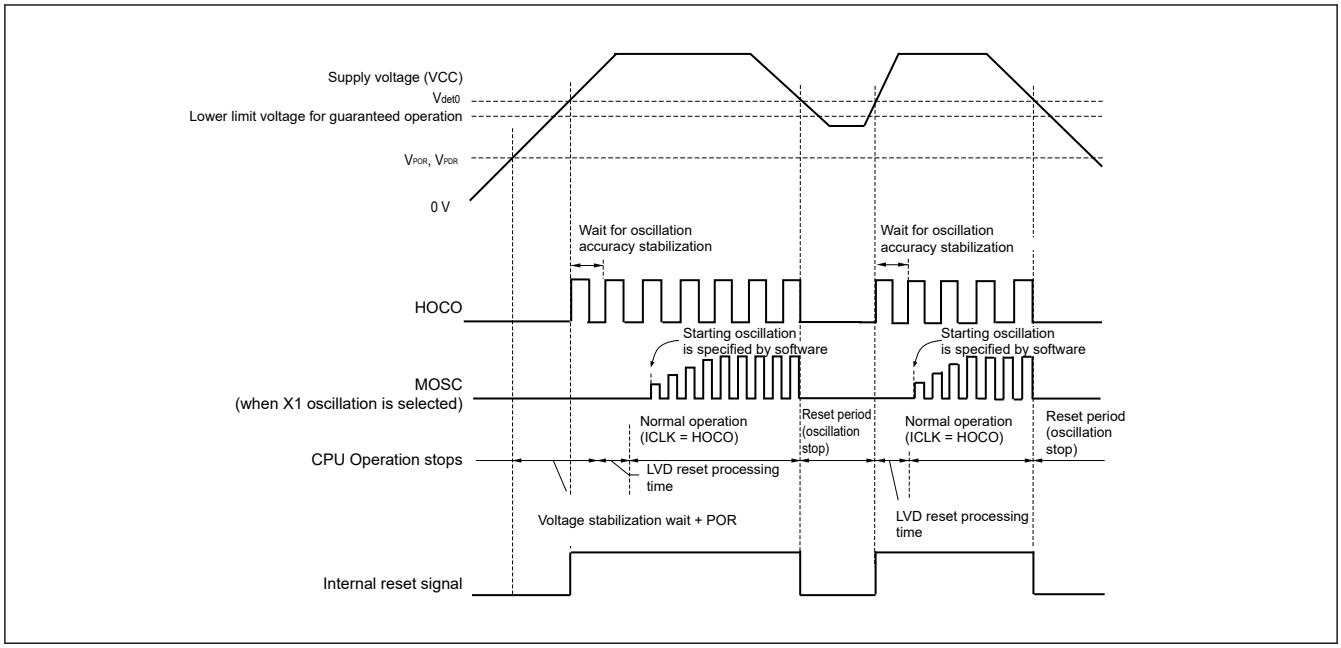


Figure 5.1 Example of operations during a power-on reset

5.3.3 Voltage Monitor Reset

The voltage monitor i (i = 0, 1) 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 RESF.LVIRF 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.

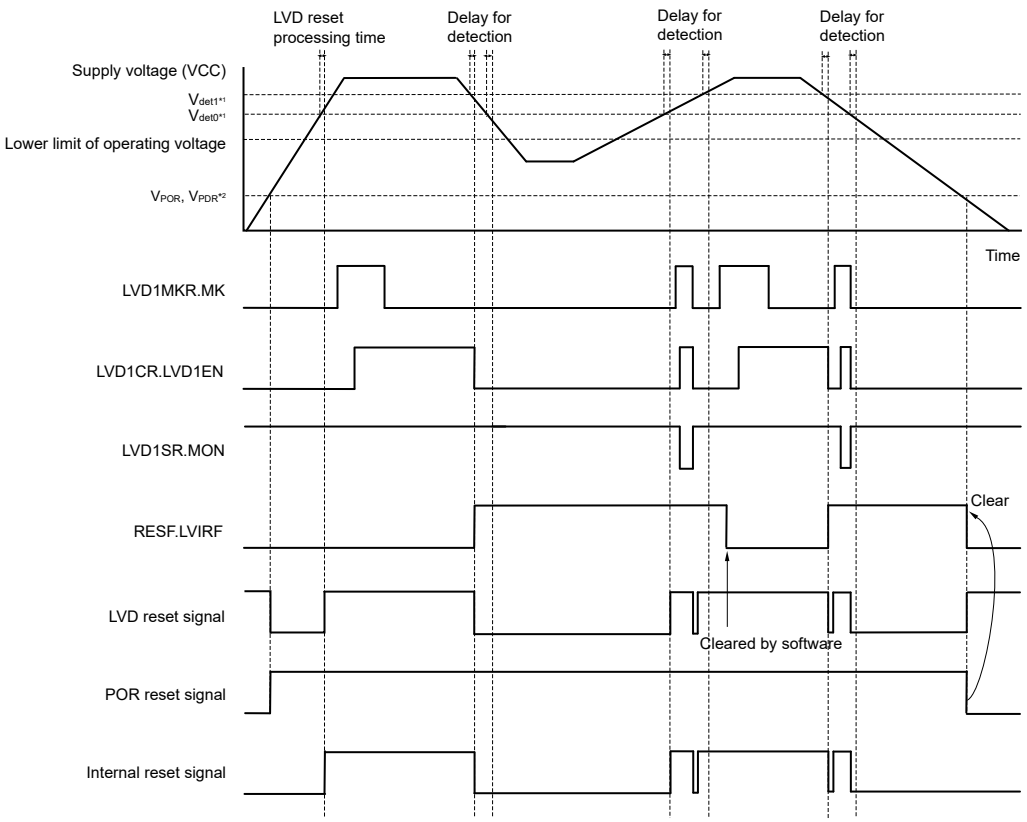
When the Enabling Operation of LVD1 bit (LVD1EN) is set to 1 (enabling generation of a reset or interrupt by the voltage detection circuit) and the LVD1 Operation Mode Select bit (LVD1SEL) is set to 1 (selecting generation of a reset in response to detection of a low voltage) in Voltage Monitor 1 Circuit Control register (LVD1CR), the RESF.LVIRF flag is set to 1 and the voltage detection circuit generates a voltage monitor 1 reset if VCC falls to or below V_{det1}*1.

If operation of LVD1 is enabled while VCC is lower than the voltage detection level (V_{det1}), it generates an internal reset at the time its operation is enabled.

Detection levels V_{det1} can be changed in the LVD1CR.LVD1V[4:0] bit.

Note 1. The LVD1MKR.MK bit should be 0. (If MK =1, reset and interrupt generation by LVD1 are masked.)

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



Note: For details on the electrical characteristics, see section 31, Electrical Characteristics.
Note 1. V_{det0} indicates the detection level of voltage monitor 0 reset and V_{det1} indicates the detection level of voltage monitor 0 reset.
Note 2. V_{POR} indicates the detection level of power-on reset at power supply rise, V_{PDR} indicates the detection level of power-on reset at power supply fall, and Vdet0 indicates the detection level of voltage monitor 0 reset.

Figure 5.2 Example of operation during voltage monitor 1 reset

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_{RESW2}) elapses after the independent watchdog timer reset is generated, the internal reset is canceled and the CPU starts the reset exception handling.

For details on the independent watchdog timer reset, see section 20, Independent Watchdog Timer (IWDT).

5.3.5 Software Reset

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

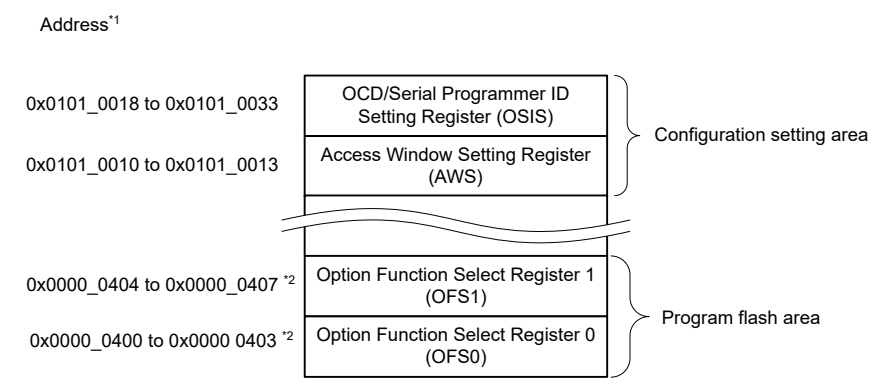
For details on the SYSRESETREQ bit, see the *ARM® Cortex®-M23 Technical Reference Manual*.

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 will be changed when the boot swap is set. See section 6.2.1. OFS0 : Option Function Select Register 0 and section 6.2.2. OFS1 : Option Function Select Register 1 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 and 0x0000_2400*1

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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:	—	IWDTSTPCTL	—	IWDTRSTIRQS	IWDTRPSS[1:0]		IWDTRPES[1:0]		IWDTCKS[3:0]				IWDTTOPS[1:0]		IWDTSTRT	—
Value after reset:	The value set by the user*2															

Bit	Symbol	Function	R/W
0	—	When read, this bit returns the written value.	R
1	IWDTSTRT	IWDT Start Mode Select 0: Automatically activate IWDT after a reset (auto start mode) 1: Disable IWDT after a reset	R
3:2	IWDTTOPS[1:0]	IWDT Timeout Period Select 0 0: 128 cycles (0x007F) 0 1: 512 cycles (0x01FF) 1 0: 1024 cycles (0x03FF) 1 1: 2048 cycles (0x07FF)	R

Bit	Symbol	Function	R/W
7:4	IWDTCKS[3:0]	IWDT 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: Non-maskable interrupt or Maskable 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
31:15	—	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 can be set with a flash writer or self-programming.

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 20, Independent Watchdog Timer (IWDT).

IWDTCKS[3:0] bits (IWDT 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 20, 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 20, 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 20, 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 20, Independent Watchdog Timer (IWDT).

IWDTSTPCTL bit (IWDT Stop Control)

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

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 / snooze/ software standby mode	Continue counting
1	Sleep / snooze / software standby mode	Stop counting

For details, see section 20, Independent Watchdog Timer (IWDT).

6.2.2 OFS1 : Option Function Select Register 1

Address: 0x0000_0404 and 0x0000_2404*1

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	FRPDIS	FRPE[5:0]						FRPS[5:0]					
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:	PORTSELB	HOCOFRQ1[2:0]			—	—	—	—	—	—	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 1 0: V_{det0_0} 0 1 1: V_{det0_1} 1 0 0: V_{det0_2} 1 0 1: V_{det0_3} 1 1 0: V_{det0_4} 1 1 1: V_{det0_5} Others: Setting prohibited	R
11:6	—	When read, these bits return the written value.	R

Bit	Symbol	Function	R/W
14:12	HOCOFRQ1[2:0]	HOCO Frequency Setting 1 0 0 0: 24 MHz 0 1 0: 32 MHz Others: Setting prohibited	R
15	PORTSELB*4	P206/RES Terminal Selection 0: Port (P206) 1: RES input (internal pull-up register is always active.)	R
21:16	FRPS[5:0]	Flash Read Protection Starting Address FRPS[5:0] specifies the bit[16:11] of the starting address of a protected region. The bit[31:17] and bit[10:0] of the starting address are filled with 0s. The value range of FRPS[5:0] is from 0x01 to 0x3F, excluding reserved areas.	R
27:22	FRPE[5:0]	Flash Read Protection Ending Address FRPE[5:0] specifies the bit[16:11] of the ending address of a protected region. The bit[31:17] of the ending address are filled with 0s and the bit[10:0] of the ending address are filled with 1s. The value range of FRPE[5:0] is from 0x01 to 0x3F, excluding reserved areas.	R
28	FRPDIS	Flash Read Protection Disable 0: Enable flash read protection 1: Disable flash read protection	R
31:29	—	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_2404 and 0x0000_0404 to the same value if boot swap is used.
Note 2. The value in a blank product is 0xFFFFFFFF. It can be set with a flash writer or self-programming.
Note 3. See section 31, Electrical Characteristics for the voltage levels to be detected. Set to 010b if LVD0 is not used.
Note 4. This setting is valid only for 32-pin product.

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.

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

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

PORTSELB bit (P206/RES Terminal Selection)

In the 32-pin product, the PORTSELB bit defines whether this port operates as port (P206) or reset. In 48-pin and 64-pin products, any value written to this bit is ignored.

FRPS[5:0] bits (Flash Read Protection Starting Address)

FRPE[5:0] bits (Flash Read Protection Ending Address)

The FRPS[5:0] and FRPE[5:0] bits specify the starting and ending address of a protected region of the Flash Read Protection (FRP). Figure 6.2 shows the starting and ending address of a protected region. Both the starting and ending address are included in a protected region (starting address ≤ protected region ≤ ending address). When the FRP is enabled, the memory space defined by the FRPS[5:0] and FRPE[5:0] bits is accessible only by instruction fetch. Since a protected region is not readable, it is prohibited that a protected region includes a vector table. Therefore, do not set FRPS[5:0] to 0x00 when using the FRP. Setting the FRPS[5:0] bits to 0x00 causes a protected region to include the initial vector table.

For details of the FRP, see section 13, Flash Read Protection (FRP).

FRPDIS bit (Flash Read Protection Disable)

The FRPDIS bit enables or disables the FRP. To enable the FRP, FRPDIS must be set to 0. To disable the FRP, FRPDIS must be set to 1.

For details of the FRP, see section 13, Flash Read Protection (FRP).

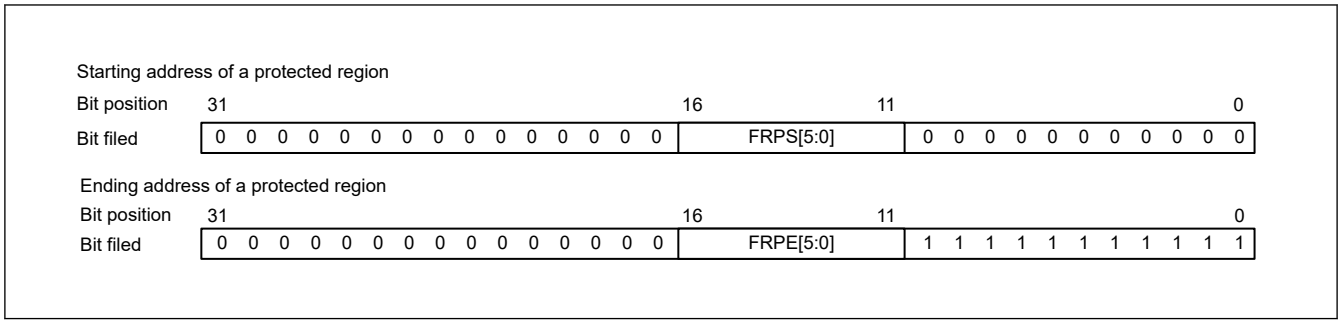


Figure 6.2 Starting and ending address of a protected

6.2.3 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:	BTFLG	—	—	—	—	FAWE[10:0]										
Value after reset:	The value set by the user															

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

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
14:11	—	When read, these bits return the written value.	R
15	FSPR	Protection of Access Window and Startup Area Select Function This bit controls the programming of the write/erase protection for the access window, the Startup Area Select Flag (BTFLG), and the temporary boot swap control with Startup Area Select (FISR.SAS[1:0] bits). When this bit is set to 0, it cannot be changed to 1. 0: Executing the configuration setting command for programming the access window (FAWE[10:0], FAWS[10:0]) and the Startup Area Select Flag (BTFLG) is invalid 1: Executing the configuration setting command for programming the access window (FAWE[10:0], FAWS[10:0]) and the Startup Area Select Flag (BTFLG) is valid	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
30:27	—	When read, these bits return the written value.	R
31	BTFLG	Startup Area Select Flag This bit specifies whether the address of the startup area is exchanged for the boot swap function. 0: First 8-KB area (0x0000_0000 to 0x0000_1FFF) and second 8-KB area (0x0000_2000 to 0x0000_3FFF) are exchanged 1: First 8-KB area (0x0000_0000 to 0x0000_1FFF) and second 8-KB area (0x0000_2000 to 0x0000_3FFF) are not exchanged	R

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 and on-chip debug mode. The access window can be locked by the FSPR 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 command is allowed to execute in the full program flash area.

FAWE[10:0] > FAWS[10:0]: The P/E command is 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 command is not allowed to execute in the program flash area.

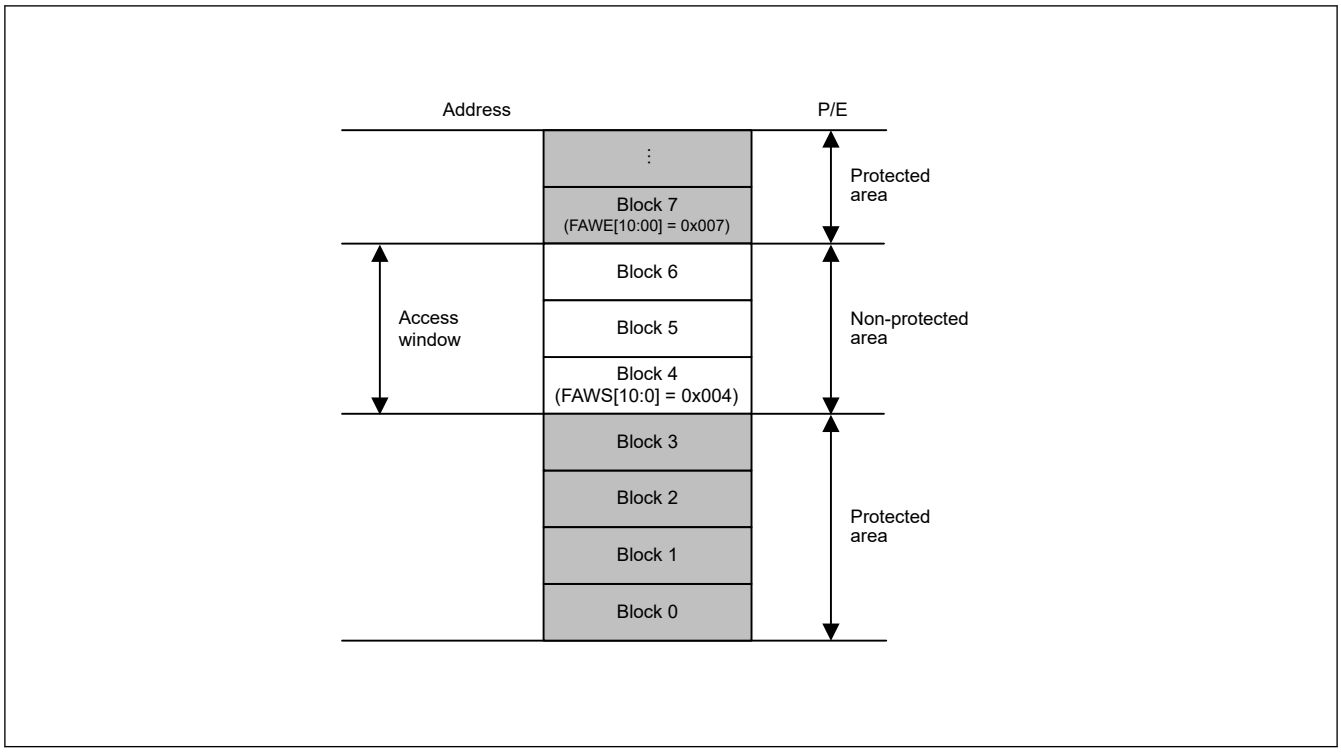


Figure 6.3 Access window overview

6.2.4 OSIS : OCD/Serial programmer ID Setting Register

The OSIS register stores the ID for ID code protection of the OCD emulator/serial programmer. When connecting the OCD emulator/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 emulator/serial programmer matches the ID code in the option-setting memory. When the ID code matches, connection with the OCD emulator/serial programmer is permitted, if not, connection with the OCD emulator/serial programmer is not possible. The OSIS register must be set in 32-bit words.

Address: 0x0101_0018, 0x0101_0020, 0x0101_0028, 0x0101_0030

Bit position:	31	0
Bit field:		
Value after reset:	The value set by the user	

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

ID code bits [127] and [126] on the 32-bit word at the address 0x0101_0018 determine whether the ID code protection is enabled, and the authentication method. Table 6.2 shows how the ID code determines the authentication method.

Table 6.2 Specifications for ID code protection

Operating mode on boot up	ID code	State of protection	Operations on connection to programmer or on-chip debugger
On-chip debug mode (SWD boot mode)	0xFF, …, 0xFF (all bytes are 0xFF)	Protection disabled	The ID code is not checked, the ID code always matches, and the connection to the on-chip debugger or serial programmer is permitted.
	Bit [127] = 1, bit [126] = 1, and at least one of the 16 bytes is not 0xFF	Protection enabled	Matching ID code indicates that authentication is complete and connection to the on-chip debugger or serial programmer is permitted. Mismatching ID code indicates transition to the ID code protection wait state. When the ID code sent from the on-chip debugger or serial programmer is ALeRASE in ASCII code (0x414C_6552_4153_45FF_FFFF_FFFF_FFFF_FFFF), the content of the user flash area is erased and all bits in the OSIS register are 1. However, when the AWS.FSPR bit is 0, the content of the user flash area is not erased.
	Bit [127] = 1 and bit [126] = 0	Protection enabled	Matching ID code indicates that authentication is complete and connection to the on-chip debugger or serial programmer 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 on-chip debugger or serial programmer is prohibited.

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. In addition, use the startup area select function to safely update the boot program that includes the option-setting memory.

For details of the programming command, the configuration setting command, and the startup area select function, see section 28, Flash Memory.

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

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

The MCU provides two setting procedures:

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

6.4 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 FSPR Bit

The AWS.FSPR bit cannot be changed to 1 once it is set to 0. At that time, access window and startup area selection cannot be set again.

7. Low Voltage Detection (LVD)

7.1 Overview

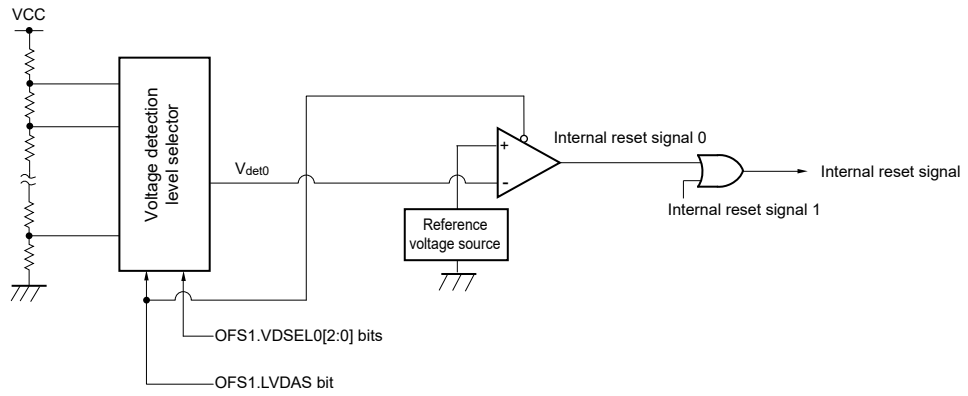
The Low Voltage Detection (LVD) module monitors the voltage level input to the VCC pin. The detection level can be selected by register settings. The LVD module consists of two separate voltage level detectors (LVD0, LVD1). LVD0 and LVD1 measure the voltage level input to the VCC pin. LVD registers allow 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 and Figure 7.2 shows a block diagram of the voltage monitor 1 interrupt and reset circuit.

Table 7.1 LVD specifications

Parameter		Voltage monitor 0	Voltage monitor 1
Means for setting up operation		OFS1 register	Registers
Target for monitoring		VCC pin input voltage	VCC pin input voltage
Monitored voltage		V_{det0}	V_{det1}
Detected event		Voltage falls past V_{det0}	Voltage rises or falls past V_{det1}
Detection voltage		Selectable from 6 different levels in the OFS1.VDSEL0[2:0] bits	Selectable from 18 different levels in the LVD1CR.LVD1V[4:0] bits
Monitoring flag		None	LVD1SR.MON flag: Monitors whether voltage is higher or lower than V_{det1}
			LVD1SR.DET flag: V_{det1} passage detection
Process on voltage detection	Reset	Voltage monitor 0 reset	Voltage monitor 1 reset
		Deasserts an internal reset signal on detecting VCC ≥ V_{det0}. Generates an internal reset on detecting VCC < V_{det0} and retains the reset state until VCC ≥ V_{det0} is detected.	Deasserts an internal reset signal on detecting VCC ≥ V_{det1}. Generates an internal reset on detecting VCC < V_{det1} and retains the reset. The reset is deasserted after specified time.
	Interrupt	No interrupt	Voltage monitor 1 interrupt
			Non-maskable or maskable interrupt selectable
			Interrupt request is issued when VCC crosses V_{det1}
Event link function		None	Available Output of event signals on detection of V_{det1} crossings



Note: For details of OFS1.LVDAS and OFS1.VDSEL0[2:0], see section 6, Option-Setting Memory.

Figure 7.1 Block diagram of voltage monitor 0 reset generation circuit

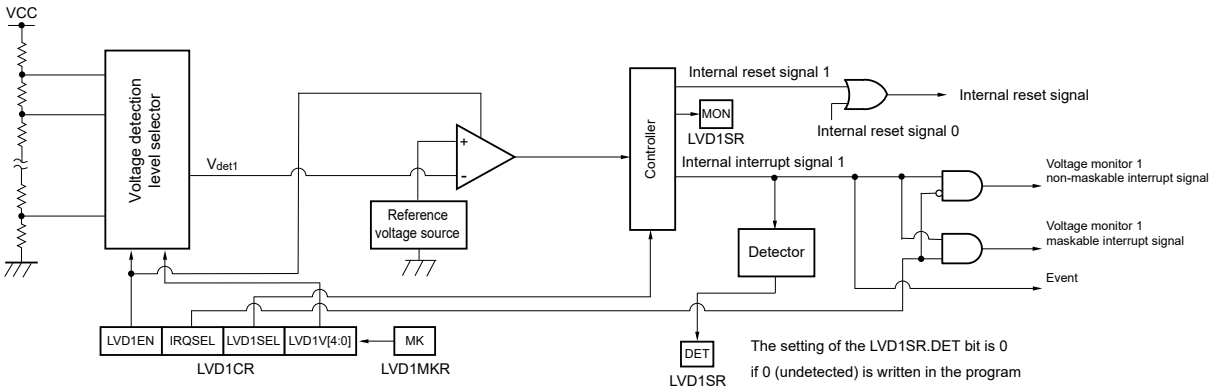


Figure 7.2 Block diagram of voltage monitor 1 interrupt and reset circuit

7.2 Register Descriptions

7.2.1 LVD1CR : Voltage Monitor 1 Circuit Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x0840

Bit position:	7	6	5	4	3	2	1	0
Bit field:	LVD1EN	LVD1SEL	IRQSEL	LVD1V[4:0]				
Value after reset:	0	0	0	1	1	0	0	1

Bit	Symbol	Function	R/W
4:0	LVD1V[4:0]	Voltage Detection 1 Level Select*1 *2 0x0E: V_{det1_0} 0x0F: V_{det1_1} 0x10: V_{det1_2} 0x11: V_{det1_3} 0x12: V_{det1_4} 0x13: V_{det1_5} 0x14: V_{det1_6} 0x15: V_{det1_7} 0x16: V_{det1_8} 0x17: V_{det1_9} 0x18: V_{det1_A} 0x19: V_{det1_B} 0x1A: V_{det1_C} 0x1B: V_{det1_D} 0x1C: V_{det1_E} 0x1D: V_{det1_F} 0x1E: V_{det1_10} 0x1F: V_{det1_11} Others: Setting prohibited	R/W
5	IRQSEL	Voltage Monitor 1 Interrupt Type Select 0: Non-maskable interrupt 1: Maskable interrupt	R/W
6	LVD1SEL	Operation mode of LVD1 0: Interrupt mode 1: Reset mode	R/W
7	LVD1EN	Enabling Operation of LVD1 When using voltage detection 1 interrupt/reset or the LVD1SR.MON flag, set the LVD1EN bit to 1. The voltage detection 1 circuit starts when LVD1 operation stabilization time ($t_{d(E-A)}$) elapses after the LVD1EN bit value is changed from 0 to 1. For details on $t_{d(E-A)}$, see section 31, Electrical Characteristics. 0: Operation stopped 1: Operation enabled	R/W

Note: • Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.
• Set the LVD1MKR.MK bit to 1 (write enabled) before rewriting this register.
Note 1. See section 31, Electrical Characteristics for the voltage levels to be detected.
Note 2. When using LVD0, set the detection voltage of LVD1 higher than the detection voltage of LVD0.

7.2.2 LVD1MKR : Voltage Monitor 1 Circuit Mask Register

Base address: SYSC = 0x4001_E000

Offset address: 0x0841

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

Bit	Symbol	Function	R/W
0	MK	Specification of Whether to Enable or Disable Rewriting the LVD1CR Register The MK bit enables or disables the rewriting of the Voltage Monitor 1 Circuit Control Register (LVD1CR). While the MK bit is 1, the reset and interrupt generation by LVD1 are masked. Therefore, clear the MK bit to 0 after having written a new value to the LVD1CR register. 0: Rewriting of the LVD1CR register is disabled. 1: Rewriting of the LVD1CR register is enabled (reset and interrupt generation by LVD1 are masked)	R/W
7:1	—	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.

7.2.3 LVD1SR : Voltage Monitor 1 Circuit Status Register

Base address: SYSC = 0x4001_E000

Offset address: 0x0843

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	MON	DET*1
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*2
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. The DET bit is invalid when reset mode is selected.
Note 2. Only 0 can be written to this bit.

DET flag (Voltage Monitor 1 Voltage Variation Detection Flag)

The DET flag is enabled when LVD1CR.LVD1EN = 1 (LVD1 operation enabled) and LVD1CR.LVD1SEL = 0 (LVD1 is set to interrupt mode).

MON flag (Voltage Monitor 1 Signal Monitor Flag)

The MON flag is enabled when LVD1CR.LVD1EN = 1 (LVD1 operation 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.

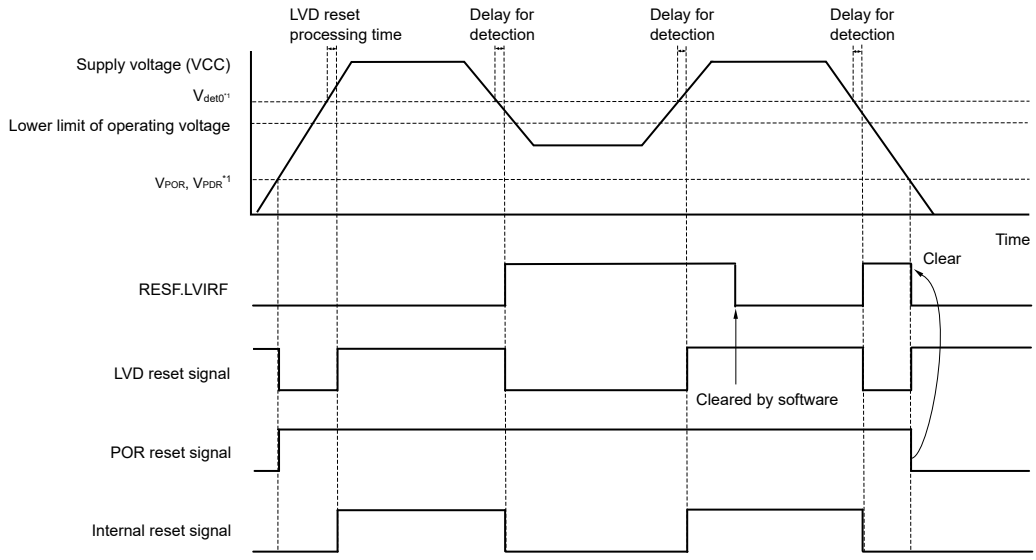
Table 7.2 Procedures to set up monitoring against V_{det1}

Step	Monitoring the comparison results from voltage monitor 1
1	Set the LVD1MKR.MK bit to 1 to enable writing to the LVD1CR register.
2	LVD1CR register setting • Set LVD1V[4:0] bits to set the LVD1 detection voltage • Set LVD1EN bit = 1 to enable LVD1 operation
3	LVD1 is enabled after $t_{d(E-A)}$ elapses as stability wait time.
4	Set the LVDMKR.MK bit to 0 to disable writing to the LVD1CR register.

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.

Figure 7.3 shows the timing of the internal reset signal generated in the LVD0 reset mode.



Note: For details of the electrical characteristics, see section 31, Electrical Characteristics.
Note 1. V_{POR} indicates the detection level of power-on reset at power supply rise, V_{PDR} indicates the detection level of power-on reset at power supply fall, and V_{det0} indicates the detection level of voltage monitor 0 reset.

Figure 7.3 Timing of LVD0 internal reset signal generation

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 7.3 shows the procedures for setting bits related to the voltage monitor 1 interrupt/reset so that voltage monitoring occurs. Table 7.4 shows the procedures for setting bits related to the voltage monitor 1 interrupt/reset so that voltage monitoring stops. Figure 7.4 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.

Table 7.3 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
1	Set the LVD1MKR.MK bit to 1 to enable writing to the LVD1CR register.	
2	LVD1CR register setting ● Set LVD1V[4:0] bits to set the LVD1 detection voltage ● Select the interrupt request condition in the IRQSEL bit ● Set LVD1SEL = 0 to set LVD1 operation mode to interrupt mode ● Set LVD1EN = 1 to enable LVD1 operation	LVD1CR register setting ● Set LVD1V[4:0] bits to set the LVD1 detection voltage ● Set LVD1SEL = 1 to set LVD1 operation mode to reset mode ● Set LVD1EN = 1 to enable LVD1 operation
3	LVD1 is enabled after $t_{d(E-A)}$ elapses as stability wait time.	
4	Set the LVD1MKR.MK bit to 0 to disable writing to the LVD1CR register.	

Table 7.4 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
1	Set the LVD1MKR.MK bit to 1 to enable writing to the LVD1CR register.
2	Set LVD1CR.LVD1EN = 0 to disable LVD1 operation.
3	Set the LVD1MKR.MK bit to 0 to disable writing to the LVD1CR register.

Figure 7.4 show the timing of the interrupt request signal generated in the LVD1 interrupt mode.

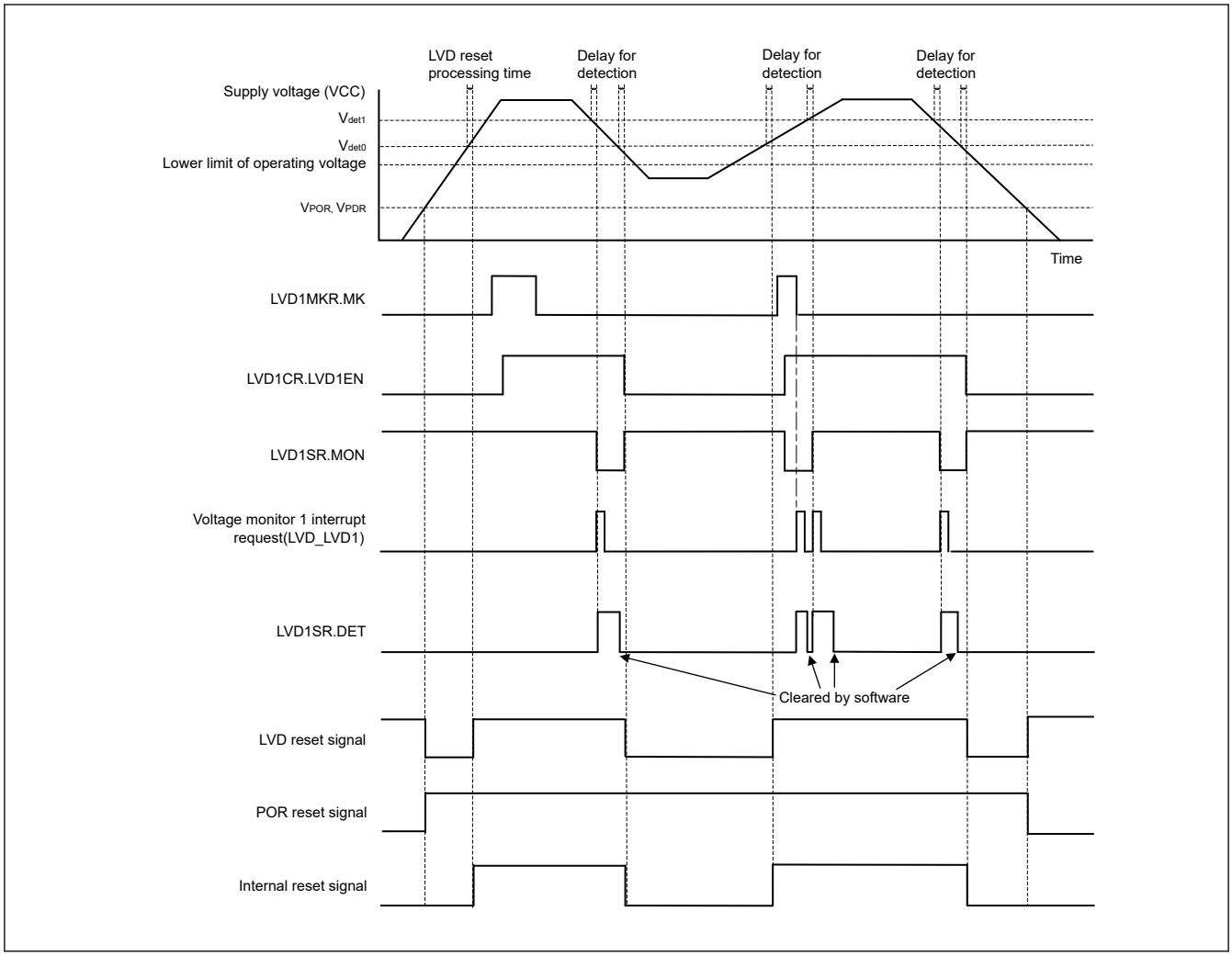


Figure 7.4 Timing of LVD1 interrupt request signal generation

Note: If operation of LVD1 is enabled while VCC is lower than the voltage detection level (V_{det1}), it generates an interrupt request signal (LVD_LVD1) at the time its operation is enabled.

7.6 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 the LVD1 interrupt mode is selected.

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.

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 shows a block diagram, and Table 8.3 lists the I/O pins.

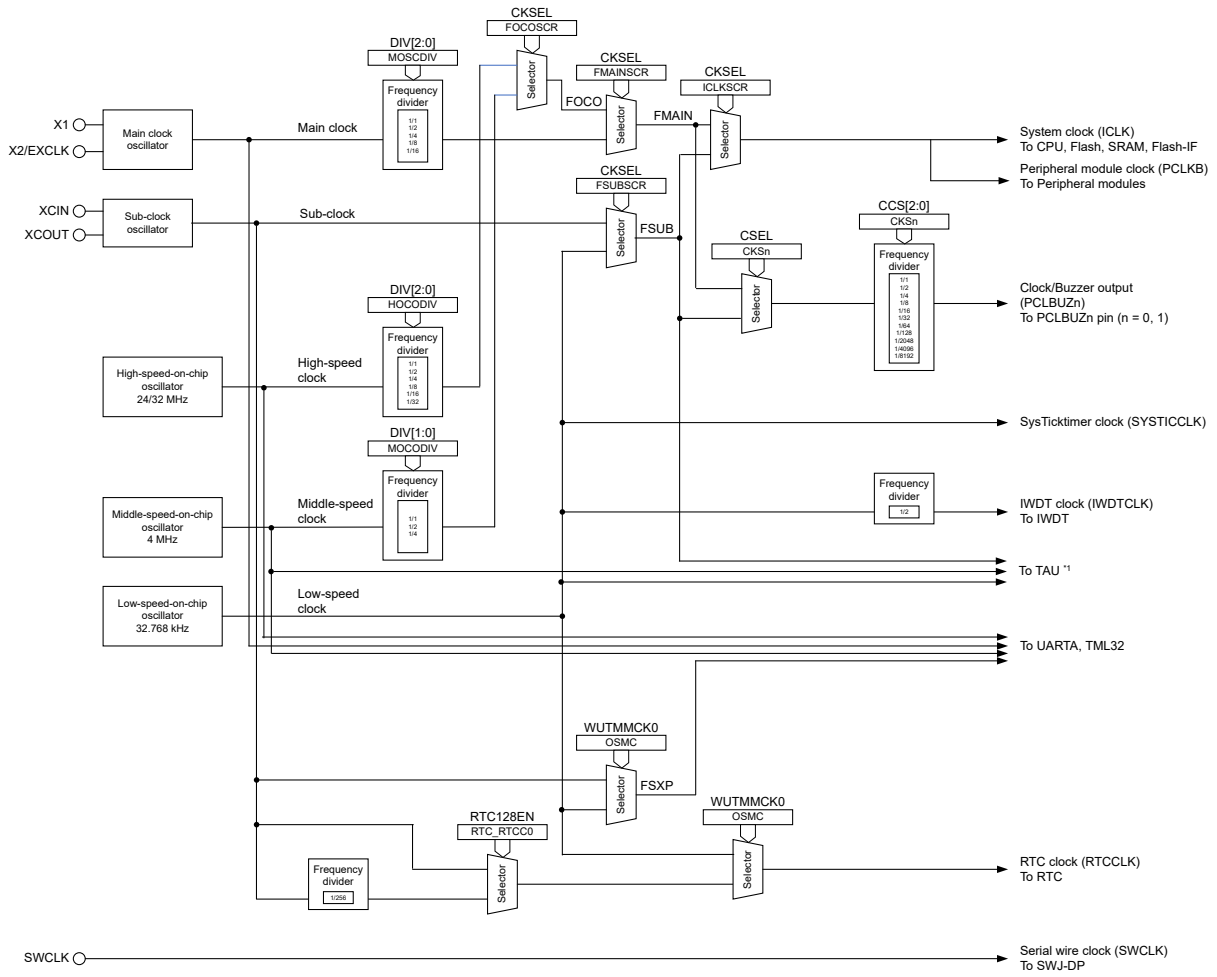
Table 8.1 Clock generation circuit specifications for the clock sources

Clock source	Description	Specification
Main clock oscillator (MOSC)	Resonator frequency	1 MHz to 20 MHz
	External clock input frequency	Up to 20 MHz
	External resonator or additional circuit	ceramic resonator, crystal
	Connection pins	X1, X2/EXCLK
	Drive capability switching	Available
Sub-clock oscillator (SOSC)	Resonator frequency	32.768 kHz
	External resonator or additional circuit	crystal resonator
	Connection pins	XCIN, XCOUT
	Drive capability switching	Available
High-speed on-chip oscillator (HOCO)	Oscillation frequency	24/32 MHz
	User trimming	Available
Middle-speed on-chip oscillator (MOCO)	Oscillation frequency	4 MHz
	User trimming	Available
Low-speed on-chip oscillator (LOCO)	Oscillation frequency	32.768 kHz
	User trimming	Available
External clock input for SWD (SWCLK)	Input clock frequency	Up to 12.5 MHz

Table 8.2 Clock generation circuit specifications for the internal clocks

Item	Clock source	Clock supply	Specification
System clock (ICLK)	MOSC*1/SOSC/HOCO/ MOCO/ LOCO	CPU, DTC, FLASH, Flash-IF, SRAM	Up to 32 MHz 1 MHz to 32 MHz (P/E)
Peripheral module clock (PCLKB)	MOSC*1/SOSC/ HOCO/ MOCO/LOCO	Peripheral modules	Up to 32 MHz
RTC clock (RTCCLK)	SOSC/LOCO	RTC	32.768 kHz, 128 Hz (SOSC/256)
IWDT clock (IWDTCLK)	LOCO	IWDT	16.384 kHz (LOCO/2)
SysTick timer clock (SYSTICCLK)	LOCO	SysTick timer	32.768 kHz
Clock/buzzer output (CLKOUT)	MOSC*1/SOSC/ HOCO/ MOCO/LOCO	PCLBUZn pin (n = 0, 1)	Up to 16 MHz Division ratios: 1/2/4/8/16/2048/4096/8192 (MOSC/MOCO/HOCO) 1/2/4/8/16/32/64/128 (SOSC/LOCO)
Serial wire clock (SWCLK)	SWCLK pin	OCD	Up to 12.5 MHz

Note 1. Using MOSC as a clock source in the Low-speed mode is prohibited.



Note 1. This is used as timer input of TAU channel 5.

Figure 8.1 Clock generation circuit block diagram (64-, 48-, 32-pin)

Table 8.3 Clock generation circuit input/output pins

Pin name	I/O	Description
X1	Output	These pins are used to connect a crystal resonator. The EXCLK pin can also be used to input an external clock. For details, see section 8.3.2. External Clock Input.
X2/EXCLK	Input	
XCIN	Input	These pins are used to connect a 32.768-kHz crystal resonator
XCOUT	Output	
PCLBUZn (n = 0, 1)	Output	This pin is used to output the CLKOUT/BUZZER clock
SWCLK	Input	This pin is used to input from the SWD

8.2 Register Descriptions

8.2.1 CMC : Clock Operation Mode Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x0800

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

Bit	Symbol	Function	R/W
0	MODRV	Main Clock Oscillator Drive Capability Switching 0: 1 MHz to 10 MHz 1: 10 MHz to 20 MHz	R/W
2:1	SODRV[1:0]	Sub-clock Oscillator Drive Capability Switching 0 0: Low Power Mode 1 0 1: Normal Mode 1 0: Low Power Mode 2 1 1: Low Power Mode 3	R/W
3	XTSEL	This bit is read as 0.	R
4	SOSEL	Sub-clock Oscillator Switching 0: Port mode 1: Resonator	R/W
5	—	This bit is read as 0. The write value should be 0.	R/W
7:6	MOSEL[1:0]	Main Clock Oscillator Switching 0 1: Resonator 1 1: External clock input mode Others: Port mode	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note: The CMC register can be written only once after release from the reset state.

MODRV bit (Main Clock Oscillator Drive Capability Switching)

The MODRV bit switches the drive capability of the main clock oscillator.

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

XTSEL bit

This bit is read as 0. Any value written to it is ignored.

SOSEL bit (Sub-clock Oscillator Switching)

The SOSEL bit switches the source for the sub-clock oscillator.

MOSEL[1:0] bits (Main Clock Oscillator Switching)

The MOSEL[1:0] bits switch the source for the main clock oscillator.

8.2.2 SOMRG : Sub-clock Oscillator Margin Check Register

Base address: SYSC = 0x4001_E000

Offset address: 0x0803

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

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

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.

8.2.3 FOCOSCR : FOCO Clock Source Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x0820

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

Bit	Symbol	Function	R/W
0	CKSEL	FOCO Clock Source Select 0: HOCO 1: MOCO	R/W
1	CKST	FOCO Clock Source Status 0: HOCO 1: MOCO	R
7:2	—	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 FOCOSCR register selects the clock source for the Main on-chip oscillator clock (FOCO).

CKSEL bit (FOCO Clock Source Select)

The CKSEL bit selects the source for the following clock:

- Main on-chip oscillator clock (FOCO)

The bit selects one of the following sources:

- Middle-speed on-chip oscillator (MOCO)
- High-speed on-chip oscillator (HOCO)

CKST bit (FOCO Clock Source Status)

The CKST flag indicates the source for the following clock:

- Main on-chip oscillator clock (FOCO)

8.2.4 FMAINSCR : FMAIN Clock Source Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x0821

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

Bit	Symbol	Function	R/W
0	CKSEL	FMAIN Clock Source Select 0: FOCO 1: MOSC*1	R/W
1	CKST	FMAIN Clock Source Status 0: FOCO 1: MOSC	R
7:2	—	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. Using MOSC as the FMAIN clock source (CKSEL = 1) in Low-speed mode is prohibited.

The FMAINSCR register selects the clock source for the Main System clock (FMAIN).

CKSEL bit (FMAIN Clock Source Select)

The CKSEL bit selects the source for the following clock:

- Main System clock (FMAIN)

The bit selects one of the following sources:

- Main on-chip oscillator clock (FOCO)
- Main clock oscillator (MOSC)

CKST bit (FMAIN Clock Source Status)

The CKST flag indicates the source for the following clock:

- Main System clock (FMAIN)

8.2.5 FSUBSCR : FSUB Clock Source Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x0822

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

Bit	Symbol	Function	R/W
0	CKSEL	FSUB Clock Source Select 0: SOSC 1: LOCO	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.

CKSEL bit (FSUB Clock Source Select)

The CKSEL bit selects the source for the following clock:

- Sub System clock (FSUB)

Writing to FSUBSCR.CKSEL is prohibited while MCU is under the following conditions:

1. ICLKSCR.CKSEL = 1
2. ICLKSCR.CKST = 1

The bit selects one of the following sources:

- Sub-clock oscillator (SOSC)
- Low-speed on-chip oscillator clock (LOCO)

8.2.6 ICLKSCR : ICLK Clock Source Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x0823

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

Bit	Symbol	Function	R/W
0	CKSEL	ICLK Clock Source Select 0: FMAIN 1: FSUB	R/W
1	CKST	ICLK Clock Source Status 0: FMAIN 1: FSUB	R
7:2	—	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 ICLKSCR register selects the clock source for the system clock.

CKSEL bit (ICLK Clock Source Select)

The CKSEL bit selects the source for the following clock:

- Peripheral module clock (PCLKB)
- System clock (ICLK)

The bit selects one of the following sources:

- Main System clock (FMAIN)
- Sub System clock (FSUB)

CKST bit (ICLK Clock Source Status)

The CKST flag indicates the source for the following clock:

- Peripheral module clock (PCLKB)
- System clock (ICLK)

8.2.7 MOSCCR : Main Clock Oscillator Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x080B

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

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

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note 1. CMC register must be set before setting MOSTP to 0.

The MOSCCR register controls the main clock oscillator.

MOSTP bit (Main Clock Oscillator Stop)

The MOSTP bit starts or stops the main clock oscillator.

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

When using the main clock, the Clock operation mode Control Register (CMC) and the Oscillation stabilization time select register (OSTS) must be set before setting MOSTP to 0. When the MOSCCR.MOSTP bit is modified for the main clock to run, only use the main clock after confirming that the OSTS register.

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

Writing 1 to MOSTP is prohibited under the following condition:

- FMAINSCR.CKST = 1b and ICLKSCR.CKST = 0b (system clock source = MOSC).

Operation of the main clock oscillator (MOSTP = 0) is prohibited in the Low-speed mode.

8.2.8 SOSCCR : Sub-clock Oscillator Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x080C

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 CMC 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 Clock Operation Mode Control Register (CMC) before setting SOSTP to 0.

After setting SOSTP to 0, only use the sub-clock oscillator after the sub-clock oscillation stabilization wait time elapses. A fixed stabilization wait time is required after selecting the sub-clock operation with the SOSTP bit. A fixed wait time is also required for oscillation to stop after setting the SOSTP bit.

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
- Confirm that the sub-clock oscillator is stable when stopping the sub-clock oscillator
- 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:

- FSUBSCR.CKSEL = 0 and ICLKSCR.CKST = 1 (system clock source = SOSC).

8.2.9 LOCOCR : Low-speed On-chip Oscillator Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x080A

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

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

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

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_{LOCO}) 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:

- FSUBSCR.CKSEL = 1 (Sub system clock source = LOCO).

During IWDT operation, LOCO oscillates regardless of the value of LCSTP.

8.2.10 HOCOCR : High-speed On-chip Oscillator Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x0808

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

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

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

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 1, 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 bit is 0 before restarting the HOCO clock.
- Confirm that the HOCO clock operates and that the OSCSF.HOCOSF bit is 1 before stopping the HOCO clock.
- Regardless of whether the HOCO clock is selected as the system clock, confirm that the OSCSF.HOCOSF bit 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 bit 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:

- FOCOSCR.CKST = 0, FMAINSCR.CKST = 0 and ICLKSCR.CKST = 0 (system clock source = HOCO).

During On chip Debug operation, HOCO oscillates regardless of the value of HCSTP.

8.2.11 MOCOCR : Middle-speed On-chip Oscillator Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x0809

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

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

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

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:

- FOCOSCR.CKST = 1, FMAINSCR.CKST = 0 and ICLKSCR.CKST = 0 (system clock source = MOCO).

8.2.12 OSTC : Oscillation Stabilization Time Counter Status Register

Base address: SYSC = 0x4001_E000

Offset address: 0x0810

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

Bit	Symbol	Function	R/W
7:0	MOST[7:0]	Selection of the Oscillation Stabilization Time 0x00: Less than $2^8/f_{MOSC}$ 0x80: $2^8/f_{MOSC}$ min 0xC0: $2^9/f_{MOSC}$ min 0xE0: $2^{10}/f_{MOSC}$ min 0xF0: $2^{11}/f_{MOSC}$ min 0xF8: $2^{13}/f_{MOSC}$ min 0xFC: $2^{15}/f_{MOSC}$ min 0xFE: $2^{17}/f_{MOSC}$ min 0xFF: $2^{18}/f_{MOSC}$ min	R

Note: After the above time has elapsed, the bits are set to 1 in order from the MOST[7] bit and remain 1.
Note: The value counted by the OSTC register will only have reached the oscillation stabilization time setting in the oscillation stabilization time select register (OSTS).
Note: Set the oscillation stabilization time of the OSTS register to the value greater than the counter value which is to be checked by using the OSTC register.
Note: Note that the value counted by the OSTC register will only have reached the oscillation stabilization time setting in the OSTS register after release from the Software Standby mode.
Note: The MOSC clock oscillation stabilization time does not include the time until clock oscillation starts (see a in Figure 8.2).

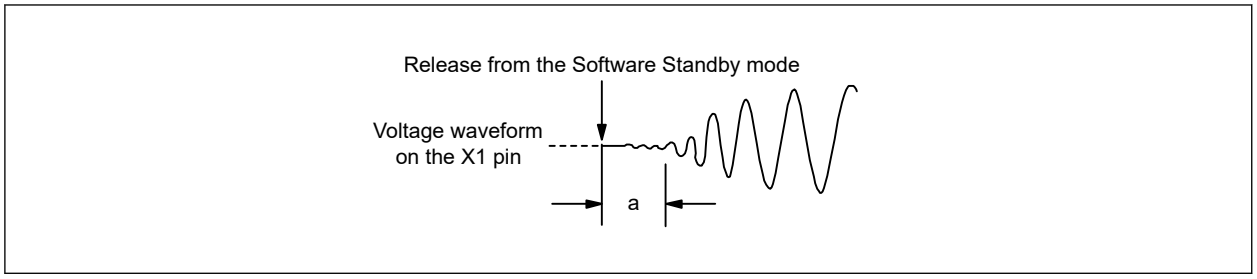


Figure 8.2 Initial oscillation image

Note: f_{MOSC}: MOSC clock oscillation frequency

This register indicates the counter value by the MOSC clock oscillation stabilization time counter.

The MOSC clock oscillation stabilization time can be checked in the following cases:

- If the MOSC clock starts oscillation while the main on-chip oscillator clock or subsystem clock is in use as the CPU clock.
- If entry to and then release from the Software Standby mode proceed while the main on-chip oscillator clock is in use as the CPU clock and the MOSC clock is oscillating.

The OSTC register can be read by an 8-bit memory manipulation instruction.

The value of this register is 0x00 following a reset, WFI instruction on SBYCR.SSBY = 1, or the MOSCCR.MOSTP bit to 1.

Note: The oscillation stabilization time counter starts counting in the following cases.

- When oscillation of the MOSC clock starts (MOSEL[1:0] = 01b → MOSTP = 0)
- When the Software Standby mode is released

MOST[7:0] bits (Selection of the Oscillation Stabilization Time)

The clock oscillation stabilization time when f_{MOSC} = 10 MHz and f_{MOSC} = 20 MHz is shown in Table 8.4.

Table 8.4 Example clock oscillation stabilization time (MOST)

MOST[7:0]	State of the oscillation stabilization time		
		f_{MOSC} = 10 MHz	f_{MOSC} = 20 MHz
0x00	Less than $2^8/f_{MOSC}$	Less than 25.6 µs	Less than 12.8 µs
0x80	$2^8/f_{MOSC}$ min.	25.6 µs min.	12.8 µs min.
0xC0	$2^9/f_{MOSC}$ min.	51.2 µs min.	25.6 µs min.
0xE0	$2^{10}/f_{MOSC}$ min.	102 µs min.	51.2 µs min.
0xF0	$2^{11}/f_{MOSC}$ min.	204 µs min.	102 µs min.
0xF8	$2^{13}/f_{MOSC}$ min.	819 µs min.	409 µs min.
0xFC	$2^{15}/f_{MOSC}$ min.	3.27 ms min.	1.63 ms min.
0xFE	$2^{17}/f_{MOSC}$ min.	13.1 ms min.	6.55 ms min.
0xFF	$2^{18}/f_{MOSC}$ min.	26.2 ms min.	13.1 ms min.

8.2.13 OSTS : Oscillation Stabilization Time Select Register

Base address: SYSC = 0x4001_E000

Offset address: 0x0811

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

Bit	Symbol	Function	R/W
2:0	OSTSB[2:0]	Selection of the Oscillation Stabilization Time 0 0 0: $2^8/f_{MOSC}$ 0 0 1: $2^9/f_{MOSC}$ 0 1 0: $2^{10}/f_{MOSC}$ 0 1 1: $2^{11}/f_{MOSC}$ 1 0 0: $2^{13}/f_{MOSC}$ 1 0 1: $2^{15}/f_{MOSC}$ 1 1 0: $2^{17}/f_{MOSC}$ 1 1 1: $2^{18}/f_{MOSC}$	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.

This register is used to select the MOSC clock oscillation stabilization time.

When the MOSC clock oscillator is set to start oscillation by clearing the MOSTP bit, actual operation is automatically delayed for the time set in the OSTS register.

Use the oscillation stabilization time counter status register (OSTC) to confirm that the specified oscillation stabilization time has elapsed when the CPU clock is switched from the main on-chip oscillator clock or the subsystem clock to the MOSC clock or entry to and then release from the Software Standby mode proceed while the main on-chip oscillator clock is in use as the CPU clock and the MOSC clock is oscillating. The OSTC register can be used to check the counter value when counting has reached the time set beforehand in the OSTS register.

The OSTS register can be set by an 8-bit memory manipulation instruction.

The value of this register is 0x07 following a reset.

The clock oscillation stabilization time when f_{MOSC} = 10 MHz and f_{MOSC} = 20 MHz is shown in Table 8.5.

Table 8.5 Example clock oscillation stabilization time (OSTS)

OSTS[2:0]	Selection of the oscillation stabilization time		
		f_{MOSC} = 10 MHz	f_{MOSC} = 20 MHz
000b	$2^8/f_{MOSC}$	25.6 µs	12.8 µs
001b	$2^9/f_{MOSC}$	51.2 µs	25.6 µs
010b	$2^{10}/f_{MOSC}$	102 µs	51.2 µs
011b	$2^{11}/f_{MOSC}$	204 µs	102 µs
100b	$2^{13}/f_{MOSC}$	819 µs	409 µs
101b	$2^{15}/f_{MOSC}$	3.27 ms	1.63 ms
110b	$2^{17}/f_{MOSC}$	13.1 ms	6.55 ms
111b	$2^{18}/f_{MOSC}$	26.2 ms	13.1 ms

8.2.14 OSCSF : Oscillation Stabilization Flag Register

Base address: SYSC = 0x4001_E000

Offset address: 0x0812

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

Bit	Symbol	Function	R/W
0	HOCOSF	HOCO Clock Oscillation Stabilization Flag 0: The HOCO clock is being started at high speed and waiting for the precision of its oscillation to become stable is in progress. 1: The HOCO clock is operating with high precision.	R
7:1	—	These bits are read as 0.	R

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).

[Setting condition]

- When the HOCO oscillation stabilization time count is completed after the HOCOCR.HCSTP bit becomes 0 while HOCO is stopped. For the HOCO oscillation stabilization time, see section 31, Electrical Characteristics.
- When the HOCO oscillation stabilization time has been counted after Software Standby mode is released.

[Clearing condition]

- When the HOCO clock is operating and then is deactivated because the HOCOCR.HCSTP bit is set to1.
- When the MCU enters Software Standby mode due to the WFI instruction.

8.2.15 HOCODIV : High-speed On-chip Oscillator Frequency Select Register

Base address: SYSC = 0x4001_E000

Offset address: 0x0818

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

Bit	Symbol	Function	R/W
2:0	DIV[2:0]	High-speed On-chip Oscillator Clock 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 1 0 1: × 1/32*1 Others: 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. Setting prohibited when OFS1.HOCOFRQ1[2:0] = 000b

8.2.16 MOCODIV : Middle-speed On-chip Oscillator Frequency Select Register

Base address: SYSC = 0x4001_E000

Offset address: 0x0819

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

Bit	Symbol	Function	R/W
1:0	DIV[1:0]	Selection of the Middle-speed On-chip Oscillator Clock Frequency 0 0: × 1/1 0 1: × 1/2 1 0: × 1/4 Others: Setting prohibited	R/W
7:2	—	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: Set the MOCODIV register while ensuring that the voltage is within the usable range for the flash operation mode set in the flash operating mode select register (FLMODE) both before and after the frequency change.

The MOCODIV register is used to select the frequency of the middle-speed on-chip oscillator.

The MOCODIV register can be set by an 8-bit memory manipulation instruction.

The value of this register is 0x00 following a reset.

8.2.17 MOSCDIV : MOSC Clock Division Register

Base address: SYSC = 0x4001_E000

Offset address: 0x081A

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

Bit	Symbol	Function	R/W
2:0	DIV[2:0]	Selection Division Ratio for the MOSC Clock 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 Others: 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: Set the MOSCDIV register while ensuring that the voltage is within the usable range for the flash operation mode set in the flash operating mode select register (FLMODE) both before and after the frequency change.

This register is used to select the division ratio of the MOSC clock.

The MOSCDIV register can be set by an 8-bit memory manipulation instruction. The value of this register is 0x00 following a reset.

DIV[2:0] bits (Selection Division Ratio for the MOSC Clock)

The clock division ratio when f_{MOSC} = 20 MHz is shown in Table 8.6.

Table 8.6 Example division ratio for the MOSC clock (MOSCDIV)

DIV[2:0]	Selected division ratio for the MOSC clock	f_{MOSC} = 20 MHz
000b	f_{MOSC} × 1/1	20 MHz
001b	f_{MOSC} × 1/2	10 MHz
010b	f_{MOSC} × 1/4	5 MHz
011b	f_{MOSC} × 1/8	2.5 MHz
100b	f_{MOSC} × 1/16	1.25 MHz
Other than above	Setting prohibited	

8.2.18 OSMC : Subsystem Clock Supply Mode Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x0824

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
3:0	—	These bits are read as 0. The write value should be 0.	R/W
4	WUTMMCK0	Selection of the Operating clock source for the Realtime Clock, 32-bit Interval Timer, Serial Interface UARTA 0: SOSC 1: LOCO*1 *2	R/W
7:5	—	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. After stopping SOSC, the clock source can be changed from SOSC to LOCO.
Note 2. Switching between SOSC and LOCO clock can be enabled by the WUTMMCK0 bit only when all of the realtime clock, 32-bit interval timer, and serial interface UARTA are stopped.

8.2.19 CKSn : Clock Out Control Register n (n = 0, 1)

Base address: PCLBUZ = 0x400A_3B00

Offset address: 0x0001 + 0x1 × n

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

Bit	Symbol	Function	R/W
2:0	CCS[2:0]	Clock Out Divide Select 0 0 0: value after reset FMAIN (When CKSn.CSEL = 0) FSUB (When CKSn.CSEL = 1) 0 0 1: FMAIN × 1/2 (When CKSn.CSEL = 0) FSUB × 1/2 (When CKSn.CSEL = 1) 0 1 0: FMAIN × $1/2^2$ (When CKSn.CSEL = 0) FSUB × $1/2^2$ (When CKSn.CSEL = 1) 0 1 1: FMAIN × $1/2^3$ (When CKSn.CSEL = 0) FSUB × $1/2^3$ (When CKSn.CSEL = 1) 1 0 0: FMAIN × $1/2^4$ (When CKSn.CSEL = 0) FSUB × $1/2^4$ (When CKSn.CSEL = 1) 1 0 1: FMAIN × $1/2^{11}$ (When CKSn.CSEL = 0) FSUB × $1/2^5$ (When CKSn.CSEL = 1) 1 1 0: FMAIN $1/2^{12}$ (When CKSn.CSEL = 0) FSUB × $1/2^6$ (When CKSn.CSEL = 1) 1 1 1: FMAIN × $1/2^{13}$ (When CKSn.CSEL = 0) FSUB × $1/2^7$ (When CKSn.CSEL = 1)	R/W
3	CSEL	Clock Out Select 0: FMAIN 1: FSUB	R/W
6:4	—	These bits are read as 0. The write value should be 0.	R/W
7	PCLOE	Clock Out Enable 0: Disable clock out 1: Enable clock out	R/W

CCS[2:0] bits (Clock Out Divide Select)

The CCS[2:0] bits specify the clock division ratio. Set the PCLOE 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 PCLBUZn pin output frequency. For details on the characteristics of the PCLBUZn pin, see section 31, Electrical Characteristics.

CSEL bit (Clock Out Select)

The CSEL bit selects the source of the clock to be output from the PCLBUZn pin. When changing the clock source, set the PCLOE bit to 0.

PCLOE bit (Clock Out Enable)

The PCLOE enables output from the PCLBUZn 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 CSEL and CCS[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.

8.2.20 LIOTRM : Low-speed On-chip Oscillator Trimming Register

Base address: SYSC = 0x4001_E000

Offset address: 0x0805

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

Bit	Symbol	Function	R/W
7:0	LIOTRM[7:0]	LOCO User Trimming 0x00: Minimum speed 0x01: ⋮ 0x80 Initial value ⋮ 0xFE: 0xFF: Maximum speed	R/W

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

The frequency of the low-speed on-chip oscillator can be trimmed by setting the LIOTRM register.

When LIOTRM is modified, the frequency stabilization time corresponds to the frequency stabilization time at the start of MCU operation. For low-speed on-chip oscillator clock frequency trimming resolution, see Table 31.6.

Low-speed oscillator calibration can be performed by self-measuring the frequency of the low-speed oscillator using a timer (such as a Timer Array Unit or a 32-bit interval timer) with a high-accuracy external clock input, or by other means.

Note: The frequency will vary if the temperature and power supply voltage change after frequency trimming. In such cases, it is essential to perform trimming regularly or before high frequency accuracy is required.

8.2.21 MIOTRM : Middle-speed On-chip Oscillator Trimming Register

Base address: SYSC = 0x4001_E000

Offset address: 0x0804

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

Bit	Symbol	Function	R/W
7:0	MIOTRM[7:0]	MOCO User Trimming 0x00: Minimum speed 0x01: ⋮ 0x90: Initial value ⋮ 0xFE: 0xFF: Maximum speed	R/W

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

The frequency of the middle-speed on-chip oscillator can be trimmed by setting the MIOTRM register.

When MIOTRM is modified, the frequency stabilization time corresponds to the frequency stabilization time at the start of MCU operation. For middle-speed on-chip oscillator clock frequency trimming resolution, see Table 31.6.

Middle-speed oscillator calibration can be performed by self-measuring the frequency of the middle-speed oscillator using a timer (such as a Timer Array Unit or a 32-bit interval timer) with a high-accuracy external clock input, or by other means.

Note: The frequency will vary if the temperature and power supply voltage change after frequency trimming. In such cases, it is essential to perform trimming regularly or before high frequency accuracy is required.

8.2.22 HIOTRM : High-speed On-chip Oscillator Trimming Register

Base address: FLCN = 0x407E_C000

Offset address: 0x0200

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

Bit	Symbol	Function	R/W
5:0	HIOTRM[5:0]	HOCO User Trimming 0x00: Minimum speed ⋮ 0x3F: Maximum speed	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W

Note: The original HOCO trimming data is set in the HIOTRM register at the time of shipment.
Note: After a reset, the HIOTRM register is initialized to the original HOCO trimming data.

The frequency of the high-speed on-chip oscillator can be trimmed by setting the HIOTRM register.

When HIOTRM is modified, the frequency stabilization time corresponds to the frequency stabilization time at the start of MCU operation. For high-speed on-chip oscillator clock frequency trimming resolution, see Table 31.6.

High-speed oscillator calibration can be performed by self-measuring the frequency of the high-speed oscillator using a timer (such as a Timer Array Unit or a 32-bit interval timer) with a high-accuracy external clock input, or by other means.

Note: The frequency will vary if the temperature and power supply voltage change after frequency trimming. In such cases, it is essential to perform trimming regularly or before high frequency accuracy is required.

8.3 Main Clock Oscillator

To supply the clock signal to the main clock oscillator, use one of the following ways:

- Connect an oscillator
- Connect the input of an external clock signal.

8.3.1 Connecting a Crystal Resonator

Figure 8.3 shows an example of connecting a crystal resonator. A damping resistor (Rd) can be added, if required.

Because the resistor values vary according to the resonator and the oscillation drive capability, use values recommended by the resonator manufacturer. If the manufacturer recommends using an external feedback resistor (Rf), insert an Rf between X1 and X2 by following the instructions.

When connecting a resonator to supply the clock, the frequency of the resonator must be in the frequency range of the resonator for the main clock oscillator as described in Table 8.1.

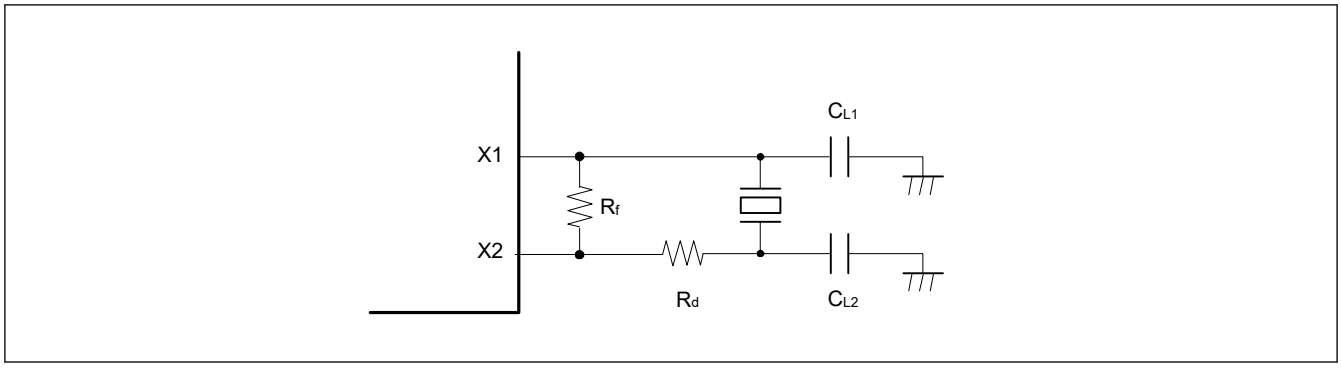


Figure 8.3 Example of crystal resonator connection

Figure 8.4 shows an equivalent circuit of the crystal resonator.

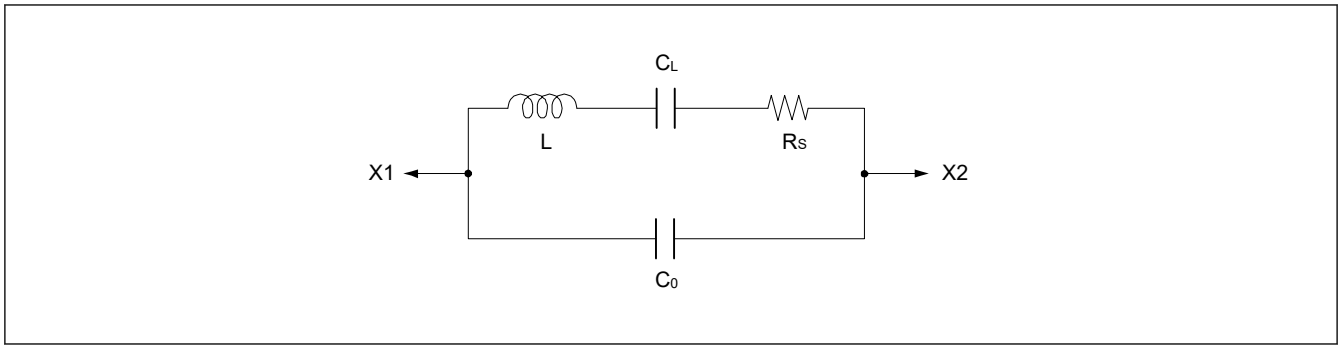


Figure 8.4 Equivalent circuit of the crystal resonator

8.3.2 External Clock Input

Figure 8.5 shows an example of connecting an external clock input. To operate the oscillator with an external clock signal, set the CMC.MOSEL bit to 11b. The X1 pin can be used as an I/O port.

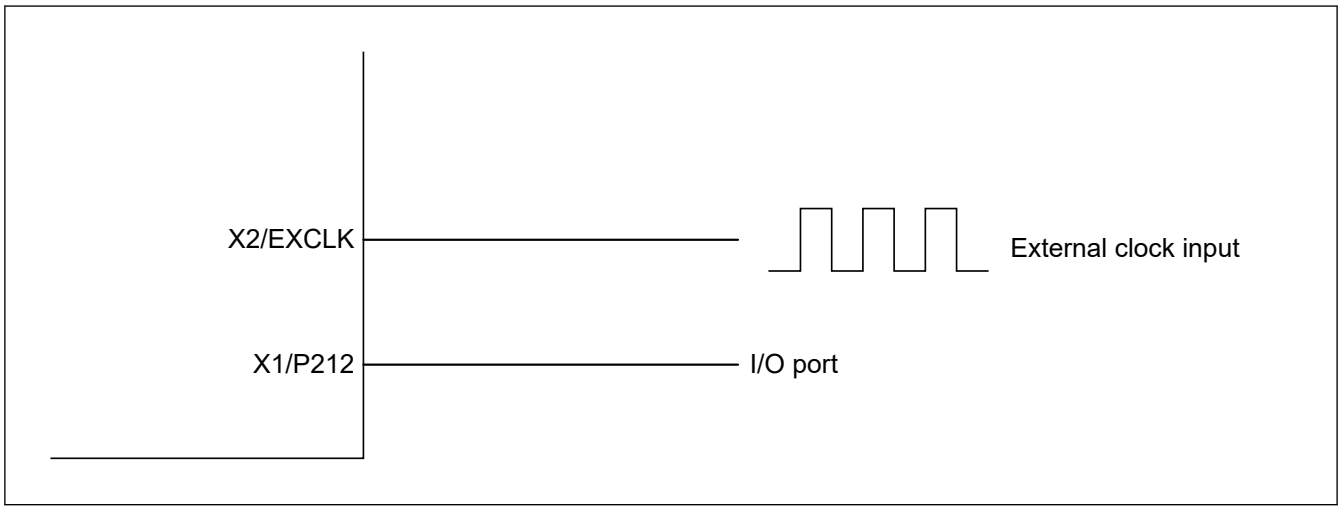


Figure 8.5 Equivalent circuit for external clock

8.3.3 Notes on External Clock Input

The frequency of the external clock input can only be changed when the main clock oscillator is stopped. Do not change the frequency of the external clock input when the setting of the Main Clock Oscillator Stop bit (MOSCCR.MOSTP) is 0.

8.4 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 XCIN and XCOUT 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.

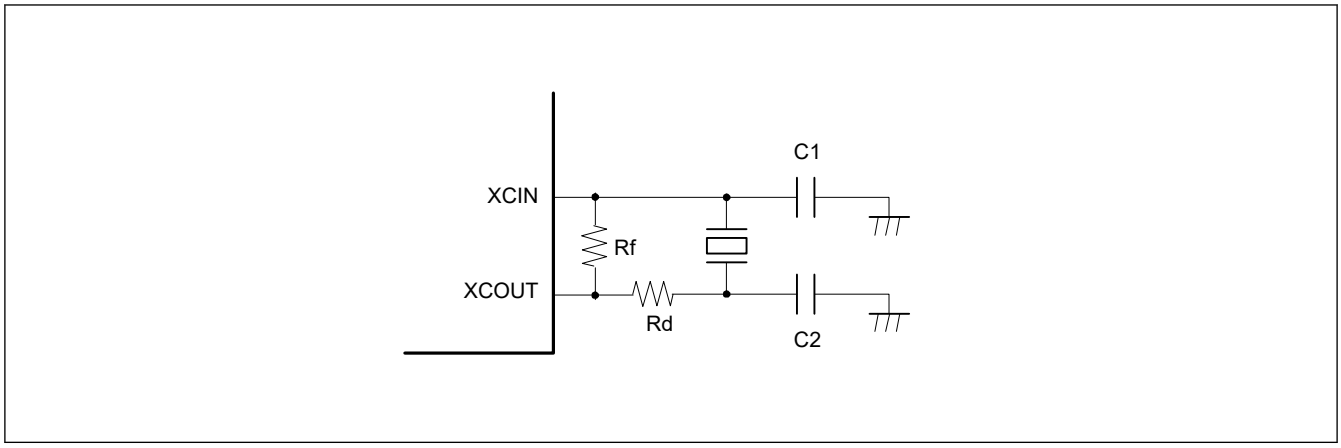


Figure 8.6 Connection example of 32.768-kHz crystal resonator

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

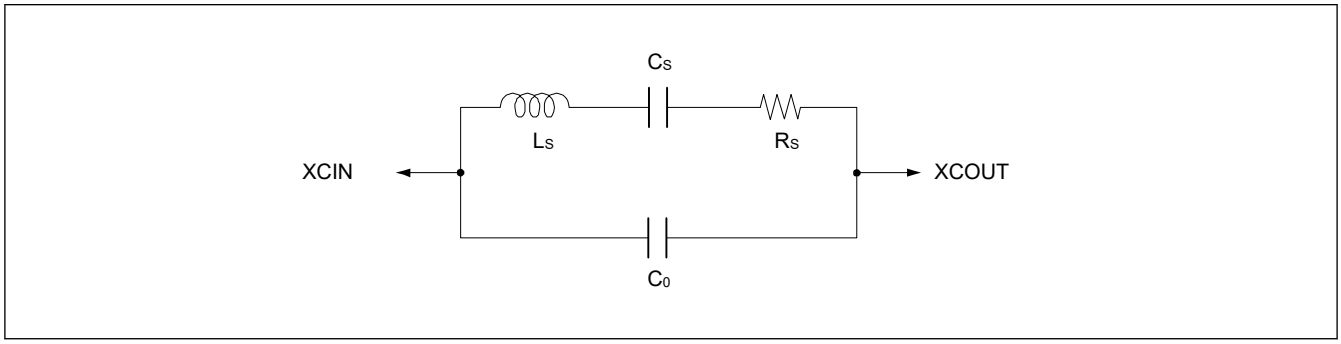


Figure 8.7 Equivalent circuit for the 32.768-kHz crystal resonator

8.5 Internal Clock

Clock sources for the internal clock signals include:

- Main clock oscillator
- Sub-clock oscillator
- HOCO clock
- MOCO clock
- LOCO 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 clock for peripheral modules — Peripheral module clock (PCLKB)
- Operating clock for the RTC clock — RTCCLK
- Operating clock for the IWDT — IWDTCLK
- 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 8.5.6. 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, SRAM, and the peripheral modules.

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

- HOCOFRQ1[2:0] in OFS1
- DIV[2:0] in HOCODIV
- DIV[2:0] in MOCODIV
- DIV[2:0] in MOSCDIV
- CKSEL in FOCOSCR
- CKSEL in FMAINSCR
- CKSEL in FSUBSCR
- CKSEL in ICLKSCR

When the clock source of ICLK is being switched, the duration of ICLK clock cycle become longer during the clock source transition period. See Figure 8.8 and Figure 8.9.

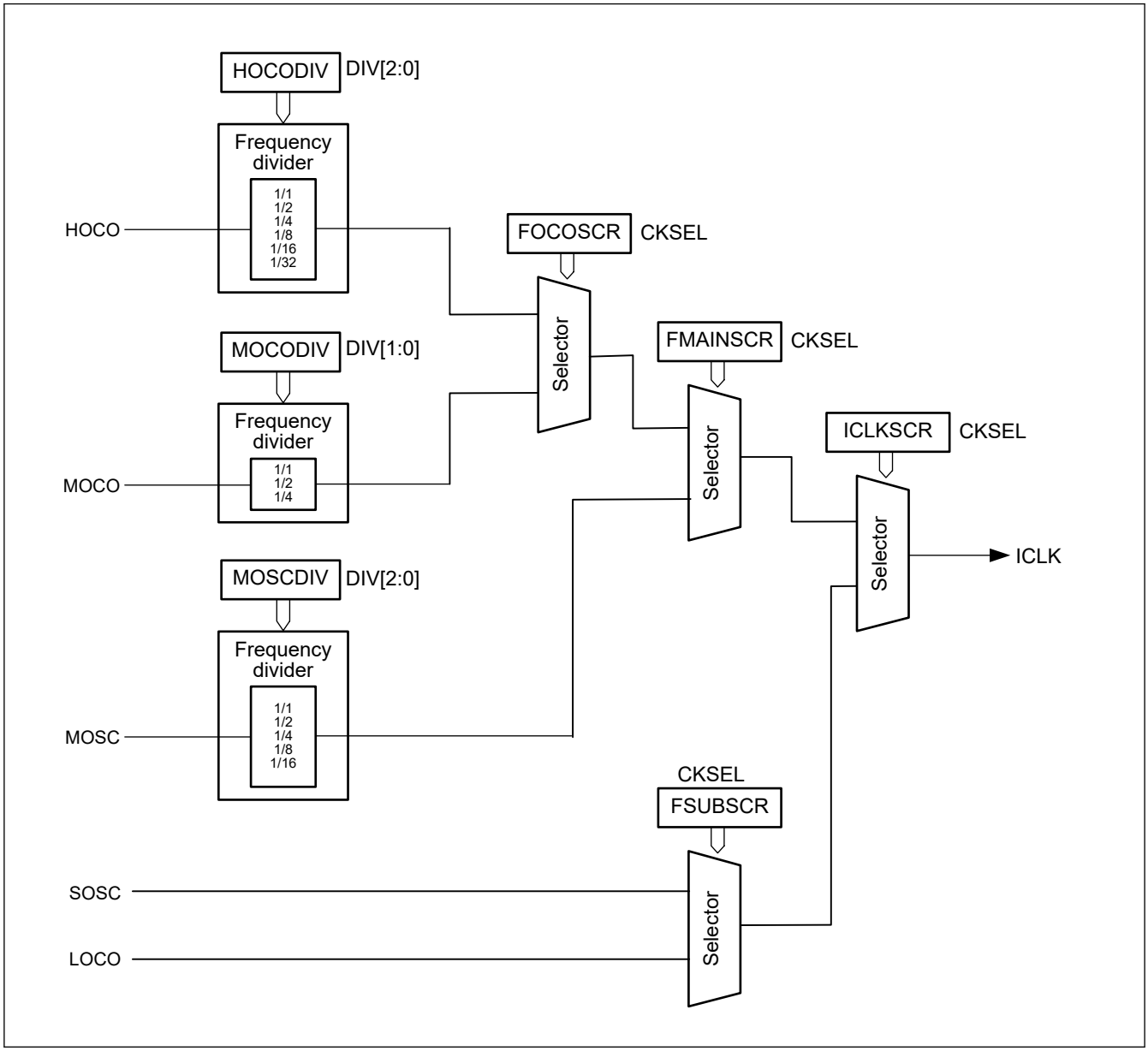


Figure 8.8 Block diagram of clock source selector

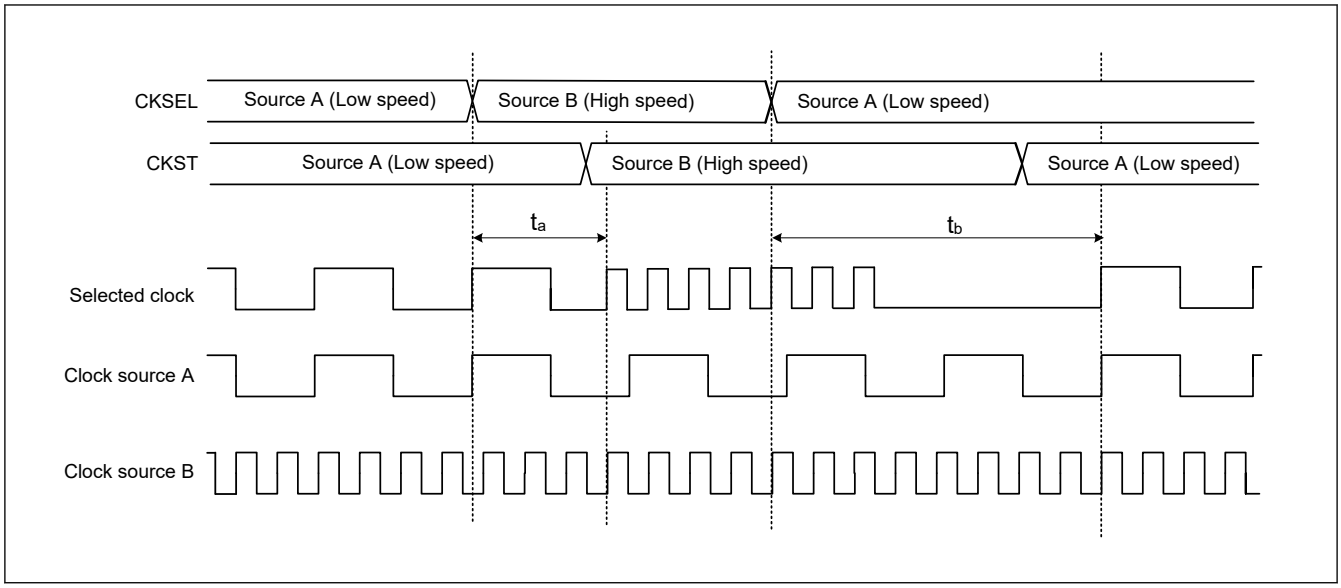


Figure 8.9 Timing of clock source switching

- Maximum Number of Clock Cycles Required for HOCO ↔ MOCO
  t_a : 2 cycles
  t_b : 2 × (source A frequency) / (source B frequency) cycles
- Maximum Number of Clock Cycles Required for FOCO ↔ MOSC
  t_a : 2 cycles
  t_b : 2 × (source A frequency) / (source B frequency) cycles
- Maximum Number of Clock Cycles Required for SOSC ↔ LOCO
  t_a : 0 cycle
  t_b : 0 cycle
- Maximum Number of Clock Cycles Required for FMAIN ↔ FSUB
  t_a : 3 cycles
  t_b : 1 + 2 × (FMAIN frequency) / (FSUB frequency) cycles

8.5.2 Peripheral Module Clock (PCLKB)

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

The peripheral module clock (PCLKB) is equal to the system clock (ICLK) in this microcontroller.

8.5.3 RTC-dedicated Clock (RTCCLK)

The RTC-dedicated clocks, RTCCLK, are the operating clocks for the RTC. RTCCLK is generated by the sub-clock oscillator or LOCO clock.

8.5.4 IWDT Clock (IWDTCLK)

The IWDT clock (IWDTCLK) is the operating clock for the IWDT. IWDTCLK is internally generated by the LOCO clock.

8.5.5 SysTick Timer-dedicated Clock (SYSTICCLK)

The SysTick timer-dedicated clock, SYSTICCLK, is the operating clock for the SysTick timer. SYSTICCLK is generated by the LOCO clock.

8.5.6 External Pin Output Clock (CLKOUT)

The CLKOUT is output externally for the clock or buzzer output. CLKOUT is output to the PCLBUZn pin (n = 0, 1) when CKSn.PCLOE is set to 1. Only change the value in the CSEL or CCS[2:0] bits in CKSn when the CKSn.PCLOE bit is 0.

The CLKOUT clock frequency is specified in the following bits:

- CSEL or CCS[2:0] in CKSn
- HOCOFRQ1[2:0] in OFS1
- DIV[2:0] in HOCODIV
- DIV[2:0] in MOCODIV
- DIV[2:0] in MOSCDIV
- CKSEL in FOCOSCR
- CKSEL in FMAINSCR
- CKSEL in FSUBSCR

8.6 Usage Notes

8.6.1 Register Access

1. Do not write to registers listed in this section for the following condition:
   [Registers]
   - FOCOSCR, FMAINSCR, FSUBSCR, HOCODIV

   [Condition]
   - ICLKSCR.CKSEL = 1 (ICLK = LOCO or SOSC)
2. Do not write to registers listed in this section for the following condition:
   [Registers]
   - FOCOSCR, HOCODIV

   [Condition]
   - FMAINSCR.CKSEL = 1 (FMAIN = MOSC)
3. Do not write to registers listed in this section for the following condition:
   [Registers]
   - HOCODIV

   [Condition]
   - FOCOSCR.CKSEL = 1 (FOCO = MOCO)

8.6.2 Notes on Clock Generation Circuit

To ensure correct processing after the clock frequency changes, first modify to the relevant Clock Control register to change the frequency, then read the value from the register, and finally perform the subsequent processing.

8.6.3 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.

8.6.4 Notes on Board Design

When using a crystal resonator, place the resonator and its load capacitors as close to the X1 and X2 pins as possible. Other signal lines should be routed away from the oscillation circuit as shown in Figure 8.10 to prevent electromagnetic induction from interfering with correct oscillation. Figure 8.10 shows the case which the main clock oscillator is used. In case of sub-clock oscillator, it is also same as Figure 8.10.

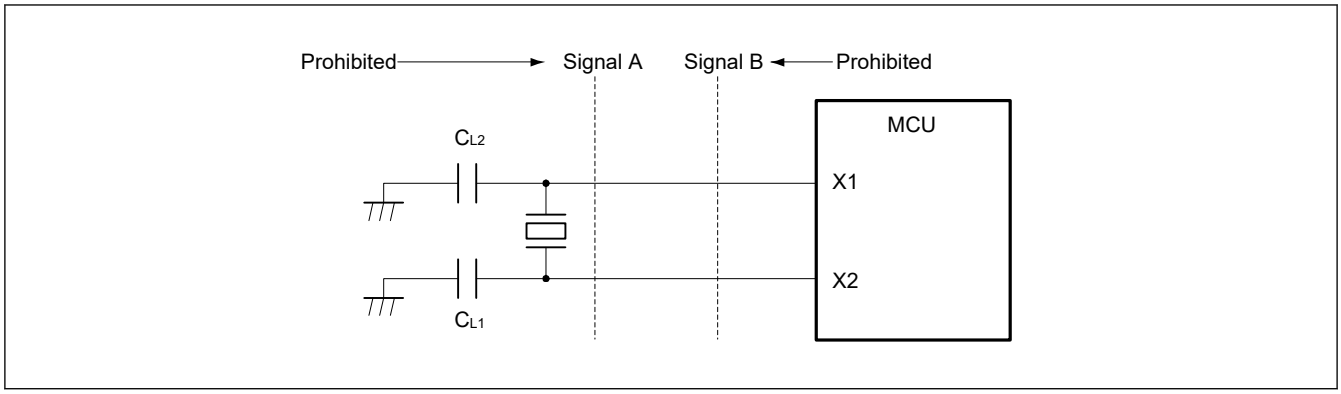


Figure 8.10 Signal routing in board design for oscillation circuit

8.6.5 Notes on Resonator Connect Pin

When the main clock is not used (CMC.MOSEL[0] bit is 0), the X1 and X2 pins can be used as general ports. When the main clock is External clock input mode (CMC.MOSEL[1:0] bits are 11b), the X1 pin can be used as general ports.

9. Low Power Modes

9.1 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 9.1 lists the specifications of the low power mode functions. Table 9.2 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 9.1 Specifications of the low power mode functions

Item	Specification
Reducing power consumption by switching clock signals	The frequency division ratio can be selected for HOCO, MOCO, and MOSC*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 9.2 Operating conditions of each low power mode (1 of 2)

Item	Sleep mode	Software Standby mode	Snooze mode*1
Transition condition	When [Condition 1] or [Condition 2] while SBYCR.SSBY = 0 [Condition 1] • WFI instruction • A valid interrupt request*2 is not accepted by the CPU. (including a transition from the time WFI instruction is executed to the time the transition to Sleep mode is completed) [Condition 2] • SCR.SLEEPONEXIT = 1 • Complete execution of all exception handlers • A valid interrupt request*2 is not accepted by the CPU. (including a transition from the time WFI instruction is executed to the time the transition to Sleep mode is completed)	When [Condition 1] or [Condition 2] while SBYCR.SSBY = 1 [Condition 1] • WFI instruction • A valid interrupt request*2 is not accepted by the CPU. (including a transition from the time WFI instruction is executed to the time the transition to Software Standby mode is completed) [Condition 2] • SCR.SLEEPONEXIT = 1 • Complete execution of all exception handlers • A valid interrupt request*2 is not accepted by the CPU. (including a transition from the time WFI instruction is executed to the time the transition to Software Standby mode is completed)	Snooze request in Software Standby mode.
Canceling method	All interrupts. Any reset available in the mode.	Interrupts shown in Table 9.3. Any reset available in the mode.	Interrupts shown in Table 9.3. Any reset available in the mode.
State after cancellation by an interrupt	Program execution state	Program execution state	Program execution state
State after cancellation by a reset	Reset state	Reset state	Reset state
Main clock oscillator	Selectable	Stop	Stop
Sub-clock oscillator	Selectable	Selectable	Selectable

Table 9.2 Operating conditions of each low power mode (2 of 2)

Item	Sleep mode	Software Standby mode	Snooze mode*1
High-speed on-chip oscillator	Selectable	Stop	Selectable*3
Middle-speed on-chip oscillator	Selectable	Stop	Selectable*3
Low-speed on-chip oscillator	Selectable	Selectable	Selectable
Clock/buzzer output function	Selectable	Selectable*4	Selectable*4
CPU	Stop (Retained)	Stop (Retained)	Stop (Retained)
SRAM	Selectable	Stop (Retained)	Selectable
Flash memory	Selectable*8	Stop (Retained)	Selectable*8
Data Transfer Controller (DTC)	Selectable	Stop (Retained)	Selectable
Independent Watchdog Timer (IWDT)	Selectable*5	Selectable*5	Selectable*5
Realtime clock (RTC)	Selectable	Selectable	Selectable
32-bit Interval Timer (TML32)	Selectable	Selectable*6	Selectable*6
12-bit A/D Converter (ADC12)	Selectable	Stop (Retained)	Selectable*11
Serial Array Unit (SAU0)	Selectable	Stop (Retained)	Selectable*9
Serial Array Unit (SAU1)	Selectable	Stop (Retained)	Operation prohibited
Serial Interface UARTA (UARTA0, UARTA1)	Selectable	Selectable*12	Selectable*12
I2C Bus Interface (IICA0, IICA1)	Selectable	Selectable*10	Selectable*10
Event Link Controller (ELC)	Selectable	Stop (Retained)	Selectable*7
NMI, IRQn (n = 0 to 7) pin interrupt	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 ICLK 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. Valid interrupt requests are any interrupts or exceptions that are not masked by the priority level of current exception and the priority level set by BASEPRI. In addition, if the interrupt request is based on IELSRn, the interrupt must be enabled by NVIC_ISERn.

Note 3. Oscillates only when ICLK source clock is set.

Note 4. Stopped when the Clock Output Source Select bit (CKS0.CSEL/CKS1.CSEL) is set to a value other than 1 (FSUB).

Note 5. 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.

Note 6. TML32 can operate when 100b (FSXP) or 101b (Event input form the ELC) is selected in the ITLCSEL0.CSEL[2:0] bits.

Note 7. Event lists the restrictions described in section 15.4.7. Link Availability in Sleep, Software Standby, and Snooze Mode.

Note 8. Flash, operating or stopping is selected by setting the Flash Stop Control bit (SBYCR.FLSTP).

Note 9. When using the Serial Array Unit 0 (SAU0) in Snooze mode, the SSC0.SWC bit must be 1.

Note 10. Only wakeup interrupt is available.

Note 11. When using the 12-bit A/D Converter (ADC12) in Snooze mode, the ADM2.AWC bit must be 1.

Note 12. UARTA can operate when 1000b (FSXP) is selected in the UTAnCK.CK[3:0].

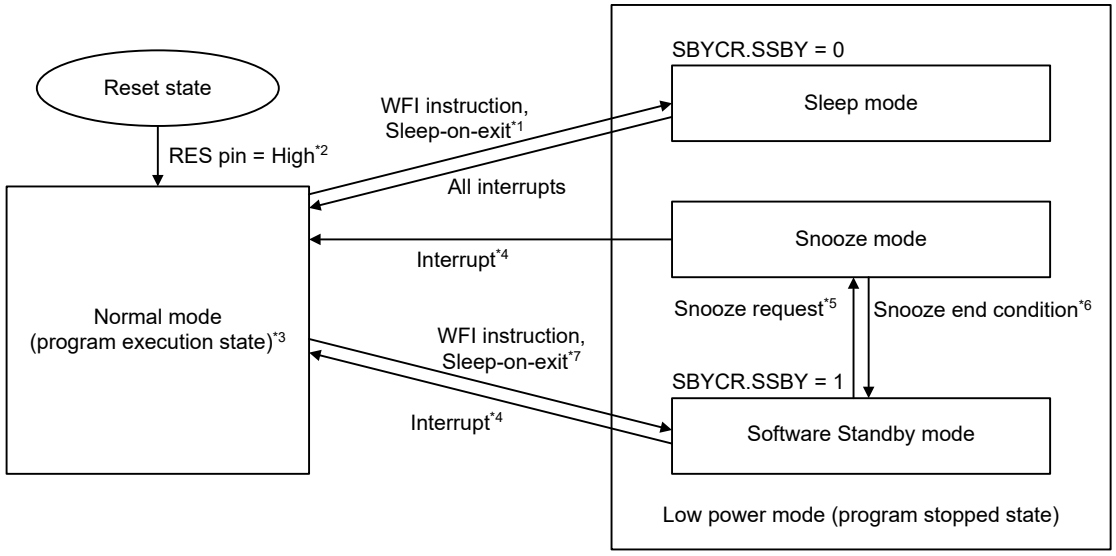
Table 9.3 Available interrupt sources to transition to Normal mode from Snooze mode and Software Standby mode (1 of 2)

Interrupt source	Name	Software Standby mode	Snooze mode
NMI		Yes	Yes

Table 9.3 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
Port	PORT_IRQn (n = 0 to 7)	Yes	Yes
LVD	LVD_LVD1	Yes	Yes
IWDT	IWDT_NMIUNDF	Yes	Yes
RTC	RTC_ALM_OR_PRD	Yes	Yes
TML32	TML32_ITL_OR	Yes	Yes
UARTA	UARTA0_ERRI	Yes	Yes
	UARTA0_TXI	Yes	Yes
	UARTA0_RXI	Yes	Yes
	UARTA1_ERRI	Yes	Yes
	UARTA1_TXI	Yes	Yes
	UARTA1_RXI	Yes	Yes
IICA	IICA0_TXRXI	Yes	Yes
	IICA1_TXRXI	Yes	Yes
ADC12	ADC12_ADI	No	Yes
SAU0	SAU0_SPI_TXRXI00	No	Yes
	SAU0_UART_ERRI0	No	Yes
	SAU0_UART_RXI0	No	Yes
DTC	DTC_COMPLETE	No	Yes

Figure 9.1 shows the transition between Normal mode to low power mode.



Note 1. When an interrupt that acts as a trigger for cancellation of the Sleep mode state is received during a transition to the Sleep mode state, the MCU does not transit to Sleep mode state but go back to Normal mode state.
Note 2. The HOCO 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 9.3.
Note 5. See section 9.8.1. Transition to Snooze Mode.
Note 6. See section 9.8.3. Returning from Snooze Mode to Software Standby Mode.
Note 7. When an interrupt that acts as a trigger for cancellation of the Sleep mode state is received during a transition to the Software Standby mode state, the MCU does not transit to Software Standby mode state but go back to Normal mode state.

Figure 9.1 Low power mode transitions

9.2 Register Descriptions

9.2.1 SBYCR : Standby Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x0860

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SSBY	—	—	—	—	—	RTCLPC	FWKUP	FLSTP	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	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	FLSTP	Flash Mode in Sleep Mode or in Snooze Mode 0: Flash active 1: Flash stop	R/W
8	FWKUP	Setting for Starting the High-speed On-chip Oscillator at the times of release from Software Standby Mode and of Transitions to Snooze Mode 0: Starting of the high-speed on-chip oscillator is at normal speed 1: Starting of the high-speed on-chip oscillator is at high speed	R/W
9	RTCLPC	SOSC Setting in Software Standby Mode or in Snooze Mode 0: Enables supply of SOSC clock to peripheral functions 1: Stops supply SOSC clock to peripheral functions other than the Realtime clock	R/W
14:10	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	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.

FLSTP bit (Flash Mode in Sleep Mode or in Snooze Mode)

When the FLSTP bit is set to 1, power consumption can be reduced because Flash stops during Sleep mode or Snooze mode. Instead, the Sleep mode release or Snooze mode release time will be extended. Also flash programming/erasure is not possible.

FWKUP bit (Setting for Starting the High-speed On-chip Oscillator at the times of release from Software Standby Mode and of Transitions to Snooze Mode)

When the FWKUP bit is set to 1, High-speed On-chip Oscillator enters high-speed startup mode, shortening standby release time and Snooze transition time. Instead, the frequency accuracy of HOCO changes during OSCSF.HOCOSF = 0 after ICLK starts operating, see section 31, Electrical Characteristics. This bit can be set only when ICLK = HOCO (32MHz).

RTCLPC bit (SOSC Setting in Software Standby Mode or in Snooze Mode)

When the RTCLPC bit is set to 1, peripheral functions operating on SOSC other than RTC stop at Software Standby mode.

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 or FENTRYR.FENTRYD bit is 1 setting of the SSBY bit is ignored. Even if SSBY bit is 1, the MCU enters Sleep mode on execution of a WFI instruction.

9.2.2 MSTPCRA : Module Stop Control Register A

Base address: SYSC = 0x4001_E000

Offset address: 0x0C02

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	MSTPA22	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1

Bit	Symbol	Function	R/W
5:0	—	These bits are read as 1. The write value should be 1.	R/W
6	MSTPA22	DTC Module Stop*1 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
15:7	—	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.

9.2.3 MSTPCRB : Module Stop Control Register B

Base address: MSTP = 0x4004_7000

Offset address: 0x0000

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:	MSTPB15	—	—	—	MSTPB11	MSTPB10	—	—	MSTPB7	MSTPB6	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
5:0	—	These bits are read as 1. The write value should be 1.	R/W
6	MSTPB6	Serial Array Unit 0 Module Stop Target module: SAU0 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
7	MSTPB7	Serial Array Unit 1 Module Stop Target module: SAU1 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
9:8	—	These bits are read as 1. The write value should be 1.	R/W
10	MSTPB10	I^2C Bus Interface Module Stop Target module: IICA0 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
11	MSTPB11	I^2C Bus Interface Module Stop Target module: IICA1 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
14:12	—	These bits are read as 1. The write value should be 1.	R/W
15	MSTPB15	Serial Interface UARTA Module Stop Target module: UARTA0, UARTA1 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
31:16	—	These bits are read as 1. The write value should be 1.	R/W

9.2.4 MSTPCRC : Module Stop Control Register C

Base address: MSTP = 0x4004_7000

Offset address: 0x0004

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	MSTPC28	—	—	—	—	—	—	—	—	—	—	—	—
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	—	—	—	—	—	—	—	—	—	—	—	—	MSTPC1	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	—	This bit is read as 1. The write value should be 1.	R/W
1	MSTPC1	Cyclic Redundancy Check Calculator Module Stop Target module: CRC 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
13:2	—	These bits are read as 1. The write value should be 1.	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
27:15	—	These bits are read as 1. The write value should be 1.	R/W
28	MSTPC28	True Random Number Generator Module Stop 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

9.2.5 MSTPCRD : Module Stop Control Register D

Base address: MSTP = 0x4004_7000

Offset address: 0x0008

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	MSTPD23	—	—	—	—	—	—	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:	—	—	—	—	—	—	—	—	—	—	—	MSTPD4	—	—	—	MSTPD0
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	MSTPD0	Timer Array Unit 0 Module Stop Target module: TAU0 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
3:1	—	These bits are read as 1. The write value should be 1.	R/W
4	MSTPD4	32-bit Interval Timer Module Stop Target module: TML32 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
15:5	—	These bits are read as 1. The write value should be 1.	R/W
16	MSTPD16	A/D Converter Module Stop Target module: ADC12 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
22:17	—	These bits are read as 1. The write value should be 1.	R/W
23	MSTPD23	Realtime Clock Module Stop Target module: RTC 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
31:24	—	These bits are read as 1. The write value should be 1.	R/W

9.2.6 FLMODE : Flash Operating Mode Control Register

Base address: FLCN = 0x407E_C000

Offset address: 0x020A

Bit position:	7	6	5	4	3	2	1	0
Bit field:	MODE[1:0]		—	—	—	—	—	—
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
7:6	MODE[1:0]	Operating Mode Select 0 0: Setting prohibited 0 1: Low-speed mode 1 0: Middle-speed mode 1 1: High-speed mode	R/W

Note: Direct transition between High-speed mode and Low-speed mode is prohibited (Writing 0x40 (0xC0) over 0xC0 (0x40) is ignored). The transition between High-speed mode and Low-speed mode should be made through the Middle-speed mode.

The FLMODE 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 FLMODE setting. For the procedure to change the operating power control modes, see section 9.5. Function for Lower Operating Power Consumption.

MODE[1:0] bits (Operating Mode Select)

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

Table 9.4 shows the relationship between the operating power control modes, the MODE[1:0], and ICLKSCR.CKSEL bits settings. Writing to FLMODE.MODE[1:0] is prohibited while MCU is under the following conditions:

1. The value of the FLMODE register can be changed when the FLMWEN bit in the flash operation mode protect register (FLMWRP) is 1. After the value of the FLMODE register is changed, set the FLMWEN bit to 0.
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 (ICLKSCR.CKST bit and CKSEL bit is 1).

Table 9.4 Operating power control mode

Operating power control mode	MODE[1:0] bits	ICLKSCR.CKST bit	Power consumption
High-speed mode	11b	0	High
Middle-speed mode	10b	0	↓
Low-speed mode	01b	0	↓
Subosc-speed mode	xxb	1	Low

9.2.7 FLMWRP : Flash Operating Mode Protect Register

Base address: FLCN = 0x407E_C000

Offset address: 0x020B

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

Bit	Symbol	Function	R/W
0	FLMWEN	Control of Flash Operation Mode Select Register 0: Rewriting the FLMODE register is disabled 1: Rewriting the FLMODE register is enabled	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

9.2.8 PSMCR : Power Save Memory Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x0862

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

Bit	Symbol	Function	R/W
1:0	RAMSD[1:0]	Operating Mode of the RAM 0 0: Normal mode (continues to operate) 0 1: Setting prohibited 1 0: Standby mode 1 1: Shutdown mode	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: Shutdown mode applies to all RAM other than that in the range from 0x2000_4000 to 0x2000_5FFF. The RAM that in the range from 0x2000_4000 to 0x2000_5FFF continues to operate and retains data.
Note: Do not access RAM while it is in the standby mode or shutdown mode.
Note: When the RAM returns to normal mode from shutdown mode, the contents of the RAM other than in the range from 0x2000_4000 to 0x2000_5FFF are undefined.
Note: When SYOCDCR.DBGEN is set to 1, the RAM does not enter shutdown mode.
Note: Setting the RAMSD[1:0] bits. This register is protected by the PRCR.PRC1 bit.

Follow the procedure shown in Table 9.5 to switch the operating mode of the RAM from normal mode to shutdown mode.

Table 9.5 Procedure for settings to switch from Normal mode to Shutdown mode

Step	Operation	Mode
0	—	Normal mode
1	RAMSD[1:0] = 10b	—
2	—	Wait mode
3	Waiting (80 ns)	—
4	RAMSD[1:0] = 11b	—
5	—	Shutdown mode

Follow the procedure shown in Table 9.6 to switch the operating mode of the RAM from shutdown mode to normal mode.

Table 9.6 Procedure for settings to switch from Shutdown mode to Normal mode

Step	Operation	Mode
0	—	Shutdown mode
1	RAMSD[1:0] = 10b	—
2	—	Wait mode
3	Waiting (1.2 µs)	—
4	RAMSD[1:0] = 00b	—
5	—	Normal mode

Note: When the RAM returns to normal mode from shutdown mode, the contents of the RAM other than in the range from 0x2000_4000 to 0x2000_5FFF are undefined. Initialize the RAM area to be used.

9.2.9 SYOCDCR : System Control OCD Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x0863

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 set to 1. For details, see section 2.7.3. Restrictions on Connecting an OCD emulator.

9.3 Reducing Power Consumption by Switching Clock Signals

The clock frequency changes when the HOCODIV, MOCODIV, and MOSCDIV register are set.

9.4 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.

9.5 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.

9.5.1 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 9.7 Available oscillators in each mode

Mode	Oscillator				
	High-speed on-chip oscillator	Middle-speed on-chip oscillator	Low-speed on-chip oscillator	Main clock oscillator	Sub-clock oscillator
High-speed	Available	Available	Available	Available	Available
Middle-speed	Available	Available	Available	Available	Available
Low-speed	Available	Available	Available	N/A	Available
Subosc-speed	N/A	N/A	Available	N/A	Available

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

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

(Operation begins in High-speed mode)

1. Change the oscillator to what is used in Middle-speed mode. Set the frequency of each clock lower than or equal to the maximum operating frequency in Middle-speed mode.
2. Turn off the oscillator that is not required in Middle-speed mode.
3. Set the FLMWRP.FLMWEN bit to 1 (Rewriting the FLMODE register is enabled.).
4. Set the FLMODE.MODE[1:0] bits to 10b (Middle-speed mode).
5. Set the FLMWRP.FLMWEN bit to 0 (Rewriting the FLMODE register is disabled.).

(Operation is now in Middle-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 or LOCO.
2. Turn off HOCO, MOCO and main oscillator.

(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[*1]

(Operation begins in Subosc-speed mode)

1. Turn on the required oscillator in High-speed mode.
2. 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)

Note: When transitioning from Subosc-speed mode to a higher speed mode, it can only revert to the mode that was active before entering Subosc-speed mode.

Note 1. When the mode before entering Subosc-speed mode was High-speed mode.

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

(Operation begins in Middle-speed mode)

1. Set the FLMWRP.FLMWEN bit to 1 (Rewriting the FLMODE register is enabled.).
2. Set the FLMODE.MODE[1:0] bits to 11b (High-speed mode).
3. Set the FLMWRP.FLMWEN bit to 0 (Rewriting the FLMODE register is disabled.).
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)

Note: There is an automatic wait time.

- Middle-speed to High-speed: 225 clocks
- Low-speed to Middle-speed: 10 clocks
- Middle-speed to Low-speed: 10 clocks
- High-speed to Middle-speed: 30 clocks
- other to Subosc-speed: 1 + 2 fMAIN/fSUB clock
- Subosc-speed to other: 3 clocks

9.5.2 Operating Range

Figure 9.2 to Figure 9.5 show the ICLK operating voltages and frequencies.

High-speed mode

The maximum operating frequency during a flash read is 32 MHz for ICLK. The operating voltage range during a flash read is 1.8 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 (P/E), the operating frequency range is 1 to 32 MHz and the operating voltage range is 1.8 to 5.5 V.

Figure 9.2 shows the operating voltages and frequencies in High-speed mode.

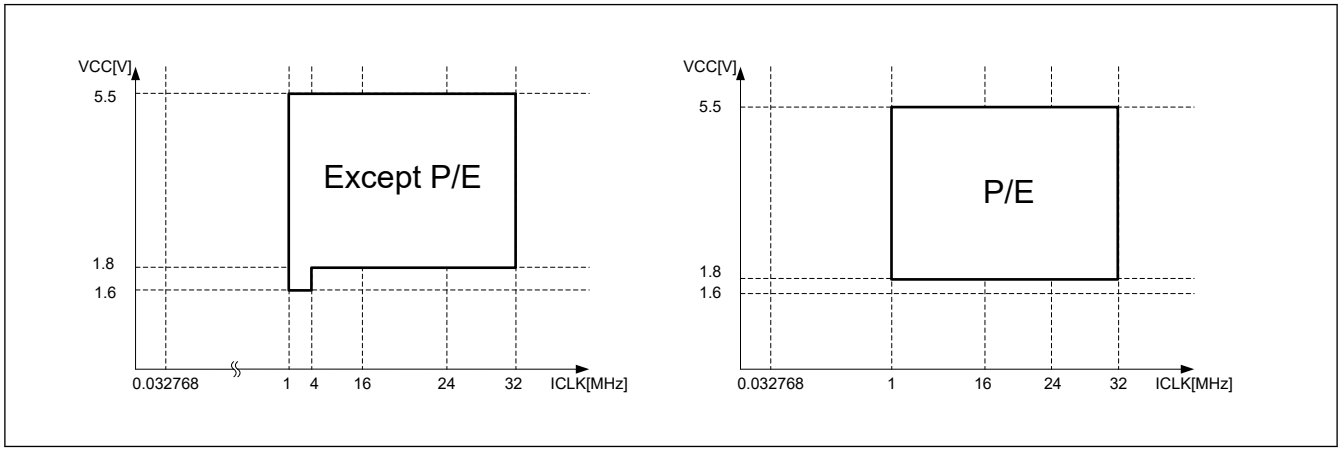


Figure 9.2 Operating voltages and frequencies in High-speed mode

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 (P/E), the operating frequency range is 1 to 24 MHz and the operating voltage range is 1.6 to 5.5 V.

Figure 9.3 shows the operating voltages and frequencies in Middle-speed mode.

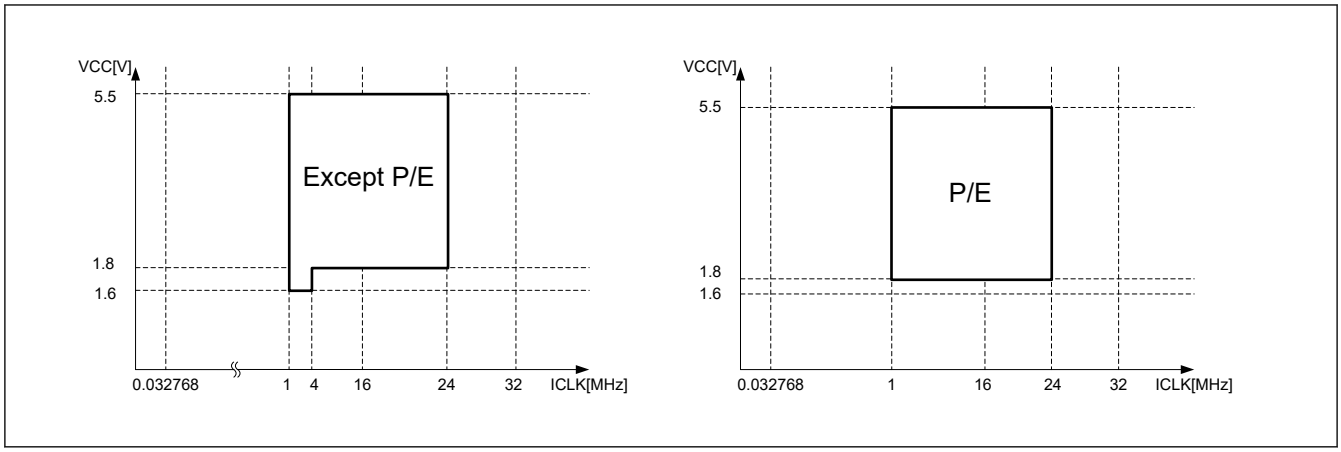


Figure 9.3 Operating voltages and frequencies in Middle-speed mode

Low-speed mode

The maximum operating frequency during a flash read is 2 MHz for ICLK. The operating voltage range during a flash read is 1.6 to 5.5 V.

Figure 9.4 shows the operating voltages and frequencies in Low-speed mode.

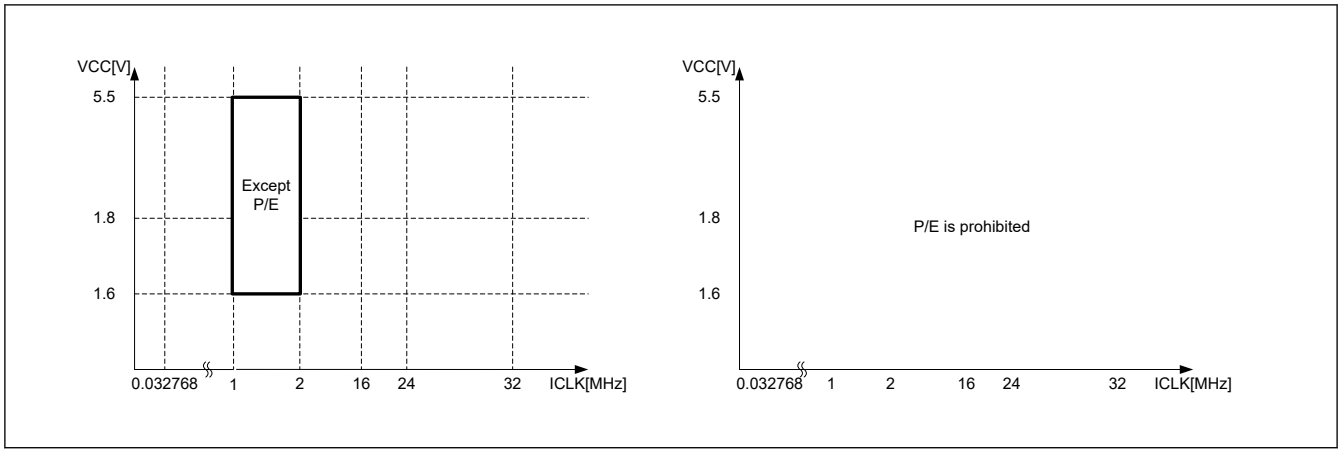


Figure 9.4 Operating voltages and frequencies in Low-speed mode

Using Main clock oscillator as the ICLK source in the Low-speed mode is prohibited.

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.

Figure 9.5 shows the operating voltages and frequencies in Subosc-speed mode.

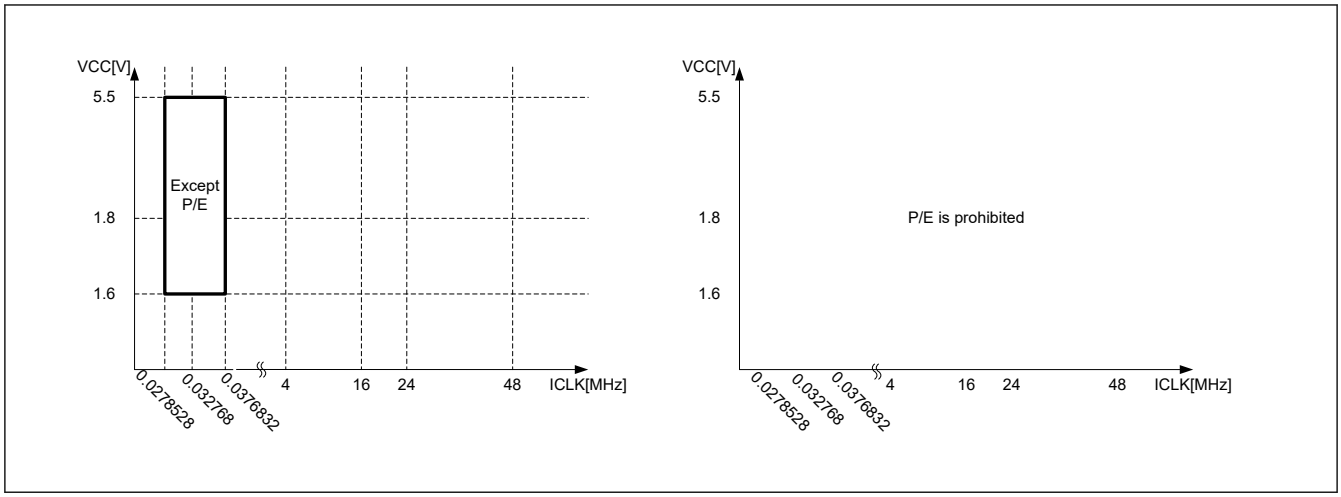


Figure 9.5 Operating voltages and frequencies in Subosc-speed mode

9.6 Sleep Mode

9.6.1 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.

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).

9.6.2 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
- A reset caused by an IWDT underflow

The operations are as follows:

1. Canceling by an interrupt
   When an interrupt request is generated, Sleep mode is canceled and the MCU starts the interrupt handling.
2. Canceling by RES pin reset
   When the RES pin is driven low, the MCU enters the reset state. Be sure to keep the RES pin low for the time period specified in section 31, Electrical Characteristics. When the RES pin is driven high after the specified time period, the CPU starts the reset exception handling.
3. Canceling by IWDT reset
   - Sleep mode is canceled by an internal reset generated by an IWDT underflow and the MCU starts the reset exception handling. However, IWDT stops in Sleep mode and an internal reset for canceling Sleep mode is not generated in the following conditions:
   - OFS0.IWDTSTRT = 0 and OFS0.IWDTSTPCTL = 1.
4. Canceling by 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 11, Interrupt Controller Unit (ICU).

9.7 Software Standby Mode

9.7.1 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. Table 9.2 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 Table 9.3 for available interrupt sources and section 11.2.14. SBYEDCR0 : Software Standby/Snooze End Control Register 0 and section 11.2.15. SBYEDCR1 : Software Standby/Snooze End Control Register 1 for information on how to wake up the MCU from Software Standby mode. For details, see section 11, 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).

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.

9.7.2 Canceling Software Standby Mode

Software Standby mode is canceled by:

- An available interrupt shown in Table 9.3
- A RES pin reset
- A power-on reset
- A voltage monitor reset
- A reset caused by an IWDT underflow.

After exiting Software Standby mode, the oscillators that were operating before the transition restart. After the oscillator set as the ICLK source clock has stabilized, the MCU returns from Software standby mode to normal mode. Oscillators that are not set as the ICLK source should wait for stabilization before using them. See section 11.2.14. SBYEDCR0 : Software Standby/Snooze End Control Register 0 and section 11.2.15. SBYEDCR1 : Software Standby/Snooze End Control Register 1 for information on how to wake up the MCU from Software Standby mode.

You can cancel Software Standby mode in any of the following ways:

1. Canceling by an interrupt
   When an available interrupt request (see Table 9.3) is generated, an oscillator that operates before the transition to Software Standby mode restarts. After all the oscillators are stabilized, the MCU returns to Normal mode from Software Standby mode and starts the interrupt handling.
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 31, 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.

9.7.3 Example of Software Standby Mode Application

Figure 9.6 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) in Normal mode, and the IRQCRi.IRQMD[1:0] bits are set to 01b (rising edge). After that, the SBYCR.SSBY bit is set to 1 and a WFI instruction is executed. As a result, entry to Software Standby mode completes and exit from Software Standby mode is initiated by a rising edge of the IRQn pin.

Setting the ICU is also required to exit Software Standby mode. For details, see section 11, Interrupt Controller Unit (ICU). The oscillation stabilization time in Figure 9.6 is specified in section 31, Electrical Characteristics.

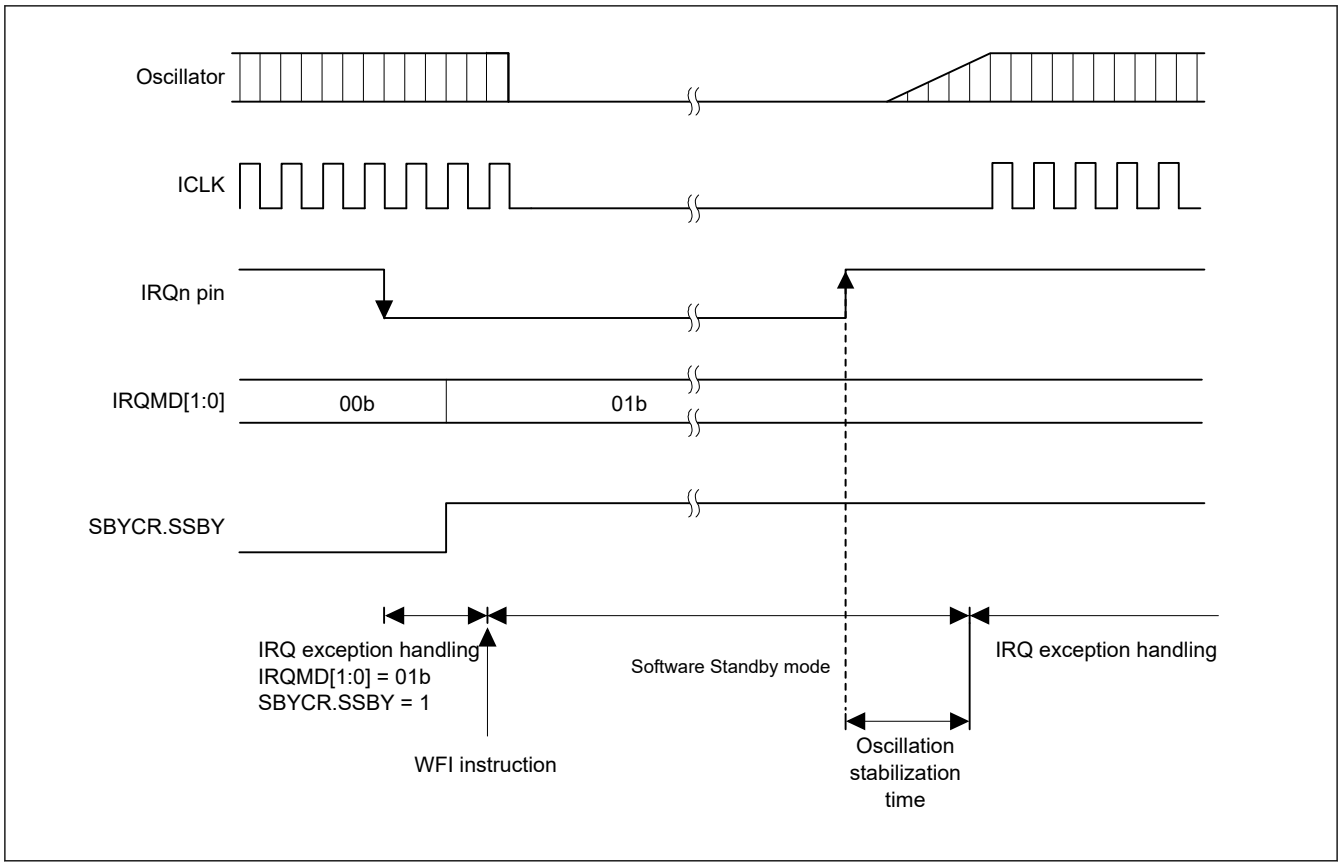


Figure 9.6 Example of Software Standby mode application

9.8 Snooze Mode

9.8.1 Transition to Snooze Mode

Figure 9.7 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 9.2 shows the peripheral modules that can operate in Snooze mode.

This can only be specified when the high-speed on-chip oscillator or middle-speed on-chip oscillator is selected for the CPU/peripheral hardware clock (ICLK).

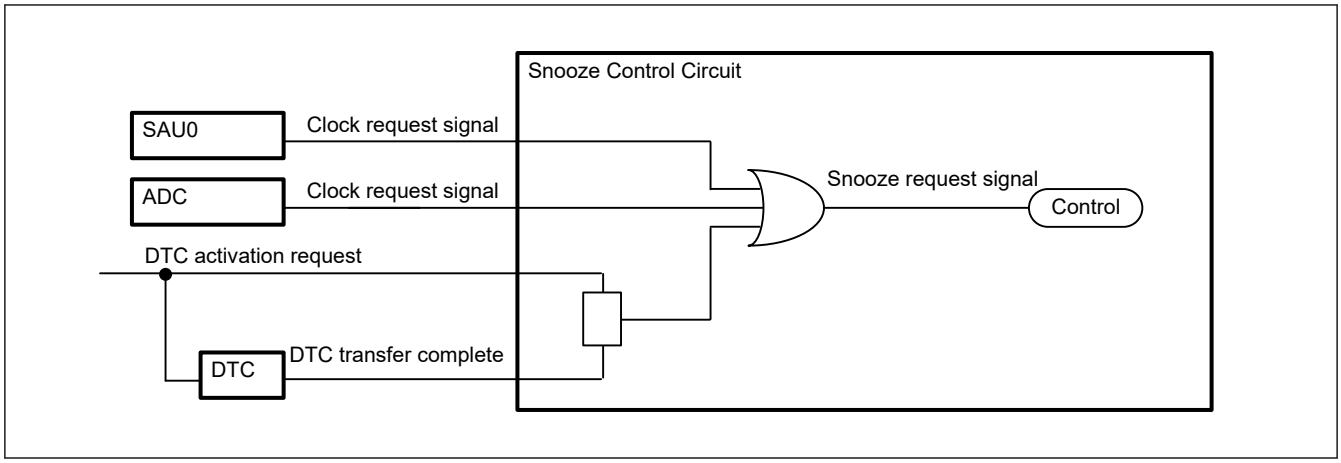


Figure 9.7 Snooze mode entry configuration

Table 9.8 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.

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

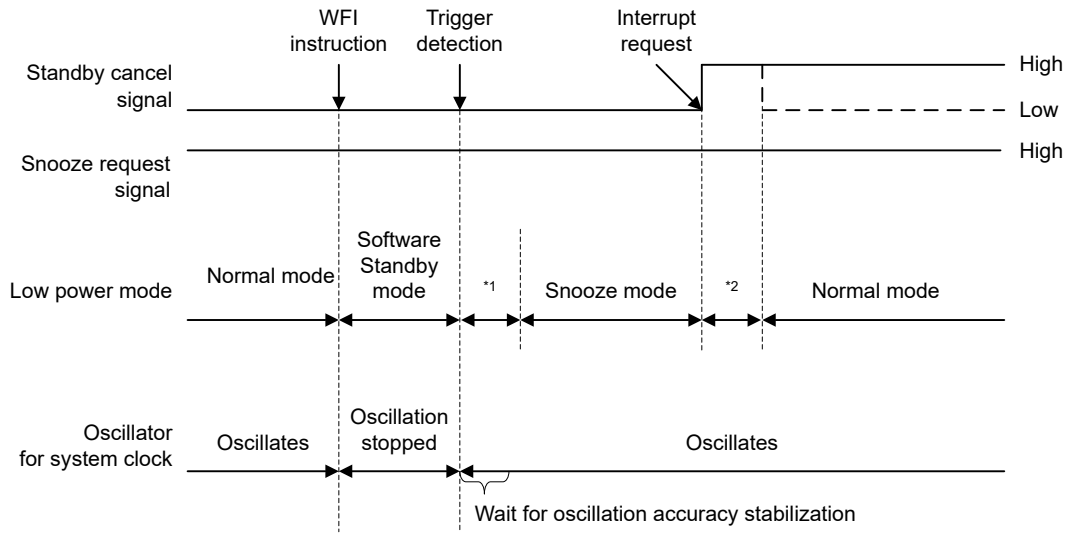
Table 9.8 Available Snooze requests to switch to Snooze mode

Snooze request output source	Control Register	
	Register	Bit
SAU0	SSC0	SWC
ADC12	ADM2	AWC
ICU (for DTC)	DTCENSTn	STm

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.

9.8.2 Canceling Snooze Mode

Snooze mode is canceled by an interrupt request that is available in Software Standby mode or a reset. Table 9.3 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.

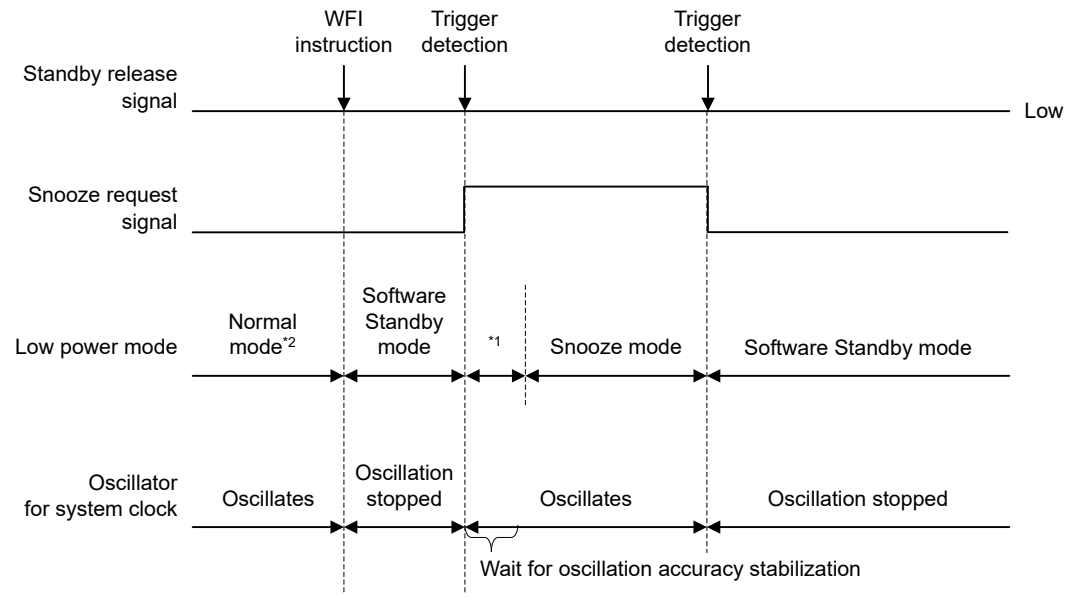


Note 1. Transition time from Software Standby mode to Snooze mode.
Note 2. Transition time from Snooze mode to Normal mode.

Figure 9.8 Canceling of Snooze mode when an interrupt request signal is generated

9.8.3 Returning from Snooze Mode to Software Standby Mode

Figure 9.9 shows the timing diagram for the transition from Snooze mode to Software Standby mode. This mode transition occurs cleared a Snooze request



Note 1. Transition time from Software Standby mode to Snooze mode.
Note 2. Enable Snooze mode immediately before switching to Software Standby mode.

Figure 9.9 Canceling of Snooze mode when an interrupt request signal is not generated

9.8.4 Snooze Operation Example

Figure 9.10 shows an example setting for using DTC in Snooze mode.

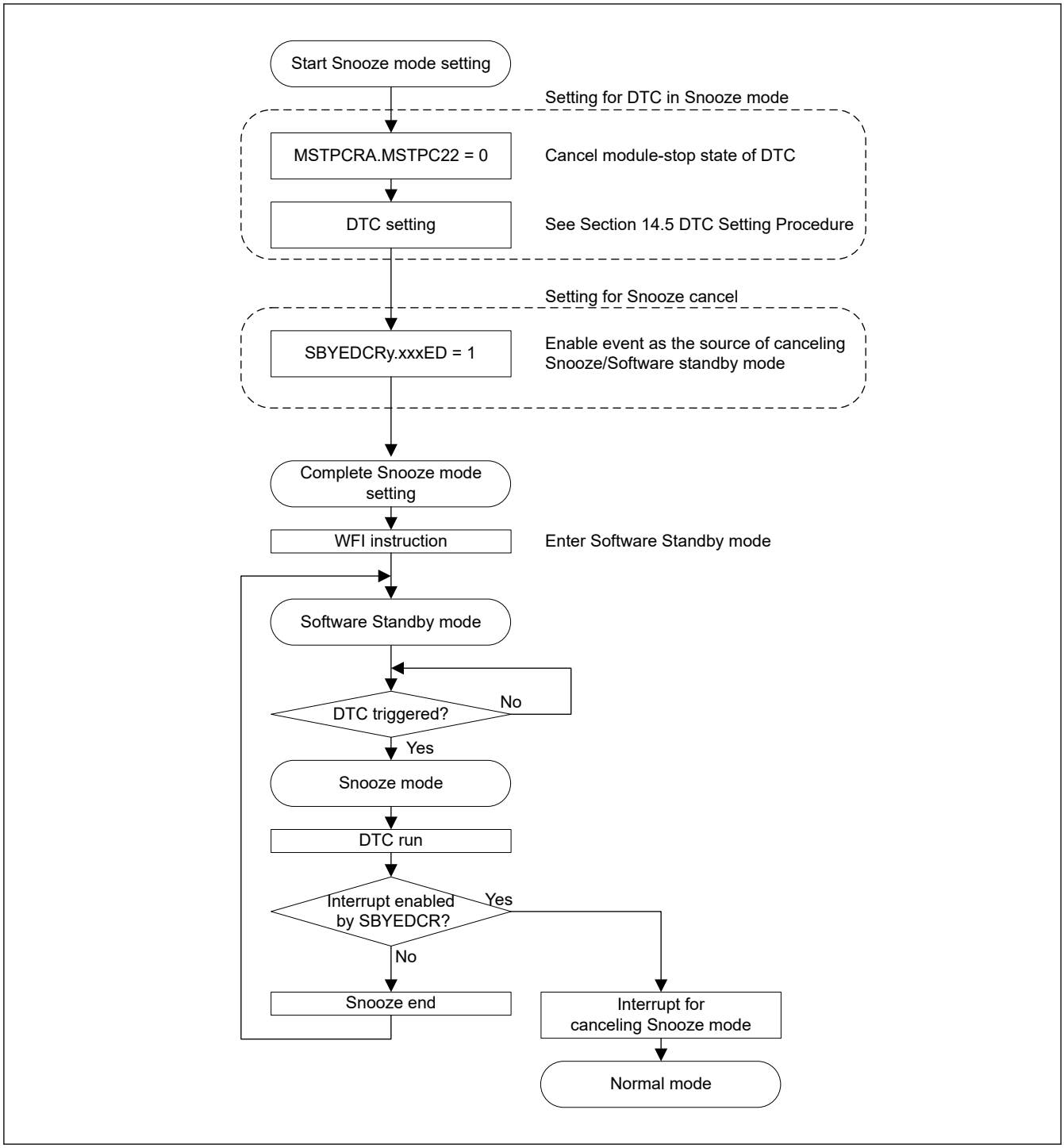


Figure 9.10 Setting example of using DTC in Snooze mode

9.9 Usage Notes

9.9.1 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 and the FLMODE register

[Conditions]

- 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 9.9 shows 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 9.9 Valid settings for the clock-related registers

Mode	Valid settings					
	ICLKSCR.CKSEL	HOCOCR.HCSTP	MOCOCR.MCSTP	LOCOCR.LCSTP	MOSCCR.MOSTP	SOSCCR.SOSTP
Subosc-speed	1 (LOCO or SOSC)	1 (stopped)	1 (stopped)	0 (operating) 1 (stopped)	1 (stopped)	0 (operating) 1 (stopped)

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

[Registers]

- FLMODE

[Condition]

- ICLKSCR.CKSEL = 1 (ICLK = LOCO or SOSC).
- DFLCTL.DFLEN = 0 (Data flash is disabled)

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

[Registers]

- MSTPCRA, MSTPCRB, MSTPCRC, MSTPCRD, FLMODE

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

[Registers]

- SBYCR, PSMCR, SYOCDCR.

9.9.2 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.

9.9.3 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).

9.9.4 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.

9.9.5 Transitioning to Low Power Modes

Because the MCU does not support wakeup by events, do not enter the low power modes such as Sleep mode, Software Standby mode by executing a WFE instruction. Also, do not set the SLEEPDEEP bit of the System Control Register in the Cortex-M23 core because the MCU does not support low power modes by SLEEPDEEP.

9.9.6 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.

9.9.7 Writing to the IWDT Registers by DTC in Sleep Mode or Snooze Mode

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

9.9.8 Oscillators in Snooze Mode

ICLK sourced oscillator that stop on entering Software Standby mode automatically restart when a trigger for switching to Snooze mode is generated. Other oscillators maintain their state in Software standby mode.

9.9.9 Using SAU0 in Snooze Mode

A transition to Software Standby mode must not occur during an SAU0 communication.

9.9.10 Using UART0 in Snooze Mode

When using UART0 in Snooze mode, ensure that the Snooze request (RXD0 falling edge) does not conflict with the wakeup requests set by the SBYEDCRn register, otherwise UART cannot be guaranteed.

When using UART in Snooze mode, the following conditions must be satisfied:

- The clock source must be HOCO.
- The RXD0 pin must be kept high before entering Software Standby mode.

9.9.11 Sleep-on-exit Function

There are 2 ways to transition to low power modes. One is WFI instruction and the other is Sleep-on-exit. When Sleep-on-exit is used for transition to low power modes, WFI instruction comments written in User’s Manual Hardware are applicable to Sleep-on-exit.

10. Register Write Protection

10.1 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 10.1 lists the association between the bits in the PRCR register and the registers to be protected.

Table 10.1 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: CMC, SOMRG, FOCOSCR, FMAINSCR, FSUBSCR, ICLKSCR, MOSCCR, SOSCCR, LOCOCR, HOCOCR, MOCOCR, OSTS, HOCODIV, MOCODIV, MOSCDIV, OSMC, LIOTRM, MIOTRM
PRC1	• Registers related to the low power modes: SBYCR, PSMCR, SYOCDCR
PRC3	• Registers related to the LVD: LVD1CR, LVD1MKR, LVD1SR

10.2 Register Descriptions

10.2.1 PRCR : Protect Register

Base address: SYSC = 0x4001_E000

Offset address: 0x08FE

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 10.1. Setting the PRCn bits to 1 or 0 enables or disables writing, respectively.

11. Interrupt Controller Unit (ICU)

11.1 Overview

The Interrupt Controller Unit (ICU) controls which event signals are linked to the Nested Vector Interrupt Controller (NVIC), and the Data Transfer Controller (DTC) modules. The ICU also controls non-maskable interrupts.

Table 11.1 lists the ICU specifications, Figure 11.1 shows a block diagram, and Table 11.2 lists the I/O pins.

Table 11.1 ICU specifications

Item		Description
Maskable interrupts	Peripheral function interrupts	• Interrupts from peripheral modules Number of sources: 37
	External pin interrupts	• Interrupt detection on falling edge, rising edge, rising and falling edges. One of these detection methods can be set for each source • 8 sources, with interrupts from IRQi (i = 0 to 7) pins.
	Interrupt requests to CPU (NVIC)	• 45 interrupt requests are output to NVIC.
	DTC control	• The DTC can be activated using interrupt sources*1 • The method for selecting an interrupt source is the same as that of the interrupt request to NVIC.
Non-maskable interrupts*2	NMI pin interrupt	• Interrupt from the NMI pin • Interrupt detection on falling edge or rising edge
	IWDT underflow/refresh error*3	Interrupt on an underflow of the down-counter or occurrence of a refresh error
	Low voltage detection 1*3	Voltage monitor 1 interrupt of the voltage monitor 1 circuit (LVD_LVD1)
	RPEST	Interrupt on SRAM parity error
Low power modes		• Sleep mode: return is initiated by non-maskable interrupts or any other interrupt source • Software Standby mode: return is initiated by non-maskable interrupts or any other interrupt source. Interrupt can be selected in the SBYEDCRn register. • Snooze mode: return is initiated by non-maskable interrupts or any other interrupt source. Interrupt can be selected in the SBYEDCRn register. See section 11.2.14. SBYEDCR0 : Software Standby/Snooze End Control Register 0 and section 11.2.15. SBYEDCR1 : Software Standby/Snooze End Control Register 1.

Note 1. For the DTC activation sources, see Table 11.5.
Note 2. Non-maskable interrupts can be enabled only once after a reset release.
Note 3. These non-maskable interrupts can also be used as maskable interrupts. When used as maskable interrupts, do not change the value of the NMIER register from the reset state. For voltage monitor 1 interrupts, additionally set the LVD1CR.IRQSEL bits to 1.

Figure 11.1 shows the ICU block diagram.

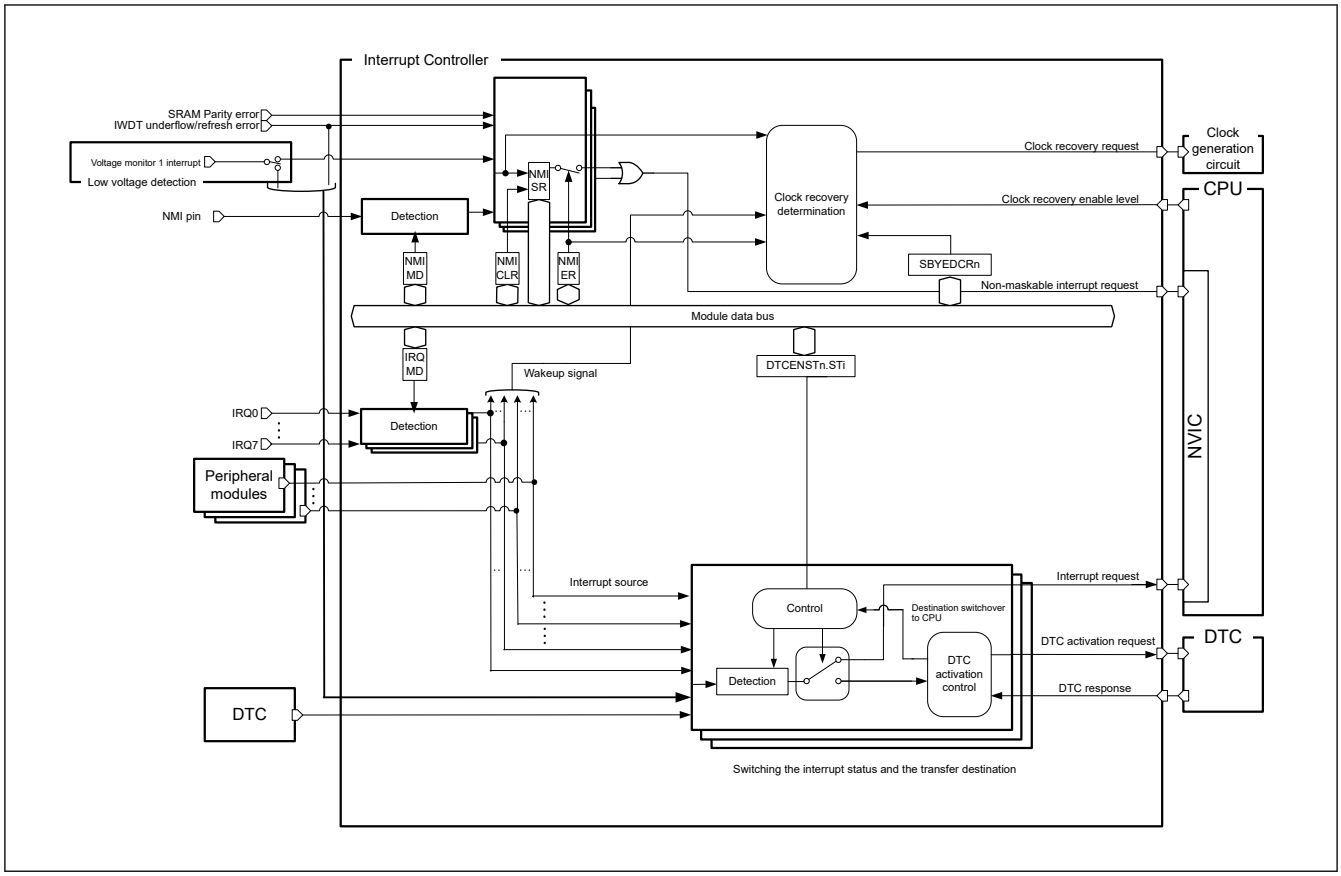


Figure 11.1 ICU block diagram

Table 11.2 lists the ICU input/output pins.

Table 11.2 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

11.2 Register Descriptions

This chapter does not describe the Arm® NVIC internal registers. For information about these registers, see ARM® Cortex®-M23 Processor Technical Reference Manual (ARM DDI 0550C).

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

Base address: ICU = 0x4000_6000

Offset address: 0x0000 + i

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	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: Setting prohibited	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W

IRQCRi register changes must satisfy the following conditions:

- For a wakeup enable signal:
  Change the IRQCRi register setting before setting the target SBYEDCR0.IRQnED (n = 0 to 7). The register value should be changed when the target SBYEDCR0.IRQnED 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 11.5.5. External Pin Interrupts.

11.2.2 NMISR : Non-maskable Interrupt Status Register

Base address: ICU = 0x4000_6000

Offset address: 0x0140

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	RPEST	NMIST	—	—	—	—	LVD1ST	—	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	—	This bit is read as 0.	R
2	LVD1ST	Voltage Monitor 1 Interrupt Status Flag 0: Interrupt not requested 1: Interrupt requested	R
6:3	—	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
15:9	—	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 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.

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.LVDCLR 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.

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.

11.2.3 NMIER : Non-maskable Interrupt Enable Register

Base address: ICU = 0x4000_6000

Offset address: 0x0120

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	RPEEN	NMIEN	—	—	—	—	LVD1EN	—	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	—	This bit is read as 0. The write value should be 0.	R/W
2	LVD1EN	Voltage Monitor 1 Interrupt Enable 0: Disabled 1: Enabled	R/W*1 *2
6:3	—	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
15:9	—	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 an event signal.

IWDTEN bit (IWDT Underflow/Refresh Error Interrupt Enable)

The IWDTEN bit enables IWDT 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.

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.

11.2.4 NMICLR : Non-maskable Interrupt Status Clear Register

Base address: ICU = 0x4000_6000

Offset address: 0x0130

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	RPECLR	NMICLR	—	—	—	—	LVD1CLR	—	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	—	This bit is read as 0. The write value should be 0.	R/W
2	LVD1CLR	Voltage Monitor 1 Interrupt Status Flag Clear 0: No effect 1: Clear the NMISR.LVD1ST flag	R/W*1
6:3	—	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
15:9	—	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.

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.

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.

11.2.5 NMICR : NMI Pin Interrupt Control Register

Base address: ICU = 0x4000_6000

Offset address: 0x0100

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	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
7:1	—	These bits are read as 0. The write value should be 0.	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.

11.2.6 DTCENST0 : DTC Enable Status Register 0

Base address: ICU = 0x4000_6000

Offset address: 0x0300

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	ST31	ST30	ST29	ST28	ST27	—	—	—	ST23	ST22	—	—	ST19	ST18	—	ST16
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:	ST15	—	ST13	ST12	—	—	ST9	ST8	ST7	ST6	ST5	ST4	ST3	ST2	ST1	—
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.	R
9:1	ST9 to ST1	DTC Enable Status by Event Number i The suffix number of each bit symbol corresponds to the DTC vector number i. 0: DTC Disable by Event number i 1: DTC Enable by Event number i	R
11:10	—	These bits are read as 0.	R
13:12	ST13 to ST12	DTC Enable Status by Event Number i The suffix number of each bit symbol corresponds to the DTC vector number i. 0: DTC Disable by Event number i 1: DTC Enable by Event number i	R
14	—	This bit is read as 0.	R
16:15	ST16 to ST15	DTC Enable Status by Event Number i The suffix number of each bit symbol corresponds to the DTC vector number i. 0: DTC Disable by Event number i 1: DTC Enable by Event number i	R
17	—	This bit is read as 0.	R

Bit	Symbol	Function	R/W
19:18	ST19 to ST18	DTC Enable Status by Event Number i The suffix number of each bit symbol corresponds to the DTC vector number i. 0: DTC Disable by Event number i 1: DTC Enable by Event number i	R
21:20	—	These bits are read as 0.	R
23:22	ST23 to ST22	DTC Enable Status by Event Number i The suffix number of each bit symbol corresponds to the DTC vector number i. 0: DTC Disable by Event number i 1: DTC Enable by Event number i	R
26:24	—	These bits are read as 0.	R
31:27	ST31 to ST27	DTC Enable Status by Event Number i The suffix number of each bit symbol corresponds to the DTC vector number i. 0: DTC Disable by Event number i 1: DTC Enable by Event number i	R

STi bits (DTC Enable Status by Event Number i) (i = 1 to 9, 12 to 13, 15 to 16, 18 to 19, 22 to 23, 27 to 31)

The STi bit indicates whether the corresponding event is disabled or enabled as a DTC activation factor. This register is read-only and is set by the DTCENSETn.SETi bit and cleared by the DTCENCLRn.CLRi bit.

After check the DTC transfer end (DTCENSTn.STi = 0), stop the DTC module by setting MSTPCRA.MSTPA22 or DTCST.DTCST register.

[Setting condition]

- When 1 is written to the DTCENSETn.SETi bit.

[Clearing condition]

- When 1 is written to the DTCENCLRn.CLRi bit.
- When the specified number of transfers is complete. For chain transfers, when the specified number of transfers for the last chain transfer is complete.

11.2.7 DTCENST1 : DTC Enable Status Register 1

Base address: ICU = 0x4000_6000

Offset address: 0x0304

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:	—	—	ST45	ST44	—	—	ST41	ST40	—	ST38	ST37	ST36	ST35	ST34	ST33	ST32
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
6:0	ST38 to ST32	DTC Enable Status by Event Number i The suffix number of each bit symbol corresponds to the DTC vector number i. 0: DTC Disable by Event number i 1: DTC Enable by Event number i	R
7	—	This bit is read as 0.	R
9:8	ST41 to ST40	DTC Enable Status by Event Number i The suffix number of each bit symbol corresponds to the DTC vector number i. 0: DTC Disable by Event number i 1: DTC Enable by Event number i	R

Bit	Symbol	Function	R/W
11:10	—	These bits are read as 0.	R
13:12	ST45 to ST44	DTC Enable Status by Event Number i The suffix number of each bit symbol corresponds to the DTC vector number i. 0: DTC Disable by Event number i 1: DTC Enable by Event number i	R
31:14	—	These bits are read as 0.	R

STi bits (DTC Enable Status by Event Number i) (i = 32 to 38, 40 to 41, 44 to 45)

The STi bit indicates whether the corresponding event is disabled or enabled as a DTC activation factor. This register is read-only and is set by the DTCENSETn.SETi bit and cleared by the DTCENCLRn.CLRi bit.

After check the DTC transfer end (DTCENSTn.STi = 0), stop the DTC module by setting MSTPCRA.MSTPA22 or DTCST.DTCST register.

[Setting condition]

- When 1 is written to the DTCENSETn.SETi bit.

[Clearing condition]

- When 1 is written to the DTCENCLRn.CLRi bit.
- When the specified number of transfers is complete. For chain transfers, when the specified number of transfers for the last chain transfer is complete.

11.2.8 DTCENSET0 : DTC Enable Set Register 0

Base address: ICU = 0x4000_6000

Offset address: 0x0310

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	SET31	SET30	SET29	SET28	SET27	—	—	—	SET23	SET22	—	—	SET19	SET18	—	SET16
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:	SET15	—	SET13	SET12	—	—	SET9	SET8	SET7	SET6	SET5	SET4	SET3	SET2	SET1	—
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
9:1	SET9 to SET1	DTC Enable Set by Event Number i The suffix number of each bit symbol corresponds to the DTC vector number i. 0: No effect 1: DTC Enable by Event number i	R/W
11:10	—	These bits are read as 0. The write value should be 0.	R/W
13:12	SET13 to SET12	DTC Enable Set by Event Number i The suffix number of each bit symbol corresponds to the DTC vector number i. 0: No effect 1: DTC Enable by Event number i	R/W
14	—	This bit is read as 0. The write value should be 0.	R/W
16:15	SET16 to SET15	DTC Enable Set by Event Number i The suffix number of each bit symbol corresponds to the DTC vector number i. 0: No effect 1: DTC Enable by Event number i	R/W
17	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
19:18	SET19 to SET18	DTC Enable Set by Event Number i The suffix number of each bit symbol corresponds to the DTC vector number i. 0: No effect 1: DTC Enable by Event number i	R/W
21:20	—	These bits are read as 0. The write value should be 0.	R/W
23:22	SET23 to SET22	DTC Enable Set by Event Number i The suffix number of each bit symbol corresponds to the DTC vector number i. 0: No effect 1: DTC Enable by Event number i	R/W
26:24	—	These bits are read as 0. The write value should be 0.	R/W
31:27	SET31 to SET27	DTC Enable Set by Event Number i The suffix number of each bit symbol corresponds to the DTC vector number i. 0: No effect 1: DTC Enable by Event number i	R/W

SETi bits (DTC Enable Set by Event Number i) (i = 1 to 9, 12 to 13, 15 to 16, 18 to 19, 22 to 23, 27 to 31)

By writing 1 to the SETi bit, the corresponding event is selected as the DTC activation source. Writing 0 has no effect. It reads as 0.

11.2.9 DTCENSET1 : DTC Enable Set Register 1

Base address: ICU = 0x4000_6000

Offset address: 0x0314

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:	—	—	SET45	SET44	—	—	SET41	SET40	—	SET38	SET37	SET36	SET35	SET34	SET33	SET32
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
6:0	SET38 to SET32	DTC Enable Set by Event Number i The suffix number of each bit symbol corresponds to the DTC vector number i. 0: No effect 1: DTC Enable by Event number i	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
9:8	SET41 to SET40	DTC Enable Set by Event Number i The suffix number of each bit symbol corresponds to the DTC vector number i. 0: No effect 1: DTC Enable by Event number i	R/W
11:10	—	These bits are read as 0. The write value should be 0.	R/W
13:12	SET45 to SET44	DTC Enable Set by Event Number i The suffix number of each bit symbol corresponds to the DTC vector number i. 0: No effect 1: DTC Enable by Event number i	R/W
31:14	—	These bits are read as 0. The write value should be 0.	R/W

SETi bits (DTC Enable Set by Event Number i) (i = 32 to 38, 40 to 41, 44 to 45)

By writing 1 to the SETi bit, the corresponding event is selected as the DTC activation source. Writing 0 has no effect. It reads as 0.

11.2.10 DTCENCLR0 : DTC Enable Clear Register 0

Base address: ICU = 0x4000_6000

Offset address: 0x0320

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CLR31	CLR30	CLR29	CLR28	CLR27	—	—	—	CLR23	CLR22	—	—	CLR19	CLR18	—	CLR16
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:	CLR15	—	CLR13	CLR12	—	—	CLR9	CLR8	CLR7	CLR6	CLR5	CLR4	CLR3	CLR2	CLR1	—
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
9:1	CLR9 to CLR1	DTC Enable Clear by Event Number i The suffix number of each bit symbol corresponds to the DTC vector number i. 0: No effect 1: DTC Disable by Event number i	R/W
11:10	—	These bits are read as 0. The write value should be 0.	R/W
13:12	CLR13 to CLR12	DTC Enable Clear by Event Number i The suffix number of each bit symbol corresponds to the DTC vector number i. 0: No effect 1: DTC Disable by Event number i	R/W
14	—	This bit is read as 0. The write value should be 0.	R/W
16:15	CLR16 to CLR15	DTC Enable Clear by Event Number i The suffix number of each bit symbol corresponds to the DTC vector number i. 0: No effect 1: DTC Disable by Event number i	R/W
17	—	This bit is read as 0. The write value should be 0.	R/W
19:18	CLR19 to CLR18	DTC Enable Clear by Event Number i The suffix number of each bit symbol corresponds to the DTC vector number i. 0: No effect 1: DTC Disable by Event number i	R/W
21:20	—	These bits are read as 0. The write value should be 0.	R/W
23:22	CLR23 to CLR22	DTC Enable Clear by Event Number i The suffix number of each bit symbol corresponds to the DTC vector number i. 0: No effect 1: DTC Disable by Event number i	R/W
26:24	—	These bits are read as 0. The write value should be 0.	R/W
31:27	CLR31 to CLR27	DTC Enable Clear by Event Number i The suffix number of each bit symbol corresponds to the DTC vector number i. 0: No effect 1: DTC Disable by Event number i	R/W

CLRi bits (DTC Enable Clear by Event Number i)(i = 1 to 9, 12 to 13, 15 to 16, 18 to 19, 22 to 23, 27 to 31)

Writing 1 to the CLRi bit disables DTC activation by the corresponding event. Writing 0 has no effect. It reads as 0.

11.2.11 DTCENCLR1 : DTC Enable Clear Register 1

Base address: ICU = 0x4000_6000

Offset address: 0x0324

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:	—	—	CLR45	CLR44	—	—	CLR41	CLR40	—	CLR38	CLR37	CLR36	CLR35	CLR34	CLR33	CLR32
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
6:0	CLR38 to CLR32	DTC Enable Clear by Event Number i The suffix number of each bit symbol corresponds to the DTC vector number i. 0: No effect 1: DTC Disable by Event number i	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
9:8	CLR41 to CLR40	DTC Enable Clear by Event Number i The suffix number of each bit symbol corresponds to the DTC vector number i. 0: No effect 1: DTC Disable by Event number i	R/W
11:10	—	These bits are read as 0. The write value should be 0.	R/W
13:12	CLR45 to CLR44	DTC Enable Clear by Event Number i The suffix number of each bit symbol corresponds to the DTC vector number i. 0: No effect 1: DTC Disable by Event number i	R/W
31:14	—	These bits are read as 0. The write value should be 0.	R/W

CLRi bits (DTC Enable Clear by Event Number i) (i = 32 to 38, 40 to 41, 44 to 45)

Writing 1 to the CLRi bit disables DTC activation by the corresponding event. Writing 0 has no effect. It reads as 0.

11.2.12 INTFLAG0 : Interrupt Request Flag Monitor Register 0

Base address: ICU = 0x4000_6000

Offset address: 0x0330

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	IF31	IF30	IF29	IF28	IF27	IF26	IF25	IF24	IF23	IF22	IF21	IF20	IF19	IF18	IF17	IF16
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:	IF15	IF14	IF13	IF12	IF11	IF10	IF9	IF8	IF7	IF6	IF5	IF4	IF3	IF2	IF1	IF0
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	IF31 to IF0	Interrupt Request Flag Monitor 0: Interrupt request of event number i is not being accepted by the ICU 1: Interrupt request of event number i is being accepted by the ICU	R

IFi flags (Interrupt Request Flag Monitor) (i = 0 to 31)

The IFi flag indicates whether an interrupt request or a DTC request of the event i is being accepted by the ICU or not. This register is read-only.

[Setting condition]

- When an interrupt request or a DTC activation request occurs.

[Clearing condition]

- When the ICU notifies the interrupt request to the NVIC.
- When a DTC transfer that does not notify the CPU of an interrupt started.
  - MRB.DISEL = 0 and Remaining transfer operations ≠ 0
- When a DTC transfer finished.
  - MRB.DISEL = 0 and Remaining transfer operations = 0
  - MRB.DISEL = 1
- When 1 is written to the DTCENSETn.SETi bit or DTCENCLRn.CLRi bit.

To clear the interrupt request flag, write 1 to the DTCENSETn.SETi bit or DTCENCLRn.CLRi bit.

By using the INTFLAG0 register and the DTCENST0 register, pending DTC requests can be confirmed. If the DTCENST0.STi = 1, the INTFLAG0.IFi = 1 and the DTCSTS.VECN[7:0] ≠ i, the DTC request of the event i is pending.

11.2.13 INTFLAG1 : Interrupt Request Flag Monitor Register 1

Base address: ICU = 0x4000_6000

Offset address: 0x0334

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:	—	—	IF45	IF44	IF43	IF42	IF41	IF40	IF39	IF38	IF37	IF36	IF35	IF34	IF33	IF32
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	IF45 to IF32	Interrupt Request Flag Monitor 0: Interrupt source of event number i is not accepted by the ICU 1: Interrupt source of event number i is accepted by the ICU	R
31:14	—	These bits are read as 0.	R

IFi flags (Interrupt Request Flag Monitor) (i = 32 to 45)

The IFi flag indicates whether an interrupt request or a DTC request of the event i is being accepted by the ICU or not. This register is read-only.

[Setting condition]

- When an interrupt request or the DTC activation request occurs.

[Clearing condition]

- When the ICU notifies the interrupt request to the NVIC.
- When a DTC transfer that does not notify the CPU of an interrupt started.
  - MRB.DISEL = 0 and Remaining transfer operations ≠ 0
- When a DTC transfer finished.

– MRB.DISEL = 0 and Remaining transfer operations = 0
– MRB.DISEL = 1

- When 1 is written to the DTCENSETn.SETi bit or DTCENCLRn.CLRi bit.

To clear the interrupt request flag, write 1 to the DTCENSETn.SETi bit or DTCENCLRn.CLRi bit.

By using the INTFLAG1 register and the DTCENST1 register, pending DTC requests can be confirmed. If the DTCENST1.STi = 1, the INTFLAG1.IFi = 1 and the DTCSTS.VECN[7:0] ≠ i, the DTC request of the event i is pending.

11.2.14 SBYEDCR0 : Software Standby/Snooze End Control Register 0

Base address: ICU = 0x4000_6000

Offset address: 0x0340

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	ADC12ED	—	—	—	UART0RXED	IICA0ED	—	—	—	—	—	UART0ERRED	—	SPI00RXED	—	—
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:	—	—	—	—	—	DTCED	IRQ7ED	IRQ6ED	IRQ5ED	IRQ4ED	IRQ3ED	IRQ2ED	IRQ1ED	IRQ0ED	LVD1ED	IWDTED
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	IWDTED	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
1	LVD1ED	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
9:2	IRQiED (i = 7 to 0)	IRQ Interrupt Software Standby/Snooze Mode Returns Enable The suffix number of each bit symbol corresponds to the IRQ number i. 0: Software Standby/Snooze Mode returns by IRQi interrupt disabled 1: Software Standby/Snooze Mode returns by IRQi interrupt enabled	R/W
10	DTCED	DTC Transfer Complete Interrupt Snooze Mode Returns Enable 0: Snooze Mode returns by DTC transfer complete interrupt disabled 1: Snooze Mode returns by DTC transfer complete interrupt enabled	R/W
17:11	—	These bits are read as 0. The write value should be 0.	R/W
18	SPI00RXED	SPI00 Transfer End or Buffer Empty Interrupt Snooze Mode Returns Enable 0: Snooze Mode returns by SPI00 transfer end or buffer empty interrupt disabled 1: Snooze Mode returns by SPI00 transfer end or buffer empty interrupt enabled	R/W
19	—	This bit is read as 0. The write value should be 0.	R/W
20	UART0ERRED	UART0 Reception Communication Error Occurrence Interrupt Snooze Mode Returns Enable 0: Snooze Mode returns by UART0 reception communication error occurrence interrupt disabled 1: Snooze Mode returns by UART0 reception communication error occurrence interrupt enabled	R/W
25:21	—	These bits are read as 0. The write value should be 0.	R/W
26	IICA0ED	IICA0 Address Match Interrupt Software Standby/Snooze Mode Returns Enable 0: Software Standby/Snooze Mode returns by IICA0 address match interrupt disabled 1: Software Standby/Snooze Mode returns by IICA0 address match interrupt enabled	R/W

Bit	Symbol	Function	R/W
27	UART0RXED	UART0 Reception Transfer End Interrupt Snooze Mode Returns Enable 0: Snooze Mode returns by UART0 reception transfer end interrupt disabled 1: Snooze Mode returns by UART0 reception transfer end interrupt enabled	R/W
30:28	—	These bits are read as 0. The write value should be 0.	R/W
31	ADC12ED	End of A/D Conversion Interrupt Snooze Mode Returns Enable 0: Snooze Mode returns by End of A/D conversion interrupt disabled 1: Snooze Mode returns by End of A/D conversion interrupt enabled	R/W

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

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

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

LVD1ED bit (LVD1 Interrupt Software Standby/Snooze Mode Returns Enable)

The LVD1ED bit enables the use of LVD1 interrupts to cancel Software Standby/Snooze mode.

IRQiED bits (IRQ Interrupt Software Standby/Snooze Mode Returns Enable) (i = 0 to 7)

The IRQiED bits enable the use of IRQn interrupts to cancel Software Standby/Snooze mode.

DTCED bit (DTC Transfer Complete Interrupt Snooze Mode Returns Enable)

The DTCED bits enable the use of DTC transfer complete interrupts to cancel Snooze mode.

SPI00RXED bit (SPI00 Transfer End or Buffer Empty Interrupt Snooze Mode Returns Enable)

The SPI00RXED bits enable the use of SPI00 transfer end or buffer empty interrupts to cancel Snooze mode.

UART0ERRED bit (UART0 Reception Communication Error Occurrence Interrupt Snooze Mode Returns Enable)

The UART0ERRED bits enable the use of UART0 reception communication error occurrence interrupts to cancel Snooze mode.

IICA0ED bit (IICA0 Address Match Interrupt Software Standby/Snooze Mode Returns Enable)

The IICA0ED bit enables the use of IICA0 interrupts to cancel Software Standby/Snooze mode.

UART0RXED bit (UART0 Reception Transfer End Interrupt Snooze Mode Returns Enable)

The UART0RXED bits enable the use of UART0 reception transfer end interrupts to cancel Snooze mode.

ADC12ED bit (End of A/D Conversion Interrupt Snooze Mode Returns Enable)

The ADC12ED bits enable the use of End of A/D conversion interrupts to cancel Snooze mode.

11.2.15 SBYEDCR1 : Software Standby/Snooze End Control Register 1

Base address: ICU = 0x4000_6000

Offset address: 0x0344

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:	—	—	UR1ED	UT1ED	URE1ED	IICA1ED	UR0ED	UT0ED	URE0ED	—	—	—	—	—	ITLED	RTCED
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	RTCED	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
1	ITLED	Interval Signal of 32-bit Interval Timer Interrupt Software Standby/Snooze Mode Returns Enable 0: Software Standby/Snooze Mode returns by Interval signal of 32-bit interval timer interrupt disabled 1: Software Standby/Snooze Mode returns by Interval signal of 32-bit interval timer interrupt enabled	R/W
6:2	—	These bits are read as 0. The write value should be 0.	R/W
7	URE0ED	UARTA0 Reception Communication Error Interrupt Software Standby/Snooze Mode Returns Enable 0: Software Standby/Snooze Mode returns by UARTA0 reception communication error interrupt disabled 1: Software Standby/Snooze Mode returns by UARTA0 reception communication error interrupt enabled	R/W
8	UT0ED	UARTA0 Transmission Transfer End or Buffer Empty Interrupt Software Standby/Snooze Mode Returns Enable 0: Software Standby/Snooze Mode returns by UARTA0 transmission transfer end or buffer empty interrupt disabled 1: Software Standby/Snooze Mode returns by UARTA0 transmission transfer end or buffer empty interrupt enabled	R/W
9	UR0ED	UARTA0 Reception Transfer End Interrupt Software Standby/Snooze Mode Returns Enable 0: Software Standby/Snooze Mode returns by UARTA0 reception transfer end interrupt disabled 1: Software Standby/Snooze Mode returns by UARTA0 reception transfer end interrupt enabled	R/W
10	IICA1ED	IICA1 Address Match Interrupt Software Standby/Snooze Mode Returns Enable 0: Software Standby/Snooze Mode returns by IICA1 address match interrupt disabled 1: Software Standby/Snooze Mode returns by IICA1 address match interrupt enabled	R/W
11	URE1ED	UARTA1 Reception Communication Error Interrupt Software Standby/Snooze Mode Returns Enable 0: Software Standby/Snooze Mode returns by UARTA1 reception communication error interrupt disabled 1: Software Standby/Snooze Mode returns by UARTA1 reception communication error interrupt enabled	R/W
12	UT1ED	UARTA1 Transmission Transfer End or Buffer Empty Interrupt Software Standby/Snooze Mode Returns Enable 0: Software Standby/Snooze Mode returns by UARTA1 transmission transfer end or buffer empty interrupt disabled 1: Software Standby/Snooze Mode returns by UARTA1 transmission transfer end or buffer empty interrupt enabled	R/W
13	UR1ED	UARTA1 Reception Transfer End Interrupt Software Standby/Snooze Mode Returns Enable 0: Software Standby/Snooze Mode returns by UARTA1 reception transfer end interrupt disabled 1: Software Standby/Snooze Mode returns by UARTA1 reception transfer end interrupt enabled	R/W
31:14	—	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.

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

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

ITLED bit (Interval Signal of 32-bit Interval Timer Interrupt Software Standby/Snooze Mode Returns Enable)

The ITLED bit enables the use of Interval signal of 32-bit interval timer interrupts to cancel Software Standby/Snooze mode.

URE0ED bit (UARTA0 Reception Communication Error Interrupt Software Standby/Snooze Mode Returns Enable)

The URE0ED bit enables the use of the UARTA0 reception communication error interrupts to cancel Software Standby/Snooze mode.

UT0ED bit (UARTA0 Transmission Transfer End or Buffer Empty Interrupt Software Standby/Snooze Mode Returns Enable)

The UT0ED bit enables the use of UARTA0 transmission transfer end or buffer empty interrupts to cancel Software Standby/Snooze mode.

UR0ED bit (UARTA0 Reception Transfer End Interrupt Software Standby/Snooze Mode Returns Enable)

The UR0ED bit enables the use of UARTA0 reception transfer end interrupts to cancel Software Standby/Snooze mode.

IICA1ED bit (IICA1 Address Match Interrupt Software Standby/Snooze Mode Returns Enable)

The IICA1ED bit enables the use of IICA1 interrupts to cancel Software Standby/Snooze mode.

URE1ED bit (UARTA1 Reception Communication Error Interrupt Software Standby/Snooze Mode Returns Enable)

The URE1ED bit enables the use of UARTA1 reception communication error interrupts to cancel Software Standby/Snooze mode.

UT1ED bit (UARTA1 Transmission Transfer End or Buffer Empty Interrupt Software Standby/Snooze Mode Returns Enable)

The UT1ED bit enables the use of UARTA1 transmission transfer end or buffer empty interrupts to cancel Software Standby/Snooze mode.

UR1ED bit (UARTA1 Reception Transfer End Interrupt Software Standby/Snooze Mode Returns Enable)

The UR1ED bit enables the use of UARTA1 reception transfer end interrupts to cancel Software Standby/Snooze mode.

11.3 Vector Table

The ICU detects maskable and non-maskable interrupts. Interrupt priorities are set up in the Arm NVIC. For information about these registers, see section 11.9. Reference.

11.3.1 Interrupt Vector Table

Table 11.3 describes the interrupt vector table. The interrupt vector addresses conform to the NVIC specifications.

Table 11.3 Interrupt vector table (1 of 3)

Exception number	IRQ number	Vector offset	Source	Description
0	—	0x000	Arm	Initial stack pointer
1	—	0x004	Arm	Initial program counter (reset vector)
2	—	0x008	Arm	Non-Maskable Interrupt (IWDT Underflow/Refresh Error Interrupt, Voltage monitor 1 Interrupt, NMI Pin Interrupt, SRAM Parity Error Interrupt)
3	—	0x00C	Arm	Hard Fault
4	—	0x010	Arm	Reserved
5	—	0x014	Arm	Reserved
6	—	0x018	Arm	Reserved
7	—	0x01C	Arm	Reserved
8	—	0x020	Arm	Reserved
9	—	0x024	Arm	Reserved
10	—	0x028	Arm	Reserved

Table 11.3 Interrupt vector table (2 of 3)

Exception number	IRQ number	Vector offset	Source	Description
11	—	0x02C	Arm	Supervisor Call (SVCall)
12	—	0x030	Arm	Reserved
13	—	0x034	Arm	Reserved
14	—	0x038	Arm	Pendable request for system service (PendableSrvReq)
15	—	0x03C	Arm	System Tick Timer (SysTick)
16	0	0x040	IWDT_NMIUNDF	Watchdog timer interval
17	1	0x044	LVD_LVD1	Low voltage detection 1
18	2	0x048	PORT_IRQ0	External pin interrupt 0
19	3	0x04C	PORT_IRQ1	External pin interrupt 1
20	4	0x050	PORT_IRQ2	External pin interrupt 2
21	5	0x054	PORT_IRQ3	External pin interrupt 3
22	6	0x058	PORT_IRQ4	External pin interrupt 4
23	7	0x05C	PORT_IRQ5	External pin interrupt 5
24	8	0x060	PORT_IRQ6	External pin interrupt 6
25	9	0x064	PORT_IRQ7	External pin interrupt 7
26	10	0x068	DTC_COMPLETE	DTC transfer complete
27	11	0x06C	FLASH_FRDYI	Flash ready interrupt
28	12	0x070	SAU1_UART_TXI2/ SAU1_SPI_TXRXI20/ SAU1_IIC_TXRXI20	UART2 transmission transfer end or buffer empty interrupt/SPI20 transfer end or buffer empty interrupt/IIC20 transfer end
29	13	0x074	SAU1_UART_RXI2/ SAU1_SPI_TXRXI21/ SAU1_IIC_TXRXI21	UART2 reception transfer end/SPI21 transfer end or buffer empty interrupt/IIC21 transfer end
30	14	0x078	SAU1_UART_ERRI2	UART2 reception communication error occurrence
31	15	0x07C	ELC_SWEVT0	Software event 0
32	16	0x080	ELC_SWEVT1	Software event 1
33	17	0x084	TRNG_RDREQ	TRNG read request
34	18	0x088	SAU0_UART_TXI0/ SAU0_SPI_TXRXI00/ SAU0_IIC_TXRXI00	UART0 transmission transfer end or buffer empty interrupt/SPI00 transfer end or buffer empty interrupt/IIC00 transfer end
35	19	0x08C	TAU0_TMI00	End of timer channel 00 count or capture
36	20	0x090	SAU0_UART_ERRI0	UART0 reception communication error occurrence
37	21	0x094	TAU0_TMI01H	End of timer channel 01 count or capture (at higher 8-bit timer operation)
38	22	0x098	SAU0_UART_TXI1/ SAU0_SPI_TXRXI10/ SAU0_IIC_TXRXI10	UART1 transmission transfer end or buffer empty interrupt/SPI10 transfer end or buffer empty interrupt/IIC10 transfer end
39	23	0x09C	SAU0_UART_RXI1/ SAU0_SPI_TXRXI11/ SAU0_IIC_TXRXI11	UART1 reception transfer end/SPI11 transfer end or buffer empty interrupt/IIC11 transfer end
40	24	0x0A0	SAU0_UART_ERRI1	UART1 reception communication error occurrence
41	25	0x0A4	TAU0_TMI03H	End of timer channel 03 count or capture (at higher 8-bit timer operation)
42	26	0x0A8	IICA0_TXRXI	End of IICA0 communication

Table 11.3 Interrupt vector table (3 of 3)

Exception number	IRQ number	Vector offset	Source	Description
43	27	0x0AC	SAU0_UART_RXI0/ SAU0_SPI_TXRXI01/ SAU0_IIC_TXRXI01	UART0 reception transfer end/SPI01 transfer end or buffer empty interrupt/IIC01 transfer end
44	28	0x0B0	TAU0_TMI01	End of timer channel 01 count or capture (at 16-bit/lower 8-bit timer operation)
45	29	0x0B4	TAU0_TMI02	End of timer channel 02 count or capture
46	30	0x0B8	TAU0_TMI03	End of timer channel 03 count or capture (at 16-bit/lower 8-bit timer operation)
47	31	0x0BC	ADC12_ADI	End of A/D conversion
48	32	0x0C0	RTC_ALM_OR_PRD	Fixed-cycle signal of realtime clock/alarm match detection
49	33	0x0C4	TML32_ITL_OR	Interval signal of 32-bit interval timer detection
50	34	0x0C8	Reserved	Reserved
51	35	0x0CC	TAU0_TMI04	End of timer channel 04 count or capture
52	36	0x0D0	TAU0_TMI05	End of timer channel 05 count or capture
53	37	0x0D4	TAU0_TMI06	End of timer channel 06 count or capture
54	38	0x0D8	TAU0_TMI07	End of timer channel 07 count or capture
55	39	0x0DC	UARTA0_ERRI	UARTA0 reception communication error occurrence
56	40	0x0E0	UARTA0_TXI	UARTA0 transmission transfer end or buffer empty interrupt
57	41	0x0E4	UARTA0_RXI	UARTA0 reception transfer end
58	42	0x0E8	IICA1_TXRXI	End of IICA1 communication
59	43	0x0EC	UARTA1_ERRI	UARTA1 reception communication error occurrence
60	44	0x0F0	UARTA1_TXI	UARTA1 transmission transfer end or buffer empty interrupt
61	45	0x0F4	UARTA1_RXI	UARTA1 reception transfer end
62	46	0x0F8	Reserved	Reserved
63	47	0x0FC	Reserved	Reserved
64	48	0x100	Reserved	Reserved
65	49	0x104	Reserved	Reserved
66	50	0x108	Reserved	Reserved
67	51	0x10C	Reserved	Reserved
68	52	0x110	Reserved	Reserved
69	53	0x114	Reserved	Reserved
70	54	0x118	Reserved	Reserved
71	55	0x11C	Reserved	Reserved
72	56	0x120	Reserved	Reserved
73	57	0x124	Reserved	Reserved
74	58	0x128	Reserved	Reserved
75	59	0x12C	Reserved	Reserved
76	60	0x130	Reserved	Reserved
77	61	0x134	Reserved	Reserved
78	62	0x138	Reserved	Reserved
79	63	0x13C	Reserved	Reserved

11.3.2 Event Number

The following table lists heading details for Table 11.5, which describes each event number.

Table 11.4 Event number

Heading	Description
Interrupt request source	Name of the source generating the interrupt request
Name	Name of the interrupt
Connect to NVIC	" ✓" indicates the interrupt can be used as a CPU interrupt
Invoke DTC	" ✓" indicates the interrupt can be used to request DTC activation
Canceling Snooze mode	" ✓" 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 11.5 Event table (1 of 2)

Event number	Interrupt request source	Name	Connect to NVIC	Invoke DTC	Canceling Snooze	Canceling Software Standby
0	IWDT	IWDT_NMIUNDF	✓	—	✓	✓
1	LVD	LVD_LVD1	✓	✓	✓	✓
2	PORT	PORT_IRQ0	✓	✓	✓	✓
3		PORT_IRQ1	✓	✓	✓	✓
4		PORT_IRQ2	✓	✓	✓	✓
5		PORT_IRQ3	✓	✓	✓	✓
6		PORT_IRQ4	✓	✓	✓	✓
7		PORT_IRQ5	✓	✓	✓	✓
8		PORT_IRQ6	✓	✓	✓	✓
9		PORT_IRQ7	✓	✓	✓	✓
10	DTC	DTC_COMPLETE	✓	—	✓	—
11	FLASH	FLASH_FRDYI	✓	—	—	—
12	SAU1	SAU1_UART_TXI2/ SAU1_SPI_TXRXI20/ SAU1_IIC_TXRXI20	✓	✓	—	—
13		SAU1_UART_RXI2/ SAU1_SPI_TXRXI21/ SAU1_IIC_TXRXI21	✓	✓	—	—
14		SAU1_UART_ERRI2	✓	—	—	—
15	ELC	ELC_SWEVT0	✓*1	✓	—	—
16		ELC_SWEVT1	✓*1	✓	—	—
17	TRNG	TRNG_RDREQ	✓	—	—	—
18	SAU0	SAU0_UART_TXI0/ SAU0_SPI_TXRXI00/ SAU0_IIC_TXRXI00	✓	✓	✓	—
19	TAU0	TAU0_TMI00	✓	✓	—	—
20	SAU0	SAU0_UART_ERRI0	✓	—	✓	—
21	TAU0	TAU0_TMI01H	✓	—	—	—

Table 11.5 Event table (2 of 2)

Event number	Interrupt request source	Name	Connect to NVIC	Invoke DTC	Canceling Snooze	Canceling Software Standby
22	SAU0	SAU0_UART_TXI1/ SAU0_SPI_TXRXI10/ SAU0_IIC_TXRXI10	✓	✓	—	—
23		SAU0_UART_RXI1/ SAU0_SPI_TXRXI11/ SAU0_IIC_TXRXI11	✓	✓	—	—
24		SAU0_UART_ERRI1	✓	—	—	—
25	TAU0	TAU0_TMI03H	✓	—	—	—
26	IICA0	IICA0_TXRXI	✓	—	✓	✓
27	SAU0	SAU0_UART_RXI0/ SAU0_SPI_TXRXI01/ SAU0_IIC_TXRXI01	✓	✓	✓	—
28	TAU0	TAU0_TMI01	✓	✓	—	—
29		TAU0_TMI02	✓	✓	—	—
30		TAU0_TMI03	✓	✓	—	—
31	ADC12	ADC12_ADI	✓	✓	✓	—
32	RTC	RTC_ALM_OR_PRD	✓	✓	✓	✓
33	TML32	TML32_ITL_OR	✓	✓	✓	✓
34		TML32_ITL0	—	✓	—	—
35	TAU0	TAU0_TMI04	✓	✓	—	—
36		TAU0_TMI05	✓	✓	—	—
37		TAU0_TMI06	✓	✓	—	—
38		TAU0_TMI07	✓	✓	—	—
39	UARTA0	UARTA0_ERRI	✓	—	✓	✓
40		UARTA0_TXI	✓	✓	✓	✓
41		UARTA0_RXI	✓	✓	✓	✓
42	IICA1	IICA1_TXRXI	✓	—	✓	✓
43	UARTA1	UARTA1_ERRI	✓	—	✓	✓
44		UARTA1_TXI	✓	✓	✓	✓
45		UARTA1_RXI	✓	✓	✓	✓

Note 1. Only interrupts after DTC transfer are supported.

11.4 Interrupt Operation

The ICU performs the following functions:

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

11.4.1 Detecting Interrupts

The ICU accepted interrupt source from a peripheral function interrupt or an external pin interrupt and sends an interrupt request to the NVIC.

External pin interrupt requests are detected by either:

- Edges (falling edge, rising edge, or rising and falling edges)

Set the IRQCRi.IRQMD[1:0] bits to select the detection mode for the IRQi pins. For interrupt sources associated with peripheral modules, see Table 11.3. Events must be accepted by the NVIC before an interrupt occurs and is accepted by the CPU.

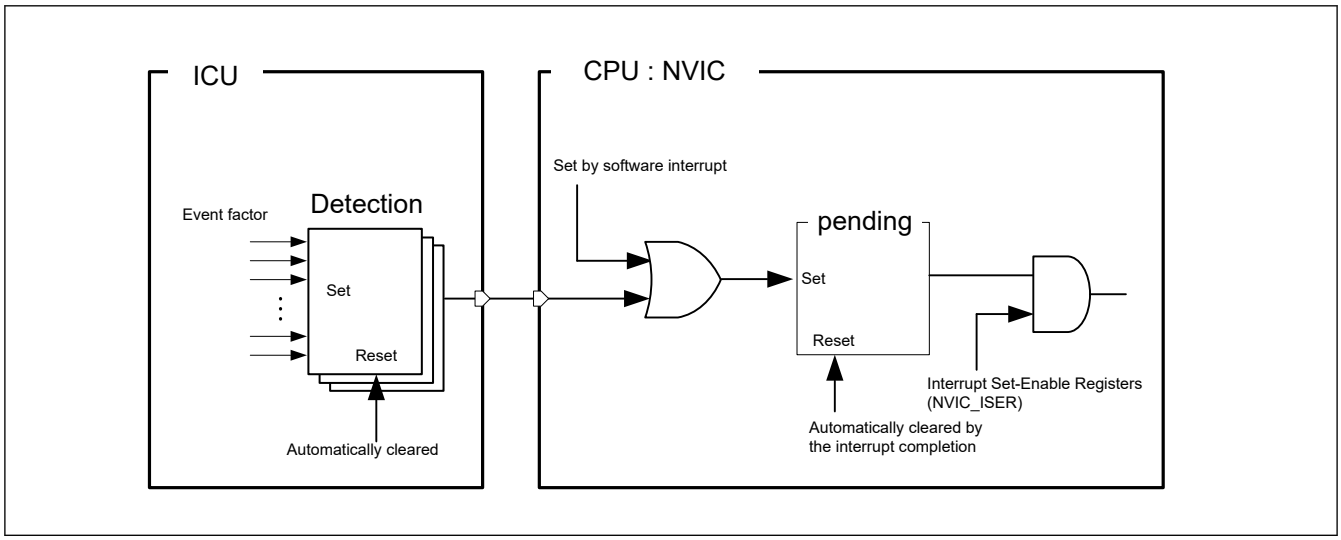


Figure 11.2 Interrupt path of the ICU and CPU (NVIC)

11.5 Interrupt setting procedure

11.5.1 Enabling Interrupt Requests

The procedure for enabling an interrupt request is as follows:

1. Set the Interrupt Set-Enable register (NVIC_ISER).
2. If you use the DTC, write 1 to the DTCENSETn.SETi bit to set DTCENSTn.STi bit.
3. Specify the operation settings for the event source, snooze mode or software standby mode cancellation (SBYEDCRn register setting).

11.5.2 Disabling Interrupt Requests

The procedure to disable the interrupt request is as follows:

1. Disable the operation settings for the event source, snooze mode or software standby mode cancellation (SBYEDCRn register setting).
2. If the DTCENSTn.STi bit is set, write 1 to the DTCENCLRn.CLRi bit to clear DTCENSTn.STi bit.
3. Clear the interrupt Clear-Enable register (NVIC_ICER) and interrupt Clear-Pending register (NVIC_ICPR).

11.5.3 Polling for Interrupts

The procedure for polling for interrupt requests is as follows:

1. Set the Interrupt Clear-Enable register (NVIC_ICER).
2. Specify the operation settings for the event source, snooze mode or software standby mode cancellation (SBYEDCRn register setting).
3. Poll the interrupt Set-Pending register (NVIC_ISPR).

11.5.4 Selecting Interrupt Request Destinations

The available destinations are fixed for each interrupt, as described in Table 11.3 and Table 11.5. The interrupt output destination, CPU, or DTC can be independently selected for each interrupt source.

11.5.4.1 CPU interrupt request

When DTCENSTn.STi = 0, the interrupt is output to the NVIC.

11.5.4.2 DTC activation

When DTCENSTn.STi = 1, the interrupt is output to the DTC. Use the following procedure:

1. Write 1 to the DTCENSETn.SETi bit to DTC enable the DTCENSTn.STi bit.
2. Set the DTC Module Start bit (DTCST.DTCST) to 1.

Table 11.6 shows operation when the DTC is the interrupt request destination.

Table 11.6 Operation when DTC becomes interrupt request destination

Interrupt request destination	DISEL*1	Remaining transfer operations	Operation per request	Interrupt request destination after transfer
DTC*2	1	≠ 0	DTC transfer → CPU interrupt	DTC
		= 0	DTC transfer → CPU interrupt	CPU (DTCENSTn, STi bit is automatically cleared)
	0	≠ 0	DTC transfer	DTC
		= 0	DTC transfer → CPU interrupt	CPU (DTCENSTn.STi bit is automatically cleared)

Note: When the INTFLAGn.IFi is 1, an interrupt request (DTC activation request) that occurs again is ignored.
Note 1. MRB.DISEL bit controls the interrupt generates timing from DTC to CPU.
Note 2. For chain transfers, DTC transfer continues until the last chain transfer ends. The MRB.DISEL bit state and the remaining transfer count determine whether a CPU interrupt occurs, the INTFLAGn.IFi clear timing, and the interrupt request destination after transfer. See Table 14.2 in section 14, Data Transfer Controller (DTC).

11.5.5 External Pin Interrupts

To use external pin interrupts:

1. Configure I/O ports settings.
2. Set the IRQMD[1:0] bits of the given IRQCRi register (i = 0 to 7) to select the senses of detection.
3. Select the IRQ pin as follows:
   - If the IRQ pin is to be used for CPU interrupt requests, write 1 to the DTCENCLRn.CLRi bit to DTC disable the DTCENSTn.STi bit.
   - If the IRQ pin is to be used for DTC activation, write 1 to the DTCENSETn.SETi bit to DTC enable the DTCENSTn.STi bit.

11.6 Non-maskable Interrupt Operation

The following sources can trigger a non-maskable interrupt:

- NMI pin interrupt
- IWDT underflow/refresh error interrupt
- Voltage monitor 1 interrupt
- SRAM parity 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:

1. Set the NMIMD bit of NMICR register.
2. Write 1 to the NMICLR.NMICLR bit to clear the NMISR.NMIST flag to 0.

3. 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.

11.7 Return from Low Power Modes

Table 11.5 lists the interrupt sources that can be used to exit Sleep or Software Standby mode. For more information, see section 9, Low Power Modes.

11.7.1 Return from Sleep Mode

To return from Sleep mode in response to an interrupt:

non-maskable interrupt

- Use the NMIER register to enable the target interrupt request.

maskable interrupt

- Select the CPU as the interrupt request destination.
- Enable the interrupt in the NVIC.

11.7.2 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 11.5.

To return from Software Standby mode:

1. Select the interrupt source that enables return from Software Standby:
   - For non-maskable interrupts, use the NMIER register to enable the target interrupt request
   - For maskable interrupts, use the SBYEDCRn register to enable the target interrupt request.
2. Select the CPU as the interrupt request destination
3. Enable the interrupt in the NVIC.

Interrupt requests through the IRQn pins that do not satisfy these conditions are not detected while the clock is stopped in Software Standby mode.

11.7.3 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. Select the CPU as the interrupt request destination.
2. Enable the interrupt in the NVIC.

Note: In Snooze mode, a clock is supplied to the ICU. If an interrupt is detected, the CPU acknowledges the interrupt after returning to Normal mode from Software Standby mode.

11.8 Using the WFI Instruction with Non-maskable Interrupts

Whenever a WFI instruction is executed, confirm that all status flags in the NMISR register are 0.

11.9 Reference

- ARM® Cortex®-M23 Processor Technical Reference Manual (ARM DDI 0550C)

12. Buses

12.1 Overview

Table 12.1 lists the bus specifications, Figure 12.1 shows the bus configuration, and Table 12.2 lists the addresses assigned for each bus.

Table 12.1 Bus specifications

Bus type		Description
Main bus	System bus (CPU)	• Connected to CPU • 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
	Internal peripheral bus 3	• Connected to peripheral modules (ELC, IWDT, MSTP and CRC) • Connected to peripheral modules (I/O Ports, ADC12, SAU0, SAU1, TAU, RTC, IICA, UARTA, TML32, and PCLBUZ)
	Internal peripheral bus 7	Connected to TRNG
	Internal peripheral bus 9	Connected to code flash memory (in P/E (Programming/Erasure)), data flash memory

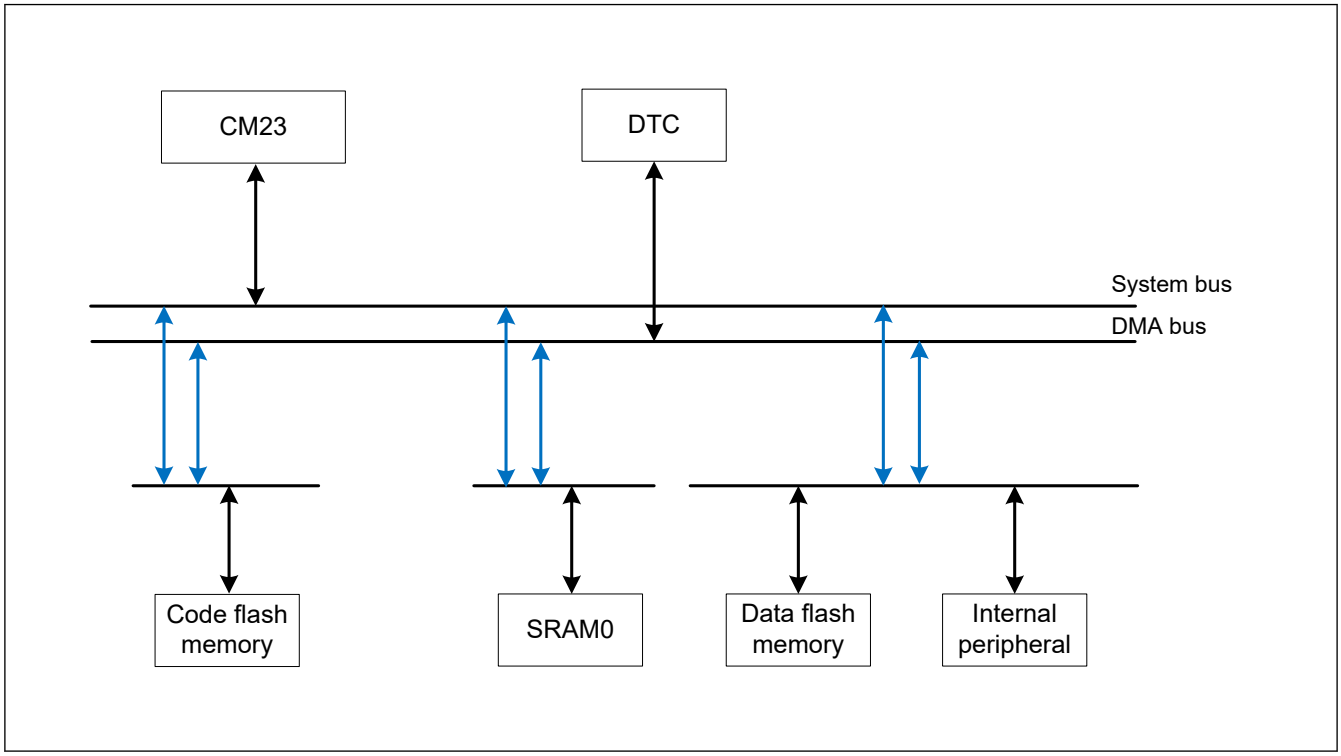


Figure 12.1 Bus configuration

Table 12.2 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 I/O registers
0x4001_9000 to 0x4001_9FFF	Memory bus 4	MTB I/O registers

Table 12.2 Addresses assigned for each bus (2 of 2)

Address	Bus	Area
0x4001_A000 to 0x4001_FFFF	Internal peripheral bus 1	Peripheral I/O registers
0x4004_0000 to 0x400B_FFFF	Internal peripheral bus 3	
0x400C_0000 to 0x400D_FFFF	Internal peripheral bus 7	Peripheral I/O registers (TRNG)
0x4010_0000 to 0x407F_FFFF	Internal peripheral bus 9	Code and data flash memory (in P/E), Data flash memory

12.2 Description of Buses

12.2.1 Main Buses

The main buses consist of the system bus and DMA bus. The system bus and DMA bus are connected to the following:

- Code flash memory
- SRAM0
- Data flash memory
- Internal peripheral bus

The system bus is used for instruction and data accesses to the CPU.

Different master and slave transfer combinations can proceed simultaneously. In addition, requests for bus access from masters other than the DTC are not accepted during reads of transfer control information for the DTC.

12.2.2 Slave Interface

For connections from the main bus to the slave interfaces, see the slave interfaces in section 12.1. Overview.

Bus access from the system bus and DMA bus is arbitrated and has the following fixed priority order:

DMA bus > system bus

Different master and slave transfer combinations can proceed simultaneously.

12.2.3 Parallel Operations

Parallel operations are possible when different bus masters request access to different slave modules. Figure 12.2 shows an example of parallel operations. In this example, the CPU uses the instruction and operand buses for simultaneous access to the flash and SRAM, respectively. Additionally, the DTC simultaneously uses the DMA bus for access to a peripheral bus during access to the flash and SRAM by the CPU.

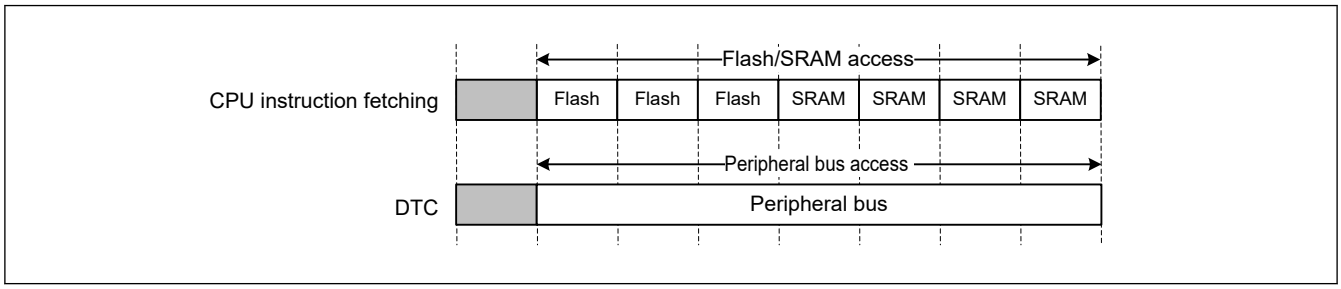


Figure 12.2 Example of parallel operations

12.2.4 Restriction on Endianness

Memory space must be little-endian to execute code on the Cortex®-M23 core.

12.2.5 Restriction on Exclusive Access

The main bus does not support exclusive transfers and there is no global monitor in the MCU. The main bus always deasserts the HEXOKAY signal (a signal in the AHB-Lite protocol) to the CPU. This means that a store exclusive

instruction such as STREX instruction always gets a failed status. When an exclusive write operation is performed by the CPU, the main bus always writes the data successfully.

12.3 Register Descriptions

12.3.1 BUSMCNTx : Master Bus Control Register x (x = SYS, DMA)

Base address: BUS = 0x4000_3000

Offset address: 0x1008 (BUSMCNTSYS)
0x100C (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: A bus error is reported. 1: A bus error is not reported.	R/W

Note: Changing reserved bits from the initial value of 0 is prohibited. Operation during the change is not guaranteed.

IERES bit (Ignore Error Responses)

The IERES bit specifies the enable or disable of an error response of the AHB-Lite protocol.

Table 12.3 lists the registers associated with each bus type.

Table 12.3 Associations between bus types and registers

Bus type	Master Bus Control Register	Bus Error Address Register	Bus Error Status Register
System bus (CPU)	BUSMCNTSYS	BUS3ERRADD	BUS3ERRSTAT
DMA bus	BUSMCNTDMA	BUS4ERRADD	BUS4ERRSTAT

12.3.2 BUSnERRADD : Bus Error Address Register n (n = 3, 4)

Base address: BUS = 0x4000_3000

Offset address: 0x1820 (n = 3)
0x1830 (n = 4)

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

Note: BUSnERRADD is only cleared by resets. For more information, see section 5, Resets.

Table 12.3 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. For more information, see the description of the ERRSTAT flag in section 12.3.3. BUSnERRSTAT : BUS Error Status Register n (n = 3, 4) and section 12.4. Bus Error Monitoring Section.

The value of the BUSnERRADD.BERAD[31:0] bits (n = 3, 4) is valid only when the BUSnERRSTAT.ERRSTAT flag (n = 3, 4) is set to 1.

12.3.3 BUSnERRSTAT : BUS Error Status Register n (n = 3, 4)

Base address: BUS = 0x4000_3000

Offset address: 0x1824 (n = 3)
0x1834 (n = 4)

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

Bit	Symbol	Function	R/W
0	ACCSTAT	Error Access Status flag Access status when the error occurred: 0: Read access 1: Write access	R
6:1	—	These bits are read as 0.	R
7	ERRSTAT	Bus Error Status flag 0: No bus error occurred. 1: Bus error occurred.	R

Note: BUSnERRSTAT is only cleared by resets. For more information, see section 5, Resets.

Table 12.3 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. For more information, see the description of the BUSnERRSTAT.ERRSTAT flag and section 12.4. Bus Error Monitoring Section.

The value is valid only when the BUSnERRSTAT.ERRSTAT flag (n = 3, 4) is set to 1.

ERRSTAT flag (Bus Error Status flag)

The ERRSTAT flag indicates whether a bus error occurred. When a bus error occurs, the access address and status of write or read access are stored. The BUSnERRSTAT.ERRSTAT flag (n = 3, 4) is set to 1.

For more information on bus errors, see section 12.4. Bus Error Monitoring Section.

12.4 Bus Error Monitoring Section

The monitoring system monitors each individual area, and whenever it detects an error, it returns the error to the requesting master IP using the AHB-Lite error response protocol.

12.4.1 Error Type that Occurs by Bus

One type of errors can occur on each bus:

- Illegal address access

section 12.4.3. Conditions for issuing illegal Address Access Errors 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.

12.4.2 Operation when a Bus Error Occurs

When a bus error occurs, operation is not guaranteed and the error is returned to the requesting master IP. The bus error information occurred in each master is stored in the BUSnERRADD and BUSnERRSTAT registers. These registers must be cleared by reset only. For more information, see section 12.3.2. BUSnERRADD : Bus Error Address Register n (n = 3, 4) and section 12.3.3. BUSnERRSTAT : BUS Error Status Register n (n = 3, 4).

Note: DTC does not receive bus errors. If the DTC accesses the bus, the transfer continues.

12.4.3 Conditions for issuing illegal Address Access Errors

Table 12.4 lists the address spaces for each bus that issue illegal address access errors.

Table 12.4 Conditions leading to illegal address access errors

Address	Slave bus name	Main buses	
		System bus(CPU)	DMA bus
0x0000_0000 to 0x01FF_FFFF	Memory bus 1	—	—
0x0200_0000 to 0x1FFF_FFFF	Reserved	E	E
0x2000_0000 to 0x2000_7FFF	Memory bus 4	—	—
0x2000_8000 to 0x3FFF_FFFF	Reserved	E	E
0x4000_0000 to 0x4001_8FFF	Internal peripheral bus 1	—	—
0x4001_9000 to 0x4001_9FFF	Memory bus 4	—	—
0x4001_A000 to 0x4001_FFFF	Internal peripheral bus 1	—	—
0x4002_0000 to 0x4003_FFFF	Reserved	E	E
0x4004_0000 to 0x400B_FFFF	Internal peripheral bus 3	—	—
0x400C_0000 to 0x400D_FFFF	Internal peripheral bus 7	—	—
0x400E_0000 to 0x400F_FFFF	Reserved	E	E
0x4010_0000 to 0x407F_FFFF	Internal peripheral bus 9	—	—
0x4080_0000 to 0xDFFF_FFFF	Reserved	E	E
0xE000_0000 to 0xFFFF_FFFF	System for Cortex®-M23	—	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: Memory bus 1 no access error detection.

12.5 References

1. *ARM®v8-M Architecture Reference Manual* (ARM DDI0553B.a)
2. *ARM® Cortex®-M23 Processor User Guide* (ARM DUI0963B)
3. *ARM® AMBA® 5 AHB-Lite Protocol Specification* (ARM IHI0033B.b)

13. Flash Read Protection (FRP)

13.1 Overview

The MCU incorporates the Flash Read Protection (FRP) function with one secure region that include the code flash. The secure region can be protected from read accesses. Any program cannot read data in a protected region.

Table 13.1 lists the specifications of the FRP and Figure 13.1 shows a block diagram.

Table 13.1 Flash Read Protection specifications

Specifications	Description
Secure regions	Code flash
Address range that can be specified as protected regions	0x0000_0800 to 0x0001_FFFF
Number of protected regions	1 region
How to specify the address of each protected region	Setting the address where regions start and end
Enable or disable setting of the protection	Settings enabled or disabled for the associated region

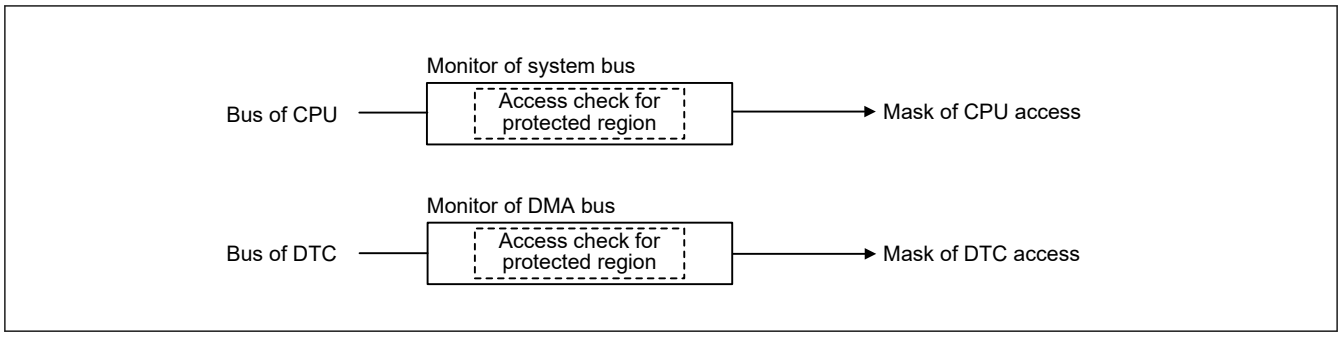


Figure 13.1 Flash Read Protection block diagram

13.1.1 Memory Protection

All the FRP 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. The FRP uses FRPS, FRPE, and FRPDIS bits of OFS1 register of option-setting memory. OFS1.FRPS and OFS1.FRPE define starting and ending address of a protected region. OFS1.FRPDIS disable or enable the FRP. For details of OFS1 register, see section 6, Option-Setting Memory.

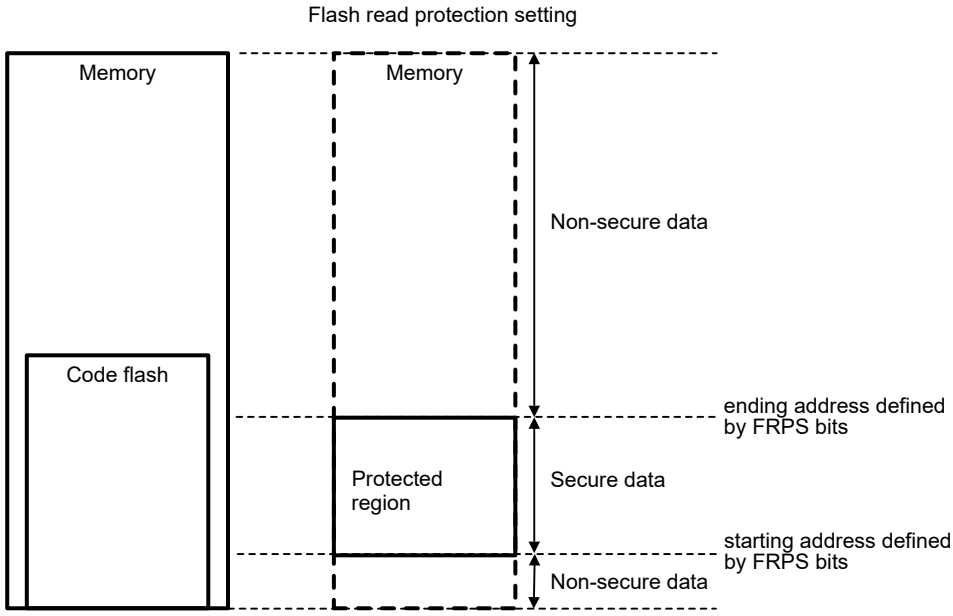
The FRP protects the secured regions (the code flash memory) from being read by programs. If a read access to a protected region is detected, the FRP invalidates it.

In each of the system bus and the DMA bus, there is a monitor that checks accesses on the bus. When OFS1.FRPDIS is set to 0, the monitor checks if an access on the bus is not instruction fetch and if its address is within a protected region defined by OFS1.FRPS and OFS1.FRPE. If the access is not instruction fetch and its address is within a protected region, the monitor returns 0xFFFF_FFFF as a read value to the bus master instead of a value read from the address. If the access is instruction fetch or its address is outside a protected region, the monitor returns a value read from the address. When OFS1.FRPDIS is set to 1, the monitor does not check accesses on the bus.

The FRP provides access protection in the following conditions:

- Secure data is read from the CPU
- Secure data is read from other than the CPU (DTC)
- Secure data is read from the debugger.
- Secure data is accessible only by instruction fetch.

Note: Secure data:
Code flash region within the limits set up by OFS1.FRPS and OFS1.FRPE.



Note: Any program cannot read secure data (Protected region).
Note: Secure data is accessible only by instruction fetch.
Note: Any program can read non-secure data.
Note: The protected region must be aligned at 2 KB boundaries.
Note: The starting address defined by FRPS bits specifies the first address of the protected region and the ending address defined by FRPE bits specifies the last address of the protected region.

Figure 13.2 Use case of flash read protection

13.2 Usage Notes

13.2.1 Notes on the Use of a Debugger

The memory cannot be debugged if the FRP is enabled. Disable the flash read protection when debug a program, OCD debug only valid when OFS1.FRPDIS bit is 1.

13.2.2 Compiler Settings

The FRP is a kind of execute-only memory (XOM). Since data in a protected region is not readable, a protected region cannot have constant data such as literal pool. Therefore, appropriate compiler settings are required.

13.2.3 Protection of OFS1 Register

Because overwriting OFS1.FRPS[5:0], OFS1.FRPE[5:0], or OFS1.FRPDIS bits can disable the protection of FRP, the OFS1 register (address = 0x0000_0404) must be protected by the access window function. However, setting the access window function, which includes setting AWS.FSPR bit to 0, also disables the changing of AWS.BTFLG and FISR.SAS[1:0] bits used for the startup area select function. Therefore, the startup area select function is not available when using the FRP function. See section 28.5.2. Startup Area Select for the startup area select function and section 28.5.3. Protection by Access Window for the access window function.

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

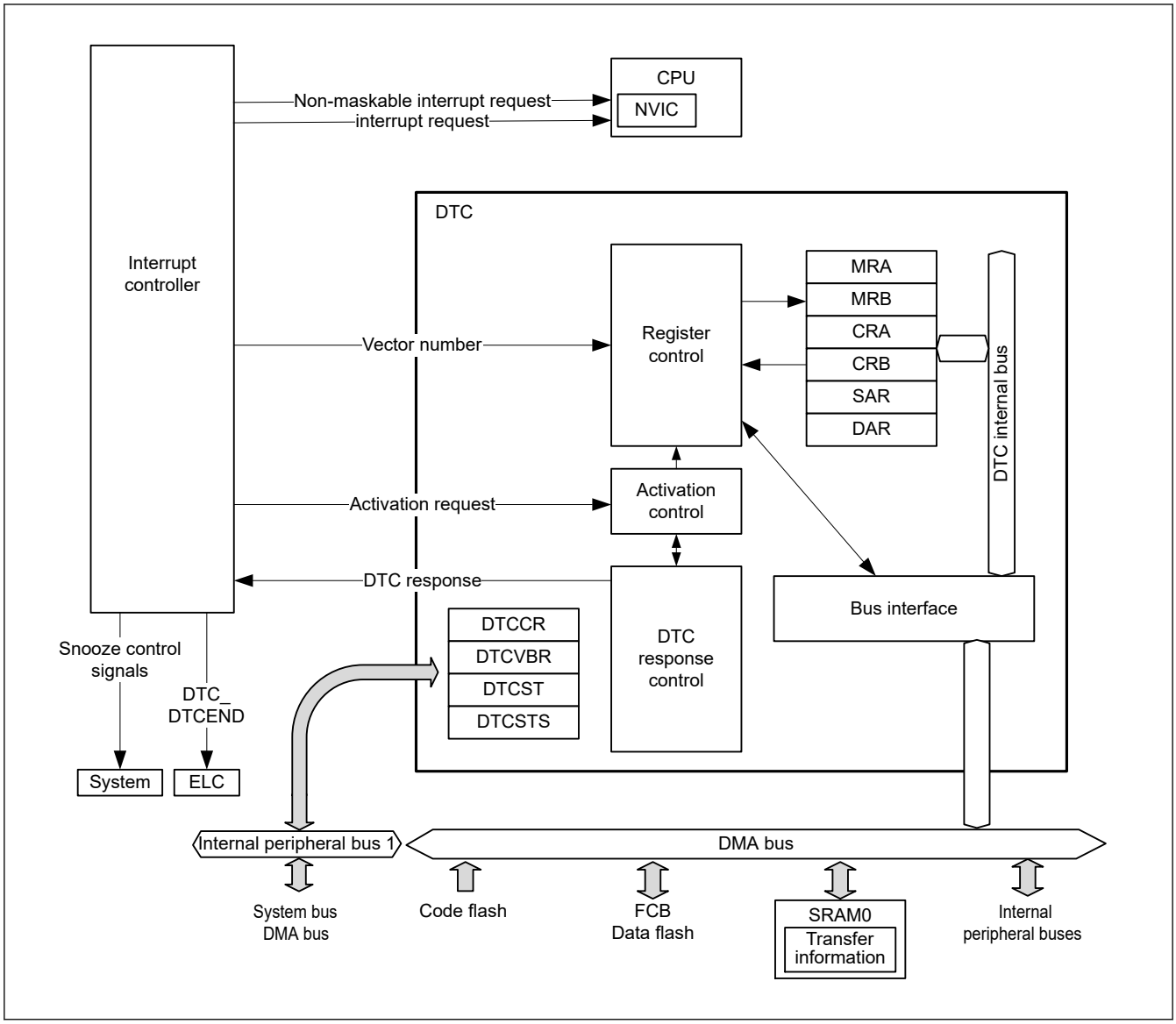


Figure 14.1 DTC block diagram

See section 11, Interrupt Controller Unit (ICU) for the connections between the DTC and NVIC 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: 0x0003 + 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: 0x0002 + 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.	—
3:2	DM[1:0]	Transfer Destination Address Addressing Mode 0 0: Address in the DAR register is fixed (write-back to DAR is skipped) 0 1: Address in the DAR register is fixed (write-back to DAR is skipped) 1 0: DAR value is incremented after data transfer: +1 when MRA.SZ[1:0] = 00b +2 when SZ[1:0] = 01b +4 when SZ[1:0] = 10b 1 1: DAR value is decremented after data transfer: -1 when MRA.SZ[1:0] = 00b -2 when SZ[1:0] = 01b -4 when SZ[1:0] = 10b	—
4	DTS	DTC Transfer Mode Select 0: Select transfer destination as repeat or block area. 1: Select transfer source as repeat or block area.	—
5	DISEL	DTC Interrupt Select 0: Generate an interrupt request to the CPU when specified data transfer is complete. 1: Generate an interrupt request to the CPU each time DTC data transfer is performed.	—

Bit	Symbol	Function	R/W
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: 0x0004 + 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

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: 0x0008 + 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

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: 0x000E + 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: 0x000C + 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: 0x0000

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: 0x0004

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	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: 0x000C

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 without Snooze mode transition

For details on these transitions, see section 14.9. Low Power Consumption Function and section 9, Low Power Modes.

14.2.10 DTCSTS : DTC Status Register

Base address: DTC = 0x4000_5400

Offset address: 0x000E

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.DTCENSTx.STy (x = 0, 1, y = 0 to 45) bit to 1 enables activation of the DTC by the associated interrupt, where y indicates the event number of the interrupt request. By using a software event, the DTC can be activated by the software. To activate the DTC by the 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.DTCENSTx.STy 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.

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.

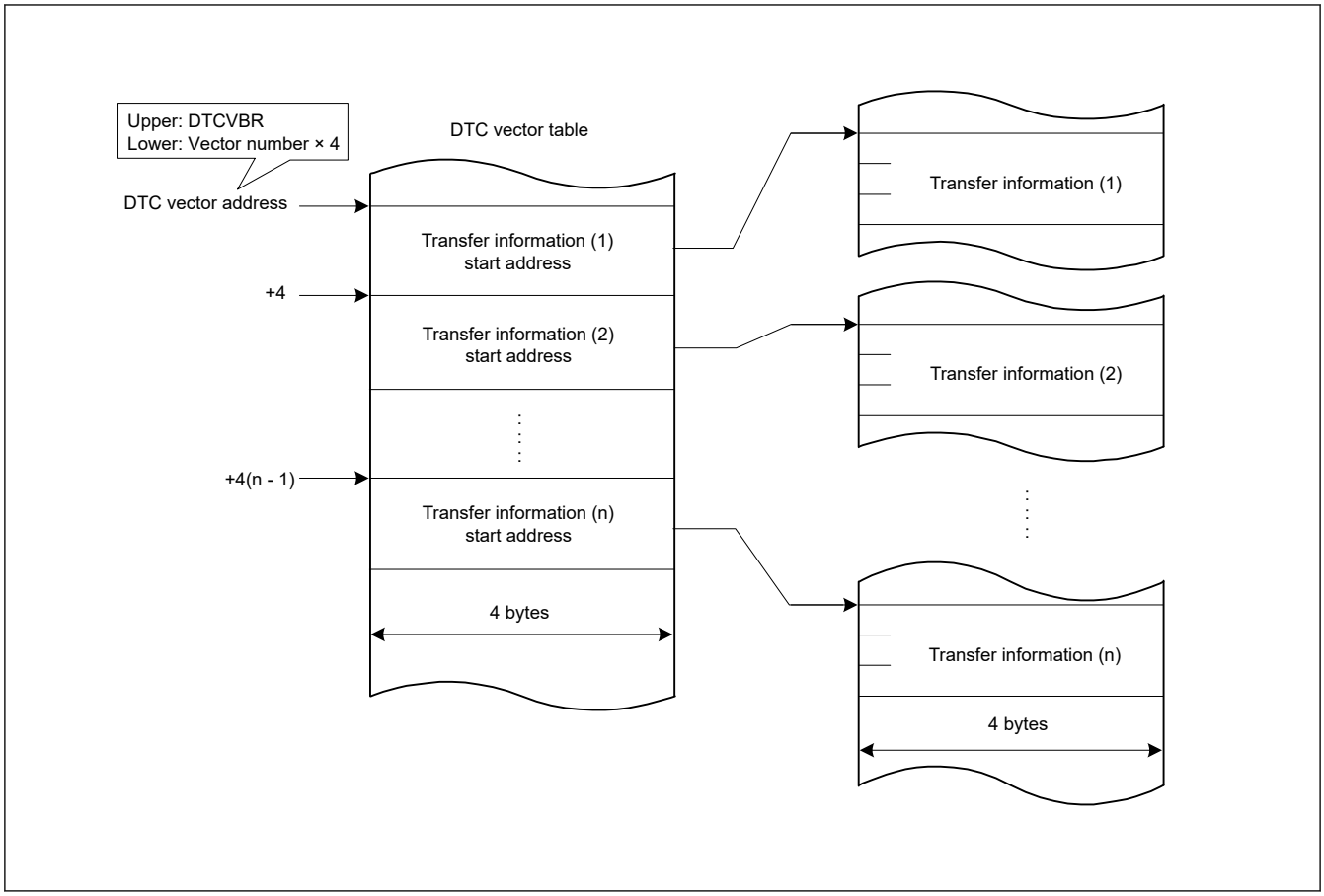


Figure 14.2 DTC vector table and transfer information

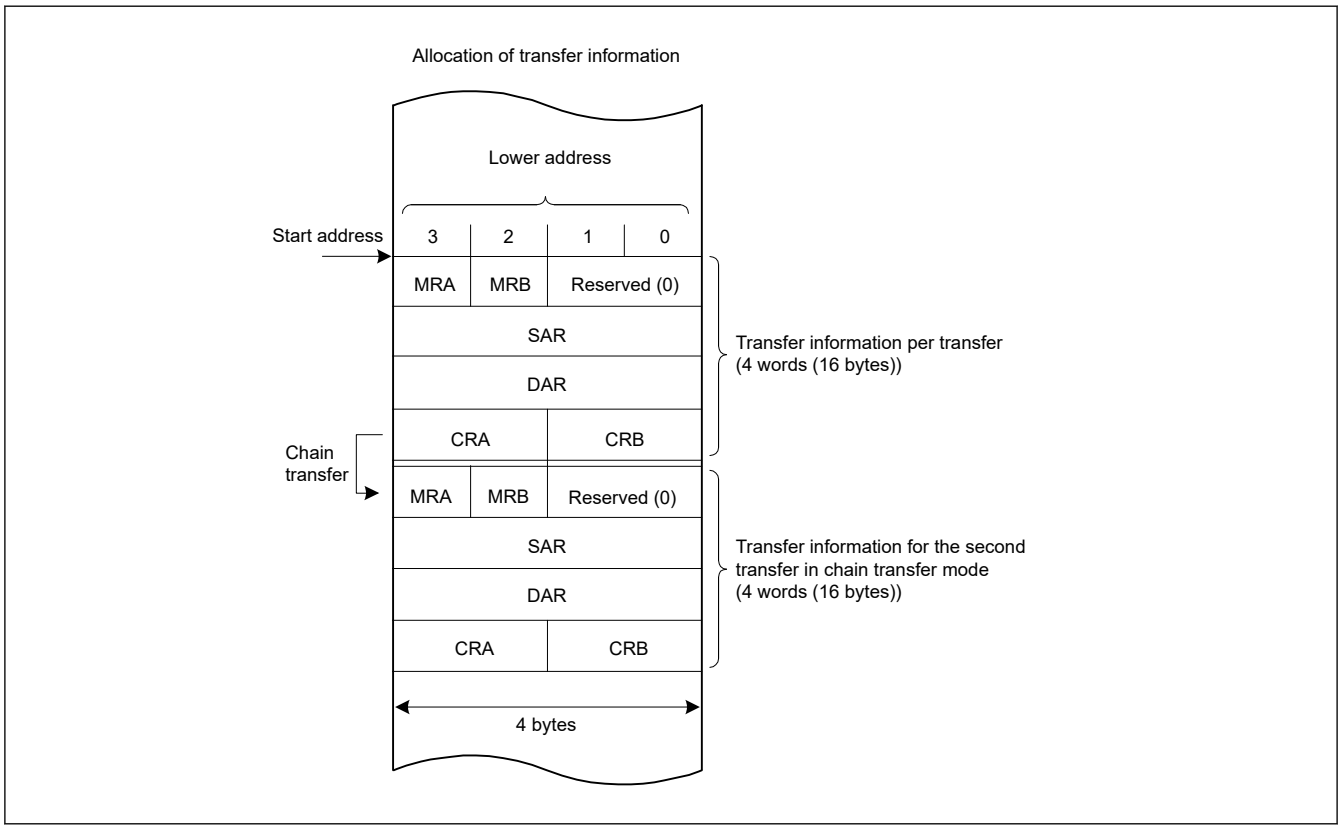


Figure 14.3 Allocation of transfer information in the SRAM area

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 lists the DTC transfer modes.

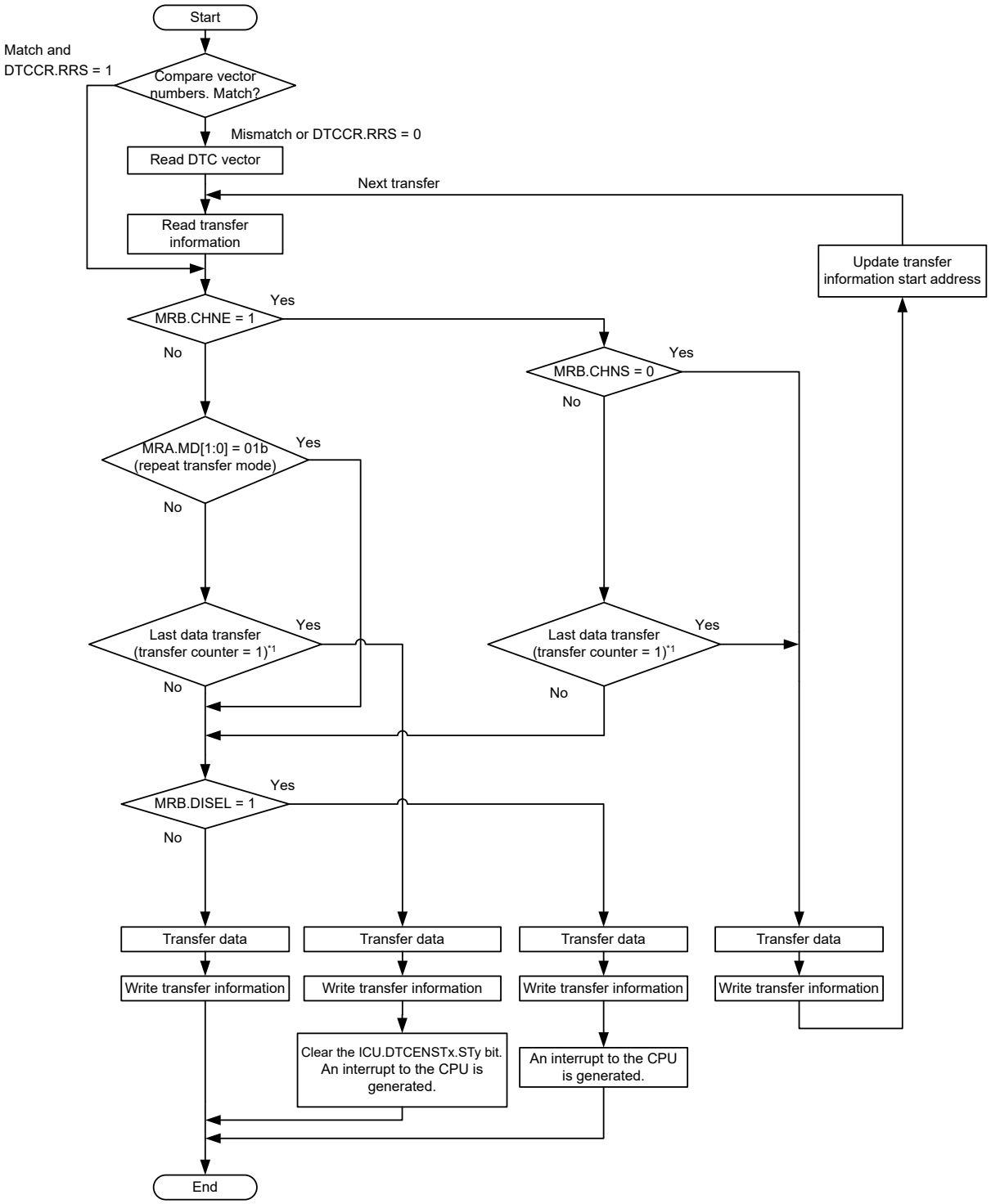
Table 14.2 DTC transfer modes

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				
CHNE bit	CHNS bit	DISEL bit	Transfer counter*1 *2	CHNE bit	CHNS bit	DISEL bit	Transfer counter*1 *2	DTC transfer
0	—	0	Other than (1 → 0)	—	—	—	—	Ends after the first transfer
0	—	0	(1 → 0)	—	—	—	—	Ends after the first transfer with an interrupt request to the CPU
0	—	1	—	—	—	—	—	
1	0	—	—	0	—	0	Other than (1 → 0)	Ends after the second transfer
				0	—	0	(1 → 0)	Ends after the second transfer with an interrupt request to the CPU
				0	—	1	—	
1	1	0	Other than (1 → *)	—	—	—	—	Ends after the first transfer
1	1	—	(1 → *)	0	—	0	Other than (1 → 0)	Ends after the second transfer
				0	—	0	(1 → 0)	Ends after the second transfer with an interrupt request to the CPU
				0	—	1	—	
1	1	1	Other than (1 → *)	—	—	—	—	Ends after the first transfer with an interrupt request to the CPU

Note 1. The transfer counter used depends on the transfer modes as follows:
Normal transfer mode — CRA register
Repeat transfer mode — CRAL register
Block transfer mode — CRB register

Note 2. On completion of a data transfer, the counters operate as follows:
1 → 0 in normal and block transfer modes
1 → CRAH in repeat transfer mode
(1 → *) in the table indicates both of these two operations, depending on the mode.

Note 3. Chain transfer can be selected for the second or subsequent transfers. The conditions for the combination of the second transfer and CHNE = 1 is omitted.

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			
b3	b2	b3	b2	SAR register	DAR register
0	0	0	0	Skip	Skip
0	0	0	1		
0	1	0	0		
0	1	0	1		
0	0	1	0	Skip	Write-back
0	0	1	1		
0	1	1	0		
0	1	1	1		
1	0	0	0	Write-back	Skip
1	0	0	1		
1	1	0	0		
1	1	0	1		
1	0	1	0	Write-back	Write-back
1	0	1	1		
1	1	1	0		
1	1	1	1		

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.

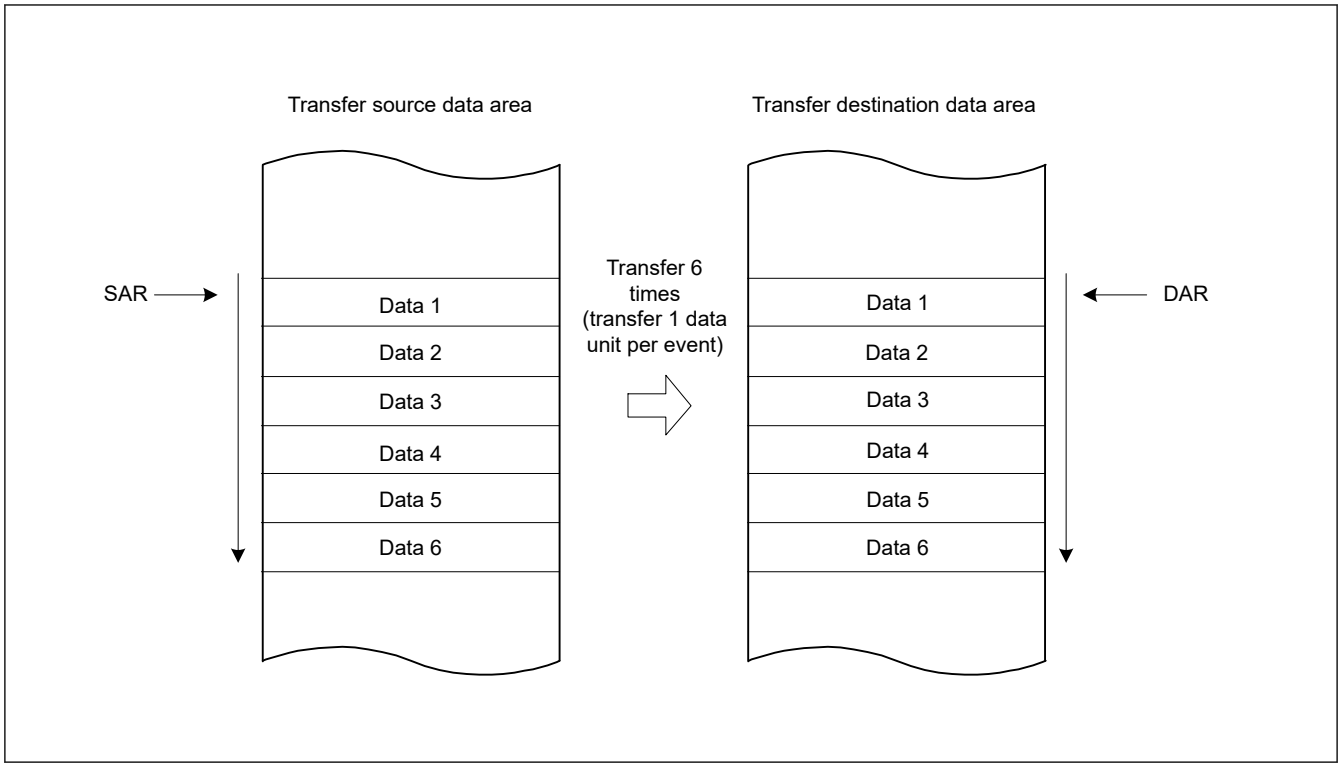


Figure 14.5 Memory map of normal transfer mode (MRA.SM[1:0] = 10b, MRB.DM[1:0] = 10b, CRA = 0x0006)

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.

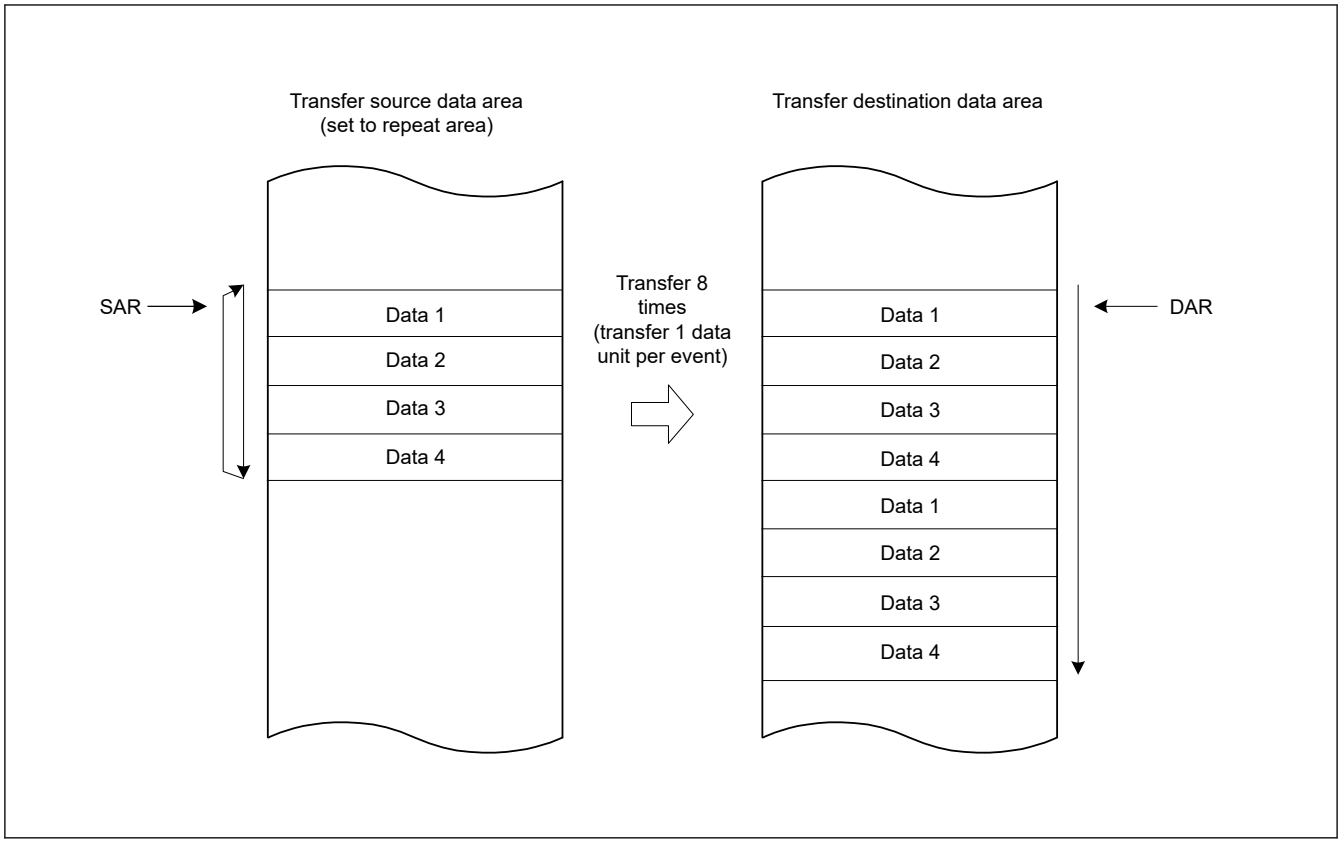


Figure 14.6 Memory map of repeat transfer mode when transfer source is a repeat area (MRA.SM[1:0] = 10b, MRB.DM[1:0] = 10b, CRAH = 0x04)

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.

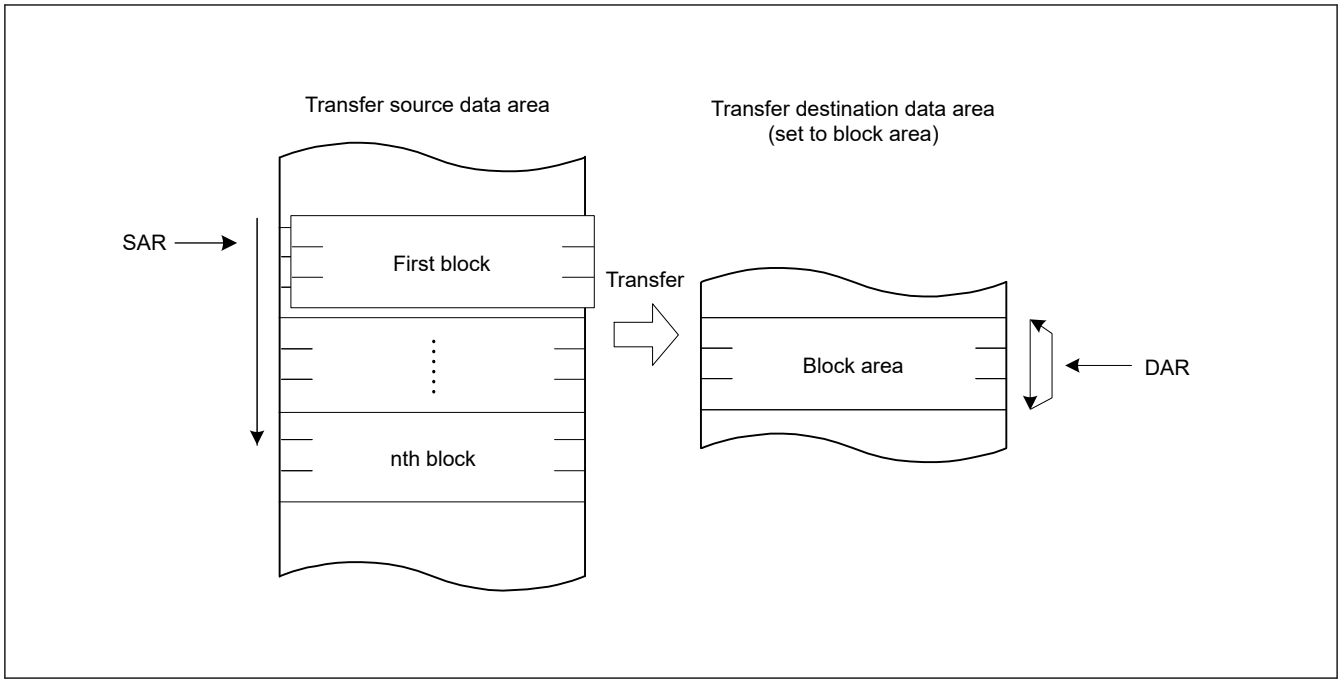


Figure 14.7 Memory map of block transfer mode

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.

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.

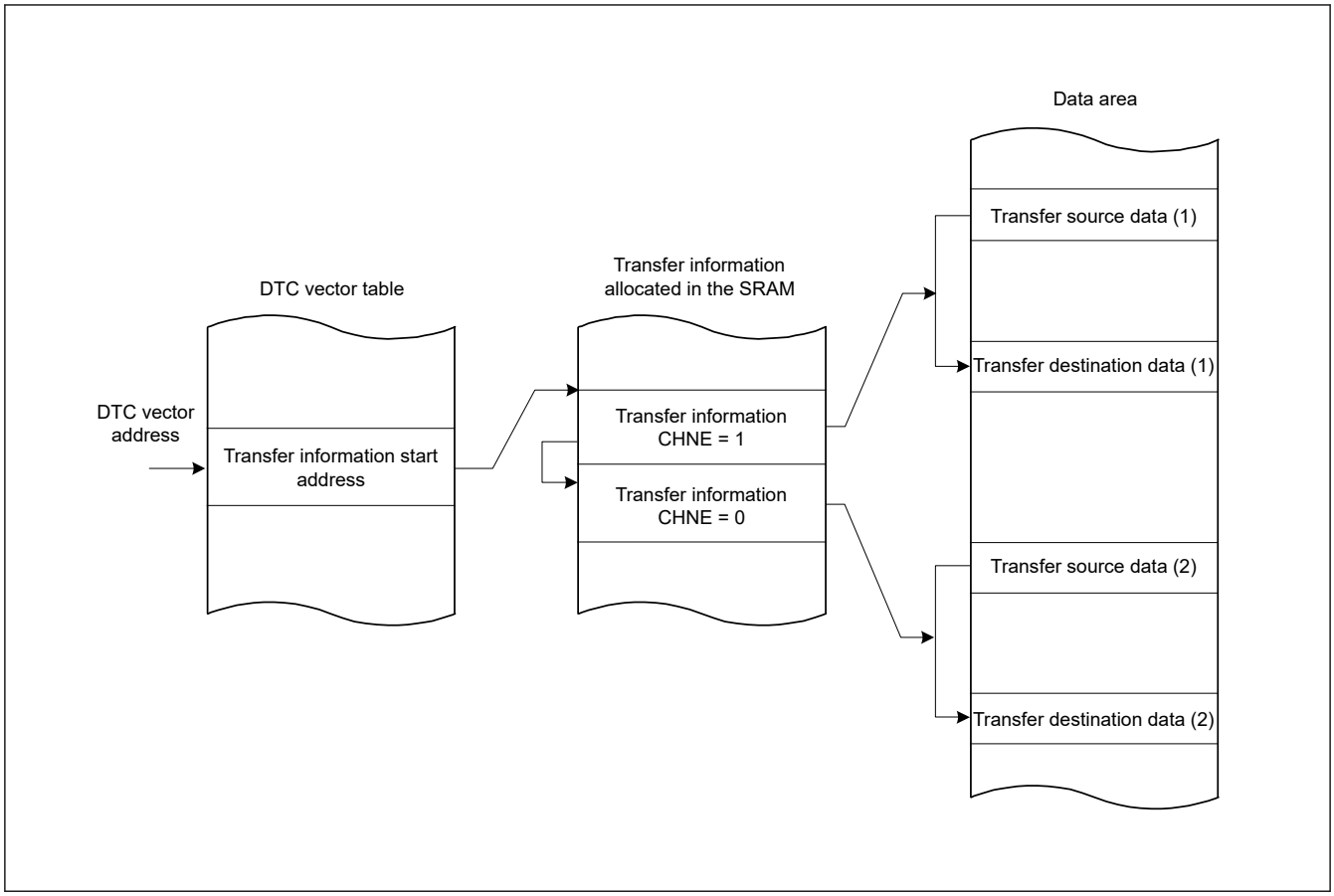


Figure 14.8 Chain transfer operation

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.

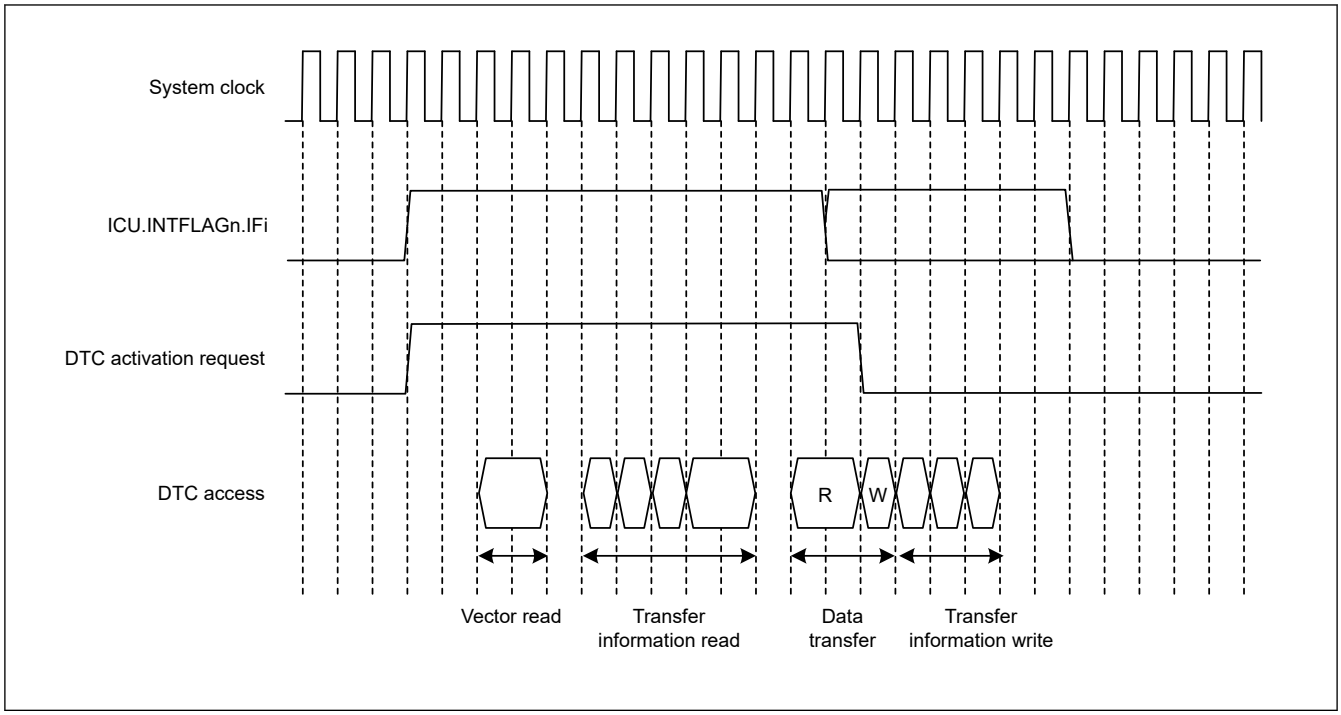


Figure 14.9 Example 1 of DTC operation timing in normal transfer and repeat transfer modes

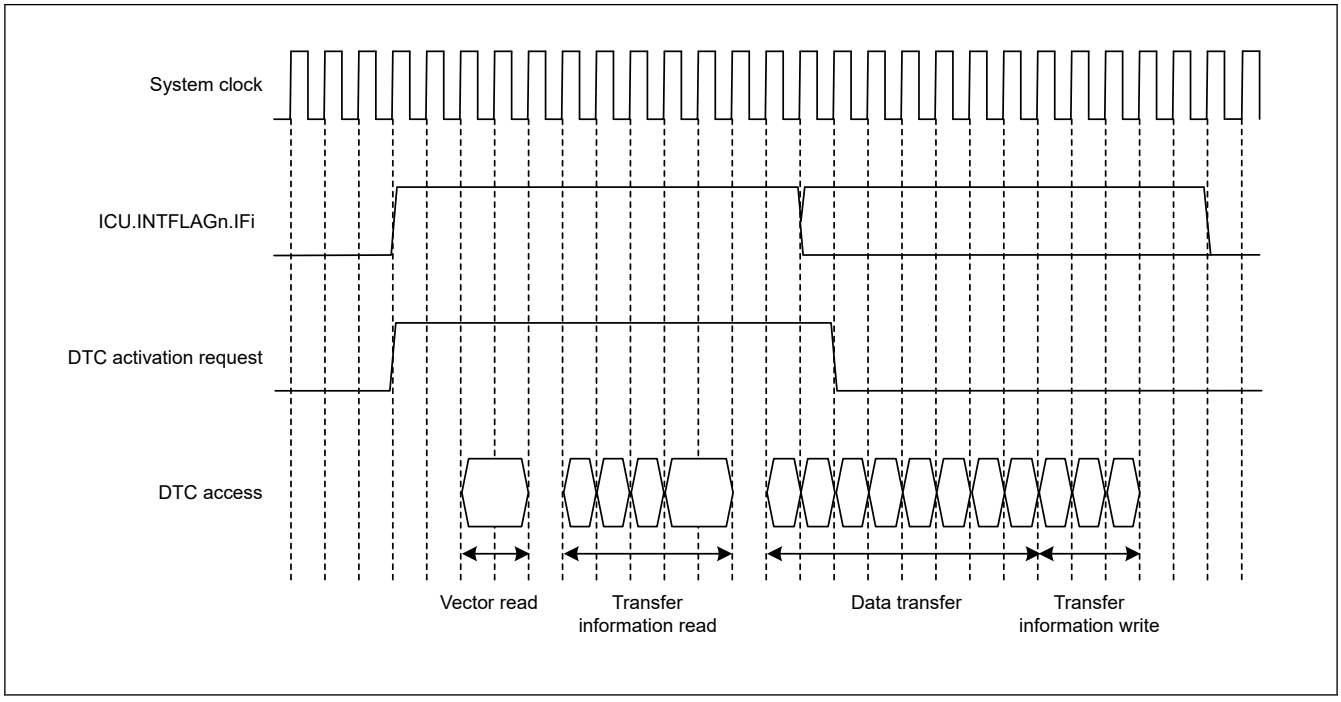


Figure 14.10 Example 2 of DTC operation timing in block transfer mode when the block size = 4

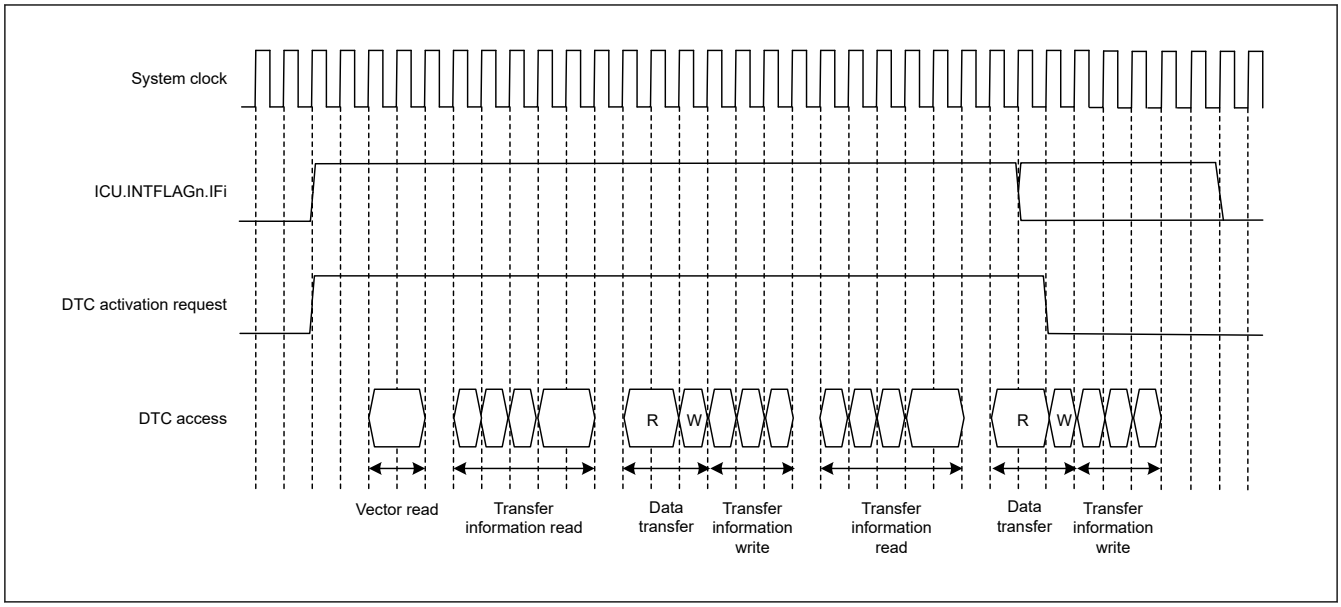
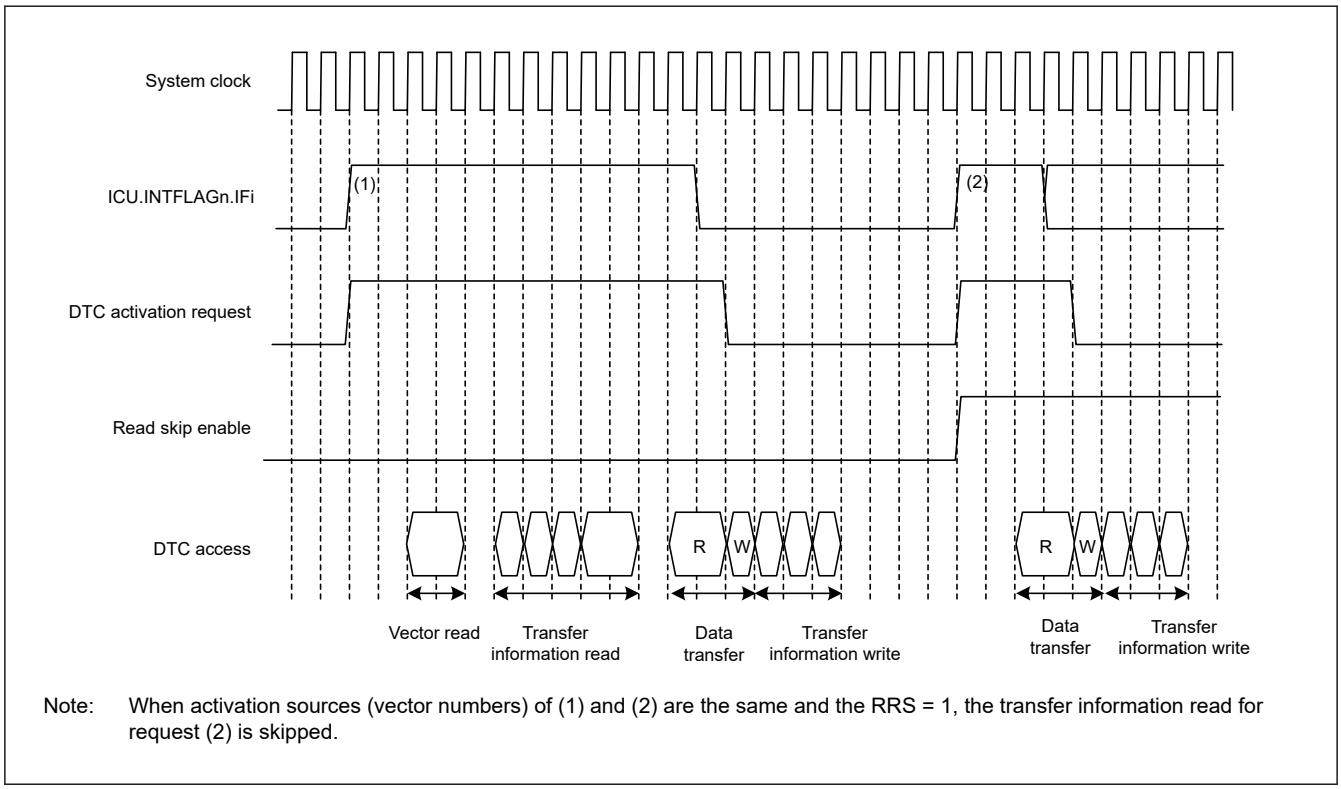


Figure 14.11 Example 3 of DTC operation timing for chain transfer



Note: When activation sources (vector numbers) of (1) and (2) are the same and the RRS = 1, the transfer information read for request (2) is skipped.

Figure 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 27, SRAM, section 28, Flash Memory, and section 12, 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 12, Buses.

14.5 DTC Setting Procedure

Before using the DTC, set the DTC Vector Base Register (DTCVBR). Set to disable the interrupt in the NVIC 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.DTCENSTx.STy bit to 1. The interrupt should be enabled in the NVIC.	Set the ICU.DTCENSTx.STy bit to 1. The interrupt must be enabled in the NVIC.
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 consecutively capturing A/D conversion results 40 times.

(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 halfword-sized transfer (MRA.SZ[1:0] = 01b). In the MRB register, specify incrementation of the destination address (MRB.DM[1:0] = 10b) and single data transfer by a single interrupt (MRB.CHNE = 0 and MRB.DISEL = 0). The MRB.DTS bit can be set to any value. Set the ADCR register address of the ADC12 in the SAR register, the start address of the SRAM area for data storage in the DAR register, and 40 (0x0028) 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 ADC12_ADI interrupt is set in the vector table for the DTC.

(3) ICU settings and DTC module activation

Set the ICU.DTCENST0.ST31 bit to 1. The interrupt must be enabled in the NVIC. Set the DTCST.DTCST bit to 1.

(4) ADC12 settings

Set the ADC12.ADUL, the ADC12.ADLL, and the ADC12.ADM2 register appropriately to enable ADC12_ADI. If ADC12_ADI does not occur, the DTC do not start the transfer. For the details of setting of the ADC12, please refer to section 25.6. A/D Converter Setup Procedure.

(5) DTC transfer

Each time A/D conversion by the ADC12 is completed, an ADC12_ADI interrupt is generated to activate the DTC. The DTC transfers the A/D conversion result from the ADCR of the ADC12 to the SRAM, after which the DAR register is incremented and the CRA register is decremented.

(6) Interrupt handling

After 40 rounds of data transfer are complete and the value in the CRA register becomes 0, an ADC12_ADI interrupt request is generated for the CPU. Complete the process in the handling routine for this interrupt.

14.6.2 Chain transfer

This section provides an example of chain transfer by the DTC and describes its use in consecutively capturing A/D conversion result and then transmitting it by UART0 of the SAU. You can use chain transfer to transfer A/D conversion results to the SRAM area and transmit it using UART0 of the SAU.

For the first of the chain transfers, normal transfer mode is specified for transfer from the ADC12.ADCR register to the SRAM area. For the second transfer, normal transfer mode is specified for transfer from the A/D conversion result in the SRAM area to the SAU0.SDR00 register. 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 second of the chain transfers, that is, transfer while MRB.CHNE = 0.

The following example shows how to use the A/D conversion end interrupt as an activating source for the DTC.

(1) First transfer information setting

Set up transfer the ADC12.ADCR register to the SRAM area.

1. In the MRA register, select the source address as fixed (MRA.SM[1:0] = 00b).
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 incrementation of the destination address (MRB.DM = 10b) and set up chain transfer (MRB.CHNE = 1 and MRB.CHNS = 0).
4. Set the SAR register to the address of the ADC12.ADCR register.

5. Set the DAR register to the address of the data table in the SRAM area where to store the A/D conversion result.
6. Set the CRA 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 SAU0.SDR00 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).
4. Set the SAR register to the first address of the data table.
5. Set the DAR register to the address of the SAU0.SDR00 register.
6. Set the CRA registers to the size of the data table. The CRB register can be set to any value.

(3) Transfer information assignment

Place the transfer information for use in the transfer to the SAU0.SDR00 immediately after the transfer control information for use in the ADC12.ADCR registers.

(4) DTC vector table

In the DTC vector table, set the address where the transfer control information for use in transfer to the ADC12.ADCR register starts.

(5) ICU setting and DTC module activation

1. Set the ICU.DTCENST0.ST31 bit associated with the A/D conversion end interrupt.
2. Set the DTCST.DTCST bit to 1.

(6) ADC12 settings

Set the ADC12.ADUL, the ADC12.ADLL, and the ADC12.ADM2 register appropriately to enable ADC12_ADI. If ADC12_ADI does not occur, the DTC do not start the transfer. For the detail of setting of the ADC12, please refer to section 25.6. A/D Converter Setup Procedure.

(7) SAU settings

Please refer to Figure 21.32 for the detailed settings of UART using the SAU.

(8) DTC transfer

Each time an A/D conversion end interrupt is generated, the value of the ADCR register is transferred to the SRAM area and the SAU0.SDR00 register. When the SAU0.SDR00 register is written, the transmission automatically starts.

(9) Interrupt handling

After the specified rounds of data transfer are complete, for example when the value in the CRA register for the ADC12 transfer becomes 0, an A/D conversion end interrupt 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 time. 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 time. 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.

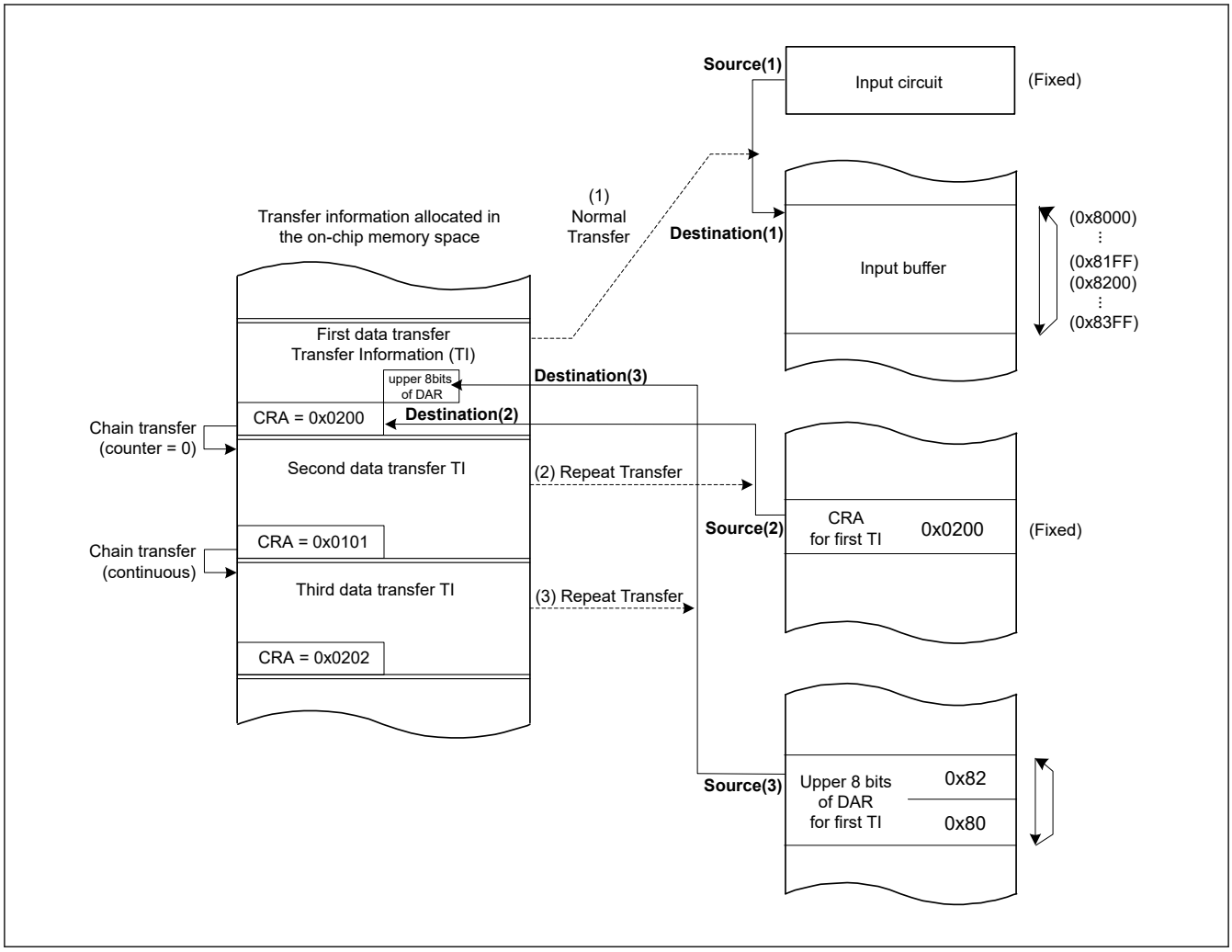


Figure 14.13 Chain transfer when counter = 0

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 NVIC. See section 11, Interrupt Controller Unit (ICU). The DTC prioritizes activation sources by granting the smaller interrupt vector numbers higher priority. The priority of interrupts to the CPU is determined by the NVIC priority.

14.8 Event Link

The DTC can produce an event link request on the completion of one transfer request.

14.9 Low Power Consumption Function

Before transitioning to the module-stop state, or Software Standby mode without Snooze mode transition, set the DTCST.DTCST bit to 0, and then perform the operations described in the following sections.

(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 9.7.1. 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) Snooze Mode

When the snooze control circuit receives a snooze request in Software Standby mode, the MCU transitions to Snooze mode. See section 9.8.1. Transition to Snooze Mode. If DTC operation is enabled in Snooze mode, before transitioning to Software Standby mode, set the DTCST.DTCST bit to 1.

Set the ICU.SBYEDCRn.xxxED bit (the bit corresponding to the interrupt that caused the DTC to activate) to 1 when transitioning to normal mode when the DTC transfer is completed.

Note: When a transfer end interrupt from the simplified SPI in SNOOZE mode is being used as the DTC activation source, use the transfer end interrupt to release the chip from the SNOOZE mode and start processing by the CPU after completion of DTC transfer, or use a chain transfer to make the settings for reception by the simplified SPI (writing 1 to the ST[0] bit, writing 0 to the SWC bit, setting the SSC0 register, and writing 1 to the SS[0] bit) again.

Note: When a transfer end interrupt from the UART in SNOOZE mode is being used as the DTC activation source, use the transfer end interrupt to release the chip from the SNOOZE mode and start processing by the CPU after completion of DTC transfer, or use a chain transfer to make the settings for reception by the UART (writing 1 to the ST[1] bit, writing 0 to the SWC bit, setting the SSC0 register, and writing 1 to the SS[1] bit) again.

Note: When an A/D conversion end interrupt from the A/D converter in SNOOZE mode is being used as the DTC activation source, use the A/D conversion end interrupt to release the chip from the SNOOZE mode and start processing by the CPU after completion of DTC transfer, or use a chain transfer to make the settings for the SNOOZE mode function of the A/D converter (writing 1 to the AWC bit after having written 0 to it) again.

(4) Notes on Low Power Consumption Function

For the WFI instruction and the register setting procedure, see section 9, Low Power Modes.

To perform a DTC transfer after returning from a low power mode without a Snooze mode transition, set the DTCST.DTCST bit to 1 again.

To use a request that is generated in Software Standby mode as an interrupt request to the CPU but not as a DTC activation request, specify the CPU as the interrupt request destination as described in section 11.4.1. Detecting Interrupts, then execute the WFI instruction. If DTC operation is enabled in Snooze mode, do not use the module-stop function of the DTC.

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	32 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.

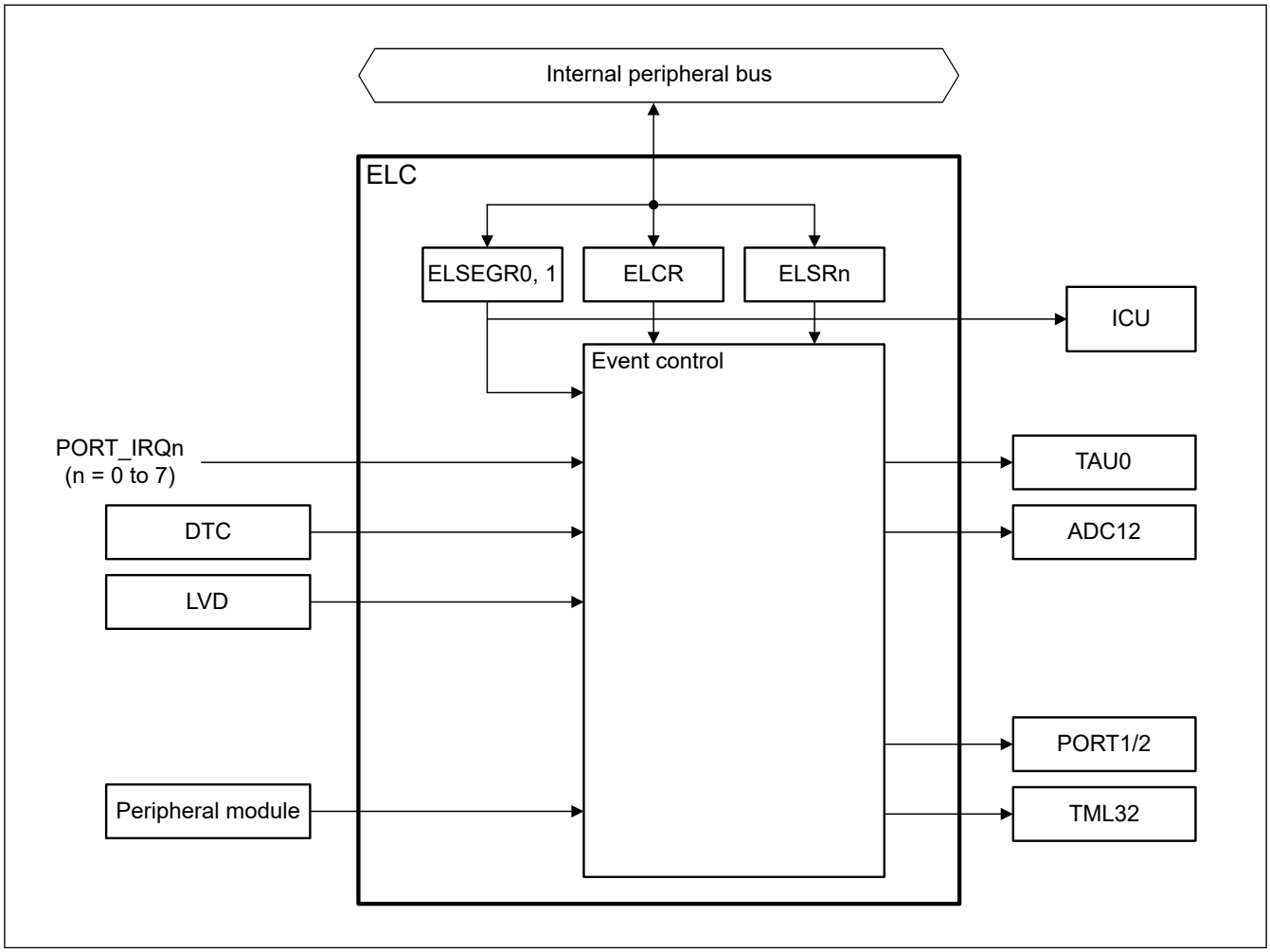


Figure 15.1 ELC block diagram

15.2 Register Descriptions

15.2.1 ELCR : Event Link Controller Register

Base address: ELC = 0x4004_1000

Offset address: 0x0000

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: All the event link operations are disabled. 1: All the event link operations are 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: 0x0002 + 0x2 × 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: Not generate a software event. 1: Generate a software event.	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 and 0 is written to the WI 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. To write this bit, write the WI bit to 0 at the same time.

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. To write this bit, write the WI bit to 0 at the same time.

[Setting condition]

- If writing the WE bit to 1 with writing the WI bit to 0, the WE bit becomes 1.

[Clearing condition]

- If writing the WE bit to 0 with writing the WI bit to 0, the WE 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. To write the WE or SEG bit, the WI bit must be written to 0 at the same time.

15.2.3 ELSRn : Event Link Setting Register n (n = 23 to 28)

Base address: ELC = 0x4004_1000

Offset address: 0x0010 + 0x4 × n

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

Bit	Symbol	Function	R/W
5:0	ELS[5:0]	Event Link Select 0x00: Event output disabled for the associated peripheral module 0x01: Number setting for the event signal to be linked ⋮ 0x20: Number setting for the event signal to be linked Others: Settings prohibited	R/W
15:6	—	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 and the event numbers set in the ELSRn register.

Table 15.2 Association between the ELSRn registers and peripheral functions

Register name	Peripheral function (module)	Event name
ELSR23	ADC12	ELC_AD
ELSR24	TAU0	ELC_TAU00*1
ELSR25	TAU0	ELC_TAU01*2
ELSR26	PORT1	ELC_PORT1
ELSR27	PORT2	ELC_PORT2
ELSR28	TML32	ELC_ITLC

Note 1. To use the timer input of timer array unit 0 channel 0 as the link destination peripheral function, set the operating clock for channel 0 to ICLK using timer clock select register 0 (TPS0), set the noise filter of the TI00 pin to OFF (TNFEN00 = 0) using noise filter enable register 1 (NFEN1), and then set the timer output used for channel 0 to an event input signal from the ELC using timer input select register 0 (TIS0).

Note 2. To use the timer input of timer array unit 0 channel 1 as the link destination peripheral function, set the operating clock for channel 1 to ICLK using timer clock select register 0 (TPS0), set the noise filter of the TI01 pin to OFF (TNFEN01 = 0) using noise filter enable register 1 (NFEN1), and then set the timer output used for channel 1 to an event input signal from the ELC using timer input select register 0 (TIS0).

Table 15.3 Association between event signal names set in ELSRn.ELS[5:0] bits and signal numbers (1 of 2)

Event number	Interrupt request source	Name	Description
0x01	PORT	PORT_IRQ0	External pin interrupt 0
0x02		PORT_IRQ1	External pin interrupt 1
0x03		PORT_IRQ2	External pin interrupt 2
0x04		PORT_IRQ3	External pin interrupt 3
0x05		PORT_IRQ4	External pin interrupt 4
0x06		PORT_IRQ5	External pin interrupt 5
0x07	RTC	RTC_ALM_OR_PRD	Fixed-cycle signal of realtime clock/alarm match detection

Table 15.3 Association between event signal names set in ELSRn.ELS[5:0] bits and signal numbers (2 of 2)

Event number	Interrupt request source	Name	Description
0x08	TML32	TML32_ITL0	32 bit interval timer0 compare-match
0x09		TML32_ITL1	32 bit interval timer1 compare-match
0x0A		TML32_ITL2	32 bit interval timer2 compare-match
0x0B		TML32_ITL3	32 bit interval timer3 compare-match
0x0C	TAU0	TAU0_TMI00	End of timer channel 00 count or capture
0x0D		TAU0_TMI01	End of timer channel 01 count or capture
0x0E		TAU0_TMI02	End of timer channel 02 count or capture
0x0F		TAU0_TMI03	End of timer channel 03 count or capture
0x10	SAU0	SAU0_UART_TXI0/ SAU0_SPI_TXRXI00/ SAU0_IIC_TXRXI00	UART0 transmission transfer end or buffer empty interrupt/ SPI00 transfer end or buffer empty interrupt/IIC00 transfer end
0x11		SAU0_UART_RXI0/ SAU0_SPI_TXRXI01/ SAU0_IIC_TXRXI01	UART0 reception transfer end/SPI01 transfer end or buffer empty interrupt/IIC01 transfer end
0x12	DTC	DTC_DTCEND*1	DTC transfer end
0x13	LVD	LVD_LVD1	Voltage monitor 1 interrupt
0x14	ELC	ELC_SWEVT0	Software event 0
0x15		ELC_SWEVT1	Software event 1
0x16	ADC12	ADC12_ADI	End of A/D conversion
0x17	UARTA0	UARTA0_ERRI	UARTA0 reception communication error occurrence
0x18		UARTA0_TXI	UARTA0 transmission transfer end or buffer empty interrupt
0x19		UARTA0_RXI	UARTA0 reception transfer end
0x1A	IICA0	IICA0_TXRXI	End of IICA0 communication
0x1B	PORT	PORT_IRQ6	External pin interrupt 6
0x1C		PORT_IRQ7	External pin interrupt 7
0x1D	UARTA1	UARTA1_ERRI	UARTA1 reception communication error occurrence
0x1E		UARTA1_TXI	UARTA1 transmission transfer end or buffer empty interrupt
0x1F		UARTA1_RXI	UARTA1 reception transfer end
0x20	IICA1	IICA1_TXRXI	End of IICA1 communication

Note 1. This event can occur in Snooze mode.

15.3 Operation

15.3.1 Relation between Interrupt Handling and Event Linking

The event number used in the ELC is different from the number used in the ICU and the DTC. 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 (1 of 2)

Module	Operations When Event is Input
ADC12	Start A/D conversion

Table 15.4 Module operations when event occurs (2 of 2)

Module	Operations When Event is Input
TAU	• Delay counter • Input pulse interval measurement • External event counter
I/O Ports	• Change pin output based on the EORR (reset) or EOSR (set) • The following ports can be used for the ELC: Port1 Port2
TML32	• Up counting • Capture trigger

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[5:0] bits for the module to be linked.
3. Set the ELCR.ELCON bit to 1 to enable linkage of all events.
4. Configure the module from which an event is output and activate the module. The link between the two modules is now active.
5. To stop event linkage of modules individually, first set the ELCR.ELCON to 0, then set 0 to the ELSRn.ELS[5:0] bits associated with the modules, finally restore the ELCR.ELCON to 1. To stop linkage of all the events, set the ELCR.ELCON bit to 0.

15.4 Usage Notes

15.4.1 Setting ELSR Register

Changing ELSRn register while some input events are active may cause a malfunction of the linking operation. Set an ELSRn register during a period when the event link operation is disabled (ELCON.ELCON = 0), all the event signals coming into the ELC are inactive, or the function of the peripheral to which the ELC outputs events is stopped.

15.4.2 Linking an Event from the same function of the Destination

Do not link an event to the same function of a module as the event is output from.

15.4.3 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.

15.4.4 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 9, Low Power Modes.

15.4.5 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 9, Low Power Modes.

15.4.6 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.

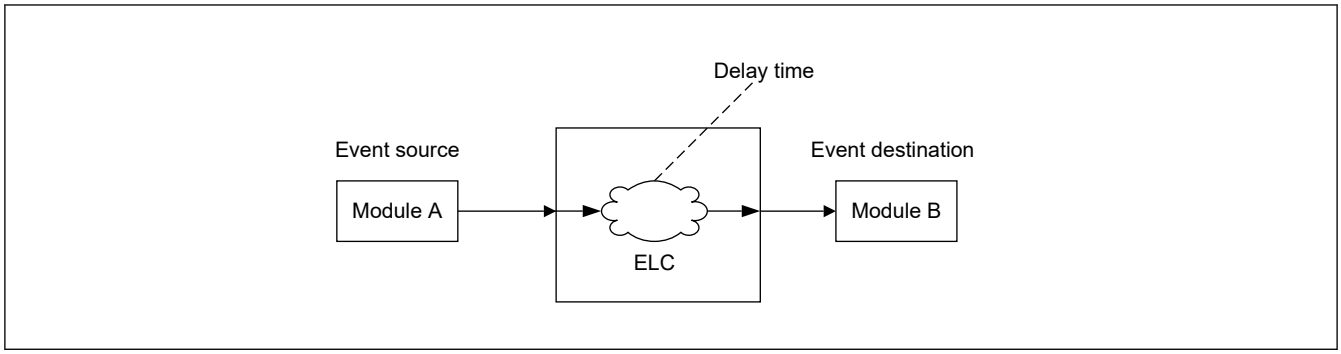


Figure 15.2 ELC delay time

Table 15.5 ELC delay time

Module	Event Name	Delay time
ADC12	ELC_AD	4 or 5 cycles of ICLK after an event input to ELC, the hardware trigger is detected in ADC12.
TAU0	ELC_TAU00	4 or 5 cycles of ICLK after an event input to ELC, the edge is detected in TAU0 channel 0.*1
TAU0	ELC_TAU01	4 or 5 cycles of ICLK after an event input to ELC, the edge is detected in TAU0 channel 1.*1
PORT1	ELC_PORT1	3 or 4 cycles of ICLK after an event input to ELC, the output of PORT1 changes.
PORT2	ELC_PORT2	3 or 4 cycles of ICLK after an event input to ELC, the output of PORT2 changes.
TML32	ELC_ITLC	There is no delay for event link to the 32-bit interval timer.

Note 1. This is the case of rising edge detection. Falling edge is detected in the following cycle of rising edge detection.

15.4.7 Link Availability in Sleep, Software Standby, and Snooze Mode

Table 15.6 shows whether ELC can link an event to each destination in each low-power mode. The availability of input events in each mode depends on the availability of its module in each mode. Please refer to Table 9.2 for the availability of modules in each mode.

Table 15.6 Link availability in each low-power mode

Module	Event Name	Sleep Mode	Software Standby Mode	Snooze Mode
ADC12	ELC_AD	Available	Not available	Using ELC event as hardware trigger of ADC12 in snooze mode is prohibited
TAU0	ELC_TAU00	Available	Not available	Using TAU in snooze mode is prohibited
TAU0	ELC_TAU01	Available	Not available	Using TAU in snooze mode is prohibited
PORT1	ELC_PORT1	Available	Not available	Available if either ADC12, DTC or SAU is working in snooze mode
PORT2	ELC_PORT2	Available	Not available	Available if either ADC12, DTC or SAU is working in snooze mode
TML32	ELC_ITLC	Available	Available	Available

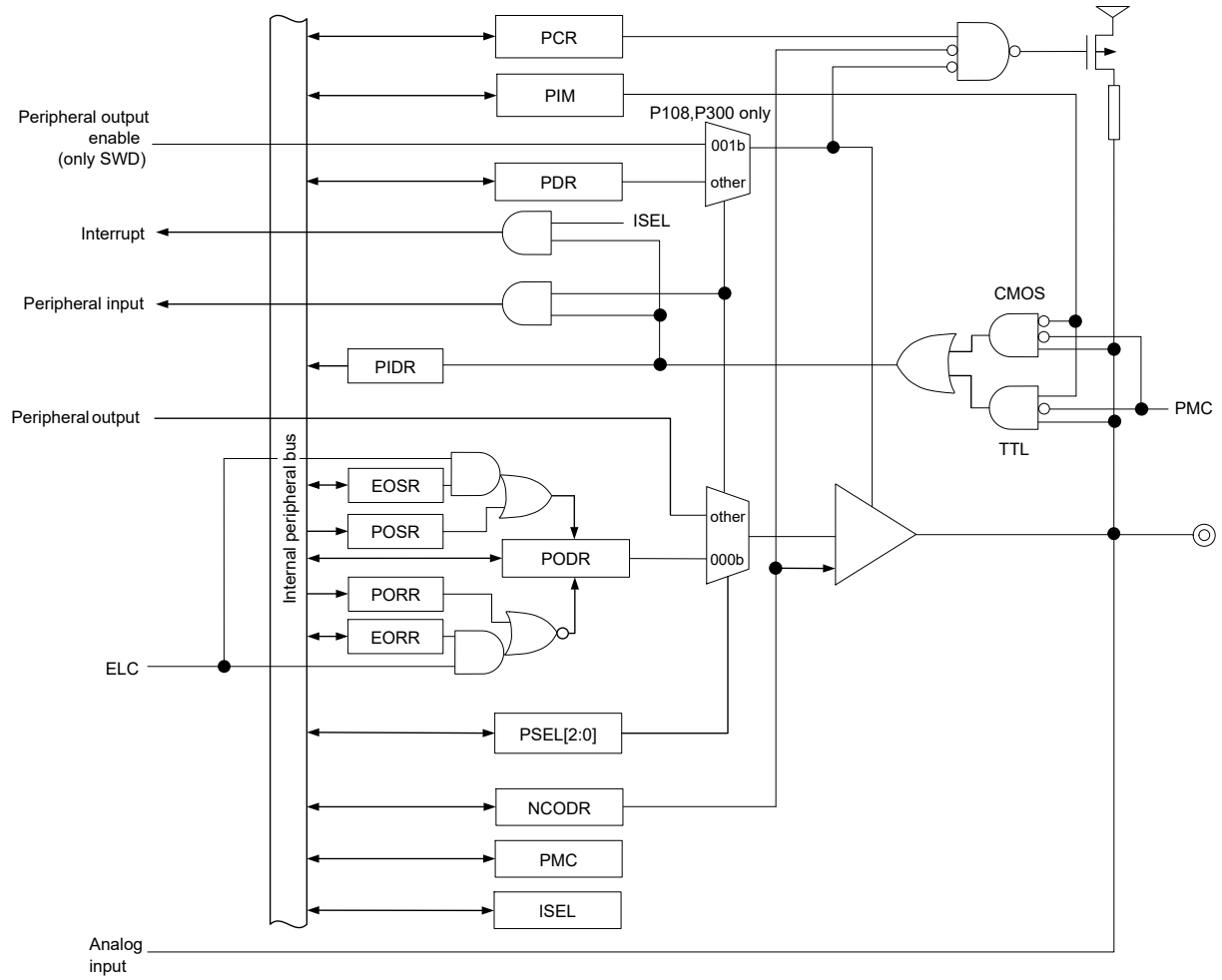
16. I/O Ports

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	
	64 pins	Number of pins	48 pins	Number of pins	32 pins	Number of pins
Port 0	P000 to P004 P008 to P015	13	P000 to P002 P008 to P015	11	P008 to P015	8

Table 16.1 I/O port specifications (2 of 2)

Port	Package		Package		Package	
	64 pins	Number of pins	48 pins	Number of pins	32 pins	Number of pins
Port 1	P100 to P115	16	P100 to P106, P108 to P112	12	P100 to P103, P108 to P110, P112	8
Port 2	P200, P201, P204 to P208, P212 to P215	11	P200, P201, P206 to P208, P212 to P215	9	P200, P201, P206 to P208, P212 to P215	9
Port 3	P300 to P304	5	P300 to P302	3	P300	1
Port 4	P400 to P403, P407 to P411	9	P400, P401, P407 to P409	5	P407	1
Port 5	P500 to P502	3	P500	1	—	0
Port 9	P913 to P915	3	P913 to P915	3	P913, P914	2

Note: —: Setting prohibited

Table 16.2 I/O port functions

Port	Port name	Input pull-up	Input mode switching	Open drain output	5V tolerant	I/O
Port 0	P000 to P004, P008 to P015	—	—	—	—	Input/Output
Port 1	P100 to P107, P109 to P115	✓	CMOS/TTL	✓	—	Input/Output
	P108	✓	CMOS/TTL	—	—	Input/Output
Port 2	P200, P214, P215	—	—	—	—	Input
	P201, P204, P205, P207, P208	✓	CMOS/TTL	✓	—	Input/Output
	P206 (32-pin product)	✓	—	—	—	Input/Output
	P206 (48-pin/64-pin products)	✓	CMOS/TTL	✓	—	Input/Output
	P212, P213	✓	—	✓	—	Input/Output
Port 3	P300	✓	CMOS/TTL	—	—	Input/Output
	P301 to P304	✓	CMOS/TTL	✓	—	Input/Output
Port 4	P400, P401 (Nch open drain)	—	—	✓	✓	Input/Output
	P402, P403, P407 to P411	✓	CMOS/TTL	✓	—	Input/Output
Port 5	P500 to P502	✓	CMOS/TTL	✓	—	Input/Output
Port 9	P913, P914 (Nch open drain)	—	—	✓	✓	Input/Output
	P915	✓	CMOS/TTL	✓	—	Input/Output

Note: ✓: Available
—: Setting prohibited

16.2 Register Descriptions

16.2.1 PODRm : Pmn Output Data Register (m = 0 to 5, 9, n = 00 to 15)

Base address: PORTm = 0x400A_0000 + 0x20 × m (m = 0 to 9)

Offset address: 0x0000

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PODR 15	PODR 14	PODR 13	PODR 12	PODR 11	PODR 10	PODR 09	PODR 08	PODR 07	PODR 06	PODR 05	PODR 04	PODR 03	PODR 02	PODR 01	PODR 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	PODR15 to PODR00	Pmn Output Data 0: Low output 1: High output	R/W

The Pmn Output Data Register (PODRm) is a 16-bit read/write register and is accessed in 16-bit units.

Note: The existent ports in this product.

- PODR0: P0n Output Data Register (n = 00 to 04, 08 to 15)
- PODR1: P1n Output Data Register (n = 00 to 15)
- PODR2: P2n Output Data Register (n = 01, 04 to 08, 12, 13)
- PODR3: P3n Output Data Register (n = 00 to 04)
- PODR4: P4n Output Data Register (n = 00 to 03, 07 to 11)
- PODR5: P5n Output Data Register (n = 00 to 02)
- PODR9: P9n Output Data Register (n = 13 to 15)

PODRn bits (Pmn Output Data)

The PODRn bits hold data to be output from the general I/O pins. Bits of non-existent port m are reserved. Reserved bits are read as 0. The write value should be 0. P200, P214, and P215 are input only, so PODR2.PODR00, PODR14, and PODR15 bits are reserved. The PODRn bit in the PODRm register serves the same function as the PODR bit in the PmnPFS_A register.

Note: In the 32-pin product, when the RES pin (OFS1.PORTSELB = 1) is selected, the PODR2.PODR06 bit (P206) is always read as 0.

16.2.2 PDRm : Pmn Direction Register (m = 0 to 5, 9, n = 00 to 15)

Base address: PORTm = 0x400A_0000 + 0x20 × m (m = 0 to 9)

Offset address: 0x0002

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PDR15	PDR14	PDR13	PDR12	PDR11	PDR10	PDR09	PDR08	PDR07	PDR06	PDR05	PDR04	PDR03	PDR02	PDR01	PDR00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	PDR15 to PDR00	Pmn Direction 0: Input (functions as an input pin) 1: Output (functions as an output pin)	R/W

The Pmn Direction Register (PDRm) is a 16-bit read/write register and is accessed in 16-bit units.

Note: The existent ports in this product.

- PDR0: P0n Direction Register (n = 00 to 04, 08 to 15)
- PDR1: P1n Direction Register (n = 00 to 15)
- PDR2: P2n Direction Register (n = 01, 04 to 08, 12, 13)
- PDR3: P3n Direction Register (n = 00 to 04)
- PDR4: P4n Direction Register (n = 00 to 03, 07 to 11)
- PDR5: P5n Direction Register (n = 00 to 02)
- PDR9: P9n Direction Register (n = 13 to 15)

PDRn bits (Pmn Direction)

The PDRn bits select the input or output direction for individual pins on the associated port when the pins are configured as general I/O pins. Each pin on port m is associated with a PDRm.PDRn bit. 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, P214, P215 are input only, so PDM2.PDR00, PDR14, and PDR15 bits are reserved. The PDRn bit in the PDRm register serves the same function as the PDRm bit in the PmnPFS_A register.

Note: In the 32-pin product, when the RES pin (OFS1.PORTSELB = 1) is selected, the PODR2.PODR06 bit (P206) is always read as 0.

16.2.3 PIDRm : Pmn State Register (m = 0 to 5, 9, n = 00 to 15)

Base address: PORTm = 0x400A_0000 + 0x20 × m (m = 0 to 9)

Offset address: 0x0006

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PIDR15	PIDR14	PIDR13	PIDR12	PIDR11	PIDR10	PIDR09	PIDR08	PIDR07	PIDR06	PIDR05	PIDR04	PIDR03	PIDR02	PIDR01	PIDR00
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
15:0	PIDR15 to PIDR00	Pmn State 0: Low level 1: High level	R

The Pmn State Register (PIDRm) is a 16-bit write register and is accessed in 16-bit units.

Note: The existent ports in this product.

- PIDR0: P0n State Register (n = 00 to 04, 08 to 15)
- PIDR1: P1n State Register (n = 00 to 15)
- PIDR2: P2n State Register (n = 00, 01, 04 to 08, 12 to 15)
- PIDR3: P3n State Register (n = 00 to 04)
- PIDR4: P4n State Register (n = 00 to 03, 07 to 11)
- PIDR5: P5n State Register (n = 00 to 02)
- PIDR9: P9n State Register (n = 13 to 15)

PIDRn bits (Pmn State)

The PIDRn bits reflect the individual pin states of the port, regardless of the values set in PDRm.PDRn. The PIDRn bit in the PIDRm register serves the same function as the PIDR bit in the PmnPFS_A register.

A pin state cannot be reflected in PIDRn when one of the following functions is enabled:

- Main clock oscillator (MOSC)
- Sub-clock oscillator (SOSC)
- Analog function (PmnPFS_A.PMC = 1)
- Input to the input buffer is disable (PmnPFS_A.PMC = 1)

Note: In the 32-pin product, when the RES pin (OFS1.PORTSELB = 1) is selected, the PIDR2.PIDR06 bit (P206) is always read as 1.

16.2.4 PORRm : Pmn Output Reset Register (m = 0 to 5, 9, n = 00 to 15)

Base address: PORTm = 0x400A_0000 + 0x20 × m (m = 0 to 9)

Offset address: 0x0008

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
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	Symbol	Function	R/W
15:0	PORR15 to PORR00	Pmn Output Reset 0: No effect on output 1: Low output	W

The Pmn Output Reset Register (PORRm) is a 16-bit write register and is accessed in 16-bit units.

Note: The existent ports in this product.

- PORR0: P0n Output Reset Register (n = 00 to 04, 08 to 15)
- PORR1: P1n Output Reset Register (n = 00 to 15)
- PORR2: P2n Output Reset Register (n = 01, 04 to 08, 12, 13)
- PORR3: P3n Output Reset Register (n = 00 to 04)
- PORR4: P4n Output Reset Register (n = 00 to 03, 07 to 11)
- PORR5: P5n Output Reset Register (n = 00 to 02)
- PORR9: P9n Output Reset Register (n = 13 to 15)

PORRn bits (Pmn Output Reset)

PORRn changes PODRn when reset by a software write. For example, for P100, when PORR1.PORR00 = 1, PODR1.PODR00 outputs 0. Bits associated with non-existent pins are reserved. The write value should always be 0.

P200, P214, and P215 are input only, so PORR2.PORR00, PORR14, and PORR15 bits are reserved.

Note: When EORRm.EORRn = 1 or EOSRm.EOSRn = 1, writing is prohibited to PODRm.PODRn, PORRm.PORRn and POSRm.POSRn.

16.2.5 POSRm : Pmn Output Set Register (m = 0 to 5, 9, n = 00 to 15)

Base address: PORTm = 0x400A_0000 + 0x20 × m (m = 0 to 9)

Offset address: 0x000A

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	POSR 15	POSR 14	POSR 13	POSR 12	POSR 11	POSR 10	POSR 09	POSR 08	POSR 07	POSR 06	POSR 05	POSR 04	POSR 03	POSR 02	POSR 01	POSR 00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	POSR15 to POSR00	Pmn Output Set 0: No effect on output 1: High output	W

The Pmn Output Set Register (POSRm) is a 16-bit write register and is accessed in 16-bit units.

Note: The existent ports in this product.

- POSR0: P0n Output Set Register (n = 00 to 04, 08 to 15)
- POSR1: P1n Output Set Register (n = 00 to 15)
- POSR2: P2n Output Set Register (n = 01, 04 to 08, 12, 13)
- POSR3: P3n Output Set Register (n = 00 to 04)
- POSR4: P4n Output Set Register (n = 00 to 03, 07 to 11)
- POSR5: P5n Output Set Register (n = 00 to 02)
- POSR9: P9n Output Set Register (n = 13 to 15)

POSRn bits (Pmn Output Set)

POSRn changes PODRn when set by a software write. For example, for P100, when POSR1.POSR00 = 1, PODR1.PODR00 outputs 1. Bits associated with non-existent pins are reserved. The write value should always be 0.

P200, P214, and P215 are input only, so POSR2.POSR00, POSR14, and POSR15 bits are reserved.

Note: When EORRm.EORRn = 1 or EOSRm.EOSRn = 1, writing is prohibited to PODRm.PODRn, PORRm.PORRn and POSRm.POSRn.

16.2.6 EORRm : Pmn Event Output Reset Register (m = 1 to 2, n = 00 to 15)

Base address: PORTm = 0x400A_0000 + 0x20 × m (m = 1 to 2)

Offset address: 0x000C

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
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	Symbol	Function	R/W
15:0	EORR15 to EORR00	Pmn Event Output Reset When an ELC_PORT1 or 2 signal occurs 0: No effect on output 1: Low output	R/W

The Pmn Event Output Reset Register (EORRm) is a 16-bit write register and is accessed in 16-bit units.

Note: The existent ports in this product.

- EORR1: P1n Event Output Reset Register (n = 00 to 15)
- EORR2: P2n Event Output Reset Register (n = 01, 04 to 08, 12, 13)

EORRn bits (Pmn Event Output Reset)

EORRn changes PODRn when reset because an ELC_PORT1 or 2 signal occurs. For example, for P100 if EORR1.EORR00 = 1 when the ELC_PORT1 or 2 occurs, PODR1.PODR00 outputs 0. Bits associated with non-existent pins are reserved. The write value should always be 0. P200, P214, and P215 are input only, so EORR2.EORR00, EORR14, and EORR15 bits are reserved.

Note: When EORRm.EORRn = 1 or EOSRm.EOSRn = 1, writing is prohibited to PODRm.PODRn, PORRm.PORRn and POSRm.POSRn.

16.2.7 EOSRm : Pmn Event Output Set Register (m = 1 to 2, n = 00 to 15)

Base address: PORTm = 0x400A_0000 + 0x20 × m (m = 1 to 2)

Offset address: 0x000E

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	EOSR 15	EOSR 14	EOSR 13	EOSR 12	EOSR 11	EOSR 10	EOSR 09	EOSR 08	EOSR 07	EOSR 06	EOSR 05	EOSR 04	EOSR 03	EOSR 02	EOSR 01	EOSR 00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	EOSR15 to EOSR00	Pmn Event Output Set When an ELC_PORT1 or 2 signal occurs 0: No effect on output 1: High output	R/W

Note: The existent ports in this product.

- EOSR1 : P1n Event Output Set Register (n = 00 to 15)
- EOSR2 : P2n Event Output Set Register (n = 01, 04 to 08, 12, 13)

The Pmn Event Output Register (EOSRm) is a 16-bit write register and is accessed in 16-bit units.

EOSRn bits (Pmn Event Output Set)

EOSRn changes PODRn when set because an ELC_PORT1 or 2 signal occurs. For example, for P100 if EOSR1.EOSR00 = 1 when the ELC_PORT1 or 2 occurs, PODR1.PODR00 outputs 0. Bits associated with non-existent pins are reserved. The write value should always be 0. P200, P214, and P215 are input only, so EOSR2.EOSR00, EOSR14, and EOSR15 bits are reserved.

Note: When EORRm.EORRn = 1 or EOSRm.EOSRn = 1, writing is prohibited to PODRm.PODRn, PORRm.PORRn and POSRm.POSRn.

16.2.8 PmnPFS_A : Port mn Pin Function Select Register (m = 1 to 9, n = 00 to 15)

Base address: PFS_A = 0x400A_0200

Offset address: 0x0000 + 0x20 × m + 0x2 × n

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PMC	ISEL	—	—	—	PSEL[2:0]			—	NCODR	PIM	PCR	—	PDR	PIDR	PODR
Value after reset:	0*1	0	0	0	0	0	0	0*1	0	0	0	0*1	0	0	x	0

Bit	Symbol	Function	R/W
0	PODR	Pmn Output Data 0: Low output 1: High output	R/W
1	PIDR	Pmn State 0: Low level 1: High level	R
2	PDR	Pmn 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	PIM	Pin Input Buffer Selection 0: Normal input buffer 1: TTL input buffer	R/W
6	NCODR	N-channel Open-drain Control 0: CMOS output 1: NMOS open-drain output	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
10:8	PSEL[2:0]	Peripheral Select These bits select the peripheral function. For individual pin functions, see the associated tables in this chapter.	R/W
13:11	—	These bits are read as 0. The write value should be 0.	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	PMC	Pin Mode Control 0: Digital I/O 1: Analog input function. Input to the input buffer is disabled	R/W

Note 1. The initial value of P100, P101, P108, P206 and P300 is not 0x0000. The initial value of P100 is 0x8000, P101 is 0x8000, P108 is 0x0110, P206 is 0x0010 and P300 is 0x0110.

Port mn Pin Function Select Register (PmnPFS_A) 16-bit read/write control register that selects the port mn pin function, and is accessed in 16-bit units.

Note: The existent ports in this product:

- P1nPFS_A: Port 1n Pin Function Select Register (n = 00 to 15)
- P2nPFS_A: Port 2n Pin Function Select Register (n = 00, 01, 04 to 08, 12 to 15)
- P3nPFS_A: Port 3n Pin Function Select Register (n = 00 to 04)
- P4nPFS_A: Port 4n Pin Function Select Register (n = 02, 03, 07 to 11)
- P5nPFS_A: Port 5n Pin Function Select Register (n = 00 to 02)
- P9nPFS_A: Port 9n Pin Function Select Register (n = 15)

PODR bit (Pmn Output Data), PIDR bit (Pmn State), PDR bit (Pmn Direction)

The PDR, PIDR, and PODR bits serve the same function as the PDRm, PIDRm, PODRm. When these bits are read, the PDRm, PIDRm, PODRm 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_A.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, the pull-up resistor for the pin is disabled regardless of the PCR setting. The pull-up resistor is also disabled in the reset state. Bits associated with non-existent pins are reserved. Reserved bits are read as 0. The write value should be 0.

P200, P214 and P215 do not have PCR bit, so the PCR bits in these Port Pin Function register are reserved.

Note: In the 32-pin product, PCR can be selected when P206 (OFS1.PORTSELB = 0) is selected. When the RES pin (OFS1.PORTSELB = 1) is selected, the input pull-up resistor is enable.

PIM bit (Pin Input Buffer Selection)

The PIM bit set the input buffer. TTL input buffer can be selected during serial communication with an external device operating at a different voltage. Bits associated with non-existent pins are reserved. Reserved bits are read as 0. The write value should be 0.

P200, P206 (32-pin product), P212, P213, P214 and P215 do not have PIM bit, so the PIM bits in these Port Pin Function register are reserved.

NCODR bit (N-channel Open-drain Control)

The NCODR bit specifies the output type for the port pins. N-ch open drain output (withstand voltage of VCC) mode can be selected during serial communication with an external device operating at a different voltage, and the SDA00, SDA01, SDA10, SDA11, SDA20 and SDA21 pins during simplified I2C communication with an external device operating at the same voltage. In addition, NCODR are used in combination with PCR to specify whether to use on-chip pull-up resistors. An on-chip pull-up resistor is not connected with a bit for which N-ch open drain output (withstand voltage of VCC) mode (NCODR = 1) is set. Bits associated with non-existent pins are reserved. Reserved bits are read as 0. The write value should be 0.

P108, P200, P206 (32-pin product), P214, P215 and P300 do not have NCODR bit, so the NCODR bits in these Port Pin Function register are reserved.

PSEL[2:0] bits (Peripheral Select)

The PSEL[2:0] bits assign the peripheral function. For details on the peripheral settings for each product, see section 16.6. Peripheral Select Settings for Each Product.

Bits associated with non-existent pins are reserved. Reserved bits are read as 0. The write value should be 0.

P200, P206 (32-pin product), P214 and P215 do not have PSEL[2:0] bits, so the PSEL[2:0] bits in these Port Pin Function register are reserved.

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.

Bits associated with non-existent pins are reserved. Reserved bits are read as 0. The write value should be 0.

P108, P113 to P115, P204, P206 (32-pin product), P214, P215, P300, P303, P304, P500, P501 and P915 do not have ISEL bit, so the ISEL bits in these Port Pin Function register are reserved.

PMC bit (Pin Mode Control)

The PMC bits specify the digital I/O or analog input function. When a pin is set as an analog pin by this bit:

1. Specify it as a general I/O port in the PSEL[2:0] bits.
2. Disable the pull-up resistor in the Pull-up Control bit (PmnPFS_A.PCR)
3. Specify the input in the Port Direction bit (PmnPFS_A.PDR). The pin state cannot be read at this point. The PmnPFS_A register is protected by the Write-Protect Register (PWPR). Release write-protect before modifying the register.

In addition, the PMC bit is used to prevent through-current flowing into input buffers.

When N-ch open drain output is selected for serial communications with an external device operating at a different voltage or an input port is not used, low power consumption can be achieved by setting the corresponding PMC bit to 1.

Bits associated with non-existent pins are reserved. Reserved bits are read as 0. The write value should be 0.

P206 (32-pin product), P214 and P215 do not have PMC bit, so the PMC bits in these Port Pin Function register are reserved.

Note: For P214 and P215, low power consumption can be achieved by setting the MOSEL[1:0] = 01b in the clock operation mode control register (CMC) and setting the SOSTP = 1 in the sub-clock oscillator control register (SOSCCR).

The ISEL bit for an unspecified IRQn is reserved.

16.2.9 P0nPFS_A : Port 0n Pin Function Select Register (n = 00 to 04, 08 to 15)

Base address: PFS_A = 0x400A_0200

Offset address: 0x0000 + 0x2 × n

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PMC	ISEL	—	—	—	PSEL[2:0]			—	—	—	—	—	PDR	PIDR	PODR
Value after reset:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	x	0

Bit	Symbol	Function	R/W
0	PODR	P0n Output Data 0: Low output 1: High output	R/W
1	PIDR	P0n State 0: Low level 1: High level	R
2	PDR	P0n Direction 0: Input (functions as an input pin) 1: Output (functions as an output pin)	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
10:8	PSEL[2:0]	Peripheral Select These bits select the peripheral function. For individual pin functions, see the associated tables in this chapter.	R/W
13:11	—	These bits are read as 0. The write value should be 0.	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	PMC	Pin Mode Control 0: Digital I/O 1: Analog input function. Input to the input buffer is disable	R/W

16.2.10 P40nPFS_A : Port 40n Pin Function Select Register (n = 0, 1)

Base address: PFS_A = 0x400A_0200

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:	PMC	—	—	—	—	PSEL[2:0]			—	—	—	—	—	PDR	PIDR	PODR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	0

Bit	Symbol	Function	R/W
0	PODR	P4n Output Data 0: Low output 1: High output	R/W
1	PIDR	P4n State 0: Low level 1: High level	R
2	PDR	P4n Direction 0: Input (functions as an input pin) 1: Output (functions as an output pin)	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
10:8	PSEL[2:0]	Peripheral Select These bits select the peripheral function. For individual pin functions, see the associated tables in this chapter.	R/W
14:11	—	These bits are read as 0. The write value should be 0.	R/W
15	PMC	Pin Mode Control 0: Digital I/O 1: Analog input function. Input to the input buffer is disable	R/W

16.2.11 P9nPFS_A : Port 9n Pin Function Select Register (n = 13, 14)

Base address: PFS_A = 0x400A_0200

Offset address: 0x0120 + 0x2 × n

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PMC	—	—	—	—	PSEL[2:0]			—	—	—	—	—	PDR	PIDR	PODR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	0

Bit	Symbol	Function	R/W
0	PODR	P9n Output Data 0: Low output 1: High output	R/W
1	PIDR	P9n State 0: Low level 1: High level	R
2	PDR	P9n Direction 0: Input (functions as an input pin) 1: Output (functions as an output pin)	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
10:8	PSEL[2:0]	Peripheral Select These bits select the peripheral function. For individual pin functions, see the associated tables in this chapter.	R/W
14:11	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
15	PMC	Pin Mode Control 0: Digital I/O 1: Analog input function. Input to the input buffer is disable	R/W

16.2.12 PWPR : Write-Protect Register

Base address: PFS_A = 0x400A_0200

Offset address: 0x0140

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_A Register Write Enable 0: Writing to the PmnPFS_A register is disabled 1: Writing to the PmnPFS_A 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_A Register Write Enable)

Writing to the PmnPFS_A 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.

16.3 Operation

16.3.1 General I/O Ports

All pins except P108 and P300 operate as general I/O ports after reset. General I/O ports are organized as 16 bits per port and can be accessed by PODRm, PDRm, PIDRm, PORRm, POSRm, EORRm, EOSRm 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:

- Pmn Direction bit (PDRn), which selects input or output direction
- Pmn Output Data bit (PODRn), which holds data for output
- Pmn Input Data bit (PIDRn), which indicates the pin states
- Pmn Output Set bit (POSRn), which indicates the output value when a software write occurs
- Pmn 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 NMOS open-drain output, 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_A), which includes the associated PODR, PIDR, and PDR bits. In addition, the PmnPFS_A 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
- PIM: Pin input buffer selection bit to specify a normal input or TTL input buffer
- ISEL: IRQ input enable bit to specify an IRQ input pin
- PMC: Pin mode control bit to specify an analog pin and prevent through-current flowing into input buffers
- PSEL[2:0]: 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 16.2.8. PmnPFS_A : Port mn Pin Function Select Register (m = 1 to 9, n = 00 to 15) to section 16.2.11. P9nPFS_A : Port 9n Pin Function Select Register (n = 13, 14).

16.3.3 Port Group Function for ELC

In the MCU, Port 1 and Port 2 are assigned for the ELC port group function.

16.3.3.1 Behavior When ELC_PORT1 or 2 is Input from ELC

The MCU supports the following functions described in this section when an ELC_PORT1 or 2 signal comes from the ELC.

(1) 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 an ELC_PORT1 or 2 signal occurs, the PODR register outputs 1 to the external pin. Otherwise, when EOSR = 0, the PODR value is retained.
- If EORR is set to 1, when an ELC_PORT1 or 2 signal occurs, the PODR register outputs 0 to the external pin. Otherwise, when EORR = 0, the PODR value is retained.

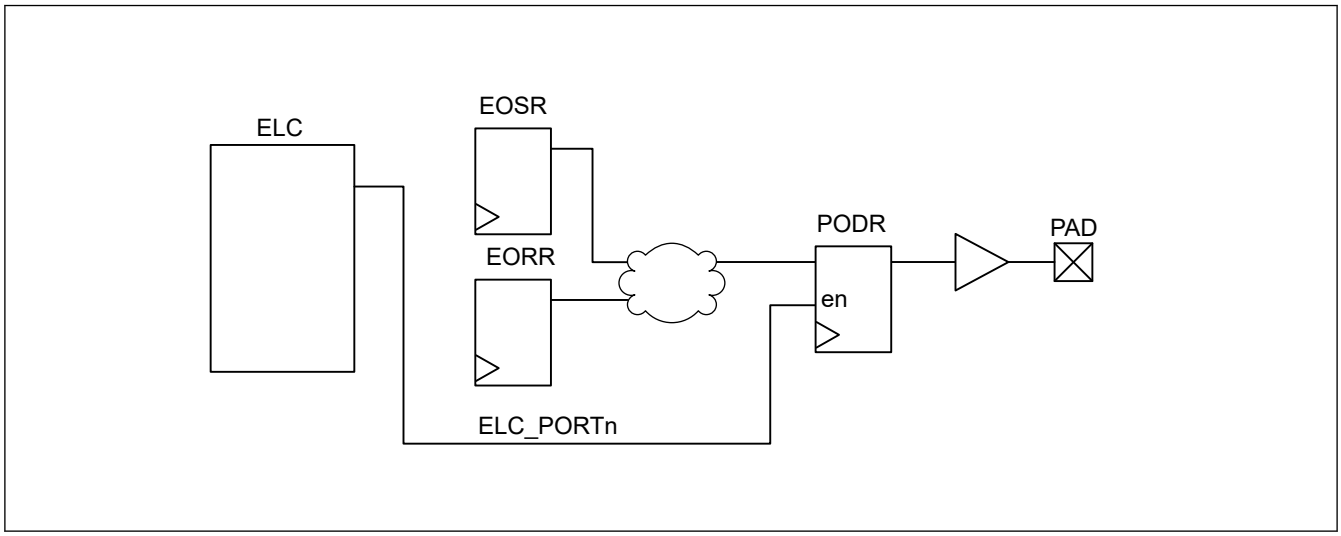


Figure 16.2 Event ports output data

Note: When linking the PORT with events, do not turn off ELC setting or rewrite PODR register with CPU while ELC event is being transferred.

Note: A glitch may occur when ELC.ELSR changes the setting to disable event output after setting the link.

16.4 Handling of Unused Pins

Table 16.3 shows how to handle unused pins.

Table 16.3 Handling of unused pins

Pin name	Description
P206/RES (32-pin product)	PORTSELB = 0: ● If the direction setting is for input (PDR2.PDR06 = 0), connect the associated pin to VCC (pulled up) through a resistor.*2 ● If the direction setting is for output (PDR2.PDR06 = 1), leave the pin open. PORTSELB = 1: ● Leave open, or connect to VCC.
P200/NMI	Set in Pin Mode Control bit (PmnPFS_A.PMC) to 1, and release the pin. Alternatively, connect the associated pin to VCC (pulled up) through a resistor or to VSS (pulled down) through a resistor.
P212/X1	When the main clock oscillator is not used, set the MOSCCR.MOSTP bit to 1 (general port P212). When this pin is not used as port P212, configure it in the same way as ports 0 to 4. When the external clock is input to the EXCLK pin, leave this pin open.
P213/X2/EXCLK	When the main clock oscillator is not used, set the MOSCCR.MOSTP bit to 1 (general port P213). When this pin is not used as port P213,it is configured in the same way as ports 0 to 4.
P215/XCIN	When the sub-clock oscillator is not used, set the SOSCCR.SOSTP bit to 1 (general port P215). When this pin is not used as port P215, configure it in the same way as ports 0 to 4.
P214/XCOUT	When the sub-clock oscillator is not used, set the SOSCCR.SOSTP bit to 1 (general port P214). When this pin is not used as port P214, configure it in the same way as ports 0 to 4.
P0x to P9x (except P400, P401, P913 and P914)	● If the direction setting is for input (PDRm.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 (PDRm.PDRn = 1), release the pin*1
P400, P401, P913, P914	● If the direction setting is for input (PDR9.PDRn = 0), connect the associated pin to VCC (pulled up) through a resistor or to VSS (pulled down) through a resistor. ● If the direction setting is for output (PDR9.PDRn = 1), set the port's output latch to 0 and leave the pins open-circuit, or set the port's output latch to 1 and independently connect the pins to VCC or VSS through resistors.

Note 1. Clear the PmnPFS_A.PSEL[2:0], PmnPFS_A.ISEL, PmnPFS_A.PCR, and PmnPFS_A.PMC bits to 0.
Note 2. P108, P206, and P300 are recommended to be pulled up to VCC through a resistor, as they are configured with input pull-up enabled (PmnPFS_A.PCR = 1) by default.

16.5 Usage Notes

16.5.1 Procedure for Specifying the Pin Functions

To specify the I/O pin functions:

1. Clear the B0WI bit in the PWPR register. This enables writing to the PFSWE bit in the PWPR register.
2. Set 1 to the PFSWE bit in the PWPR register. This enables writing to the PmnPFS_A register.
3. Specify the I/O function for the pin through the PSEL[2:0] bits settings in the PmnPFS_A register.
4. Clear the PFSWE bit in the PWPR register. This disables writing to the PmnPFS_A register.
5. Set 1 to the B0WI bit in the PWPR register. This disables writing to the PFSWE bit in the PWPR register.

16.5.2 Port Output Data Register (PODR) Summary

This register outputs data as follows:

1. Outputs 0 if EORRm.EORRn is set to 1 when ELC_PORT1 or 2 signal occurs.
2. Outputs 1 if EOSRm.EOSRn is set to 1 when ELC_PORT1 or 2 signal occurs.
3. Outputs 0 if PORRm.PORRn is set to 1.
4. Outputs 1 if POSRm.POSRn is set to 1.
5. Outputs 0 or 1 because PODRm.PODRn is set.
6. Outputs 0 or 1 because PmnPFS_A.PODRn is set.

Numbers in this list correspond to the priority for writing to the PODRn. For example, if 1. and 3. from the list occur at the same time, the higher priority event 1. is executed.

16.5.3 Notes on Register Settings and Port Pin State

The correspondence between register settings and port pin state is shown in Table 16.4.

Table 16.4 Correspondence between register settings and port pin state

PDR	PMC	PCR	PODR	Pin State
0	1	0	x	Analog input /digital input disable
0	1	1	x	Pulled up
0	0	0	x	Digital Input
0	0	1	x	Pulled up
1	1	x	1	High-level port output/digital input disable
1	0	x	1	High-level port output
1	1	x	0	Low-level port output/digital input disable
1	0	x	0	Low-level port output

16.5.4 Notes on Using Analog Functions

To use an analog function, set the Pin Mode Control bit (PMC) to 1, the N-Channel Open-Drain Control bit (NCODR) to 0, and the Port Direction bit (PDR) to 0 in the Port mn Pin Function Select Register PmnPFS_A.

16.5.5 Notes on Using Alternate Functions

To use alternate functions, set PmnPFS_A register. Examples of register settings for port and alternate functions are shown in below table. The registers used to control the port functions should be set as shown in below table.

Table 16.5 Examples of register settings for port and alternate functions (1/6) (1 of 2)

Pin Name	Function Used		PSEL[2:0]	ISEL	PMC	PDR	PODR
	Function Name	I/O					
P000	P000	Input	000b	0	0	0	x
		Output	000b	0	x	1	0/1
	AN008	Analog Input	000b	0	1	0	x
	IRQ6	Input	000b	1	0	0	x
P001	P001	Input	000b	0	0	0	x
		Output	000b	0	x	1	0/1
	AN009	Analog Input	000b	0	1	0	x
	IRQ7	Input	000b	1	0	0	x
P002	P002	Input	000b	0	0	0	x
		Output	000b	0	x	1	0/1
	AN010	Analog Input	000b	0	1	0	x
	IRQ7	Input	000b	1	0	0	x
P003	P003	Input	000b	—	0	0	x
		Output	000b	—	x	1	0/1
	AN011	Analog Input	000b	—	1	0	x
P004	P004	Input	000b	0	0	0	x
		Output	000b	0	x	1	0/1
	AN012	Analog Input	000b	0	1	0	x
	IRQ2	Input	000b	1	0	0	x

Table 16.5 Examples of register settings for port and alternate functions (1/6) (2 of 2)

Pin Name	Function Used		PSEL[2:0]	ISEL	PMC	PDR	PODR
	Function Name	I/O					
P008	P008	Input	000b	—	0	0	x
		Output	000b	—	x	1	0/1
	AN002	Analog Input	000b	—	1	0	x
P009	P009	Input	000b	—	0	0	x
		Output	000b	—	x	1	0/1
	AN003	Analog Input	000b	—	1	0	x
P010	P010	Input	000b	—	0	0	x
		Output	000b	—	x	1	0/1
	AN000	Analog Input	000b	—	1	0	x
P011	P011	Input	000b	—	0	0	x
		Output	000b	—	x	1	0/1
	AN001	Analog Input	000b	—	1	0	x
P012	P012	Input	000b	—	0	0	x
		Output	000b	—	x	1	0/1
	AN004	Analog Input	000b	—	1	0	x
P013	P013	Input	000b	—	0	0	x
		Output	000b	—	x	1	0/1
	AN005	Analog Input	000b	—	1	0	x
P014	P014	Input	000b	—	0	0	x
		Output	000b	—	x	1	0/1
	AN006	Analog Input	000b	—	1	0	x
P015	P015	Input	000b	0	0	0	x
		Output	000b	0	x	1	0/1
	AN007	Analog Input	000b	0	1	0	x
	IRQ1	Input	000b	1	0	0	x

Table 16.6 Examples of register settings for port and alternate functions (2/6) (1 of 5)

Pin Name	Function Used		PSEL[2:0]	ISEL	NCODR	PMC	PDR	PODR
	Function Name	I/O						
P100	P100	Input	000b	0	0	0	0	x
		Output	000b	0	0	x	1	0/1
		N-ch open drain output	000b	0	1	x	1	0/1
	AN022	Analog Input	000b	0	0	1	0	x
	TI04	Input	001b	0	0	0	0	x
	TO04	Output	001b	0	0	x	1	x
	TI01	Input	010b	0	0	0	0	x
	TO01	Output	010b	0	0	x	1	x
	RXD0	Input	011b	0	x	0	0	x
	SI00	Input	011b	0	x	0	0	x
	SDA00	I/O	011b	0	1	0	1	x
	SCLA0	I/O	100b	0	1	0	1	x
	RXDA0	Input	101b	0	x	0	0	x
	IRQ2	Input	000b	1	0	0	0	x
P101	P101	Input	000b	0	0	0	0	x
		Output	000b	0	0	x	1	0/1
		N-ch open drain output	000b	0	1	x	1	0/1
	AN021	Analog Input	000b	0	0	1	0	x
	TI07	Input	001b	0	0	0	0	x
	TO07	Output	001b	0	0	x	1	x
	TI00	Input	010b	0	0	0	0	x
	TXD0	Output	011b	0	0/1	0/1	1	x
	SO00	Output	011b	0	0/1	0/1	1	x
	SDAA0	I/O	100b	0	1	0	1	x
	TXDA0	Output	101b	0	0/1	0/1	1	x
	IRQ3	Input	000b	1	0	0	0	x
P102	P102	Input	000b	0	0	0	0	x
		Output	000b	0	0	x	1	0/1
		N-ch open drain output	000b	0	1	x	1	0/1
	TI06	Input	001b	0	0	0	0	x
	TO06	Output	001b	0	0	x	1	x
	TO00	Output	010b	0	0	x	1	x
	SCK00	Input	011b	0	x	0	0	x
		Output	011b	0	0/1	0/1	1	x
	SCL00	Output	011b	0	0/1	0/1	1	x
	RTCOUT	Output	100b	0	0	x	1	x
	PCLBUZ0	Output	101b	0	0	x	1	x
	IRQ4	Input	000b	1	0	0	0	x
	SCLA1	I/O	110b	0	1	0	1	x
	RXDA1	Input	111b	0	x	0	0	x

Table 16.6 Examples of register settings for port and alternate functions (2/6) (2 of 5)

Pin Name	Function Used		PSEL[2:0]	ISEL	NCODR	PMC	PDR	PODR
	Function Name	I/O						
P103	P103	Input	000b	0	0	0	0	x
		Output	000b	0	0	x	1	0/1
		N-ch open drain output	000b	0	1	x	1	0/1
	TI05	Input	001b	0	0	0	0	x
	TO05	Output	001b	0	0	x	1	x
	SSI00	Input	010b	0	0	0	0	x
	IRQ5	Input	000b	1	0	0	0	x
	SDAA1	I/O	110b	0	1	0	1	x
	TXDA1	Output	111b	0	0/1	0/1	1	x
P104	P104	Input	000b	0	0	0	0	x
		Output	000b	0	0	x	1	0/1
		N-ch open drain output	000b	0	1	x	1	0/1
	TI02	Input	001b	0	0	0	0	x
	TO02	Output	001b	0	0	x	1	x
	TI00	Input	010b	0	0	0	0	x
	SCK10	Input	011b	0	x	0	0	x
		Output	011b	0	0/1	0/1	1	x
	SCL10	Output	011b	0	0/1	0/1	1	x
	IRQ6	Input	000b	1	0	0	0	x
P105	P105	Input	000b	0	0	0	0	x
		Output	000b	0	0	x	1	0/1
		N-ch open drain output	000b	0	1	x	1	0/1
	TI01	Input	001b	0	0	0	0	x
	TO01	Output	001b	0	0	x	1	x
	TO00	Output	010b	0	0	x	1	x
	SI10	Input	011b	0	x	0	0	x
	SDA10	I/O	011b	0	1	0	1	x
	RXDA1	Input	100b	0	x	0	0	x
	IRQ1	Input	000b	1	0	0	0	x
P106	P106	Input	000b	0	0	0	0	x
		Output	000b	0	0	x	1	0/1
		N-ch open drain output	000b	0	1	x	1	0/1
	SO10	Output	011b	0	0/1	0/1	1	x
	TXDA1	Output	100b	0	0/1	0/1	1	x
	IRQ0	Input	000b	1	0	0	0	x
P107	P107	Input	000b	0	0	0	0	x
		Output	000b	0	0	x	1	0/1
		N-ch open drain output	000b	0	1	x	1	0/1
	IRQ7	Input	000b	1	0	0	0	x

Table 16.6 Examples of register settings for port and alternate functions (2/6) (3 of 5)

Pin Name	Function Used		PSEL[2:0]	ISEL	NCODR	PMC	PDR	PODR
	Function Name	I/O						
P108	P108	Input	000b	—	—	0	0	x
		Output	000b	—	—	x	1	0/1
	SWDIO	I/O	001b	—	—	x	x	x
	TI03	Input	010b	—	—	0	0	x
	TO03	Output	010b	—	—	x	1	x
P109	P109	Input	000b	0	0	0	0	x
		Output	000b	0	0	x	1	0/1
		N-ch open drain output	000b	0	1	x	1	0/1
	TI02	Input	001b	0	0	0	0	x
	TO02	Output	001b	0	0	x	1	x
	TXD2	Output	010b	0	0/1	0/1	1	x
	SO20	Output	010b	0	0/1	0/1	1	x
	SDAA0	I/O	011b	0	1	0	1	x
	TXDA0	Output	100b	0	0/1	0/1	1	x
	IRQ4	Input	000b	1	0	0	0	x
	PCLBUZ1	Output	101b	0	0	x	1	x
P110	P110	Input	000b	0	0	0	0	x
		Output	000b	0	0	x	1	0/1
		N-ch open drain output	000b	0	1	x	1	0/1
	TI01	Input	001b	0	0	0	0	x
	TO01	Output	001b	0	0	x	1	x
	RXD2	Input	010b	0	x	0	0	x
	SI20	Input	010b	0	x	0	0	x
	SDA20	I/O	010b	0	1	0	1	x
	SCLA0	I/O	011b	0	1	0	1	x
	RXDA0	Input	100b	0	x	0	0	x
	IRQ3	Input	000b	1	0	0	0	x
P111	P111	Input	000b	0	0	0	0	x
		Output	000b	0	0	x	1	0/1
		N-ch open drain output	000b	0	1	x	1	0/1
	TI07	Input	001b	0	0	0	0	x
	TO07	Output	001b	0	0	x	1	x
	IRQ1	Input	000b	1	0	0	0	x

Table 16.6 Examples of register settings for port and alternate functions (2/6) (4 of 5)

Pin Name	Function Used		PSEL[2:0]	ISEL	NCODR	PMC	PDR	PODR
	Function Name	I/O						
P112	P112	Input	000b	0	0	0	0	x
		Output	000b	0	0	x	1	0/1
		N-ch open drain output	000b	0	1	x	1	0/1
	TI03	Input	001b	0	0	0	0	x
	TO03	Output	001b	0	0	x	1	x
	SCK20	Input	010b	0	x	0	0	x
		Output	010b	0	0/1	0/1	1	x
	SCL20	Output	010b	0	0/1	0/1	1	x
	SSI00	Input	011b	0	0	0	0	x
	IRQ2	Input	000b	1	0	0	0	x
P113	P113	Input	000b	0	0	0	0	x
		Output	000b	0	0	x	1	0/1
		N-ch open drain output	000b	0	1	x	1	0/1
	SO21	Output	001b	0	0/1	0/1	1	x
P114	P114	Input	000b	0	0	0	0	x
		Output	000b	0	0	x	1	0/1
		N-ch open drain output	000b	0	1	x	1	0/1
	SI21	Input	001b	0	x	0	0	x
	SDA21	I/O	001b	0	1	0	1	x
P115	P115	Input	000b	0	0	0	0	x
		Output	000b	0	0	x	1	0/1
		N-ch open drain output	000b	0	1	x	1	0/1
	SCK21	Input	001b	0	x	0	0	x
		Output	001b	0	0/1	0/1	1	x
	SCL21	Output	001b	0	0/1	0/1	1	x
P200	P200	Input	—	0	—	0	—	—
	IRQ0	Input	—	1	—	0	—	—
P201	P201	Input	000b	0	0	0	0	x
		Output	000b	0	0	x	1	0/1
		N-ch open drain output	000b	0	1	x	1	0/1
	TI05	Input	001b	0	0	0	0	x
	TO05	Output	001b	0	0	x	1	x
	SSI00	Input	010b	0	0	0	0	x
	SCK11	Input	011b	0	x	0	0	x
		Output	011b	0	0/1	0/1	1	x
	SCL11	Output	011b	0	0/1	0/1	1	x
	RTCOUT	Output	100b	0	0	x	1	x
	PCLBUZ0	Output	101b	0	0	x	1	x
	IRQ5	Input	000b	1	0	0	0	x

Table 16.6 Examples of register settings for port and alternate functions (2/6) (5 of 5)

Pin Name	Function Used		PSEL[2:0]	ISEL	NCODR	PMC	PDR	PODR
	Function Name	I/O						
P204	P204	Input	000b	0	0	0	0	x
		Output	000b	0	0	x	1	0/1
		N-ch open drain output	000b	0	1	x	1	0/1
	SCK01	Input	001b	0	x	0	0	x
		Output	001b	0	0/1	0/1	1	x
	SCL01	Output	001b	0	0/1	0/1	1	x
P205	P205	Input	000b	0	0	0	0	x
		Output	000b	0	0	x	1	0/1
		N-ch open drain output	000b	0	1	x	1	0/1
	SI01	Input	001b	0	x	0	0	x
	SDA01	I/O	001b	0	1	0	1	x
	SCLA1	I/O	010b	0	1	0	1	x
	RXDA1	Input	011b	0	x	0	0	x
	PCLBUZ1	Output	100b	0	0	x	1	x
	IRQ5	Input	000b	1	0	0	0	x
P206 (48-pin/64-pin product)	P206	Input	000b	0	0	0	0	x
		Output	000b	0	0	x	1	0/1
		N-ch open drain output	000b	0	1	x	1	0/1
	SO01	Output	001b	0	0/1	0/1	1	x
	SDAA1	I/O	010b	0	1	0	1	x
	TXDA1	Output	011b	0	0/1	0/1	1	x
	IRQ0	Input	000b	1	0	0	0	x

Table 16.7 Examples of register settings for port and alternate functions (3/6)

Pin Name	Function Used		PORTSELB	PDR	PODR
	Function Name	I/O			
P206 (32-pin product)	P206	Input	0	0	0
		Output	0	1	0/1
	RES	Input	1	—	—

Table 16.8 Examples of register settings for port and alternate functions (4/6)

Pin Name	Function Used		PSEL[2:0]	ISEL	NCODR	PMC	PDR	PODR
	Function Name	I/O						
P207	P207	Input	000b	0	0	0	0	x
		Output	000b	0	0	x	1	0/1
		N-ch open drain output	000b	0	1	x	1	0/1
	TO00	Output	001b	0	0	x	1	x
	RXDA0	Input	010b	0	x	0	0	x
	IRQ2	Input	000b	1	0	0	0	x
	SDA01	I/O	011b	0	1	0	1	x
	SI01	Input	011b	0	x	0	0	x
	SCLA1	I/O	100b	0	1	0	1	x
P208	P208	Input	000b	0	0	0	0	x
		Output	000b	0	0	x	1	0/1
		N-ch open drain output	000b	0	1	x	1	0/1
	TI00	Input	001b	0	0	0	0	x
	TXDA0	Output	010b	0	0/1	0/1	1	x
	IRQ3	Input	000b	1	0	0	0	x
	SCK01	Input	011b	0	x	0	0	x
		Output	011b	0	0/1	0/1	1	x
	SCL01	Output	011b	0	0/1	0/1	1	x
	SDAA1	I/O	100b	0	1	0	1	x

Table 16.9 Examples of register settings for port and alternate functions (5/6) (1 of 2)

Pin Name	Function Used		CMC			PSEL[2:0]	ISEL	NCODR	PMC	PDR	PODR
	Function Name	I/O	SOSEL	MOSEL[1:0]	XTSEL						
P212	P212	Input	0/1	00b/10b	0	000b	0	0	0	0	x
		Output	0/1	00b/10b	0	000b	0	0	x	1	0/1
		N-ch open drain output	0/1	00b/10b	0	000b	0	1	x	1	0/1
	TO00	Output	0/1	00b/10b	0	001b	0	0	x	1	x
	TI03	Input	0/1	00b/10b	0	010b	0	0	0	0	x
	TO03	Output	0/1	00b/10b	0	010b	0	0	x	1	x
	RXD1	Input	0/1	00b/10b	0	011b	0	x	0	0	x
	SI11	Input	0/1	00b/10b	0	100b	0	x	0	0	x
	SDA11	I/O	0/1	00b/10b	0	100b	0	1	0	1	x
	SCLA0	I/O	0/1	00b/10b	0	101b	0	1	0	1	x
	RXDA0	Input	0/1	00b/10b	0	110b	0	x	0	0	x
	IRQ1	Input	0/1	00b/10b	0	000b	1	0	0	0	x
	X1	—	x	01b	0	000b	0	0	0	0	x

Table 16.9 Examples of register settings for port and alternate functions (5/6) (2 of 2)

Pin Name	Function Used		CMC			PSEL[2:0]	ISEL	NCODR	PMC	PDR	PODR
	Function Name	I/O	SOSEL	MOSEL[1:0]	XTSEL						
P213	P213	Input	0/1	00b/10b	0	000b	0	0	0	0	x
		Output	0/1	00b/10b	0	000b	0	0	x	1	0/1
		N-ch open drain output	0/1	00b/10b	0	000b	0	1	x	1	0/1
	TI00	Input	0/1	00b/10b	0	001b	0	0	0	0	x
	TI02	Input	0/1	00b/10b	0	010b	0	0	0	0	x
	TO02	Output	0/1	00b/10b	0	010b	0	0	x	1	x
	TXD1	Output	0/1	00b/10b	0	011b	0	0/1	0/1	1	x
	SO11	Output	0/1	00b/10b	0	100b	0	0/1	0/1	1	x
	SDAA0	I/O	0/1	00b/10b	0	101b	0	1	0	1	x
	TXDA0	Output	0/1	00b/10b	0	110b	0	0/1	0/1	1	x
	IRQ0	Input	0/1	00b/10b	0	000b	1	0	0	0	x
	X2	—	x	01b	0	000b	0	0	0	0	x
	EXCLK	Input	x	11b	0	000b	0	0	0	0	x
P214	P214	Input	0/1	xxb	0	—	—	—	—	—	—
	XCOUT	—	1	xxb	0	—	—	—	—	—	—
P215	P215	Input	0/1	xxb	0	—	—	—	—	—	—
	XCIN	—	1	xxb	0	—	—	—	—	—	—

Table 16.10 Examples of register settings for port and alternate functions (6/6) (1 of 5)

Pin Name	Function Used		PSEL[2:0]	ISEL	NCODR	PMC	PDR	PODR
	Function Name	I/O						
P300	P300	Input	000b	—	—	0	0	x
		Output	000b	—	—	x	1	0/1
	SWCLK	Input	001b	—	—	x	x	x
	TI04	Input	010b	—	—	0	0	x
	TO04	Output	010b	—	—	x	1	x
P301	P301	Input	000b	0	0	0	0	x
		Output	000b	0	0	x	1	0/1
		N-ch open drain output	000b	0	1	x	1	0/1
	TI06	Input	001b	0	0	0	0	x
	TO06	Output	001b	0	0	x	1	x
	SI21	Input	010b	0	x	0	0	x
	SDA21	I/O	010b	0	1	0	1	x
	SCLA1	I/O	011b	0	1	0	1	x
	RXDA1	Input	100b	0	x	0	0	x
	IRQ6	Input	000b	1	0	0	0	x

Table 16.10 Examples of register settings for port and alternate functions (6/6) (2 of 5)

Pin Name	Function Used		PSEL[2:0]	ISEL	NCODR	PMC	PDR	PODR
	Function Name	I/O						
P302	P302	Input	000b	0	0	0	0	x
		Output	000b	0	0	x	1	0/1
		N-ch open drain output	000b	0	1	x	1	0/1
	TI05	Input	001b	0	0	0	0	x
	TO05	Output	001b	0	0	x	1	x
	SCK21	Input	010b	0	x	0	0	x
		Output	010b	0	0/1	0/1	1	x
	SCL21	Output	010b	0	0/1	0/1	1	x
	SDAA1	I/O	011b	0	1	0	1	x
	TXDA1	Output	100b	0	0/1	0/1	1	x
	IRQ0	Input	000b	1	0	0	0	x
P303	P303	Input	000b	0	0	0	0	x
		Output	000b	0	0	x	1	0/1
		N-ch open drain output	000b	0	1	x	1	0/1
	SO21	Output	010b	0	0/1	0/1	1	x
P304	P304	Input	000b	0	0	0	0	x
		Output	000b	0	0	x	1	0/1
		N-ch open drain output	000b	0	1	x	1	0/1
P400	P400	Input	000b	—	—	0	0	x
		N-ch open drain output	000b	—	—	x	1	0/1
	SCLA1	I/O	001b	0	1	0	1	x
P401	P401	Input	000b	—	—	0	0	x
		N-ch open drain output	000b	—	—	x	1	0/1
	SDAA1	I/O	001b	0	1	0	1	x
P402	P402	Input	000b	0	0	0	0	x
		Output	000b	0	0	x	1	0/1
		N-ch open drain output	000b	0	1	x	1	0/1
	TXD2	Output	001b	0	0/1	0/1	1	x
	SO20	Output	001b	0	0/1	0/1	1	x
	TXDA0	Output	010b	0	0/1	0/1	1	x
	IRQ2	Input	000b	1	0	0	0	x
P403	P403	Input	000b	0	0	0	0	x
		Output	000b	0	0	x	1	0/1
		N-ch open drain output	000b	0	1	x	1	0/1
	RXD2	Input	001b	0	x	0	0	x
	SI20	Input	001b	0	x	0	0	x
	SDA20	I/O	001b	0	1	0	1	x
	RXDA0	Input	010b	0	x	0	0	x
	IRQ4	Input	000b	1	0	0	0	x

Table 16.10 Examples of register settings for port and alternate functions (6/6) (3 of 5)

Pin Name	Function Used		PSEL[2:0]	ISEL	NCODR	PMC	PDR	PODR
	Function Name	I/O						
P407	P407	Input	000b	0	0	0	0	x
		Output	000b	0	0	x	1	0/1
		N-ch open drain output	000b	0	1	x	1	0/1
	SCK11	Input	001b	0	x	0	0	x
		Output	001b	0	0/1	0/1	1	x
	SCL11	Output	001b	0	0/1	0/1	1	x
	RTCOUT	Output	010b	0	0	x	1	x
	PCLBUZ0	Output	011b	0	0	x	1	x
	IRQ4	Input	000b	1	0	0	0	x
	SDAA1	I/O	100b	0	1	0	1	x
P408	P408	Input	000b	0	0	0	0	x
		Output	000b	0	0	x	1	0/1
		N-ch open drain output	000b	0	1	x	1	0/1
	TI04	Input	010b	0	0	0	0	x
	TO04	Output	010b	0	0	x	1	x
	SCLA1	I/O	100b	0	1	0	1	x
	IRQ7	Input	000b	1	0	0	0	x
P409	P409	Input	000b	0	0	0	0	x
		Output	000b	0	0	x	1	0/1
		N-ch open drain output	000b	0	1	x	1	0/1
	SCK11	Input	001b	0	x	0	0	x
		Output	001b	0	0/1	0/1	1	x
	SCL11	Output	001b	0	0/1	0/1	1	x
	TI03	Input	010b	0	0	0	0	x
	TO03	Output	010b	0	0	x	1	x
	IRQ6	Input	000b	1	0	0	0	x
P410	P410	Input	000b	0	0	0	0	x
		Output	000b	0	0	x	1	0/1
		N-ch open drain output	000b	0	1	x	1	0/1
	SCK20	Input	001b	0	x	0	0	x
		Output	001b	0	0/1	0/1	1	x
	SCL20	Output	001b	0	0/1	0/1	1	x
	TI02	Input	010b	0	0	0	0	x
	TO02	Output	010b	0	0	x	1	x
	SSI00	Input	011b	0	0	0	0	x
	SCLA0	I/O	100b	0	1	0	1	x
	RXDA1	Input	101b	0	x	0	0	x
	IRQ4	Input	000b	1	0	0	0	x

Table 16.10 Examples of register settings for port and alternate functions (6/6) (4 of 5)

Pin Name	Function Used		PSEL[2:0]	ISEL	NCODR	PMC	PDR	PODR
	Function Name	I/O						
P411	P411	Input	000b	0	0	0	0	x
		Output	000b	0	0	x	1	0/1
		N-ch open drain output	000b	0	1	x	1	0/1
	SCK11	Input	001b	0	x	0	0	x
		Output	001b	0	0/1	0/1	1	x
	SCL11	Output	001b	0	0/1	0/1	1	x
	TI01	Input	010b	0	0	0	0	x
	TO01	Output	010b	0	0	x	1	x
	SDAA0	I/O	100b	0	1	0	1	x
	TXDA1	Output	101b	0	0/1	0/1	1	x
	IRQ3	Input	000b	1	0	0	0	x
P500	P500	Input	000b	0	0	0	0	x
		Output	000b	0	0	x	1	0/1
		N-ch open drain output	000b	0	1	x	1	0/1
	TI03	Input	001b	0	0	0	0	x
	TO03	Output	001b	0	0	x	1	x
	SCK00	Input	010b	0	x	0	0	x
		Output	010b	0	0/1	0/1	1	x
	SCL00	Output	010b	0	0/1	0/1	1	x
P501	P501	Input	000b	0	0	0	0	x
		Output	000b	0	0	x	1	0/1
		N-ch open drain output	000b	0	1	x	1	0/1
	TI04	Input	001b	0	0	0	0	x
	TO04	Output	001b	0	0	x	1	x
	TXD0	Output	010b	0	0/1	0/1	1	x
	SO00	Output	010b	0	0/1	0/1	1	x
	SDAA0	I/O	011b	0	1	0	1	x
	TXDA0	Output	100b	0	0/1	0/1	1	x
P502	P502	Input	000b	0	0	0	0	x
		Output	000b	0	0	x	1	0/1
		N-ch open drain output	000b	0	1	x	1	0/1
	RXD0	Input	010b	0	x	0	0	x
	SI00	Input	010b	0	x	0	0	x
	SDA00	I/O	010b	0	1	0	1	x
	SCLA0	I/O	011b	0	1	0	1	x
	RXDA0	Input	100b	0	x	0	0	x
	IRQ5	Input	000b	1	0	0	0	x
P913	P913	Input (5V tolerant)	000b	—	—	0	0	x
		N-ch open drain output	000b	—	—	1	1	0/1
	SDAA0	I/O	001b	—	—	0	1	x

Table 16.10 Examples of register settings for port and alternate functions (6/6) (5 of 5)

Pin Name	Function Used		PSEL[2:0]	ISEL	NCODR	PMC	PDR	PODR
	Function Name	I/O						
P914	P914	Input (5 V tolerant)	000b	—	—	0	0	x
		N-ch open drain output	000b	—	—	1	1	0/1
	SCLA0	I/O	001b	—	—	0	1	x
P915	P915	Input	000b	0	0	0	0	x
		Output	000b	0	0	x	1	0/1
		N-ch open drain output	000b	0	1	x	1	0/1
	SO01	Output	011b	0	0/1	0/1	1	x

16.5.6 Notes on Communications with Devices Operating at a Different Voltage (1.8 V, 2.5 V, or 3 V) by Switching I/O Buffers

The pin input buffer selection bits (PIM) and the N-Channel Open-Drain Control bits (NCODR) can be used to switch the I/O buffers to enable communications with external devices that have different operating voltages (1.8 V, 2.5 V, or 3 V) to this device.

(1) Procedure for setting input pins of UART0 to UART2, UARTA0, UARTA1, SPI00, SPI01, SPI10, SPI11, SPI20 and SPI21 for use with the TTL input buffers

1. Pull up the input pin to be used to the voltage of the target device through an external resistor. The on-chip pull-up resistor cannot be used for this purpose.
2. Set the PIM bit to 1 to switch to the TTL input buffer. For V_{IH} and V_{IL}, refer to the DC characteristics when the TTL input buffer is selected.
3. Enable the operation of the serial array unit and set the mode to the UART/simplified SPI mode.

(2) Procedure for setting output pins of UART0 to UART2, UARTA0, UARTA1, SPI00, SPI01, SPI10, SPI11, SPI20 and SPI21 for use with the N-ch open-drain output mode

1. Pull up the input pin to be used to the voltage of the target device through an external resistor. The on-chip pull-up resistor cannot be used for this purpose.
2. The port pins are set for input (Hi-Z) after the reset state is released.
3. Set the PMC bits to 1 to disable input to the input buffer.
4. Set the NCODR bits to 1 to set the N-ch open drain output (withstand voltage of VCC) mode.
5. Enable the operation of the serial array unit and set the mode to the UART/simplified SPI mode.
6. Set the PDR bits to the output mode. At this time, the output data is high level, so the pin is in the Hi-Z state.

(3) Procedure for setting I/O pins of IIC00, IIC01, IIC10, IIC11, IIC20 and IIC21 for use in connection with a device operating at a different voltage (1.8 V, 2.5 V, or 3 V)

1. Pull up the input pin to be used to the voltage of the target device through an external resistor. The on-chip pull-up resistor cannot be used for this purpose.
2. The port pins are set for input (Hi-Z) after the reset state is released.
3. Set the NCODR bits to 1 to set the N-ch open drain output (withstand voltage of VCC) mode.
4. Set the PIM bit to 1 to switch to the TTL input buffer. For V_{IH} and V_{IL}, refer to the DC characteristics when the TTL input buffer is selected.
5. Enable the operation of the serial array unit and set the mode to the simplified I^2C mode.
6. Set the PDR bits to the output mode (data I/O is possible in the output mode). At this time, the output data is high level, so the pin is in the Hi-Z state.

Note: The input buffer is enabled even if the pin is operating as an output when the N-ch open-drain output mode is selected by the corresponding bit in the N-channel open-drain control bit (NCODR). This may lead to a through current flowing through the pin when the voltage level on this pin is intermediate. However, setting the corresponding bit of the given PMC bit to 1 prevents the flow of a through current.

Note: When the pin is set to TTL input buffer by the port input buffer selection bit (PIMx) and is driven high, a through current may flow through the pin due to the configuration of the TTL input buffer. Drive the pin low to prevent the through current.

Note: Communications by using P212 and P213 with devices operating at different voltage levels are not possible since P212PFS_A and P213PFS_A registers do not have PIM bit.

Note: The input buffer is enabled even if the P400, P401, P913 and P914 pin is operating as an output. This may lead to a through current flowing through the pin when the voltage level on this pin is intermediate. However, setting the corresponding bit of the given PMC bit to 1 prevents the flow of a through current.

16.5.7 Restriction on P206 Usage in the 32-pin product

Once the power is turned on, P206 functions as the RES input. The PORTSELB bit of the option select register1 (OFS1) defines whether this port operates as P206 or RES. When this pin is set to P206, do not input the low level to this pin during a reset by the power-on-reset (POR) circuit and during the period from release from the reset by the POR circuit to the start of normal operation. If input of the low level continues during this period, the chip will remain in the reset state in response to the external reset. The on-chip pull-up resistor is enabled after power is turned on.

16.6 Peripheral Select Settings for Each Product

This section describes the pin function select configuration by the PmnPFS_A register. Some signal names have _A, _B, _C, _D, _E, or _F suffixes, but these suffixes can be ignored when assigning functionality, except for SAU and IICA. For SAU and IICA, only signals, except for SCK11, SCL11 and SSI00, bearing the same suffix can be selected. 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_A. If a value that is not allowed for the register is specified, the correct operation is not guaranteed.

Table 16.11 Register settings for the pin function select configuration (PORT0)

PSEL[2:0] settings	Pin												
	P000	P001	P002	P003	P004	P008	P009	P010	P011	P012	P013	P014	P015
000b	P0n Output Data (initial)												
PMC bit	✓ (AN008)	✓ (AN009)	✓ (AN010)	✓ (AN011)	✓ (AN012)	✓ (AN002)	✓ (AN003)	✓ (AN000/ VREFH0)	✓ (AN001/ VREFL0)	✓ (AN004)	✓ (AN005)	✓ (AN006)	✓ (AN007)
ISEL bit	IRQ6_D	IRQ7_A	IRQ7_C	—	IRQ2_E	—	—	—	—	—	—	—	IRQ1_A
NCODR bit	—	—	—	—	—	—	—	—	—	—	—	—	—
PIM bit	—	—	—	—	—	—	—	—	—	—	—	—	—
PCR bit	—	—	—	—	—	—	—	—	—	—	—	—	—
64-pin product	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
48-pin product	✓	✓	✓	—	—	✓	✓	✓	✓	✓	✓	✓	✓
32-pin product	—	—	—	—	—	✓	✓	✓	✓	✓	✓	✓	✓

✓: Available
—: Setting prohibited

Table 16.12 Register settings for the pin function select configuration (PORT1) (1 of 3)

PSEL[2:0] settings	Pin							
	P100	P101	P102	P103	P104	P105	P106	P107
000b	P1n Output Data (initial)							
001b	TI04_A/TO04_A	TI07_A/TO07_A	TI06_A/TO06_A	TI05_A/TO05_A	TI02_D/TO02_D	TI01_D/TO01_D	—	—
010b	TI01_B/TO01_B	TI00_C	TO00_C	SSI00_A	TI00_D	TO00_D	—	—
011b	RXD0_A/SI00_A/ SDA00_A	TXD0_A/ SO00_A	SCK00_A/ SCL00_A	—	SCK10_A/ SCL10_A	SI10_A/SDA10_A	SO10_A	—
100b	SCLA0_D	SDAA0_D	RTCOUT_C	—	—	RXDA1_B	TXDA1_B	—
101b	RXDA0_D	TXDA0_D	PCLBUZ0_B	—	—	—	—	—

Table 16.12 Register settings for the pin function select configuration (PORT1) (2 of 3)

PSEL[2:0] settings	Pin							
	P100	P101	P102	P103	P104	P105	P106	P107
110b	—	—	SCLA1_B	SDAA1_B	—	—	—	—
111b	—	—	RXDA1_A	TXDA1_A	—	—	—	—
PMC bit	✓ (AN022)	✓ (AN021)	✓	✓	✓	✓	✓	✓
ISEL bit	IRQ2_A	IRQ3_A	IRQ4_A	IRQ5_A	IRQ6_C	IRQ1_D	IRQ0_E	IRQ7_D
NCODR bit	✓	✓	✓	✓	✓	✓	✓	✓
PIM bit	✓	✓	✓	✓	✓	✓	✓	✓
PCR bit	✓	✓	✓	✓	✓	✓	✓	✓
64-pin product	✓	✓	✓	✓	✓	✓	✓	✓
48-pin product	✓	✓	✓	✓	✓	✓	✓	—
32-pin product	✓	✓	✓	✓	—	—	—	—

Table 16.12 Register settings for the pin function select configuration (PORT1) (3 of 3)

PSEL[2:0] settings	Pin							
	P108	P109	P110	P111	P112	P113	P114	P115
000b	P1n Output Data	P1n Output Data (initial)						
001b	SWDIO (initial)	TI02_A/TO02_A	TI01_A/TO01_A	TI07_B/TO07_B	TI03_A/TO03_A	SO21_B	SI21_B/SDA21_B	SCK21_B/ SCL21_B
010b	TI03_B/TO03_B	TXD2_A/ SO20_A	RXD2_A/SI20_A/ SDA20_A	—	SCK20_A/ SCL20_A	—	—	—
011b	—	SDAA0_C	SCLA0_C	—	SSI00_C	—	—	—
100b	—	TXDA0_C	RXDA0_C	—	—	—	—	—
101b	—	PCLBUZ1_B	—	—	—	—	—	—
PMC bit	✓	✓	✓	✓	✓	✓	✓	✓
ISEL bit	—	IRQ4_B	IRQ3_B	IRQ1_C	IRQ2_B	—	—	—
NCODR bit	—	✓	✓	✓	✓	✓	✓	✓
PIM bit	✓	✓	✓	✓	✓	✓	✓	✓
PCR bit	✓	✓	✓	✓	✓	✓	✓	✓
64-pin product	✓	✓	✓	✓	✓	✓	✓	✓
48-pin product	✓	✓	✓	✓	✓	—	—	—
32-pin product	✓	✓	✓	—	✓	—	—	—

✓: Available
—: Setting prohibited

Table 16.13 Register settings for the pin function select configuration (PORT2) (1 of 2)

PSEL[2:0] settings	Pin											
	P200	P201	P204	P205	P206 (48-pin/64-pin products)	P206 (32-pin product)	P207	P208	P212	P213	P214	P215
000b	Hi-Z (initial)	P2n Output Data (initial)									Hi-Z (initial)	
001b	—	TI05_B/ TO05_B	SCK01_A/ SCL01_A	SI01_A/ SDA01_A	SO01_A	—	TO00_B	TI00_B	TO00_A	TI00_A	—	—
010b	—	SSI00_B	—	SCLA1_E	SDAA1_E	—	RXDA0_A	TXDA0_A	TI03_C/ TO03_C	TI02_B/ TO02_B	—	—
011b	—	SCK11_B/ SCL11_B	—	RXDA1_E	TXDA1_E	—	SI01_B/ SDA01_B	SCK01_B/ SCL01_B	RXD1_A	TXD1_A	—	—
100b	—	RTCOUT_B	—	PCLBUZ1_ A	—	—	SCLA1_A	SDAA1_A	SI11_A / SDA11_A	SO11_A	—	—
101b	—	PCLBUZ0_ A	—	—	—	—	—	—	SCLA0_B	SDAA0_B	—	—
110b	—	—	—	—	—	—	—	—	RXDA0_B	TXDA0_B	—	—
PMC bit	✓	✓	✓	✓	✓	—	✓	✓	✓	✓	—	—
ISEL bit	IRQ0_A	IRQ5_B	—	IRQ5_C	IRQ0_C	—	IRQ2_C	IRQ3_C	IRQ1_B	IRQ0_B	—	—
NCODR bit	—	✓	✓	✓	✓	—	✓	✓	✓	✓	—	—
PIM bit	—	✓	✓	✓	✓	—	✓	✓	—	—	—	—
PCR bit	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	—	—

Table 16.13 Register settings for the pin function select configuration (PORT2) (2 of 2)

PSEL[2:0] settings	Pin											
	P200	P201	P204	P205	P206 (48-pin/64-pin products)	P206 (32-pin product)	P207	P208	P212	P213	P214	P215
64-pin product	✓	✓	✓	✓	✓	—	✓	✓	✓	✓	✓	✓
48-pin product	✓	✓	—	—	✓	—	✓	✓	✓	✓	✓	✓
32-pin product	✓	✓	—	—	—	✓	✓	✓	✓	✓	✓	✓

✓: Available
—: Setting prohibited

Table 16.14 Register settings for the pin function select configuration (PORT3)

PSEL[2:0] settings	Pin				
	P300	P301	P302	P303	P304
000b	P3n Output Data	P3n Output Data (initial)			
001b	SWCLK (initial)	TI06_B/TO06_B	TI05_C/TO05_C	—	—
010b	TI04_B/TO04_B	SI21_A/SDA21_A	SCK21_A/SCL21_A	SO21_A	—
011b	—	SCLA1_C	SDAA1_C	—	—
100b	—	RXDA1_C	TXDA1_C	—	—
PMC bit	✓	✓	✓	✓	✓
ISEL bit	—	IRQ6_A	IRQ0_D	—	—
NCODR bit	—	✓	✓	✓	✓
PIM bit	✓	✓	✓	✓	✓
PCR bit	✓	✓	✓	✓	✓
64-pin product	✓	✓	✓	✓	✓
48-pin product	✓	✓	✓	—	—
32-pin product	✓	—	—	—	—

✓: Available
—: Setting prohibited

Table 16.15 Register settings for the pin function select configuration (PORT4)

PSEL[2:0] settings	Pin								
	P400	P401	P402	P403	P407	P408	P409	P410	P411
000b	P4n Output Data (initial)								
001b	SCLA1_D	SDAA1_D	TXD2_B/ SO20_B	RXD2_B/ SI20_B/ SDA20_B	SCK11_A/ SCL11_A	—	SCK11_C/ SCL11_C	SCK20_B/ SCL20_B	SCK11_D/ SCL11_D
010b	—	—	TXDA0_F	RXDA0_F	RTCOUT_A	TI04_C/TO04_C	TI03_E/TO03_E	TI02_C/TO02_C	TI01_C/TO01_C
011b	—	—	—	—	PCLBUZ0_C	—	—	SSI00_D	—
100b	—	—	—	—	SDAA1_F	SCLA1_F	—	SCLA0_E	SDAA0_E
101b	—	—	—	—	—	—	—	RXDA1_D	TXDA1_D
PMC bit	✓	✓	✓	✓	✓	✓	✓	✓	✓
ISEL bit	—	—	IRQ2_D	IRQ4_E	IRQ4_C	IRQ7_B	IRQ6_B	IRQ4_D	IRQ3_D
NCODR bit	—	—	✓	✓	✓	✓	✓	✓	✓
PIM bit	—	—	✓	✓	✓	✓	✓	✓	✓
PCR bit	—	—	✓	✓	✓	✓	✓	✓	✓
64-pin product	✓	✓	✓	✓	✓	✓	✓	✓	✓
48-pin product	✓	✓	—	—	✓	✓	✓	—	—
32-pin product	—	—	—	—	✓	—	—	—	—

✓: Available
—: Setting prohibited

Table 16.16 Register settings for the pin function select configuration (PORT5) (1 of 2)

PSEL[2:0] settings	Pin		
	P500	P501	P502
000b	P5n Output Data (initial)		

Table 16.16 Register settings for the pin function select configuration (PORT5) (2 of 2)

PSEL[2:0] settings	Pin		
	P500	P501	P502
001b	TI03_D/TO03_D	TI04_D/TO04_D	—
010b	SCK00_B/SCL00_B	TXD0_B/SO00_B	RXD0_B/SI00_B/SDA00_B
011b	—	SDAA0_F	SCLA0_F
100b	—	TXDA0_E	RXDA0_E
PMC bit	✓	✓	✓
ISEL bit	—	—	IRQ5_D
NCODR bit	✓	✓	✓
PIM bit	✓	✓	✓
PCR bit	✓	✓	✓
64-pin product	✓	✓	✓
48-pin product	✓	—	—
32-pin product	—	—	—

✓: Available
—: Setting prohibited

Table 16.17 Register settings for the pin function select configuration (PORT9)

PSEL[2:0] settings	Pin		
	P913	P914	P915
000b	P9n Output Data (initial)		
001b	SDAA0_A	SCLA0_A	—
010b	—	—	—
011b	—	—	SO01_B
PMC bit	✓	✓	✓
ISEL bit	—	—	—
NCODR bit	—	—	✓
PIM bit	—	—	✓
PCR bit	—	—	✓
64-pin product	✓	✓	✓
48-pin product	✓	✓	✓
32-pin product	✓	✓	—

✓: Available
—: Setting prohibited

17. Timer Array Unit (TAU)

17.1 Overview

The timer array unit of this device has one unit with 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 functional timer.

Figure 17.1 shows the channel configuration per unit in timer array unit.

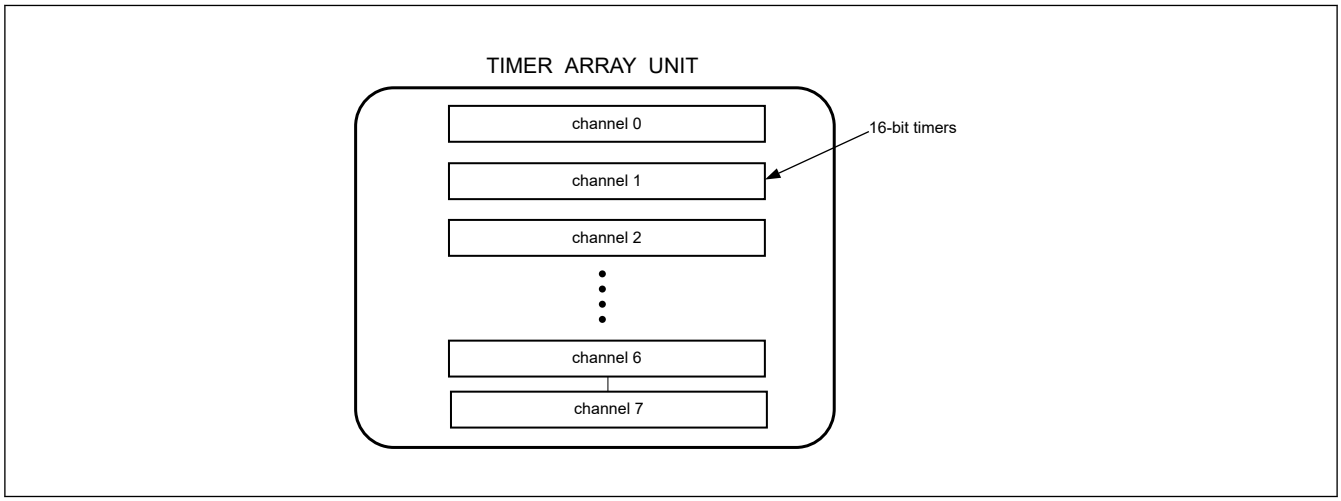


Figure 17.1 Channel configuration per unit

It is possible to use the 16-bit timer of channels 1 and 3 of unit 0 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 and 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 of unit 0 can be used to realize LIN-bus communication operating in combination with UART2 of the serial array unit.

Table 17.1 lists the TAU functions and Figure 17.2 to Figure 17.11 show each functional image.

Table 17.1 TAU functions

Parameter		Description
Independent channel operation function *1	Interval timer	Each timer of a unit can be used as a reference timer that generates an interrupt (TAU0_TMI0n) at fixed intervals.
	Square wave output	A toggle operation is performed each time TAU0_TMI0n 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 of a unit 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 of unit 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 of unit 0 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 17.7. 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 17.8. 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 17.3.2. 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 17.2.16. ISC : Input Switch Control Register and section 17.7.5. Operation for Input Signal High- or Low-Level Width Measurement.

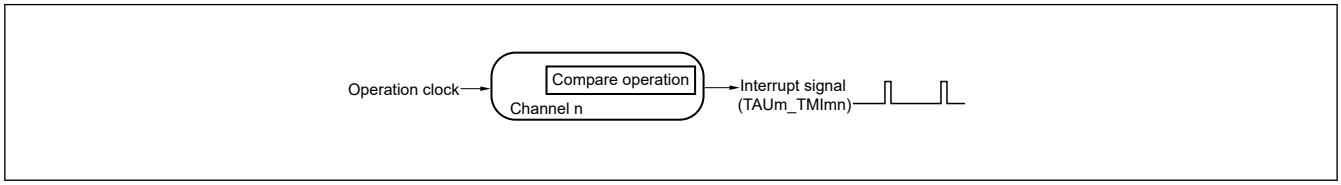


Figure 17.2 Functional image of interval timer

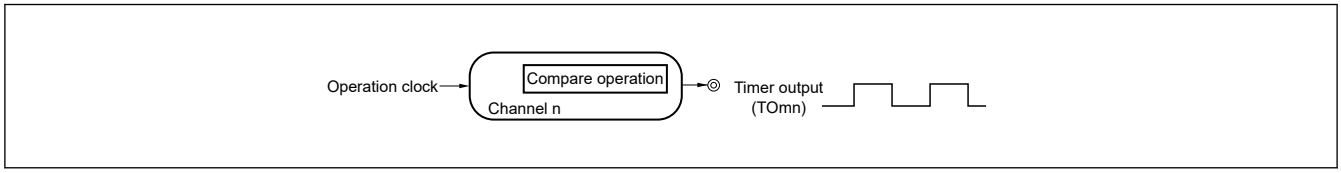


Figure 17.3 Functional image of square wave output

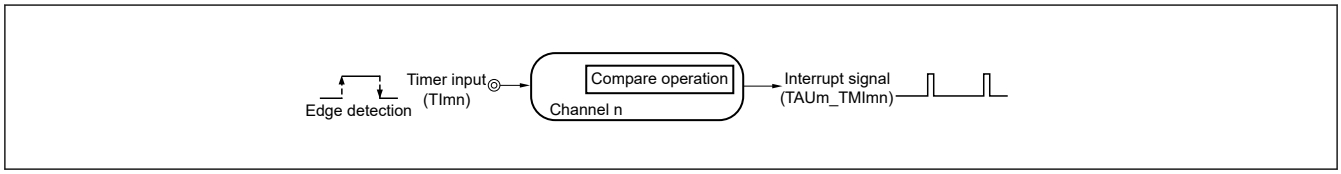


Figure 17.4 Functional image of external event counter

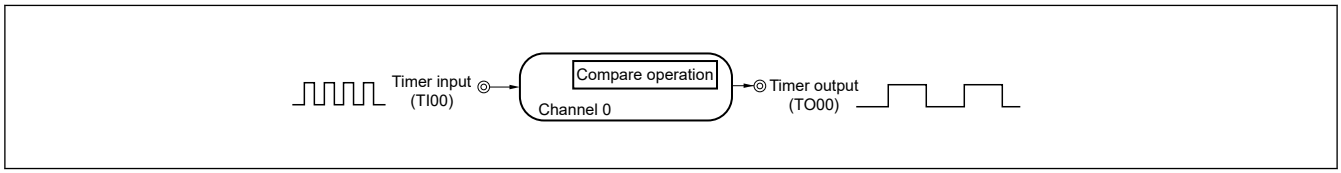


Figure 17.5 Functional image of divider function

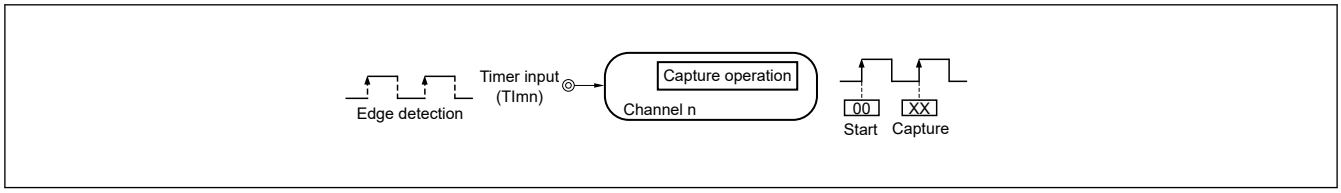


Figure 17.6 Functional image of input pulse interval measurement

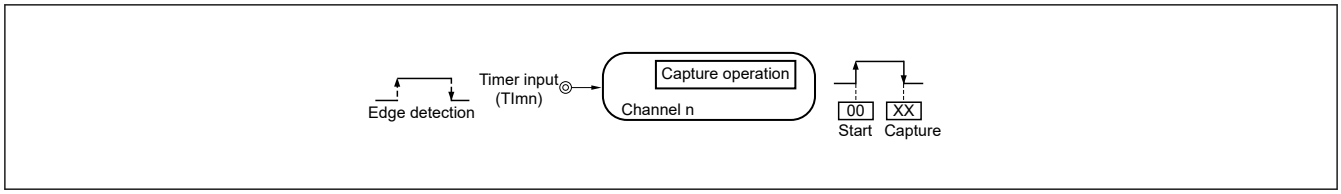


Figure 17.7 Functional image of measurement of high- or low-level width of input signal

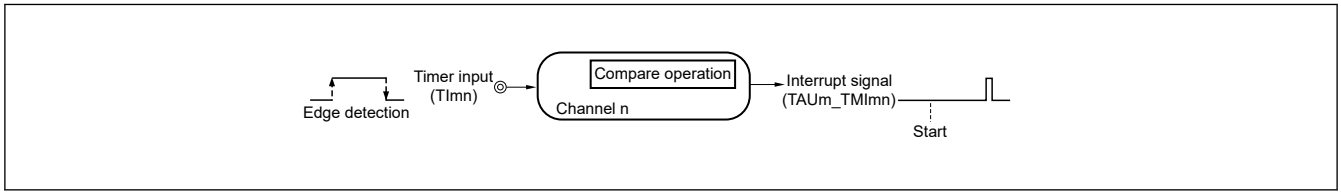


Figure 17.8 Functional image of delay counter

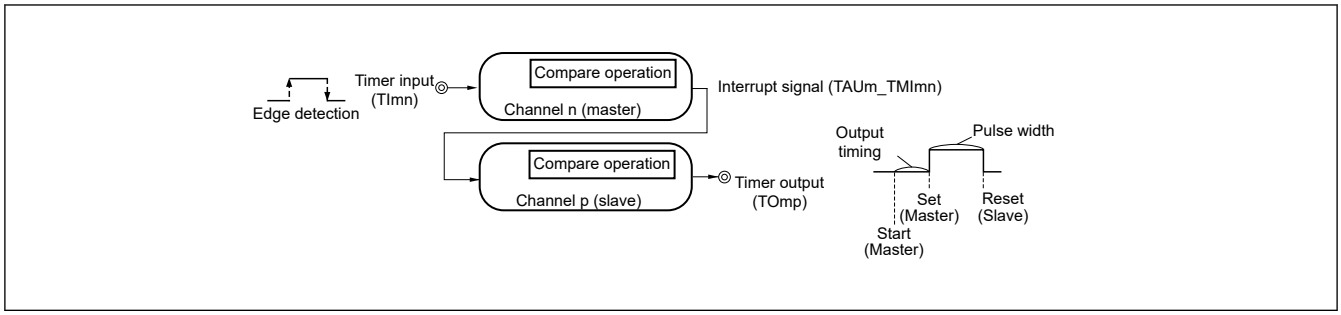


Figure 17.9 Functional image of one-shot pulse output

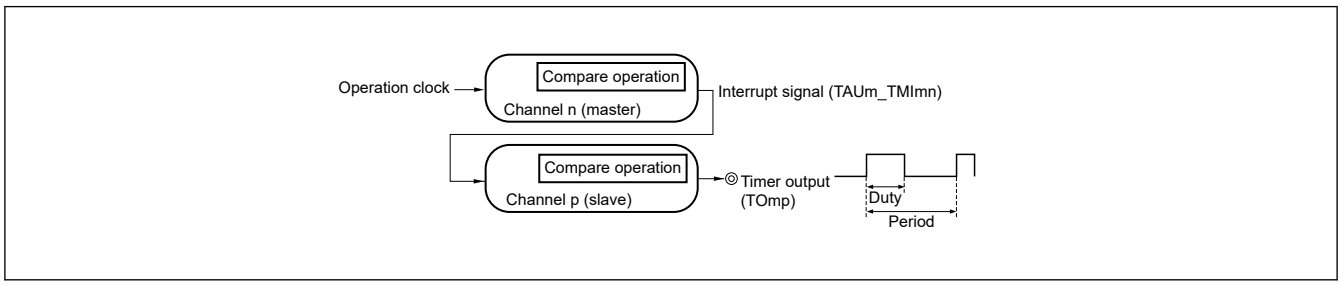


Figure 17.10 Functional image of PWM output

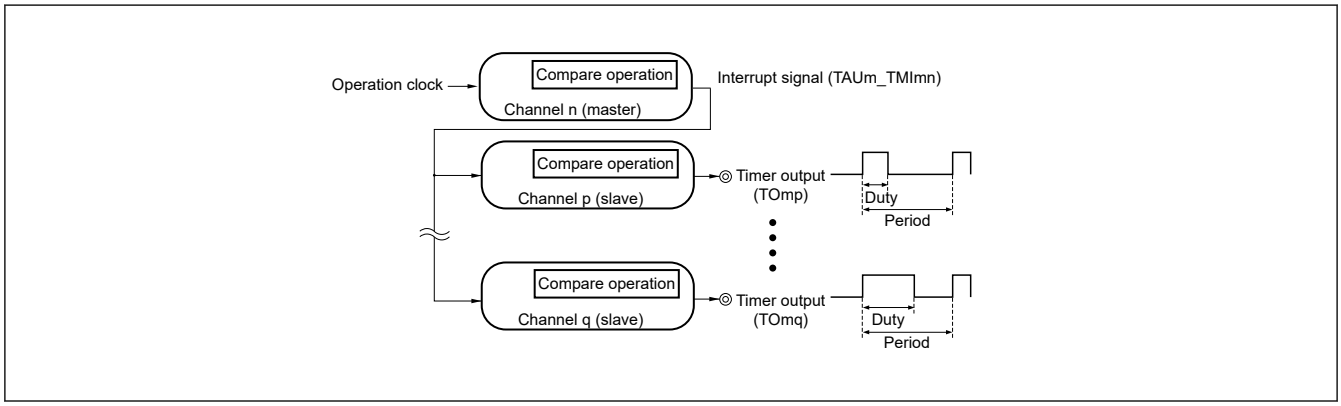


Figure 17.11 Functional image of multiple PWM output

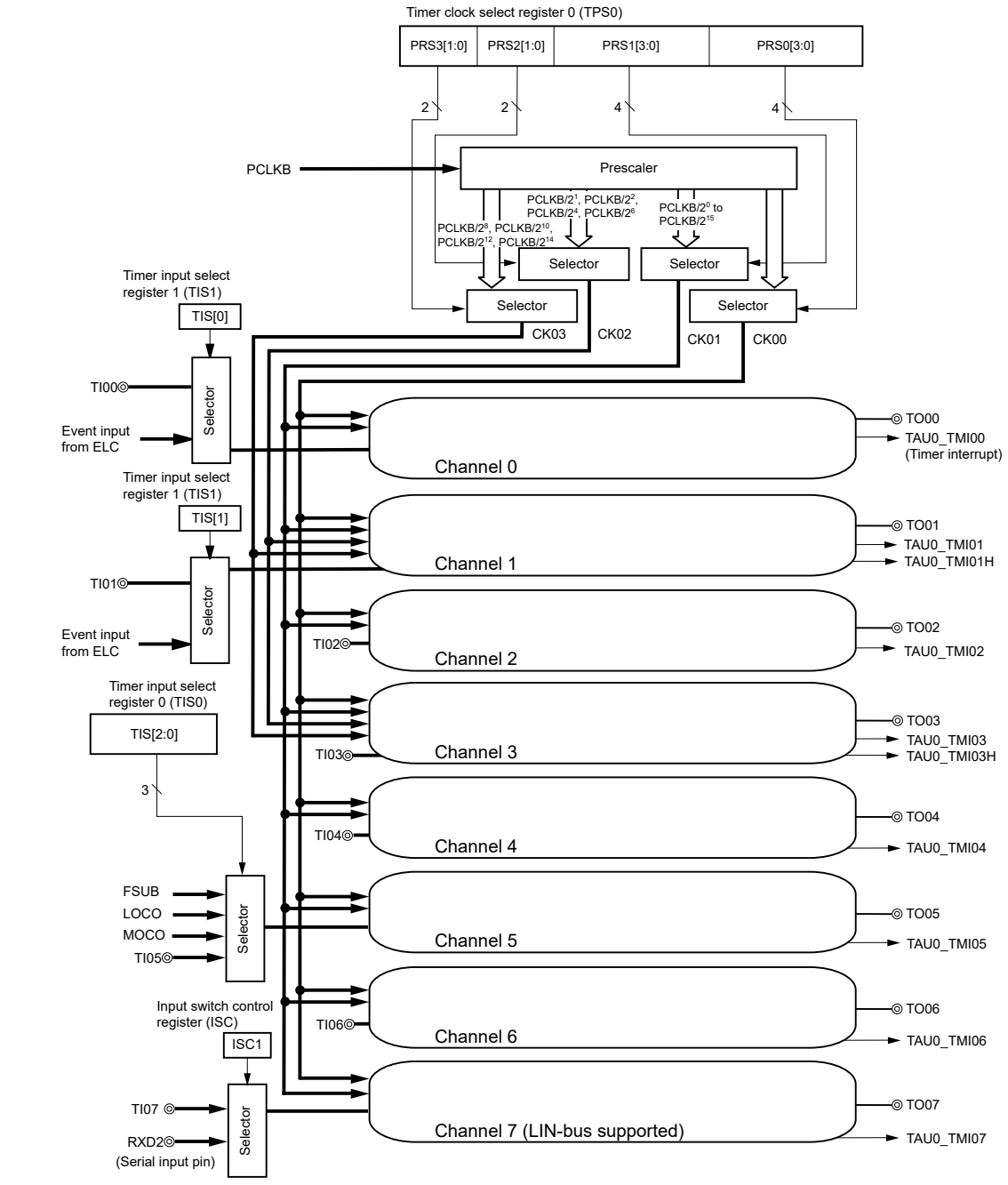
Note: m: Unit number (m = 0), n: Channel number (n = 0 to 7), p, q: Slave channel number (n < p < q ≤ 7)

Timer array unit includes the hardware shown in Table 17.2.

Table 17.2 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 pins, RXD2 pin (for LIN-bus)
Timer output	TO00 to TO07 pins, output controller

Figure 17.12 shows the block diagram of the timer array unit. Figure 17.13 to Figure 17.18 show a block diagram for each channel.



Note: FSUB: Sub System clock
LOCO: Low-speed on-chip oscillator
MOCO: Middle-speed on-chip oscillator

Figure 17.12 Entire configuration of timer array unit 0

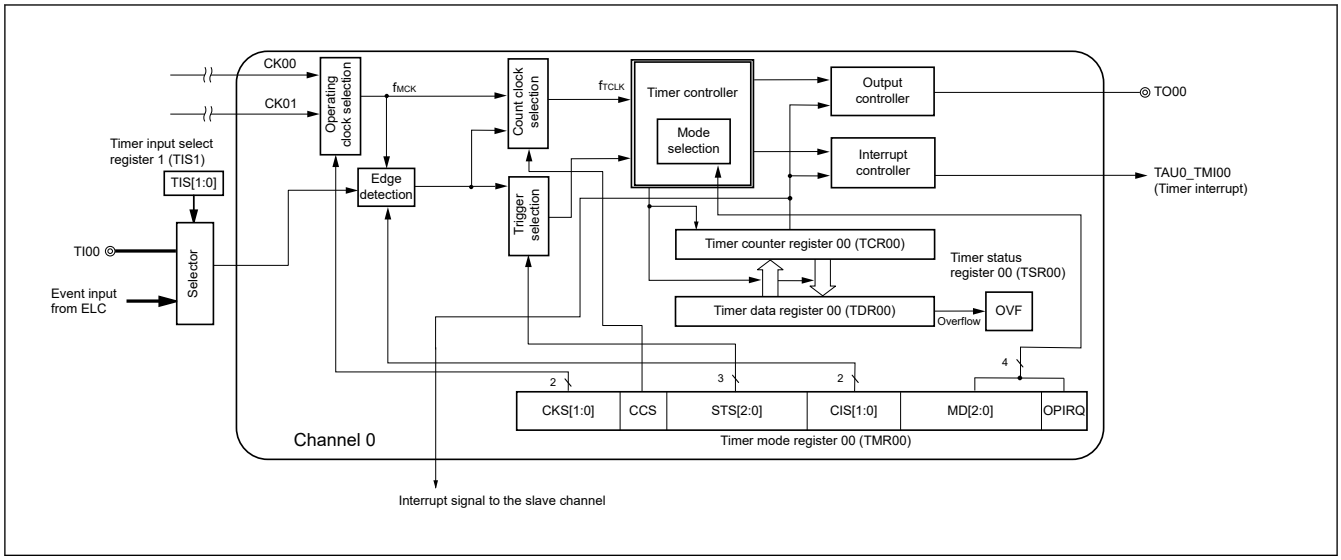


Figure 17.13 Internal block diagram of channel 0 of timer array unit 0

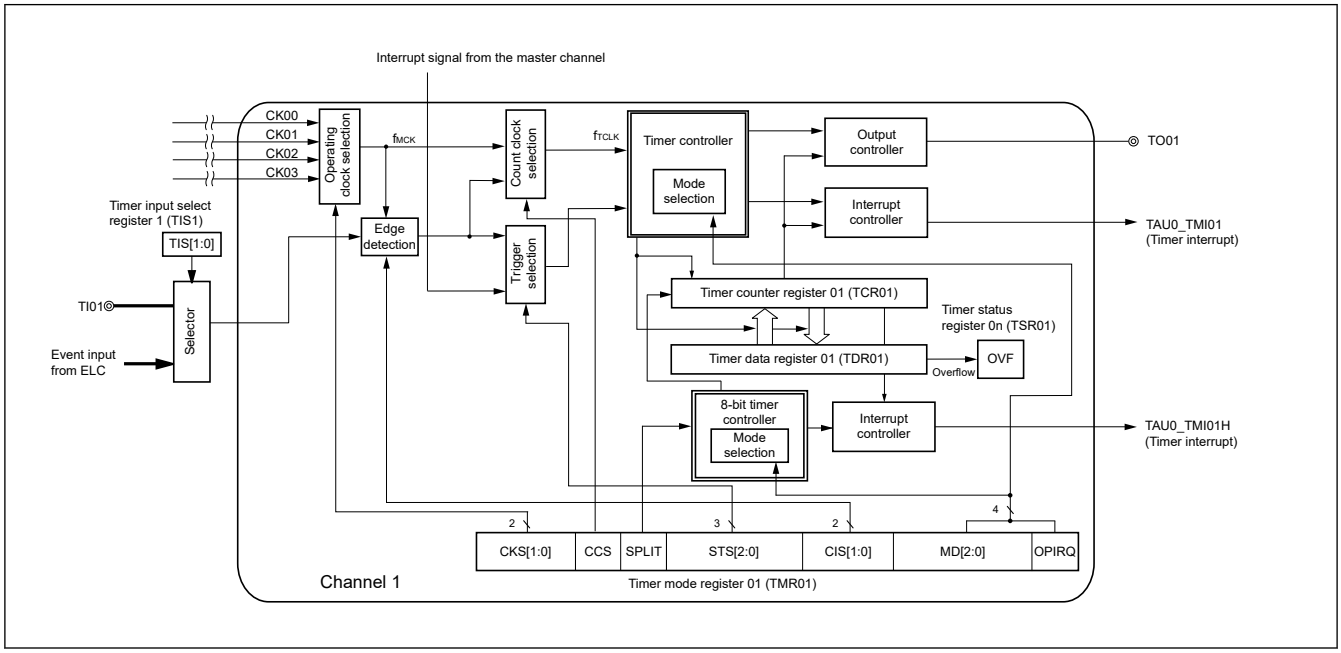
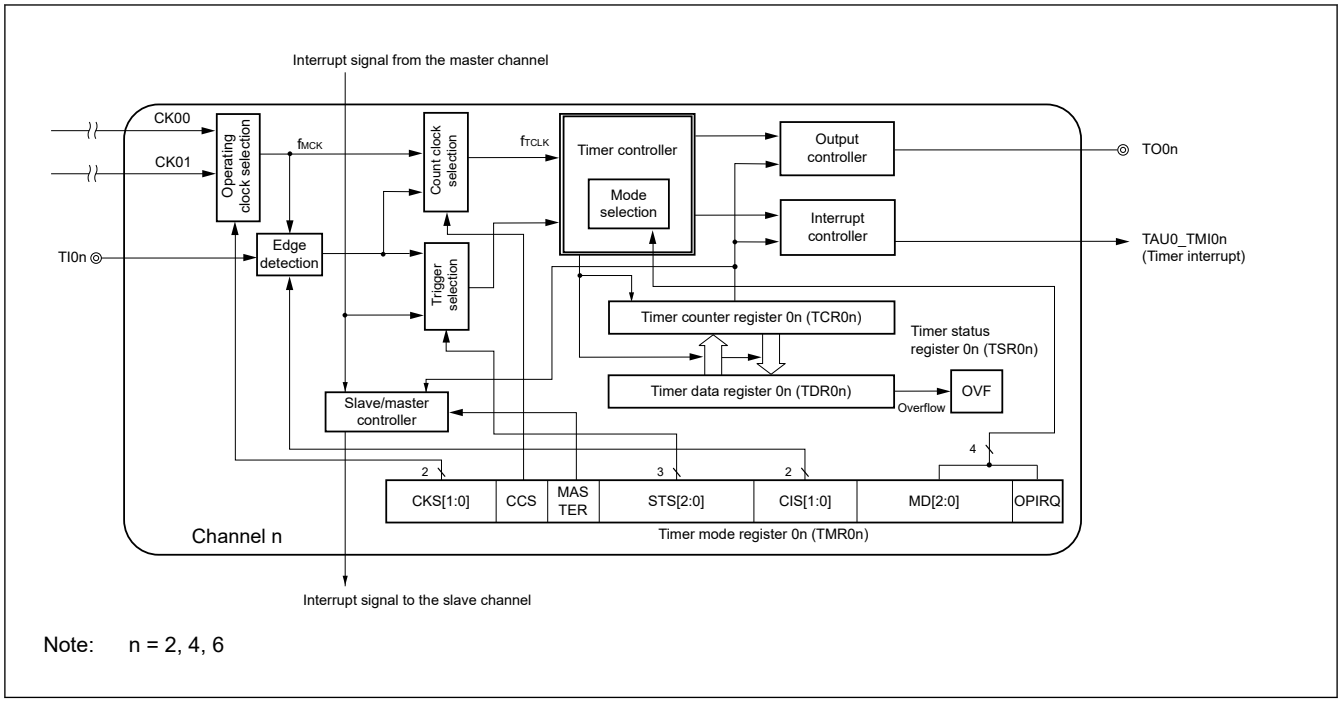


Figure 17.14 Internal block diagram of channel 1 of timer array unit 0



Note: n = 2, 4, 6

Figure 17.15 Internal block diagram of channels 2, 4, and 6 of timer array unit 0

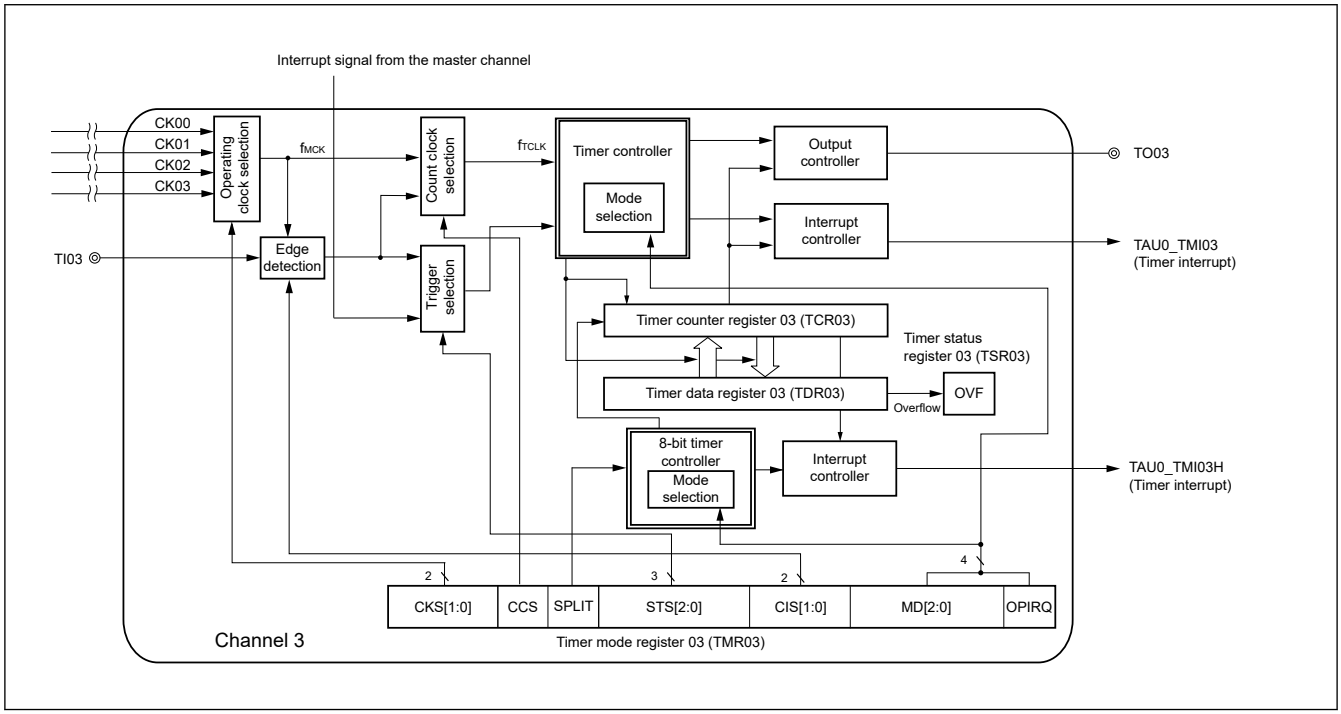


Figure 17.16 Internal block diagram of channel 3 of timer array unit 0

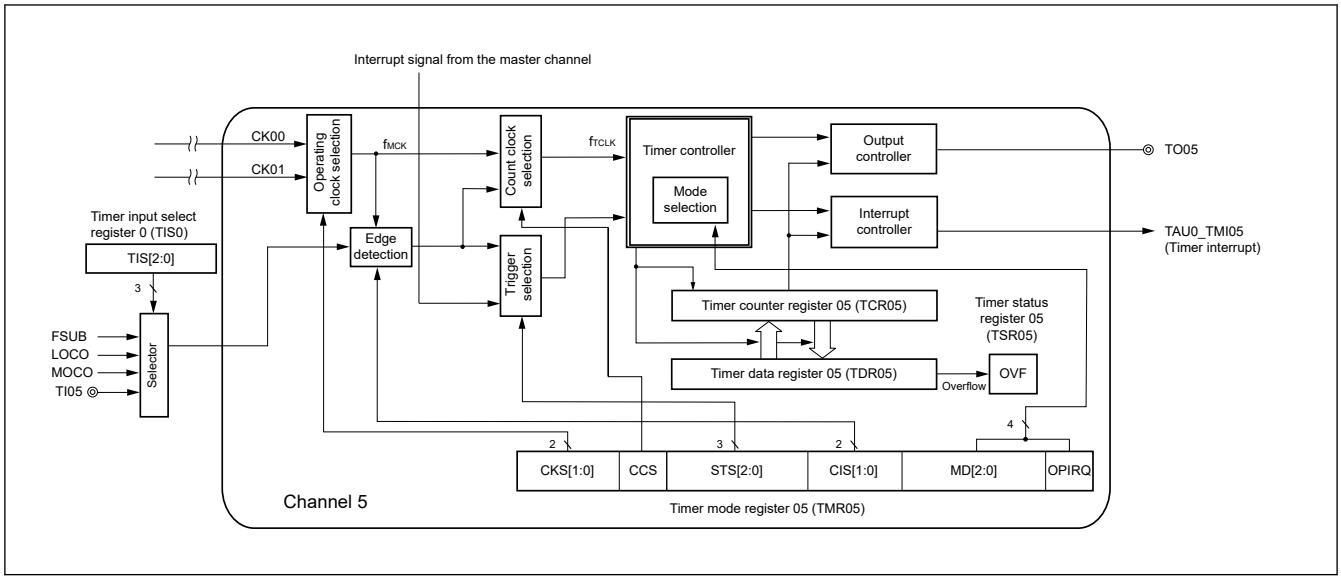


Figure 17.17 Internal block diagram of channel 5 of timer array unit 0

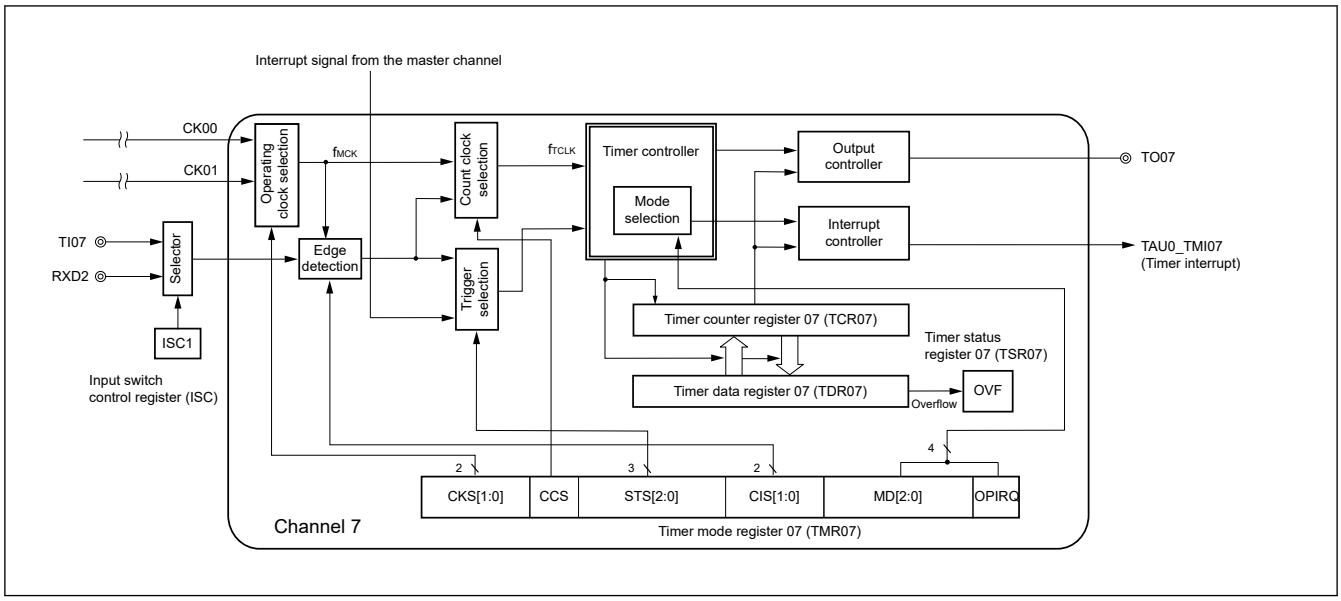


Figure 17.18 Internal block diagram of channel 7 of timer array unit 0

17.2 Register Descriptions

17.2.1 TCR0n : Timer Counter Register 0n (n = 0 to 7)

Base address: TAU = 0x400A_2600

Offset address: 0x0100 + 0x2 × n

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	n/a															
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 Unit m and 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 17.2.4. TMR0n : Timer Mode Register 0n (n = 0, 2, 4, 5, 6, 7) and section 17.2.5. 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.

Note: When reading the TCR0n register, it is necessary to access the TCR0n register with 16-bit width which is the same as its counter size to prevent from mistaking the count value.

The value read from the TCR0n register varies depending on the change to the operation mode and operating state as shown in the Table 17.3.

Table 17.3 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 for a 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 TRD0n 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.

17.2.2 TDR0n/TDR01x/TDR03x : Timer Data Register 0n (n = 0 to 7) (x = L, H)

Base address: TAU = 0x400A_2600

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 Unit m and 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 SPLIT01, SPLIT03 bits of timer mode registers 01 and 03 (TMR01, TMR03) 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_TMI0n) 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).

17.2.3 TPS0 : Timer Clock Select Register 0

Base address: TAU = 0x400A_2600

Offset address: 0x0136

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 *3 *4 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 *3 *4 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 *2 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 *2 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 timer array unit (TT0 = 0x00FF).
Note 2. 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.
Note 3. If PCLKB (undivided) is selected as the operation clock (CK00, CK01) and TDR0n is set to 0x0000 (n = 0 to 7), interrupt requests output from timer array units cannot be used.
Note 4. 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 17.4.1. Count Clock (f_{TCLK}).

Note: PCLKB: CPU and peripheral hardware clock frequency.

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 bits 3 to 0 of the TPS0 register, and CK01

is selected by using bits 7 to 4 of the TPS0 register. In addition, only for channels 1 and 3, CK02 and CK03 can be also selected. CK02 is selected by using bits 9 and 8 of the TPS0 register, and CK03 is selected by using bits 13 and 12 of the TPS0 register.

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 17.4.

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 17.4.

Table 17.4 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
0000b	PCLKB	2 MHz	5 MHz	10 MHz	20 MHz	32 MHz
0001b	PCLKB/2	1 MHz	2.5 MHz	5 MHz	10 MHz	16 MHz
0010b	PCLKB/2^2	500 kHz	1.25 MHz	2.5 MHz	5 MHz	8 MHz
0011b	PCLKB/2^3	250 kHz	625 kHz	1.25 MHz	2.5 MHz	4 MHz
0100b	PCLKB/2^4	125 kHz	313 kHz	625 kHz	1.25 MHz	2 MHz
0101b	PCLKB/2^5	62.5 kHz	156 kHz	313 kHz	625 kHz	1 MHz
0110b	PCLKB/2^6	31.3 kHz	78.1 kHz	156 kHz	313 kHz	500 kHz
0111b	PCLKB/2^7	15.6 kHz	39.1 kHz	78.1 kHz	156 kHz	250 kHz
1000b	PCLKB/2^8	7.81 kHz	19.5 kHz	39.1 kHz	78.1 kHz	125 kHz
1001b	PCLKB/2^9	3.91 kHz	9.77 kHz	19.5 kHz	39.1 kHz	62.5 kHz
1010b	PCLKB/2^{10}	1.95 kHz	4.88 kHz	9.77 kHz	19.5 kHz	31.3 kHz
1011b	PCLKB/2^{11}	977 Hz	2.44 kHz	4.88 kHz	9.77 kHz	15.6 kHz
1100b	PCLKB/2^{12}	488 Hz	1.22 kHz	2.44 kHz	4.88 kHz	7.81 kHz
1101b	PCLKB/2^{13}	244 Hz	610 Hz	1.22 kHz	2.44 kHz	3.91 kHz
1110b	PCLKB/2^{14}	122 Hz	305 Hz	610 Hz	1.22 kHz	1.95 kHz
1111b	PCLKB/2^{15}	61.0 Hz	153 Hz	305 Hz	610 Hz	977 Hz

Note 1. When changing the clock selected for PCLKB, stop timer array unit (TT0 = 0x00FF).

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: PCLKB: CPU and peripheral hardware clock frequency.

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 17.4.1. Count Clock (f_{TCLK}).

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 17.5.

Table 17.5 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
00b	PCLKB/2	1 MHz	2.5 MHz	5 MHz	10 MHz	16 MHz
01b	PCLKB/2^2	500 kHz	1.25 MHz	2.5 MHz	5 MHz	8 MHz
10b	PCLKB/2^4	125 kHz	313 kHz	625 kHz	1.25 MHz	2 MHz
11b	PCLKB/2^6	31.3 kHz	78.1 kHz	156 kHz	313 kHz	500 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 17.6.

Table 17.6 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
00b	PCLKB/2^8	7.81 kHz	19.5 kHz	39.1 kHz	78.1 kHz	125 kHz
01b	PCLKB/2^{10}	1.95 kHz	4.88 kHz	9.77 kHz	19.5 kHz	31.3 kHz
10b	PCLKB/2^{12}	488 Hz	1.22 kHz	2.44 kHz	4.88 kHz	7.81 kHz
11b	PCLKB/2^{14}	122 Hz	305 Hz	610 Hz	1.22 kHz	1.95 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 17.7 can be achieved by using the interval timer function.

Table 17.7 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 1. The margin is within 5%.

Note: PCLKB: CPU and peripheral hardware clock frequency

Note: For details of a signal of PCLKB/2^j selected with the TPS0 register, see section 17.4.1. Count Clock (f_{TCLK}).

17.2.4 TMR0n : Timer Mode Register 0n (n = 0, 2, 4, 5, 6, 7)

Base address: TAU = 0x400A_2600

Offset address: 0x0110 (TMR00)
0x0114 (TMR02)
0x0118 (TMR04)
0x011A (TMR05)
0x011C (TMR06)
0x011E (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 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	MASTER*2	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 0, valid edge of input signal selected by the TIS1 register • 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 1. In one-count mode, interrupt output (TAU0_TMI0n) when starting a count operation and TO0n output are not controlled.
Note 2. Not supported (Bit 11 is a read-only bit and fixed to 0) when n = 0, 5, 7. Writing to this bit is ignored.

Note: The bit function assigned to bit 11 of the TMR0n register depends on the channel.
TMR00, TMR05, TMR07: Fixed to 0

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 while the corresponding timer is in operation (when TE0.TE[n] = 1). However, bits 7 and 6 (CIS[1:0]) can be rewritten even while the corresponding timer is operating with some functions (when TE0.TE[n] = 1). For details, see section 17.7. Independent Channel Operation Function of Timer Array Unit and section 17.8. Simultaneous Channel Operation Function of Timer Array Unit.

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}).

OPIRQ bit (Setting of Starting Count and Interrupt)

Table 17.8 lists operation mode that can selected with MD[2:0] bits and OPIRQ bit.

Table 17.8 OPIRQ 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)	0	Start trigger is invalid during counting operation. At that time, interrupt is not generated.
	1	Start trigger is valid during counting operation *1. At that time, interrupt is not 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. 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 17.8).

Table 17.9 lists operation mode that can be selected with MD[2:0] bits.

Table 17.9 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

Table 17.9 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
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
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 the channel 2, 4, 6 can be set as a master channel (MASTER = 1).

Be sure to use channel 0, 5, 7 are 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 bits (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.

17.2.5 TMR0n : Timer Mode Register 0n (n = 1, 3)

Base address: TAU = 0x400A_2600

Offset address: 0x0112 (TMR01)
0x0116 (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

Bit	Symbol	Function	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	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 In the case of unit 0: • In channel 1, valid edge of input signal selected by the TIS1 register • 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 bit function assigned to bit 11 of the TMR0n register depends on the channel.
TMR01, TMR03: SPLIT bit (n = 1, 3)

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, bits 7 and 6 (CIS[1:0]) can be rewritten even while the register is operating with some functions (when TE0.TE[n] = 1). For details, see section 17.7. Independent Channel Operation Function of Timer Array Unit and section 17.8. Simultaneous Channel Operation Function of Timer Array Unit.

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}).

OPIRQ bit (Setting of Starting Count and Interrupt)

Table 17.8 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 TMR0n.OPIRQ bit (see Table 17.8).

Table 17.9 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.

17.2.6 TSR0n : Timer Status Register 0n (n = 0 to 7)

Base address: TAU = 0x400A_2600

Offset address: 0x0120 + 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 17.10 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 memory manipulation instruction.

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 17.10 shows the OVF bit operation, set, and clear conditions in each operation mode.

Table 17.10 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.

17.2.7 TE0 : Timer Channel Enable Status Register 0

Base address: TAU = 0x400A_2600

Offset address: 0x0130

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 Stopped State of Channel n 0: Operation is stopped 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 Stopped when Channel 1 is in the 8-bit Timer Mode 0: Operation is stopped 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 Stopped when Channel 3 is in the 8-bit Timer Mode 0: Operation is stopped 1: Operation is enabled	R
15:12	—	These bits are read as 0.	R

The TE0 register is used to enable or stop 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 set with an 8-bit memory manipulation instruction.

17.2.8 TS0 : Timer Channel Start Register 0

Base address: TAU = 0x400A_2600

Offset address: 0x0132

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 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.	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.	R/W

Bit	Symbol	Function	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.	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): Four cycles of the operation clock (f_{MCK}) When the Ti0n pin noise filter is disabled (TNFEN.TNFEN0n = 0): Two cycles of the operation clock (f_{MCK})

Note: When the TS0 register is read, 0 is always read.

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), because they are trigger bits.

TS[7:0] bits (Operation Enable (Start) Trigger of Channel n)

The TCR0n register count operation start in the count operation enabled state varies depending on each operation mode (see Table 17.11 in section 17.4.2. Timing of the Start of Counting).

This bit is the trigger to enable operation (start operation) of the lower 8-bit timer for TS01 and TS03 when channel 1 or 3 is in the 8-bit timer mode.

TSH1 bit (Trigger to Enable Operation (Start Operation) of the Higher 8-bit Timer when Channel 1 is in the 8-bit Timer Mode)

The TCR01 register count operation start in the interval timer mode in the count operation enabled state (see Table 17.11 in section 17.4.2. Timing of the Start of Counting).

TSH3 bit (Trigger to Enable Operation (Start Operation) of the Higher 8-bit Timer when Channel 3 is in the 8-bit Timer Mode)

The TCR03 register count operation start in the interval timer mode in the count operation enabled state (see Table 17.11 in section 17.4.2. Timing of the Start of Counting).

17.2.9 TT0 : Timer Channel Stop Register 0

Base address: TAU = 0x400A_2600

Offset address: 0x0134

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 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.	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.	R/W

Bit	Symbol	Function	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.	R/W

Note: When the TT0 register is read, 0 is always read.

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), because they are trigger bits.

17.2.10 TIS0 : Timer Input Select Register 0

Base address: PORGA = 0x400A_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 System clock (FSUB) Others: Setting prohibited	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

The TIS0 register is used to select the channel 5 of unit 0 timer input.

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 FSUB as PCLKB, the TIS[2] bit cannot be set to 1.

17.2.11 TIS1 : Timer Input Select Register 1

Base address: PORGA = 0x400A_1000

Offset address: 0x0005

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	TIS1	TIS0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TIS0	Selection of Timer Input Used with Channel 0 0: Input signal of timer input pin (TI00) 1: Event input signal from ELC	R/W
1	TIS1	Selection of Timer Input Used with Channel 1 0: Input signal of timer input pin (TI01) 1: Event input signal from ELC	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W

The TIS1 register is used to select channels 0 and 1 of unit 0 timer input.

Note: When selecting the event input signal from ELC in this register, select PCLKB (undivided) as the operating clock in timer clock select register 0 (TPS0).

17.2.12 TOE0 : Timer Output Enable Register 0

Base address: TAU = 0x400A_2600

Offset address: 0x013A

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 TO0.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 memory manipulation instruction.

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 a 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).

17.2.13 TO0 : Timer Output Register 0

Base address: TAU = 0x400A_2600

Offset address: 0x0138

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 PSEL[2:0] bits.

The lower 8 bits of the TO0 register can be set with an 8-bit memory manipulation instruction.

Note: n: Channel number (n = 0 to 7)

17.2.14 TOL0 : Timer Output Level Register 0

Base address: TAU = 0x400A_2600

Offset address: 0x013C

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

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). In the master channel output mode (TOM0.TOM[n] = 0), this register setting is invalid.

The lower 8 bits of the TOL0 register can be set with an 8-bit memory manipulation instruction.

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.

Note: n: Channel number (n = 0 to 7)

17.2.15 TOM0 :Timer Output Mode Register 0

Base address: TAU = 0x400A_2600

Offset address: 0x013E

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_TMI0n)) 1: Slave channel output mode (output is set by the timer interrupt request signal (TAU0_TMI0n) of the master channel, and reset by the timer interrupt request signal (TAU0_TMI0p) 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 memory manipulation instruction.

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 17.3.1. Basic Rules of Simultaneous Channel Operation Function.

17.2.16 ISC : Input Switch Control Register

See section 21.3.26. ISC : Input Switch Control Register.

17.2.17 TNFEN : TAU Noise Filter Enable Register

Base address: PORGA = 0x400A_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
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 to use the noise filter of the TI07 pin can be selected.
ISC.ISC1 = 1: Whether to use the noise filter of the RXD2 pin can be selected.

Note: The presence or absence of timer I/O pins of channels 0 to 7 depends on the product.

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 through the TI0n pin is selected (TMR0n.CCS = 1), section 17.4.2. Timing of the Start of Counting, and section 17.6. Timer Input (TI0n) Control.

17.2.18 Registers Controlling Port Functions of Pins to be Used for Timer I/O

Set the following registers to control the port functions multiplexed with the inputs and outputs of timer array units.

- Pmn Direction Register (PDRm) or PDR bit of Port mn Pin Function Select Register (PmnPFS_A)
- PSEL[2:0] bits of Port mn Pin Function Select Register (PmnPFS_A)
- Do not use the analog input (for example, AN0xx).

For details, see the following sections.

- section 16.2.2. PDRm : Pmn Direction Register (m = 0 to 5, 9, n = 00 to 15)
- section 16.2.8. PmnPFS_A : Port mn Pin Function Select Register (m = 1 to 9, n = 00 to 15)
- P00n4PFS_A : Port 00n4 Pin Function Select Register (n = 0 to 2)
- section 16.2.11. P9nPFS_A : Port 9n Pin Function Select Register (n = 13, 14)
- section 16.6. Peripheral Select Settings for Each Product.

When the pins multiplexed with TO01 to TO07 are to be used for outputs of timers, set the following registers.

- Set the PDRxx bit of Pmn Direction Register (PDRm) or PDR bit of Port mn Pin Function Select Register (PmnPFS_A)
  Set the PDRm.PDRxx bit to 1 or set the PmnPFS_A.PDR bit to 1.
- PSEL[2:0] bits of Port mn Pin Function Select Register (PmnPFS_A)
  Select the TO0x function by setting PSEL[2:0] bits of Port mn Pin Function Select Register (PmnPFS_A).

When the pins multiplexed with TI01 to TI07 are to be used for inputs of timers, set the following registers.

- Set the PDRxx bit of Pmn Direction Register (PDRm) or PDR bit of Port mn Pin Function Select Register (PmnPFS_A)
  Set the PDRm.PDRxx bit to 0 or set the PmnPFS_A.PDR bit to 0.
- PSEL[2:0] bits of Port mn Pin Function Select Register (PmnPFS_A)
  Select the TI0x function by setting PSEL[2:0] bits of Port mn Pin Function Select Register (PmnPFS_A).

17.3 Basic Rules of Timer Array Unit

17.3.1 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 CKS[1:0] bits (bit 15, 14 of timer mode register 0n (TMR0n)) 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_TMI0n (interrupt), start software trigger, and count clock to the lower channels.
8. A slave channel can use TAU0_TMI0n (interrupt), a start software trigger, or the count clock of the master channel as a source clock, but cannot transmit its own TAU0_TMI0n (interrupt), start software trigger, or count clock to channels with lower channel numbers.
9. A master channel cannot use TAU0_TMI0n (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] bit of all channels which are operating simultaneously can be set. It cannot be applied to TS0.TS[n] bit 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 17.19 shows an example of how TAU can be used.

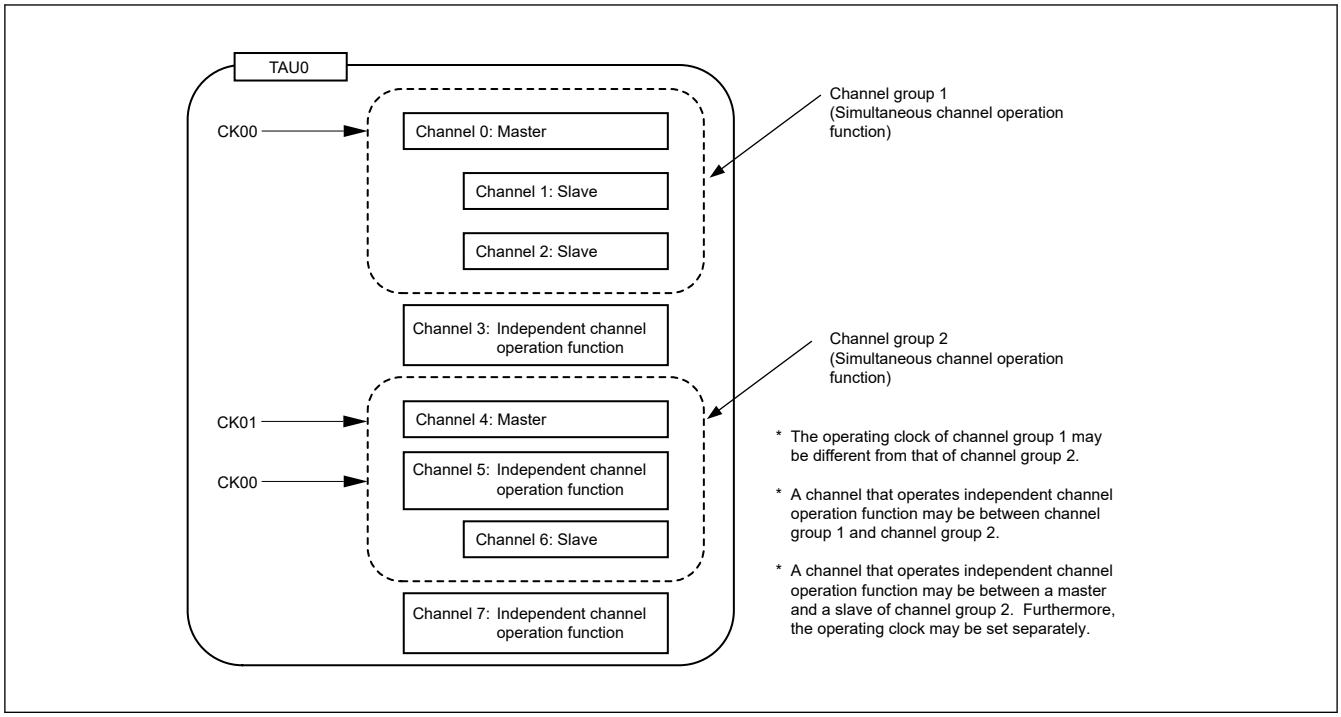


Figure 17.19 TAU utilization example

17.3.2 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_TMI01H and TAU0_TMI03H (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] and 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)

17.4 Operations of Counters

17.4.1 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

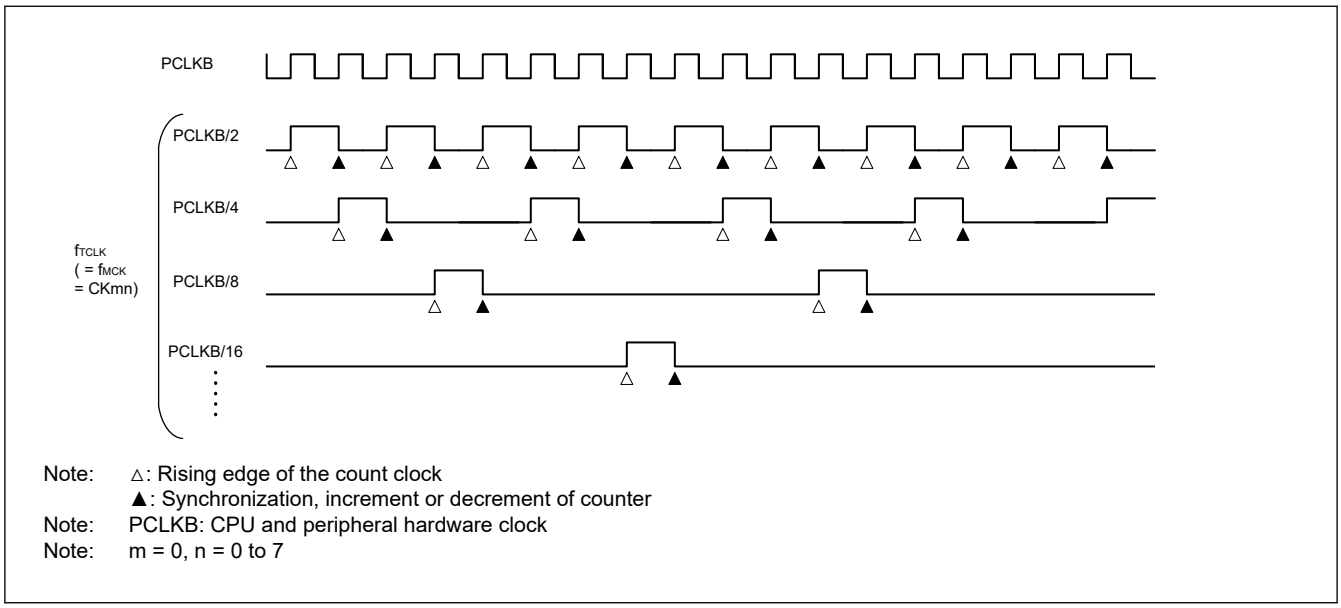
Because the timer array unit is designed to operate in synchronization with PCLKB, the timings of the count clock (f_{TCLK}) are shown below.

(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, however, the clock selected in TPS0n register, but a signal which becomes high level for one period of PCLKB from its rising edge. When a PCLKB is selected, fixed to high level

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 17.20 shows the count clock (f_{TCLK}) timing from PCLKB when TMR0n.CCS = 0.



Note: △: Rising edge of the count clock
▲: Synchronization, increment or decrement of counter
Note: PCLKB: CPU and peripheral hardware clock
Note: m = 0, n = 0 to 7

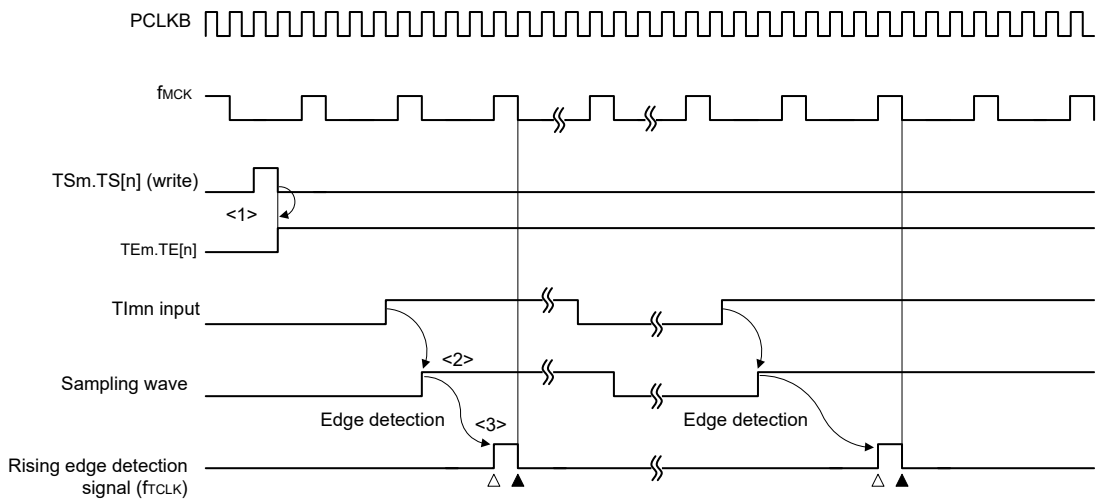
Figure 17.20 Timing of PCLKB and count clock (f_{TCLK}) (when TMR0n.CCS = 0)

(2) When valid edge of input signal through the TI0n pin is selected (TMR0n.CCS = 1)

The count clock (f_{TCLK}) becomes the signal that detects valid edge of input signal through 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 through the Ti0n pin (when a noise filter is used, the delay becomes 3 to 4 clock).

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 valid edge of input signal through the TI0n pin", as a matter of convenience.

Figure 17.21 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: PCLKB: CPU and peripheral hardware clock
f_{MCK}: Operation clock of channel n
Note: The waveform of the input signal through TImn 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.
Note: m = 0, n = 0 to 7

Figure 17.21 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 through 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.

17.4.2 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 17.11.

Table 17.11 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)).
• 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).

Table 17.11 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
• 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)).

17.4.3 Operations of Counters

Here, the counter operation in each mode is explained.

(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.

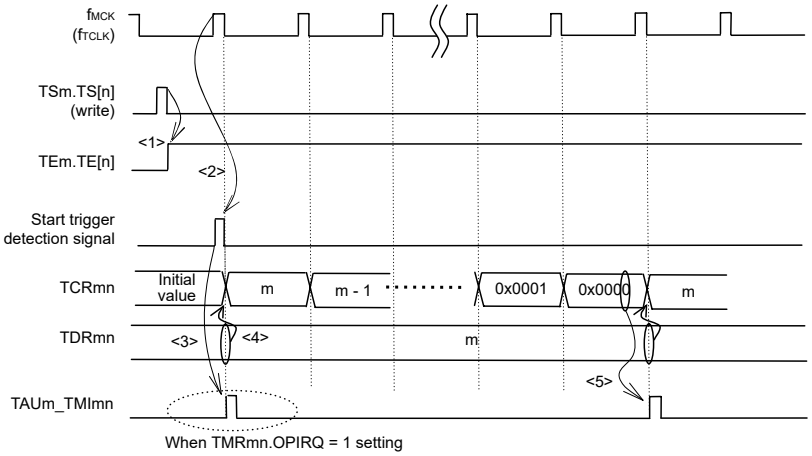
<2> A start trigger is generated at the first count clock after operation is enabled.

<3> When the TMR0n.OPIRQ bit is set to 1, TAU0_TMI0n 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_TMI0n is generated and the value of timer data register 0n (TDR0n) is loaded to the TCR0n register and counting keeps on.

Figure 17.22 shows the timing during operation in interval timer mode.



Note: In the first cycle operation of count clock after writing the TSm.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 TMRmn.OPIRQ = 1.

Note: f_{MCK}, the start trigger detection signal, and TAUm_TMImn become active between one clock in synchronization with PCLKB.

Note: m = 0, n = 0 to 7

Figure 17.22 Timing during operation in interval timer mode

(2) Operation in event counter mode

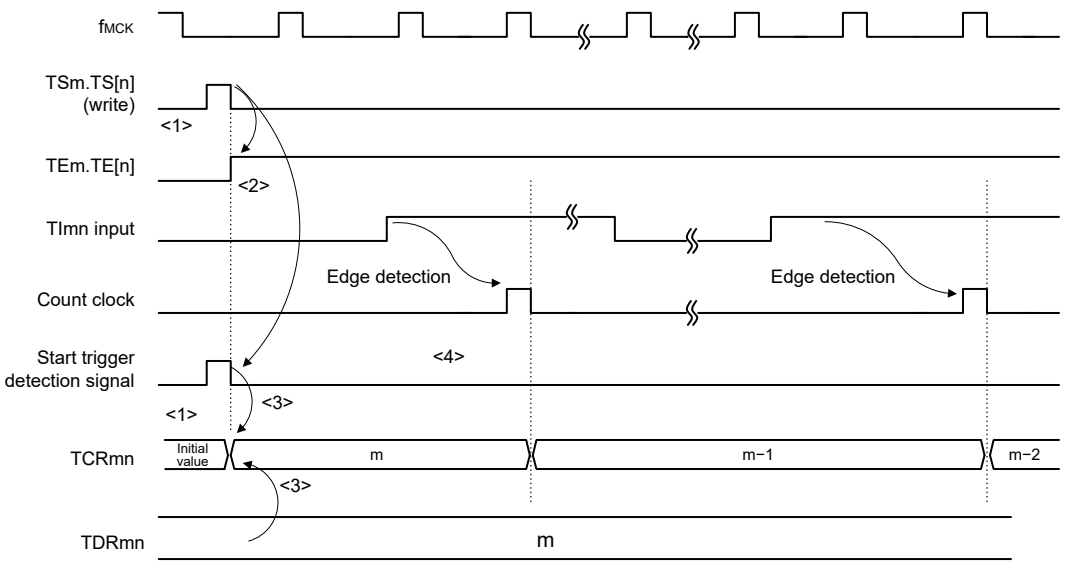
<1> Timer counter register 0n (TCR0n) holds its initial value while operation is stopped (TE0.TE[n] = 0).

<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 17.23 shows the timing during operation in event counter mode.



Note: Figure 17.23 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 TImn input. The error per one period occurs be the asynchronous between the period of the TImn input and that of the count clock (f_{MCK}).

Note: m = 0, n = 0 to 7

Figure 17.23 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.

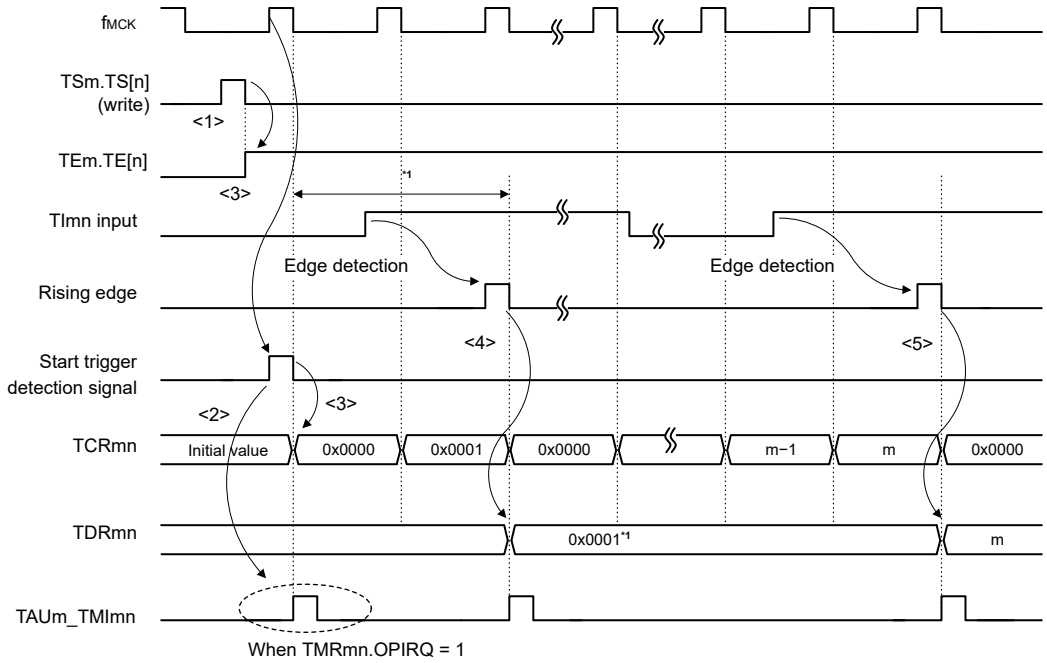
<2> Timer counter register 0n (TCR0n) holds the initial value until count clock generation.

<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_TMI0n 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_TMI0n 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_TMI0n is generated.

Figure 17.24 shows the timing during operation in capture mode (Input Pulse Interval Measurement).



Note: In the first cycle operation of count clock after writing the TSm.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 TMRmn.OPIRQ = 1.

Note: If a clock has been input to TImn (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.

Note: Figure 17.24 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 TImn input. The error per one period occurs be the asynchronous between the period of the TImn input and that of the count clock (f_{MCK}).

Note: m = 0, n = 0 to 7

Figure 17.24 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.

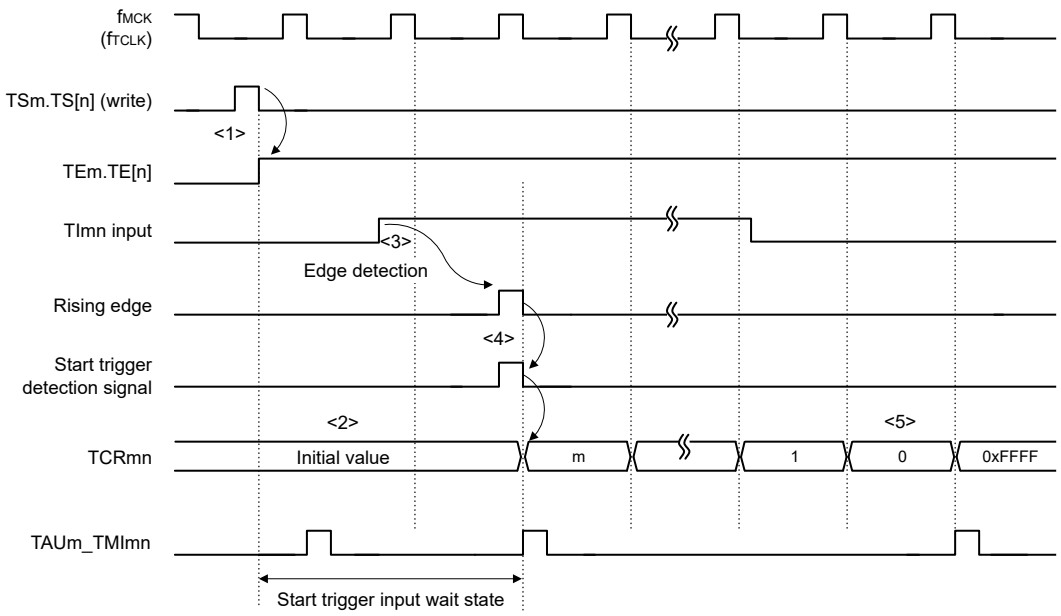
<2> Timer counter register 0n (TCR0n) holds the initial value until start trigger generation.

<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_TMI0n is generated and the value of the TCR0n register becomes 0xFFFF and counting stops.

Figure 17.25 shows the timing during operation in one-count mode.



Note: Figure 17.25 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 TImn input. The error per one period occurs be the asynchronous between the period of the TImn input and that of the count clock (f_{MCK}).

Note: m = 0, n = 0 to 7

Figure 17.25 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).

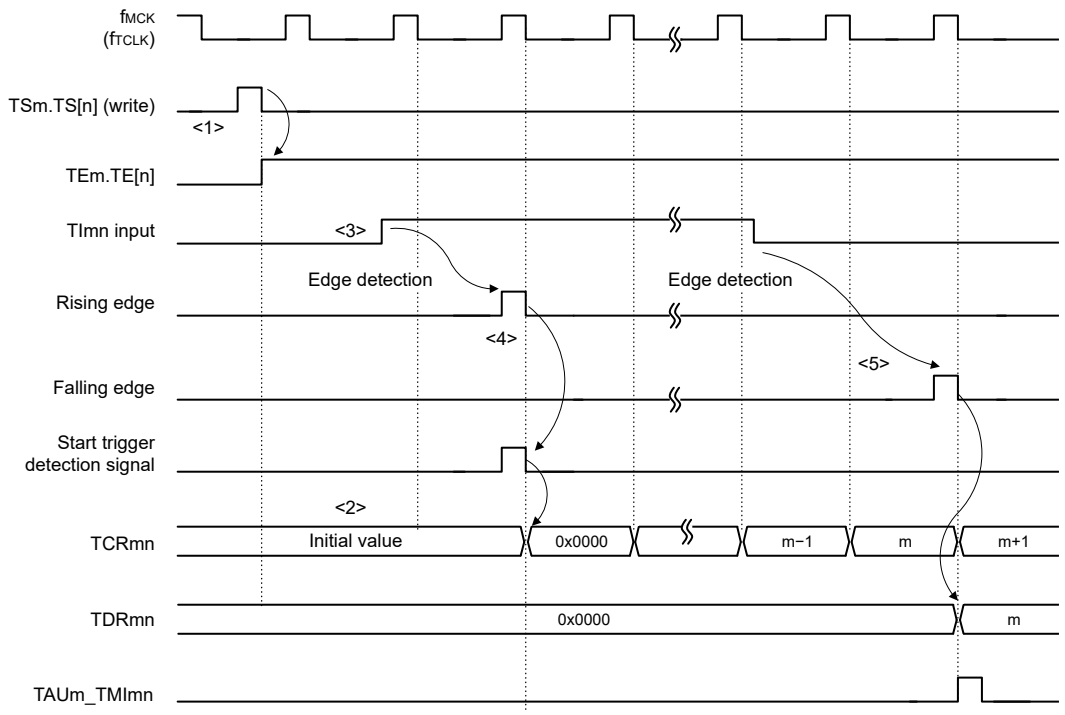
<2> Timer counter register 0n (TCR0n) holds the initial value until start trigger generation.

<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_TMI0n is generated.

Figure 17.26 shows the timing during operation in capture & one-count mode (High-Level Width Measurement).



Note: Figure 17.26 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 TImn input. The error per one period occurs be the asynchronous between the period of the TImn input and that of the count clock (f_{MCK}).

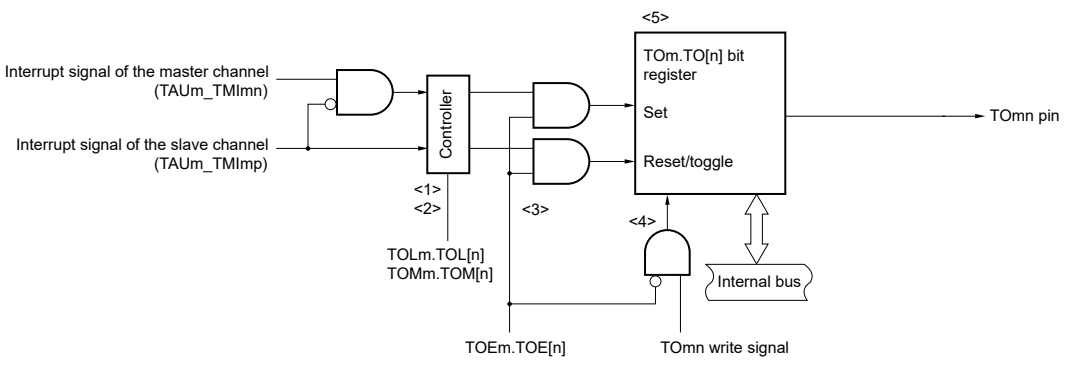
Note: m = 0, n = 0 to 7

Figure 17.26 Timing during operation in capture & one-count mode (high-level width measurement)

17.5 Channel Output (TO0n Pin) Control

17.5.1 TO0n Pin Output Circuit Configuration

Figure 17.27 shows the TO0n pin output circuit.



Note: m = 0, n = 0 to 7

Figure 17.27 Output circuit configuration

The following describes the TO0n pin output circuit.

<1> When TOM0.TOM[n] = 0 (master channel output mode), the set value of timer output level register 0 (TOL0) is ignored and only TAU0_TMI0p (slave channel timer interrupt) is transmitted to timer output register 0 (TO0).

<2> When TOM0.TOM[n] = 1 (slave channel output mode), both TAU0_TMI0n (master channel timer interrupt) and TAU0_TMI0p (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] = 0: Positive logic output (TAU0_TMI0n → set, TAU0_TMI0p → reset)

When TOL0.TOL[n] = 1: Negative logic output (TAU0_TMI0n → reset, TAU0_TMI0p → set)

When TAU0_TMI0n and TAU0_TMI0p are simultaneously generated, (0% output of PWM), TAU0_TMI0p (reset signal) takes priority, and TAU0_TMI0n (set signal) is masked.

<3> While timer output is enabled (TOE0.TOE[n] = 1), TAU0_TMI0n (master channel timer interrupt) and TAU0_TMI0p (slave channel timer interrupt) are transmitted to the TO0 register. Writing to the TO0 register (TO0.TO[n] 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 (TO0.TO[n] write signal) becomes valid. When timer output is disabled (TOE0.TOE[n] = 0), neither TAU0_TMI0n (master channel timer interrupt) nor TAU0_TMI0p (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

17.5.2 TO0n Pin Output Setting

Figure 17.28 shows the procedure and state transitions from the initial settings of a TO0n output pin to the start of timer operation.

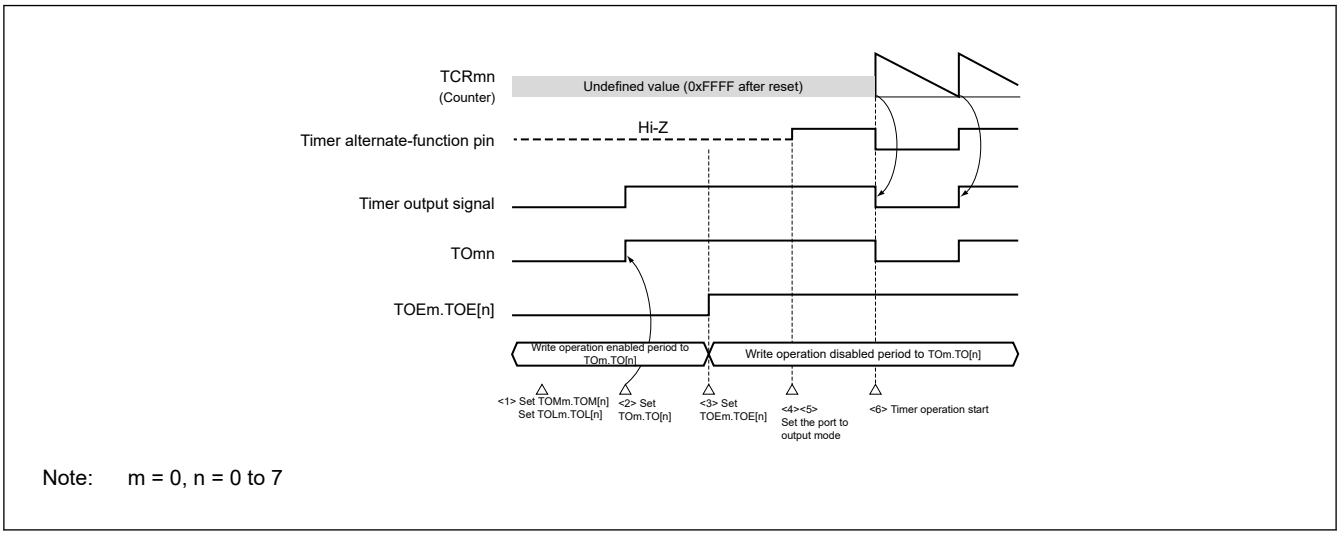


Figure 17.28 State transitions from the settings for timer output to the start of timer operation

<1> The operation mode of timer output is set.

- TOM0.TOM[n] bit (0: Master channel output mode, 1: Slave channel output mode)
- TOL0.TOL[n] 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[2:0] bits of Port mn Pin Function Select Register (PmnPFS_A) (see section 16.2.8. PmnPFS_A : Port mn Pin Function Select Register (m = 1 to 9, n = 00 to 15), P00n4PFS_A : Port 00n4 Pin Function Select Register (n = 0 to 2), section 16.2.11. P9nPFS_A : Port 9n Pin Function Select Register (n = 13, 14)).

<5> The port I/O setting is set to output by the PDR bit of PmnPFS_A register (see section 17.2.18. Registers Controlling Port Functions of Pins to be Used for Timer I/O).

<6> The timer operation is enabled (TS0.TS[n] = 1).

Note: n: Channel number (n = 0 to 7)

17.5.3 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 17.6. Timer Input (TI0n) Control and section 17.7. Independent Channel Operation Function of Timer Array Unit.

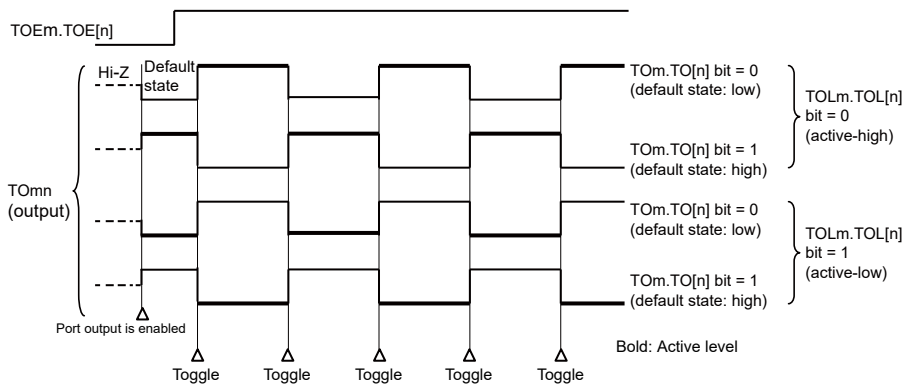
When the values set to the TOE0, and TOM0 registers (but not the TO0 register) are changed close to the occurrence of the timer interrupt (TAU0_TMI0n) of each channel, the waveform output to the TO0n pin might differ, depending on whether the values are changed immediately before or immediately after the timer interrupt (TAU0_TMI0n) 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.

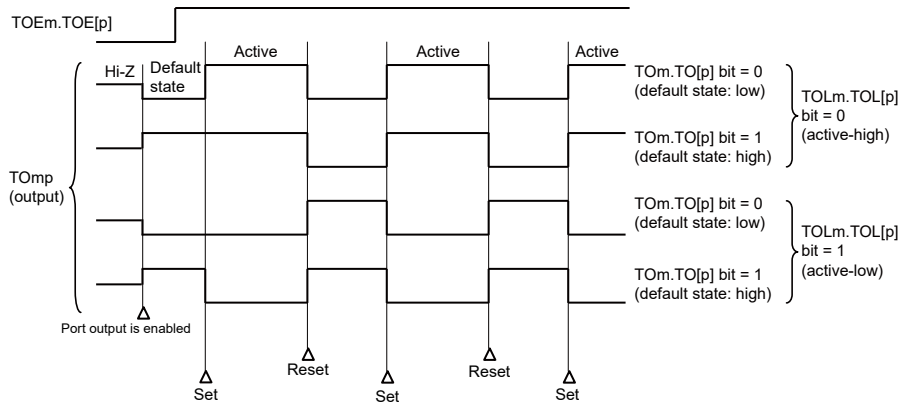
a. When operation starts with master channel output mode (TOM0.TOM[n] = 0) setting
The setting of timer output level register 0 (TOL0) is invalid when master channel output mode (TOM0.TOM[n] = 0). When the timer operation starts after setting the default level, the toggle signal is generated and the output level of the TO0n pin is inverted.
Figure 17.29 shows the output state of the TO0n pin with the output toggled (TOM0.TOM[n] = 0).



Note: Toggle: Toggle signal to invert the output on the TOmn pin
Note: m = 0, n = 0 to 7

Figure 17.29 TOmn pin output states with toggled output (TOM0.TOM[n] = 0)

b. When operation starts with slave channel output mode (TOM0.TOM[p] = 1) setting (PWM output)
When slave channel output mode (TOM0.TOM[p] = 1), the active level is determined by timer output level register 0 (TOL0) setting.
Figure 17.30 shows the output state of the TO0p pin with PWM output (TOM0.TOM[p] = 1).

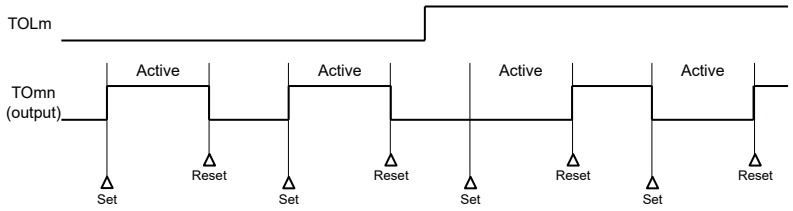


Note: Set: The output signal of the TOmp pin changes from inactive level to active level.
Reset: The output signal of the TOmp pin changes from active level to inactive level.
Note: m = 0, p = 1 to 7

Figure 17.30 TO0p pin output states with PWM output (TOM0.TOM[p] = 1)

(3) Operation of TO0n pin in slave channel output mode (TOM0.TOM[n] = 1)

a. 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] 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 17.31.

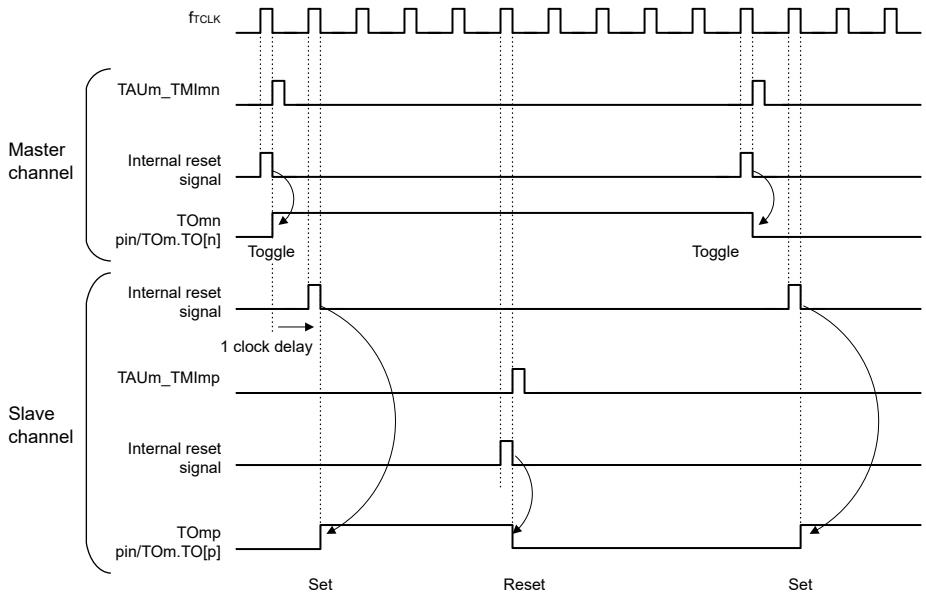


Note: Set: The output signal of the TOmn pin changes from inactive level to active level.
Reset: The output signal of the TOmn pin changes from active level to inactive level.
Note: m = 0, n = 0 to 7

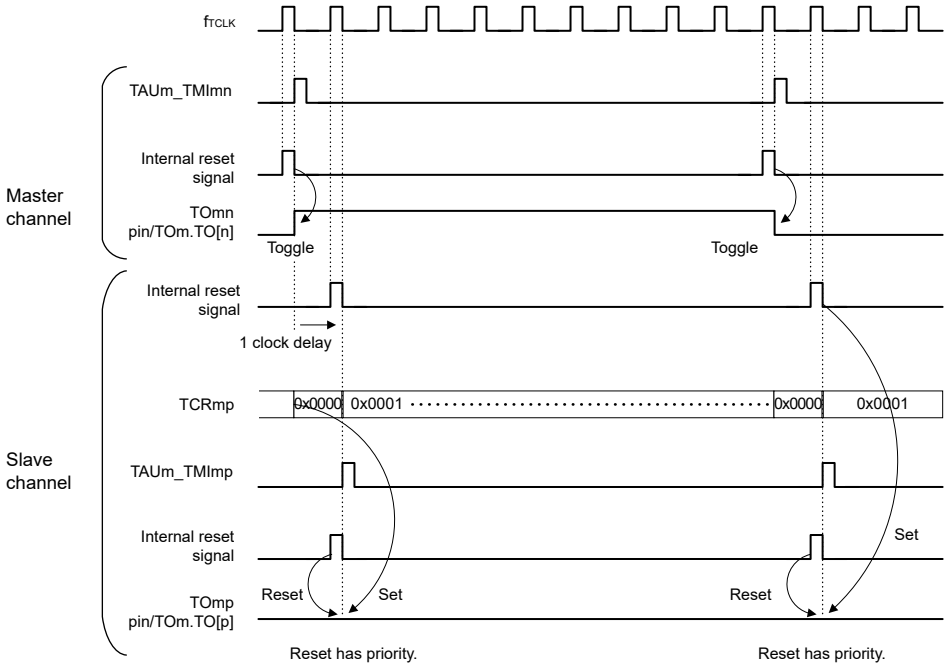
Figure 17.31 Operation when the relevant bit of the TOL0 register is changed during timer operation

b. 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_TMI0n) 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 17.32 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] = 0, TOL0.TOL[n] = 0
Slave channel: TOE0.TOE[p] = 1, TOM0.TOM[p] = 1, TOL0.TOL[p] = 0

(1) Basic timing during operation



(2) Timing during operation with the duty cycle set to 0%



Note: Internal reset signal: TOmn pin reset/toggle signal
Internal set signal: TOmn pin set signal

Note: m: Unit number (m = 0)
n: Channel number
n = 0 to 7 (n = 0, 2, 4, 6 for master channel)
p: Slave channel number
n < p ≤ 7

Figure 17.32 States of operation following set and reset signals

17.5.4 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] bit (TOE0.TOE[n] = 0) that correspond to the relevant bits of the channel used to perform output (TO0n).

Table 17.12 shows an example of a TO0n bit collective manipulation.

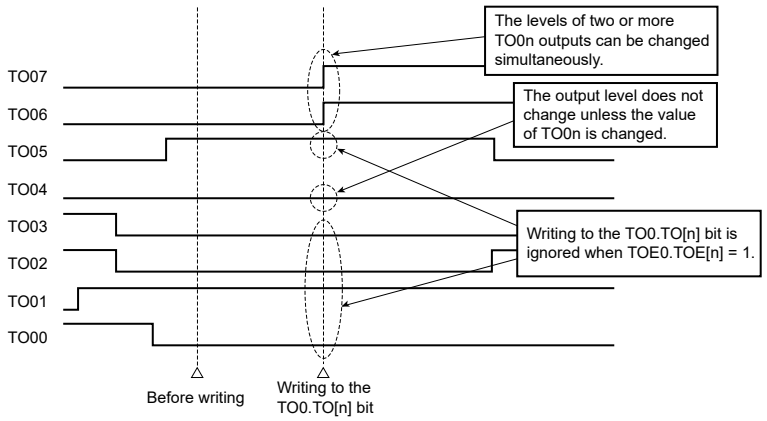
Table 17.12 Example of TO0.TO[n] bit collective manipulation

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 17.33 shows an example of a TO0.TO[n] bit collective manipulation.



Note: n: Channel number (n = 0 to 7)

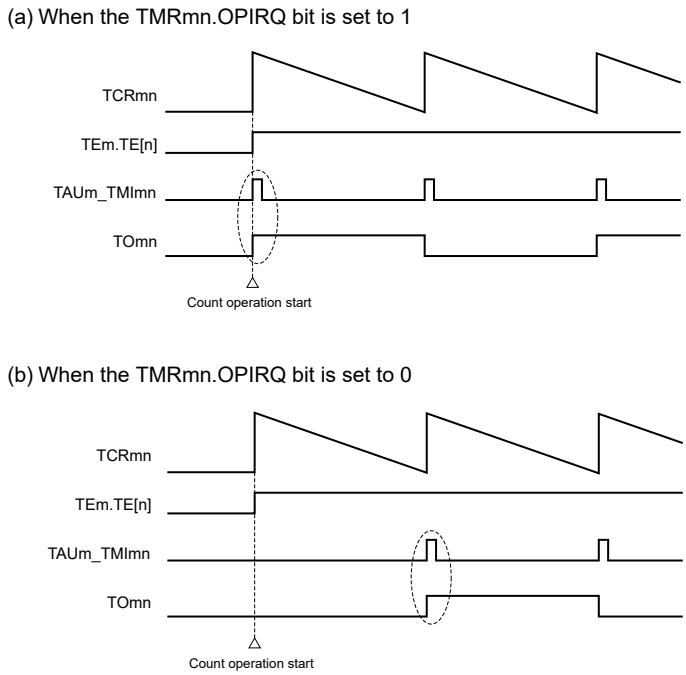
Figure 17.33 TO0n pin states by collective manipulation of TO0.TO[n] bit

17.5.5 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 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_TMI0n) generation. In the other modes, neither timer interrupt at count operation start nor TO0n output is controlled.

Figure 17.34 shows operation examples when the interval timer mode (TOE0.TOE[n] = 1, TOM0.TOM[n] = 0) is set.



Note: m = 0, n = 0 to 7

Figure 17.34 Examples of the operation of timer interrupts and TOmn outputs when counting is started

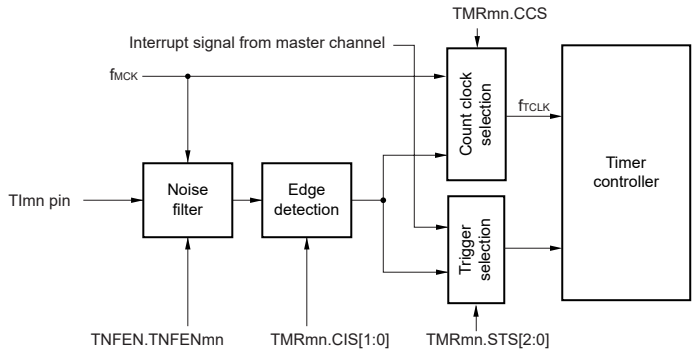
When the TMR0n.OPIRQ bit is set to 1, a timer interrupt (TAU0_TMI0n) is output at count operation start, and TO0n performs a toggle operation.

When the TMR0n.OPIRQ bit is set to 0, a timer interrupt (TAU0_TMI0n) is not output at count operation start, and TO0n does not change either. After counting one cycle, TAU0_TMI0n is output and TO0n performs a toggle operation.

17.6 Timer Input (TI0n) Control

17.6.1 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 17.35 shows the configuration of the input circuit.

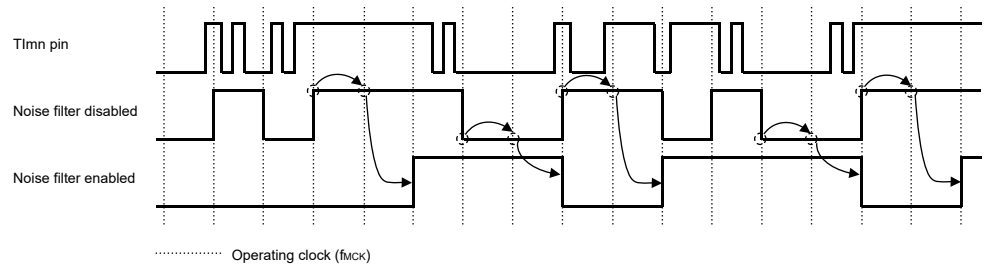


Note: m = 0, n = 0 to 7

Figure 17.35 Input circuit configuration

17.6.2 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 17.36 shows differences in waveforms output from the noise filter between when the noise filter is enabled and disabled.



Note: m = 0, n = 0 to 7

Figure 17.36 Sampling waveforms through TI0n input pin with noise filter enabled and disabled

17.6.3 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 bits 12 (CCS), 9 and 8 (STS[1:0]) in the timer mode register 0n (TMR0n) are 0 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 (TS0).
2. Noise filter is enabled
   When bits 12 (CCS), 9 and 8 (STS[1:0]) in the timer mode register 0n (TMR0n) are all 0 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 (TS0).

17.7 Independent Channel Operation Function of Timer Array Unit

17.7.1 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_TMI0n (timer interrupt) at fixed intervals. The interrupt generation period can be calculated by the following expression.

Generation period of TAU0_TMI0n (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_TMI0n 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 TDR0n + 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 TMR0n.OPIRQ bit of timer mode register

0n (TMR0n) is 0 at this time,TAU0_TMI0n is not output and the output on TO0n is not toggled. If the TMR0n.OPIRQ bit of the TMR0n register is 1, TAU0_TMI0n is output and the output on TO0n is toggled.

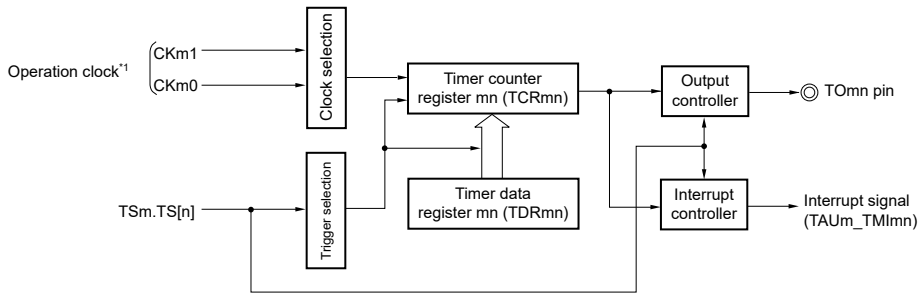
After that, the TCR0n register count down in synchronization with the count clock.

When TCR0n = 0x0000, TAU0_TMI0n 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)

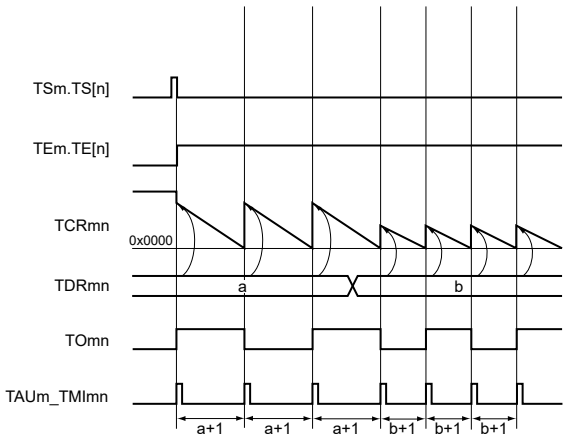
Figure 17.37 shows a block diagram of the operation as an interval timer or a square wave output.



Note: m = 0, n = 0 to 7
Note 1. For channels 1 and 3, the clock can be selected as CKm0, CKm1, CKm2, or CKm3.

Figure 17.37 Block diagram for operation as an interval timer or for square wave output

Figure 17.38 shows an example of basic timing during operation as an Interval timer or for square wave output (TMR0n.OPIRQ = 1).



Note: m = 0, n = 0 to 7

Figure 17.38 Example of basic timing during operation as an interval timer or for square wave output (TMR0n.OPIRQ = 1)

Table 17.13 to Table 17.18 show register settings and procedure for operation when the interval timer or square wave output.

Table 17.13 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_TMI0n nor inverts timer output when counting is started. 1: Generates TAU0_TMI0n 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 17.14 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 17.15 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 17.16 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] (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] = 0 (master channel output mode).

Table 17.17 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] (n = 1 to 7)		Control of timer output mode of channel n (channels 1 to 7) 0: Sets master channel output mode.

Table 17.18 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		—		Power-off state (Clock supply is stopped and writing to each register is disabled.)
	<1>	Sets the MSTPD0 bit of Module Stop Control Register D (MSTPCRD) to 0.	→	Power-on state. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	<2>	Sets timer clock select register 0 (TPS0). Determines clock frequencies of CK00 to CK03.		—
Channel default setting	<3>	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.)
	<4>	To use the TO0n output Clears the TOM0.TOM[n] bit of timer output mode register 0 (TOM0) to 0 (master channel output mode). Clears the TOL0.TOL[n] 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 Ppq Direction Register (PDRp) 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 Ppq Direction Register (PDRp) to 1.	→	The TO0n pin outputs the TO0n set level.
Operation start	<5>	(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_TMI0n is generated and TO0n performs toggle operation if the TMR0n.OPIRQ bit of the TMR0n register is 1.
During operation	<6>	Set values of the TMR0n register, TOM0.TOM[n], and TOL0.TOL[n] 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_TMI0n is generated and TO0n performs toggle operation. After that, the above operation is repeated.

Table 17.18 Procedure for operations when the interval timer or square wave output function is to be used (2 of 2)

	Step	Software operation		Hardware state
Operation stop	<7>	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.
	<8>	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 <5>. To terminate the operation, go to step <9>	→	The TO0n pin outputs the TO0.TO[n] bit set level.
TAU stop	<9>	To hold the TO0n pin output level Set PSEL[2:0] bits to 000b after the value to be held is set to the Ppq Output Data Register (PODRp). When holding the TO0n pin output level is not necessary Setting not required.	→	The TO0n pin output level is held by port function.
	<10>	Sets the MSTPD0 bit of Module Stop Control Register D (MSTPCRD) to 1.	→	This stops supply of the input clock to timer array unit 0. Power-off state.

Note: n = 0 to 7, p = 0 to 9, q = 00 to 15

17.7.2 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] bit 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_TMI0n.

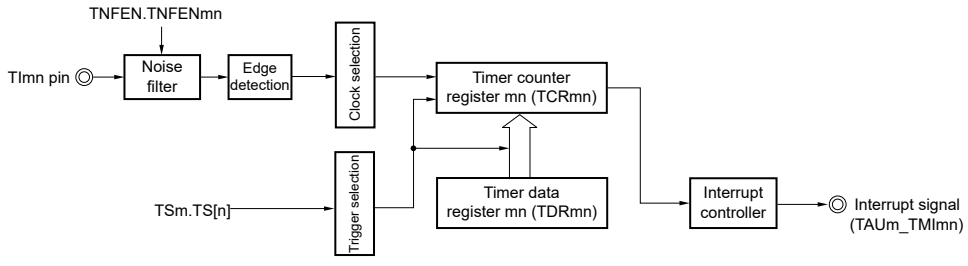
After that, the above operation is repeated.

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.

Instead of using the TI0n pin input, a channel specified for the external event counter function can also use the timer input selected in the TIS0 or TIS1 register as its input source to drive counting.

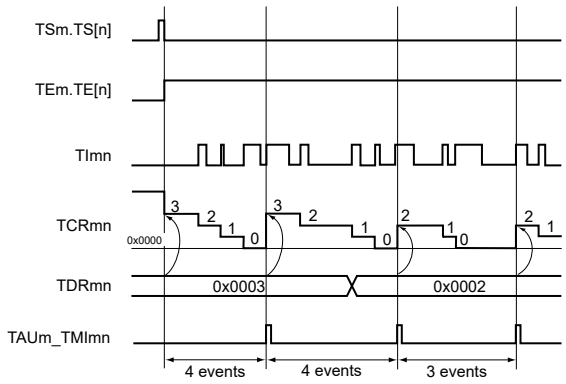
Figure 17.39 shows a block diagram of the operation as an external event counter.



Note: m = 0, n = 0 to 7

Figure 17.39 Block diagram for operation as an external event counter

Figure 17.40 shows an example of basic timing during operation as an external event counter.



Note: m = 0, n = 0 to 7

Figure 17.40 Example of basic timing during operation as an external event counter

Table 17.19 to Table 17.24 show register settings and procedure for operation when the external event counter.

Table 17.19 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_TMI0n 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 edges. 1 0: Detects both edge. 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 17.19 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 17.20 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 17.21 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 17.22 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] (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] = 0 (master channel output mode).

Table 17.23 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] (n = 1 to 7)		Control of timer output mode of channel n (channels 1 to 7) 0: Sets master channel output mode.

Table 17.24 Procedure for operations when the external event counter function is to be used (1 of 2)

	Step	Software operation		Hardware state
TAU default setting		—		Power-off state (Clock supply is stopped and writing to each register is disabled.)
	<1>	Sets the MSTPD0 bit of Module Stop Control Register D (MSTPCRD) to 0.	→	Power-on state. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	<2>	Sets timer clock select register 0 (TPS0). Determines clock frequencies of CK00 to CK03.		—
Channel default setting	<3>	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.)

Table 17.24 Procedure for operations when the external event counter function is to be used (2 of 2)

	Step	Software operation		Hardware state
Operation start	<4>	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.
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. Set values of the TMR0n register, TOM0.TOM[n], TOL0.TOL[n], 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_TMI0n output is generated. After that, the above operation is repeated.
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 <4>. To terminate the operation, go to step <7>	→	TE0.TE[n] = 0, and count operation stops. The TCR0n register holds count value and stops.
TAU stop	<7>	Sets the MSTPD0 bit of Module Stop Control Register D (MSTPCRD) to 1.	→	This stops supply of the input clock to timer array unit 0. Power-off state.

Note: n = 0 to 7

17.7.3 Operation as a Frequency Divider (Channel 0 of Unit 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_TMI00 is not output and the output on TO00 is not toggled. If the OPIRQ bit of timer mode register 00 (TMR00) is 1, TAU0_TMI00 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 17.41 shows a block diagram for operation as a frequency divider.

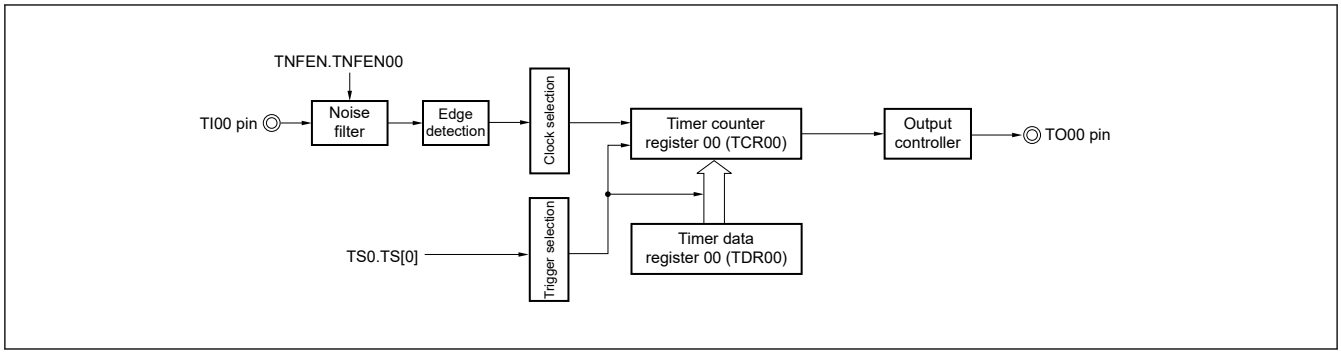


Figure 17.41 Block diagram for operation as a frequency divider

Figure 17.42 shows an example of basic timing during operation as a frequency divider (TMR00.OPIRQ = 1).

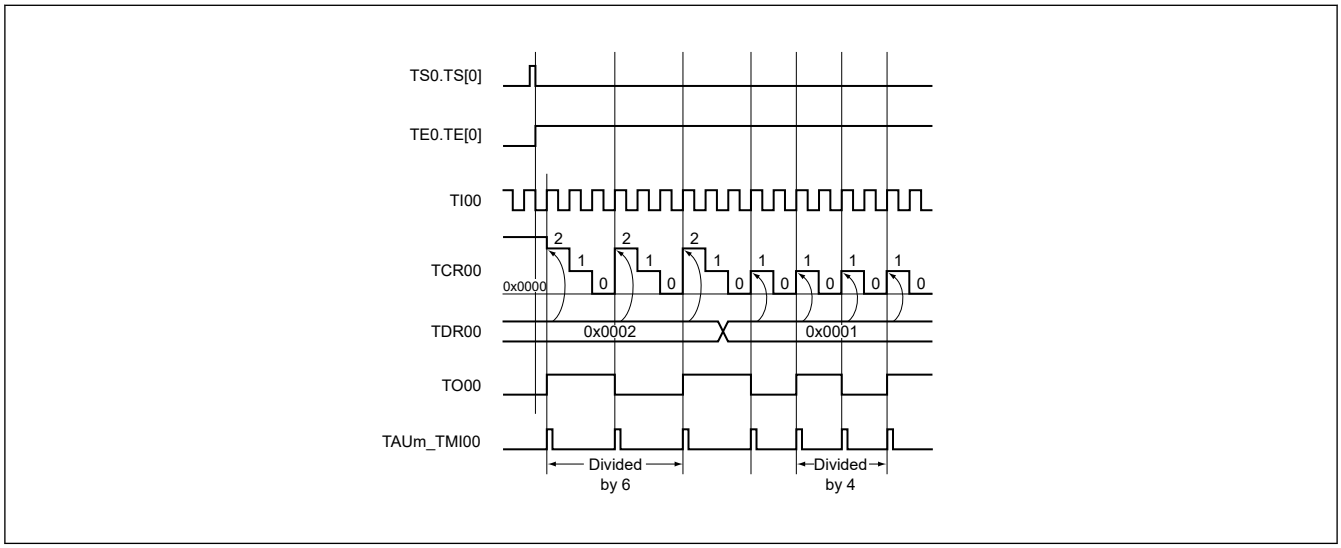


Figure 17.42 Example of basic timing during operation as a frequency divider (TMR00.OPIRQ = 1)

Table 17.25 to Table 17.30 show register settings and procedure for operation when a frequency divider.

Table 17.25 Example of TMR00 settings for operation as a frequency divider (1 of 2)

Bit	Symbol	Set value	Function
0	OPIRQ	1/0	Setting of operation when counting is started 0: Neither generates TAU0_TMI00 nor inverts timer output when counting is started. 1: Generates TAU0_TMI00 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 TImn pin input edge 0 0: Detects falling edge. 0 1: Detects rising edges. 1 0: Detects both edge. Others: Setting prohibited.
10:8	STS[2:0]	000b	Start trigger selection 0 0 0: Selects only software start.
11	—	0	Fixed to 0

Table 17.25 Example of TMR00 settings for operation as a frequency divider (2 of 2)

Bit	Symbol	Set value	Function
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 17.26 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 17.27 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 17.28 Example of TOL0 settings for operation as a frequency divider

Bit	Symbol	Set value	Function
0	TOL[0]	0	Control of timer output of channel 0 0: Set this bit to 0 when TOM0.TOM[0] = 0 (master channel output mode).

Table 17.29 Example of TOM0 settings for operation as a frequency divider

Bit	Symbol	Set value	Function
0	TOM[0]	0	Control of timer output mode of channel 0 0: Sets master channel output mode.

Table 17.30 Procedure for operations when the frequency divider function is to be used (1 of 2)

	Step	Software operation		Hardware state
TAU default setting		—		Power-off state (Clock supply is stopped and writing to each register is disabled.)
	<1>	Sets the MSTPD0 bit of Module Stop Control Register D (MSTPCRD) to 0.	→	Power-on state. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	<2>	Sets timer clock select register 0 (TPS0). Determines clock frequencies of CK00 to CK03.		—

Table 17.30 Procedure for operations when the frequency divider function is to be used (2 of 2)

	Step	Software operation		Hardware state
Channel default setting	<3>	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.)
	<4>	Sets the TO0.TO[0] bit and determines default level of the TO00 output. Sets the TOE0.TOE[0] bit to 1 and enables operation of TO00. Clears the Pmn Output Data Register and Pmn Direction Register to 0.	→ → →	The TO00 pin goes into Hi-Z output state. The TO00 default setting level is output when the Ppq Direction Register (PDRp) is in output mode. TO00 does not change because channel stops operating. The TO00 pin outputs the TO00 set level.
Operation start	<5>	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_TMI00 is generated and TO00 performs toggle operation if the MD[0] bit of the TMR00 register is 1.
During operation	<6>	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_TMI00 is generated and TO00 performs toggle operation. After that, the above operation is repeated.
Operation stop	<7>	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.
	<8>	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 <5>. To terminate the operation, go to step <9>	→	The TO00 pin outputs the TO00 set level.
TAU stop	<9>	To hold the TO0n pin output level Set PSEL[2:0] bits to 000b after the value to be held is set to the Ppq Output Data Register (PODRp). When holding the TO0n pin output level is not necessary Setting not required.	→	The TO0n pin output level is held by port function
	<10>	Sets the MSTPD0 bit of Module Stop Control Register D (MSTPCRD) to 1.	→	This stops supply of the input clock to timer array unit 0. Power-off state.

Note: p = 0 to 9, q = 00 to 15

17.7.4 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 equivalent to one operation clock 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_TMI0n 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 OVF bit is cleared. After that, the above operation is repeated.

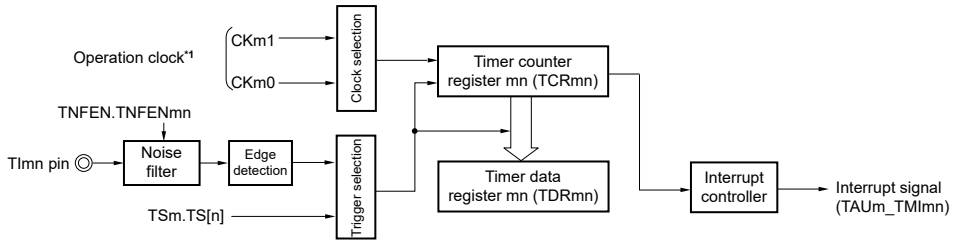
As soon as the count value has been captured to the TDR0n register, the OVF bit of the TSR0n register is updated depending on whether the counter overflows during the measurement period. Therefore, the overflow state of the captured value can be checked.

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.

Instead of using the TI0n pin input, an input pulse interval can also be measured by using the timer input selected in the TIS0 or TIS1 register or the software operation (TS0.TS[n] = 1) as a start trigger and a capture trigger.

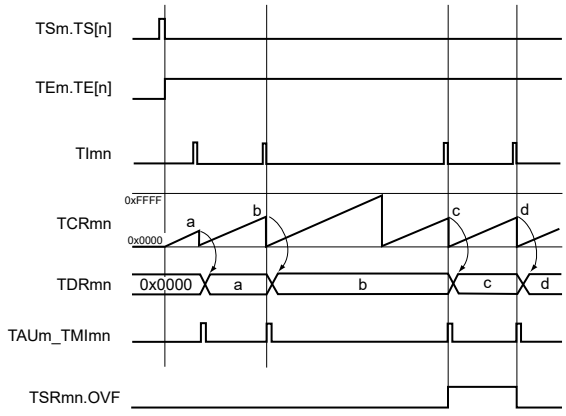
Figure 17.43 shows a block diagram for operation for input pulse interval measurement.



Note: m = 0, n = 0 to 7
Note 1. For channels 1 and 3, the clock can be selected as CKm0, CKm1, CKm2, or CKm3.

Figure 17.43 Block diagram for operation for input pulse interval measurement

Figure 17.44 shows an example of basic timing during operation for input pulse interval measurement (TMR0n.OPIRQ = 0).



Note: m = 0, n = 0 to 7

Figure 17.44 Example of basic timing during operation for input pulse interval measurement (TMR0n.OPIRQ = 0)

Table 17.31 to Table 17.36 show register settings and procedure for operation for input pulse interval measurement.

Table 17.31 Example of TMR0n settings for operation for input pulse interval measurement

Bit	Symbol	Set value	Function
0	OPIRQ	1/0	Setting of operation when counting is started 0: Neither generates TAU0_TMI0n nor inverts timer output when counting is started. 1: Generates TAU0_TMI0n 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.
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 17.32 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 17.33 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 17.34 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] (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] = 0 (master channel output mode).

Table 17.35 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] (n = 1 to 7)		Control of timer output mode of channel n (channels 1 to 7) 0: Sets master channel output mode.

Table 17.36 Procedure for operations when the input pulse interval measurement function is to be used

	Step	Software operation		Hardware state
TAU default setting		—		Power-off state (Clock supply is stopped and writing to each register is disabled.)
	<1>	Sets the MSTPD0 bit of Module Stop Control Register D (MSTPCRD) to 0.	→	Power-on state. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	<2>	Sets timer clock select register 0 (TPS0). Determines clock frequencies of CK00 to CK03.		—
Channel default setting	<3>	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	<4>	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_TMI0n is generated.
During operation	<5>	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], TOL0.TOL[n], 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_TMI0n 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 OVF bit is cleared. After that, the above operation is repeated.
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 <4>. To terminate the operation, go to step <7>.	→	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.
TAU stop	<7>	Sets the MSTPD0 bit of Module Stop Control Register D (MSTPCRD) to 1.	→	This stops supply of the input clock to timer array unit 0. Power-off state.

Note: n = 0 to 7

17.7.5 Operation for Input Signal High- or Low-Level Width Measurement

Note: When using a channel to implement the LIN-bus, set bit 1 (ISC1) of the input switch control register (ISC) to 1. In the following descriptions, read TImn 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_TMI0n 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 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 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.

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.

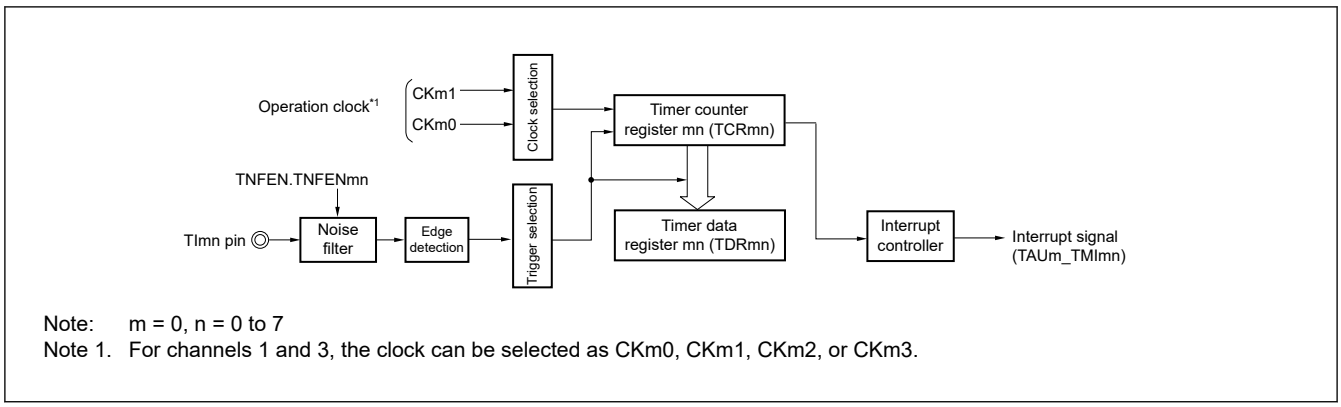
Because this function is used to measure the signal width of the TI0n pin input, the TS0.TS[n] bit cannot be set to 1 while the TE0.TE[n] bit is 1.

Instead of the TI0n pin input, the timer input selected in the TIS0 register can also be used as a start edge and a capture edge.

CIS[1:0] of TMR0n register = 10b: Low-level width is measured.

CIS[1:0] of TMR0n register = 11b: High-level width is measured.

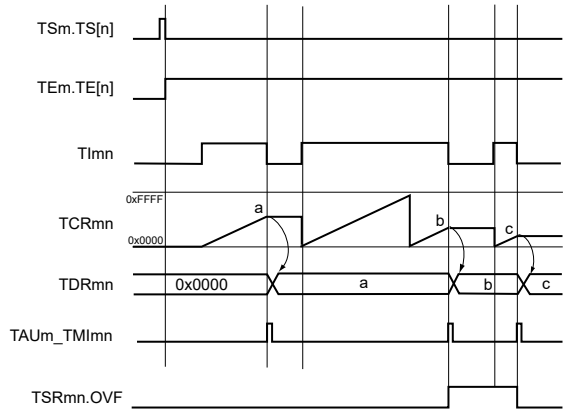
Figure 17.45 shows a block diagram for operation for Input Signal high- or low-level width measurement.



Note: m = 0, n = 0 to 7
Note 1. For channels 1 and 3, the clock can be selected as CKm0, CKm1, CKm2, or CKm3.

Figure 17.45 Block diagram for operation for input signal high- or low-level width measurement

Figure 17.46 shows an example of basic timing during operation for input signal high- or low-level width measurement.



Note: m = 0, n = 0 to 7

Figure 17.46 Example of basic timing during operation for input signal high- or low-level width measurement

Table 17.37 to Table 17.42 show register settings and procedure for operation for input signal high- or low-level width measurement.

Table 17.37 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_TMI0n 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 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 edges.
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 17.37 Example of TMR0n settings for operation for input signal high- or low-level width 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 17.38 Example of TO0 settings for operation for 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 17.39 Example of TOE0 settings for operation for 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 17.40 Example of TOL0 settings for operation for input signal high- or low-level width measurement

Bit	Symbol	Set value	Function
n	— (n = 0)	0	Fixed to 0 (channels 0)
	TOL[n] (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] = 0 (master channel output mode).

Table 17.41 Example of TOM0 settings for operation for input signal high- or low-level width measurement

Bit	Symbol	Set value	Function
n	— (n = 0)	0	Fixed to 0 (channels 0)
	TOM[n] (n = 1 to 7)		Control of timer output mode of channel n (channels 1 to 7) 0: Sets master channel output mode.

Table 17.42 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		—	→	Power-off state (Clock supply is stopped and writing to each register is disabled.)
	<1>	Sets the MSTPD0 bit of Module Stop Control Register D (MSTPCRD) to 0.		Power-on state. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	<2>	Sets timer clock select register 0 (TPS0). Determines clock frequencies of CK00 to CK03.		—
Channel default setting	<3>	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.)

Table 17.42 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
Operation start	<4>	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.
	<5>	Detects the TI0n pin input count start valid edge.	→	Clears timer counter register 0n (TCR0n) to 0x0000 and starts counting up.
During operation	<6>	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], TOL0.TOL[n], 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_TMI0n 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 OVF bit is cleared. The TCR0n register stops the count operation until the next TI0n pin start edge is detected.
Operation stop	<7>	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 <4>. To terminate the operation, go to step <8>	→	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.
TAU stop	<8>	Sets the MSTPD0 bit of Module Stop Control Register D (MSTPCRD) to 1.	→	This stops supply of the input clock to timer array unit 0. Power-off state.

Note: n = 0 to 7

17.7.6 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_TMI0n (a timer interrupt) after any specified interval.

It is also possible to start counting down and generate TAU0_TMI0n (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_TMI0n (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.

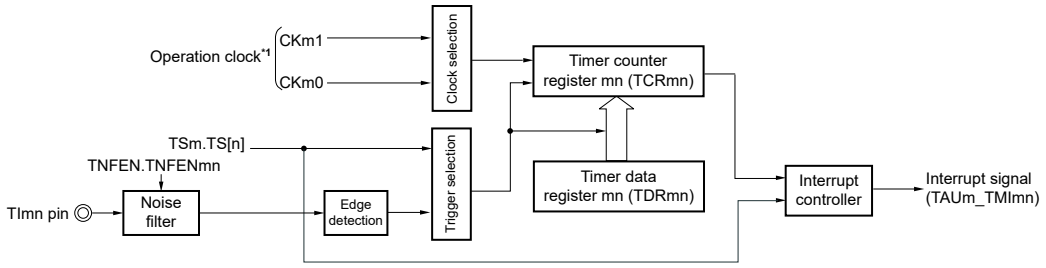
When the channel start trigger bit (TS[n], TSH1, TSH3) of timer channel start register 0 (TS0) is set to 1, the TE0.TE[n], TEH1, and TEH3 bits are set to 1 and the TI0n pin input valid edge detection wait state is set.

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_TMI0n 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.

Instead of using the TI0n pin input, a channel specified for the delay counter function can also use the timer input selected in the TIS0 or TIS1 register or the software operation (TS0.TS[n] = 1) as a start trigger for the function.

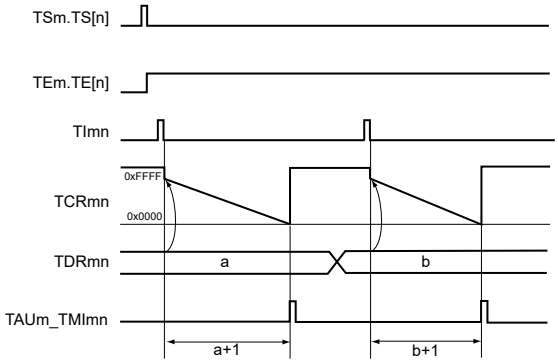
Figure 17.47 shows a block diagram for operation as a delay counter.



Note: m = 0, n = 0 to 7
Note 1. For channels 1 and 3, the clock can be selected as CKm0, CKm1, CKm2, or CKm3.

Figure 17.47 Block diagram for operation as a delay counter

Figure 17.48 shows an example of basic timing during operation as a delay counter.



Note: m = 0, n = 0 to 7

Figure 17.48 Example of basic timing during operation as a delay counter

Table 17.43 to Table 17.48 show register settings and procedure for operation as a delay counter.

Table 17.43 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 17.43 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 17.44 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 17.45 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 17.46 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] (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] = 0 (master channel output mode)

Table 17.47 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] (n = 1 to 7)		Control of timer output mode of channel n (channels 1 to 7) 0: Sets master channel output mode.

Table 17.48 Procedure for operations when the delay counter function is to be used

	Step	Software operation		Hardware state
TAU default setting		—		Power-off state (Clock supply is stopped and writing to each register is disabled.)
	<1>	Sets the MSTPD0 bit of Module Stop Control Register D (MSTPCRD) to 0.	→	Power-on state. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	<2>	Sets timer clock select register 0 (TPS0). Determines clock frequencies of CK00 to CK03.		—
Channel default setting	<3>	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_TMI0n 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	<4>	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.
	<5>	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	<6>	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_TMI0n 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	<7>	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 <4>. To terminate the operation, go to step <8>	→	TE0.TE[n] = 0, and count operation stops. The TCR0n register holds count value and stops.
TAU stop	<8>	Sets the MSTPD0 bit of Module Stop Control Register D (MSTPCRD) to 1.	→	This stops supply of the input clock to timer array unit 0. Power-off state.

Note: n = 0 to 7

17.8 Simultaneous Channel Operation Function of Timer Array Unit

17.8.1 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_TMI0n 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_TMI0n 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_TMI0p and stops counting until the next start trigger (TAU0_TMI0n of the

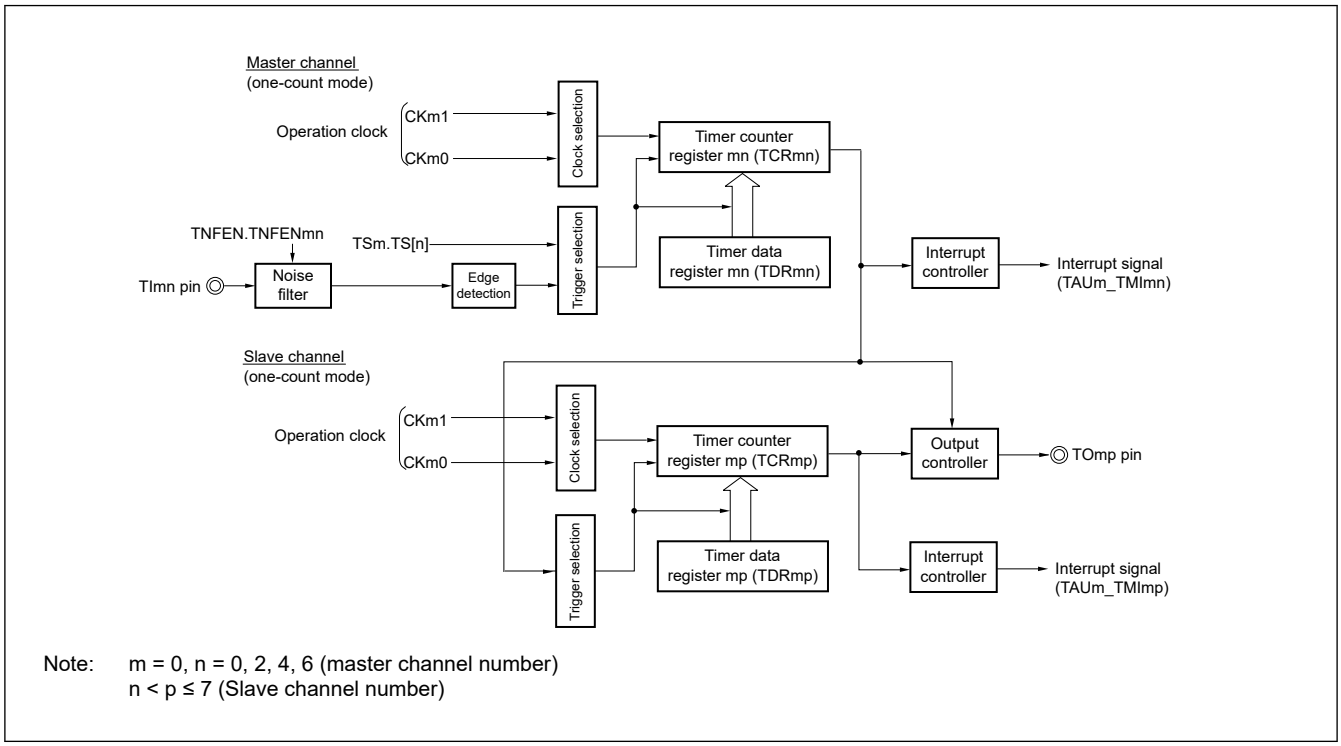
master channel) is detected. The output level of TO0p becomes active one count clock after generation of TAU0_TMI0n from the master channel, and inactive when TCR0p = 0x0000.

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_TMI0n is generated and the TDR0p register after TAU0_TMI0p is generated.

Note: n = 0, 2, 4, 6 (master channel number)
n < p ≤ 7 (Slave channel number)

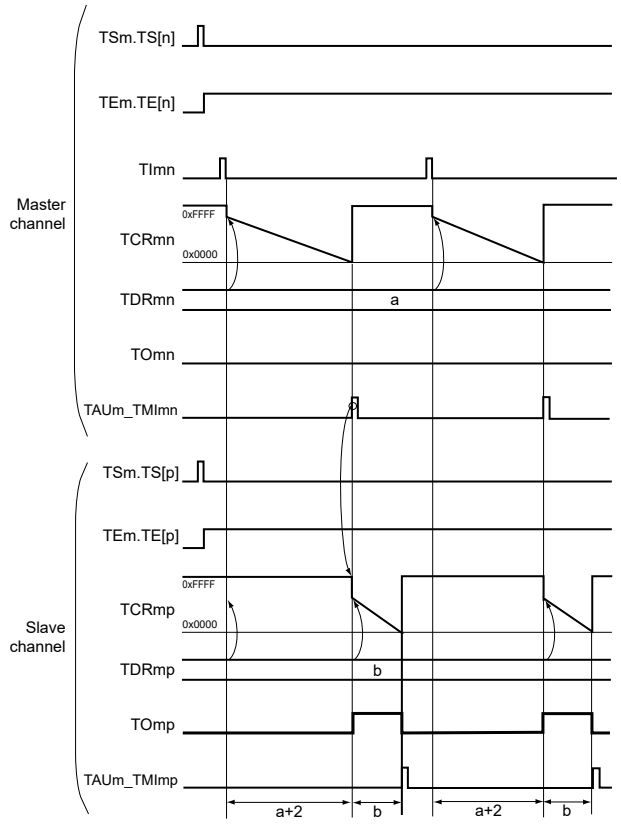
Figure 17.49 shows a block diagram for operation for the one-shot pulse output function.



Note: m = 0, n = 0, 2, 4, 6 (master channel number)
n < p ≤ 7 (Slave channel number)

Figure 17.49 Block diagram for operation for the one-shot pulse output function

Figure 17.50 shows an example of basic timing during operation for the one-shot pulse output function.



Note: m = 0, n = 0, 2, 4, 6 (master channel number)
n < p ≤ 7 (Slave channel number)

Figure 17.50 Example of basic timing during operation for the one-shot pulse output function

Table 17.49 to Table 17.53 show register settings for the master channel when the one-shot pulse output function is to be used.

Table 17.49 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 17.49 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 17.50 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 17.51 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 17.52 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] (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] = 0 (master channel output mode)

Table 17.53 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] (n = 2, 4, 6)		Control of timer output mode of channel n (channels 2, 4, 6) 0: Sets master channel output mode

Table 17.54 to Table 17.58 show register settings for the slave channel when the one-shot pulse output function is to be used.

Table 17.54 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 17.54 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_TMI0n 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 . 1 0: Selects CK01 as the operating clock for channel p

Table 17.55 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 17.56 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 17.57 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]	1/0	Control of timer output of channel p 0: Positive logic output (active-high) 1: Negative logic output (active-low)

Table 17.58 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	Control of timer output mode of channel p (channels 1 to 7) 1: Sets the slave channel output mode.

Table 17.59 show procedure for operations when the one-shot pulse output function is to be used.

Table 17.59 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		—		Power-off state (Clock supply is stopped and writing to each register is disabled.)
	<1>	Sets the MSTPD0 bit of Module Stop Control Register D (MSTPCRD) to 0.	→	Power-on state. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	<2>	Sets timer clock select register 0 (TPS0). Determines clock frequencies of CK00 and CK01.		—
Channel default setting	<3>	Sets the corresponding bit of the TAU noise filter enable register (TNFEN) to 1. Sets timer mode register 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.)
	<4>	Sets slave channel. The TOM0.TOM[p] bit of timer output mode register 0 (TOM0) is set to 1 (slave channel output mode). Sets the TOL0.TOL[p] 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 Prs Direction Register (PDRr) is in 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 Prs Direction Register (PDRr) to 1.	→	The TO0p pin outputs the TO0p set level.
Operation start	<5>	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] and TS[p] bits automatically return to 0 because they are trigger bits.	→	The TE0.TE[n] and 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.
	<6>	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	<7>	Set values of only the CIS[1:0] bits of the TMR0n register can be changed. Set values of the TMR0p, TDR0n, TDR0p registers, TOM0.TOM[n], TOM[p], TOL0.TOL[n], and TOL[p] 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_TMI0n output is generated, and the counter stops until the next start trigger detection. The slave channel, triggered by TAU0_TMI0n 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_TMI0n from the master channel. It becomes inactive when TCR0p = 0x0000, and the counting operation is stopped. After that, the above operation is repeated.

Table 17.59 Procedure for operations when the one-shot pulse output function is to be used (2 of 2)

	Step	Software operation		Hardware state
Operation stop	<8>	The TT0.TT[n] (master) and TT[p] (slave) bits are set to 1 at the same time. The TT0.TT[n] and 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.
	<9>	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 <5>. To terminate the operation, go to step <10>	→	The TO0p pin outputs the TO0p set level.
TAU stop	<10>	To hold the TO0p pin output level Set PSEL[2:0] bits to 000b after the value to be held is set to the Prs Output Data Register (PODRr). When holding the TO0p pin output level is not necessary Setting not required.	→	The TO0p pin output level is held by port function
	<11>	Sets the MSTPD0 bit of Module Stop Control Register D (MSTPCRD) to 1.	→	This stops supply of the input clock to timer array unit 0. Power-off state.

Note 1. Do not set the TS0.TS[n] bit of the slave channel to 1.

Note: n = 0, 2, 4, 6 (Master channel number)
n < p ≤ 7 (Slave channel number)
r = 0 to 9
s = 00 to 15

17.8.2 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: The duty factor exceeds 100% if the set value of TDR0p (slave) > (set value of TDR0n (master) + 1), it summarizes 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_TMI0n) 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_TMI0n 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_TMI0n 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_TMI0p and waits until the next start trigger (TAU0_TMI0n from the master channel) is generated.

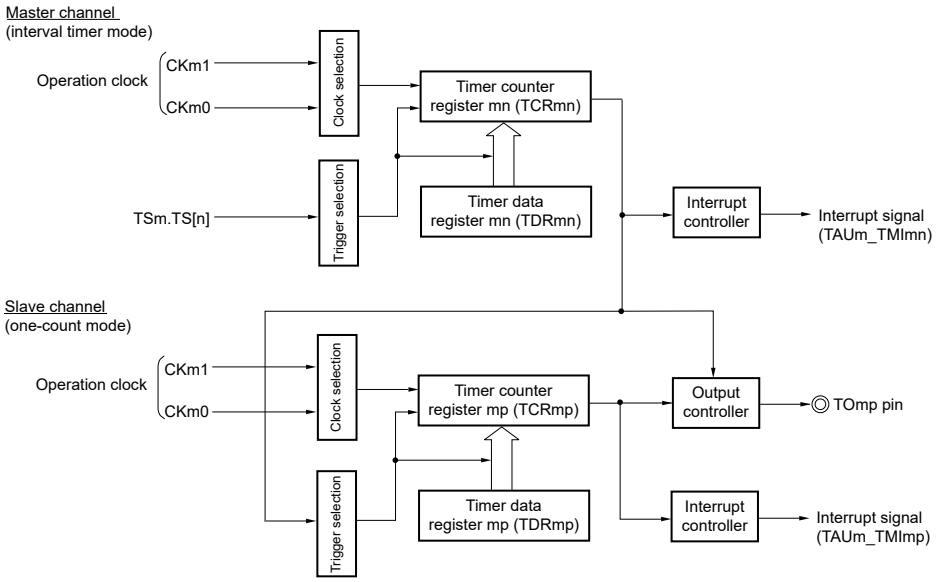
If two channels are used to output a PWM waveform, the period until the slave channel counts down to 0x0000 is the PWM output (TO0p) duty.

PWM output (TO0p) goes to the active level one clock after the master channel generates TAU0_TMI0n 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_TMI0n of the master channel. Thus, when rewriting is performed split before and after occurrence of TAU0_TMI0n 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_TMI0n is generated from the master channel.

Note: n = 0, 2, 4, 6 (master channel number)
n < p ≤ 7 (Slave channel number)

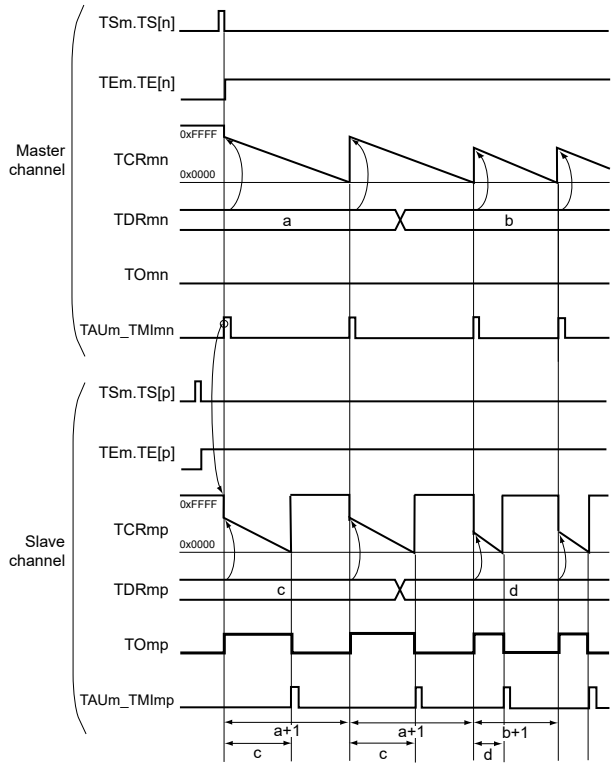
Figure 17.51 shows a block diagram for operation for the PWM function



Note: m = 0
n = 0, 2, 4, 6 (master channel number)
n < p ≤ 7 (Slave channel number)

Figure 17.51 Block diagram for operation for the PWM function

Figure 17.52 shows an example of basic timing during operation for the PWM function.



Note: m = 0
n = 0, 2, 4, 6 (master channel number)
n < p ≤ 7 (Slave channel number)

Figure 17.52 Example of basic timing during operation for the PWM function

Table 17.60 to Table 17.64 show register settings for the master channel when the PWM function is to be used.

Table 17.60 Example of TMR0n settings for the master channel when the PWM 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_TMI0n 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.

Table 17.60 Example of TMR0n settings for the master channel when the PWM function is to be used (2 of 2)

Bit	Symbol	Set value	Function
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 17.61 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 17.62 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 17.63 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] (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] = 0 (master channel output mode).

Table 17.64 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] (n = 2, 4, 6)		Control of timer output mode of channel n (channels 2, 4, 6) 0: Sets master channel output mode.

Table 17.65 to Table 17.69 show register settings for the slave channel when the PWM function is to be used.

Table 17.65 Example of TMR0p settings for the slave channel when the PWM function is to be used (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 p 1 0 0: One-count mode
5:4	—	00b	Fixed to 0
7:6	CIS[1:0]	00b	Selection of TImp 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_TMI0n 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

Table 17.65 Example of TMR0p settings for the slave channel when the PWM function is to be used (2 of 2)

Bit	Symbol	Set value	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 17.66 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 17.67 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 17.68 Example of TOL0 settings for the slave channel when the PWM function is to be used

Bit	Symbol	Set value	Function
p	TOL[p]	1/0	Control of timer output of channel p 0: Positive logic output (active-high) 1: Negative logic output (active-low)

Table 17.69 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	Control of timer output mode of channel p (channels 1 to 7) 1: Sets the slave channel output mode.

Table 17.70 show procedure for operations when the PWM function is to be used.

Table 17.70 Procedure for operations when the PWM function is to be used (1 of 2)

	Step	Software operation		Hardware state
TAU default setting		—		Power-off state (Clock supply is stopped and writing to each register is disabled.)
	<1>	Sets the MSTPD0 bit of Module Stop Control Register D (MSTPCRD) to 0.	→	Power-on state. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	<2>	Sets timer clock select register 0 (TPS0). Determines clock frequencies of CK00 and CK01.		—

Table 17.70 Procedure for operations when the PWM function is to be used (2 of 2)

	Step	Software operation		Hardware state
Channel default setting	<3>	Sets timer mode register 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.)
	<4>	Sets slave channel. The TOM0.TOM[p] bit of timer output mode register 0 (TOM0) is set to 1 (slave channel output mode). Sets the TOL0.TOL[p] 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 Prs Direction Register (PDRr) is in 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 Prs Direction Register (PDRr) to 1.	→	The TO0p pin outputs the TO0p set level.
Operation start	<5>	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] and 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_TMI0n is generated. Triggered by this interrupt, the slave channel also starts counting.
During operation	<6>	Set values of the TMR0n and TMR0p registers, TOM0.TOM[n], TOM[p], TOL0.TOL[n], and TOL[p] bits cannot be changed. Set values of the TDR0n and TDR0p registers can be changed after TAU0_TMI0n 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_TMI0n 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_TMI0n of the master channel, and the counter starts counting down. The output level of TO0p becomes active one count clock after generation of theTAU0_TMI0n 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	<7>	The TT0.TT[n] (master) and TT[p] (slave) bits are set to 1 at the same time. The TT0.TT[n] and 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 TOm.TO[p] bit. To resume operation, go to step <5>. To terminate the operation, go to step <9>	→	The TO0p pin outputs the TO0p set level.
TAU stop	<9>	To hold the TO0p pin output level Set PSEL[2:0] bits to 000b after the value to be held is set to the Prs Output Data Register (PODRr). When holding the TO0p pin output level is not necessary Setting not required.	→	The TO0p pin output level is held by port function
	<10>	Sets the MSTPD0 bit of Module Stop Control Register D (MSTPCRD) to 1.	→	This stops supply of the input clock to timer array unit 0. Power-off state.

Note: n = 0, 2, 4, 6 (Master channel number)
n < p ≤ 7 (Slave channel number)
r = 0 to 9
s = 00 to 15

17.8.3 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_TMI0n of the master channel as a start trigger, and starts counting down. When TCR0p = 0x0000, TCR0p outputs TAU0_TMI0p and stops counting until the next start trigger (TAU0_TMI0n of the master channel) has been input. The output level of TO0p becomes active one count clock after generation of TAU0_TMI0n 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_TMI0n of the master channel as a start trigger, and starts counting down.

When TCR0q = 0x0000, the TCR0q register outputs TAU0_TMI0q and stops counting until the next start trigger (TAU0_TMI0n of the master channel) has been input. The output level of TO0q becomes active one count clock after generation of TAU0_TMI0n 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.

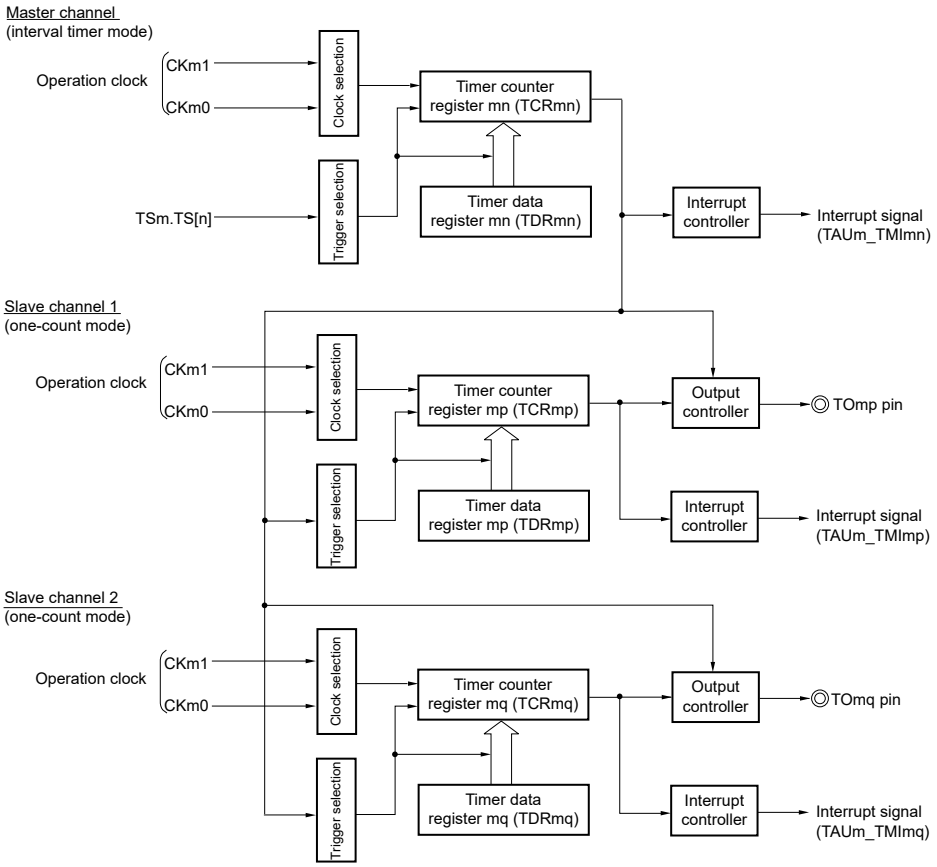
Note: To rewrite both timer data register 0n (TDR0n) of the master channel and the TDR0p register of the slave channel 1, write access is necessary at least twice. Since the values of the TDR0n and TDR0p registers are loaded to the TCR0n and TCR0p registers after TAU0_TMI0n is generated from the master channel, if rewriting is performed separately before and after generation of TAU0_TMI0n 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_TMI0n is generated from the master channel (This applies also to the TDR0q register of the slave channel 2).

Note: n = 0, 2, 4 (Master channel number)

p, q : Slave channel number

n < p < q ≤ 7 (Where p and q are integers greater than n)

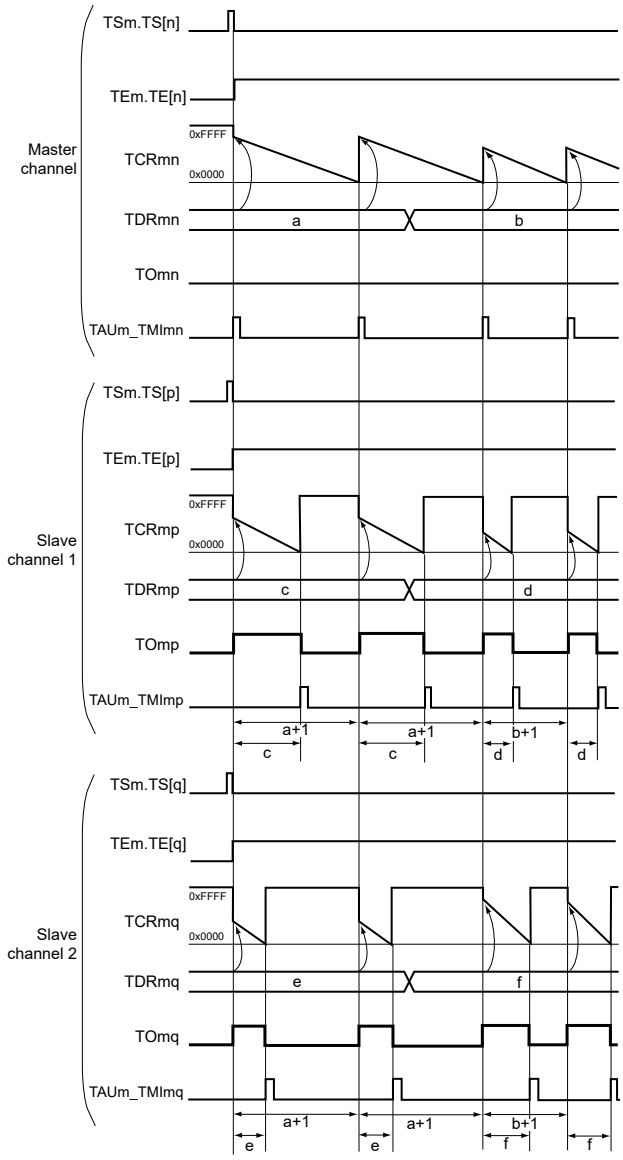
Figure 17.53 shows a block diagram for operation for the multiple PWM output function (for two types of PWM output).



Note: m = 0
n = 0, 2, 4 (Master channel number)
p, q: Slave channel number
n < p < q ≤ 7 (Where p and q are integers greater than n)

Figure 17.53 Block diagram for the multiple PWM output function (for two types of PWM output)

Figure 17.54 shows an example of basic timing during operation for the multiple PWM output function (for two types of PWM output).



Note: m = 0
n = 0, 2, 4 (Master channel number)
p, q: Slave channel number
n < p < q ≤ 7 (Where p and q are integers greater than n)

Note: TSm.TS[n], TSmp, TSmq: Bits n, p, and q of timer channel start register m (TSm)
TEmn, TEmp, TEmq: Bits n, p, and q of timer channel enable status register m (TEm)
TCRmn, TCRmp, TCRmq: Timer counter registers mn, mp, mq (TCRmn, TCRmp, TCRmq)
TDRmn, TDRmp, TDRmq: Timer data registers mn, mp, mq (TDRmn, TDRmp, TDRmq)
TOmn, TOmp, TOmq: Signals on the TOmn, TOmp, and TOmq output pins

Figure 17.54 Example of basic timing during operation for the multiple PWM output function (for two types of PWM output)

Table 17.71 to Table 17.75 show register settings for the master channel when the multiple PWM output function is to be used.

Table 17.71 Example of TMR0n settings for the master channel when the multiple PWM output function is to be used

Bit	Symbol	Set value	Function
0	OPIRQ	1	Setting of operation when counting is started 1: Generates TAU0_TMI0n 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)	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 17.72 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 17.73 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 17.74 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] (n = 2, 4)		Control of timer output of channel n (channels 2, 4) 0: Set this bit to 0 when TOM0.TOM[n] = 0 (master channel output mode).

Table 17.75 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] (n = 2, 4)		Control of timer output mode of channel n (channels 2, 4) 0: Sets master channel output mode.

Table 17.76 to Table 17.81 show register settings for the slave channel when the multiple PWM output function is to be used (for two types of PWM output).

Table 17.76 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_TMI0n 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 17.77 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 TI0q 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_TMI0n 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})

Table 17.77 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
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 q 1 0: Selects CK01 as the operating clock for channel q

Table 17.78 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 17.79 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 17.80 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]	1/0	Control of timer output of channel p 0: Positive logic output (active-high) 1: Negative logic output (active-low)
q	TOL[q]	1/0	Control of timer output of channel q 0: Positive logic output (active-high) 1: Negative logic output (active-low)

Table 17.81 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	Control of timer output mode of channel p (channels 1 to 6) 1: Sets the slave channel output mode.
q	TOM[q]	1	Control of timer output mode of channel q (channels 2 to 7) 1: Sets the slave channel output mode.

Table 17.82 show procedure for operations when the PWM function is to be used.

Table 17.82 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		—		Power-off state (Clock supply is stopped and writing to each register is disabled.)
	<1>	Sets the MSTPD0 bit of Module Stop Control Register D (MSTPCRD) to 0.	→	Power-on state. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	<2>	Sets timer clock select register 0 (TPS0). Determines clock frequencies of CK00 and CK01.		—
Channel default setting	<3>	Sets timer mode register 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.)
	<4>	Sets slave channels. The TOM0.TOM[p] and TOM[q] 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] and TOL[q] bits and determines default level of the TO0p and TO0q outputs.	→	The TO0p and TO0q default setting levels are output when Prs Direction Register (PDRr) is in output mode.
		Sets the TOE0.TOE[p] and TOE[q] bits to 1 and enables operation of TO0p and TO0q.	→	TO0p and TO0q do not change because channels stop operating.
		Sets the Prs Direction Register (PDRr) to 1.	→	The TO0p and TO0q pins output the TO0p and TO0q set levels.
Operation start	<5>	(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], and TS[q] bits automatically return to 0 because they are trigger bits.	→	TE0.TE[n] = 1, TE0.TE[p], TE[q] = 1 When the master channel starts counting, TAU0_TMI0n is generated. Triggered by this interrupt, the slave channel also starts counting.
During operation	<6>	Set values of the TMR0n, TMR0p, TMR0q registers, TOM0.TOM[n], TOM[p], TOM[q], OL0.TOL[n], TOL [p], and TOL[q] bits cannot be changed. Set values of the TDR0n, TDR0p and TDR0q registers can be changed after TAU0_TMI0n 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_TMI0n 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_TMI0n 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_TMI0n 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_TMI0n 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_TMI0n output from the master channel. It becomes inactive when TCR0q = 0x0000, and the counting operation is stopped. After that, the above operation is repeated.

Table 17.82 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
Operation stop	<7>	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] and 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.
	<8>	The TOE0.TOE[p], and TOE[q] bits of slave channels are cleared to 0 and value is set to the TO0.TO[p] and TO[q] bits. To resume operation, go to step <5>. To terminate the operation, go to step <9>	→	The TO0p and TO0q pins output the TO0p and TO0q set levels.
TAU stop	<9>	To hold the TO0p pin and TO0q pin output levels Set PSEL[2:0] bits to 000b after the value to be held is set to the Prs Output Data Register (PODRr). When holding the TO0p pin output level is not necessary Setting not required.	→	The TO0p pin and TO0q pin output levels are held by port function
	<10>	Sets the MSTPD0 bit of Module Stop Control Register D (MSTPCRD) to 1.	→	This stops supply of the input clock to timer array unit 0. Power-off state.

Note: n = 0, 2, 4 (Master channel number)
p, q: Slave channel number
n < p < q ≤ 7 (Where p and q are integers greater than n)
r = 0 to 9
s = 00 to 15

17.9 Usage Notes

17.9.1 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.

17.9.2 Point for Caution when a Timer Output is to be Used as an Event Input for the ELC

The timer outputs (TO00 to TO03) of channels 0 to 3 of timer array unit 0 can be used as event inputs for the event link controller (ELC).

18. 32-bit Interval Timer (TML32)

18.1 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 HOCO, MOCO, MOSC, FSXP (LOCO or SOSC), or the event input from the ELC, which is asynchronous with the CPU operation. Table 18.1 lists the 32-bit interval timer functions. Figure 18.1 to Figure 18.4 show images of each timer function, and Figure 18.5 shows a block diagram of the 32-bit interval timer.

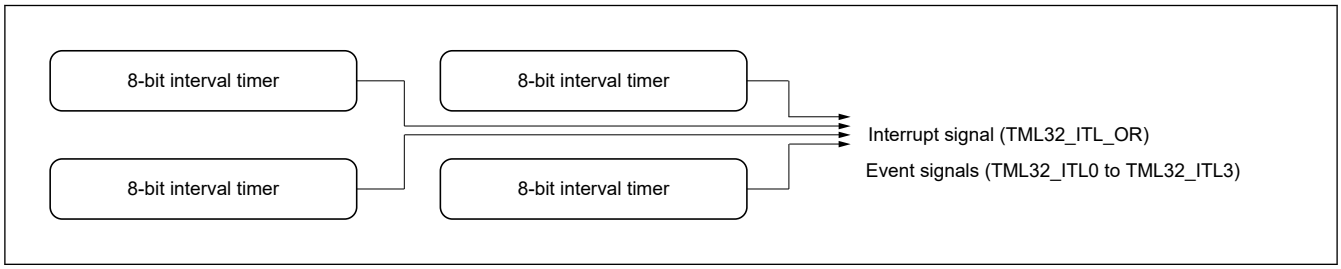


Figure 18.1 Image of four 8-bit interval timer function

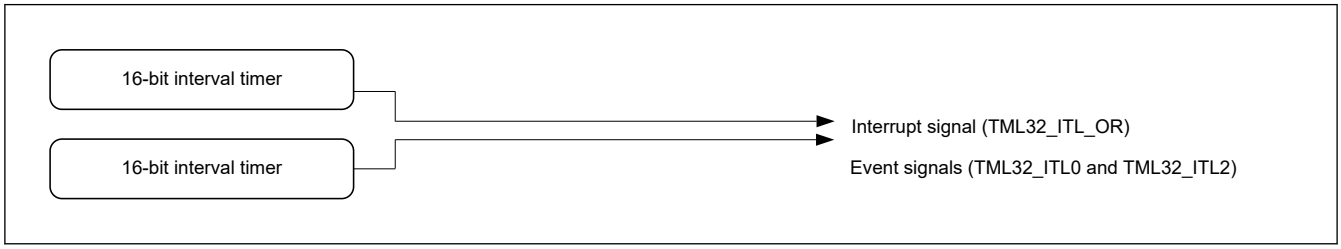


Figure 18.2 Image of two 16-bit Interval timer function

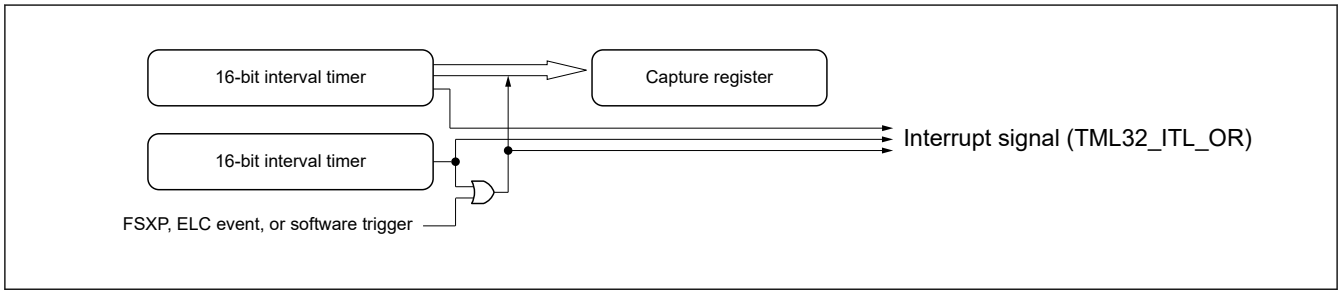


Figure 18.3 Image of the 16-bit interval timer and 16-bit capture function

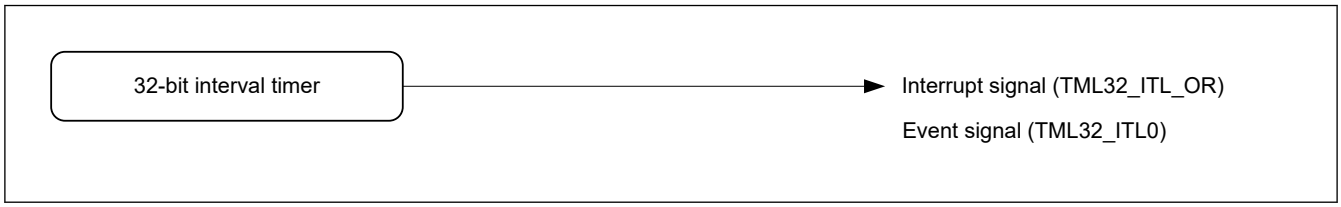


Figure 18.4 Image of a 32-bit interval timer function

Table 18.1 Specifications of 32-bit interval timer operations

Item	Description
Count source (operating clock)	• HOCO • MOCO • MOSC • FSXP (LOCO or SOSC)*1 • Event input from the ELC
Capture clock (Selectable sources for counting by the timer which can generate a capture trigger)	• HOCO • MOCO • MOSC • FSXP (LOCO or SOSC)*1 • 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_ITL_OR 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.
Event link function	• Four trigger signals TML32_ITL0 to TML32_ITL3 for the ELC are output. – Output when the counter value in any of channels 0 to 3 matches the compare value.

Note 1. Select either LOCO or SOSC as the FSXP by setting the OSMC.WUTMMCK0 bit.

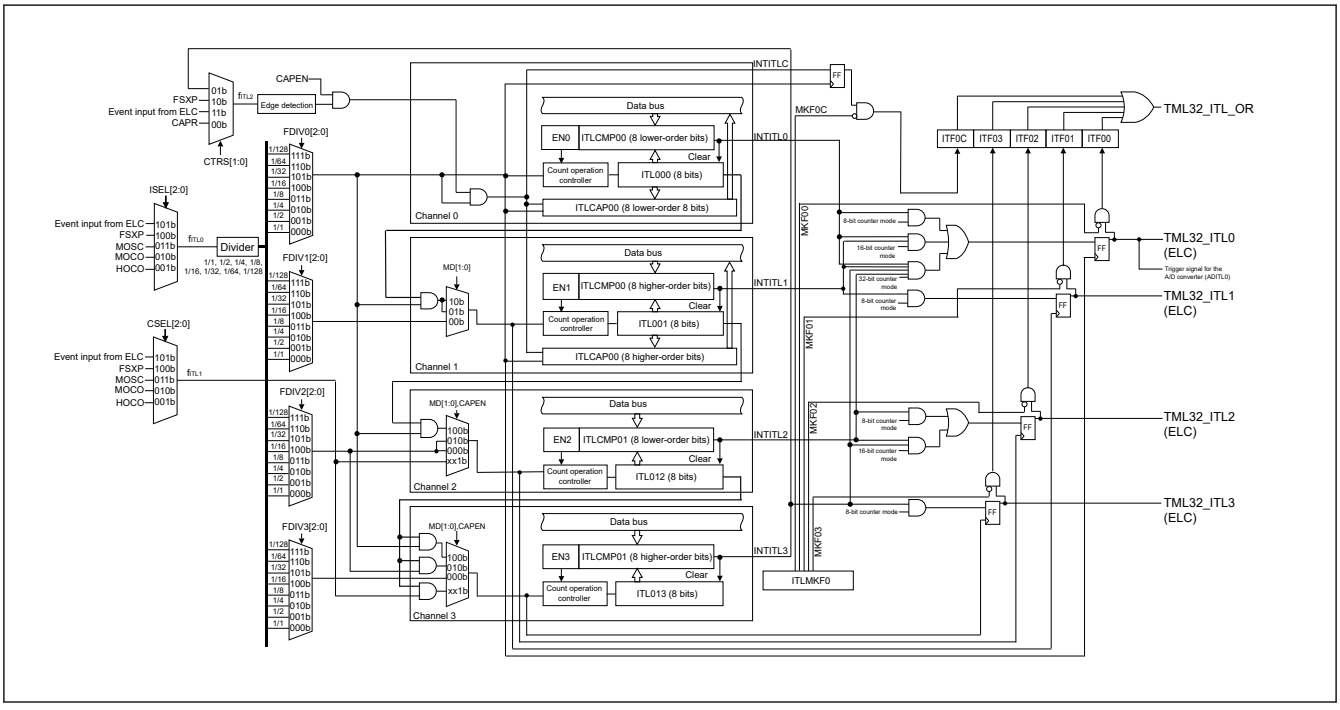


Figure 18.5 Block diagram of 32-bit interval timer

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).

18.2 Register Descriptions

18.2.1 ITLCMP0n/ITLCMP0n_L/ITLCMP0n_H : Interval Timer Compare Registers 0n (n = 0, 1)

Base address: TML32 = 0x400A_3800

Offset address: 0x0000 + 0x2 × n (ITLCMP0n/ITLCMP0n_L)
0x0001 + 0x2 × n (ITLCMP0n_H)

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	Comparison Data of 16-bit Counter (ITL0n) and 8-bit Counter (ITL0n_H/ITL0n_L) are Stored The ITLCMP00 is compared with ITL00 (ITL000 + ITL001). The ITLCMP01 is compared with ITL01 (ITL012 + ITL013).	R/W

Note: Write to the ITLCMP0n_H and ITLCMP0n_L registers while the settings of the EN0 to EN3 bits in the ITLCTL0 register are 0, respectively.

Note: Write to the ITLCMP00 register while the IEN0 bit in the ITLCTL0 register is 0. Write to the ITLCMP01 register while the EN2 bit in the ITLCTL0 register is 0 in 16-bit counter mode or while the EN0 bit in the ITLCTL0 register is 0 in 32-bit counter mode.

Interval Timer Compare Registers (ITLCMP0n/ITLCMP0n_L/ITLCMP0n_H) is compare value registers used in 8-bit, 16-bit, or 32-bit counter mode. ITLCMP0n_L (ITLCMP0n[7:0]) and ITLCMP0n_H (ITLCMP0n[15:8]) used in 8-bit counter mode.

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 ITL0n counter values.

When the ITLCTL0.MD[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.

18.2.2 ITLCAP00 : Interval Timer Capture Register 00

Base address: TML32 = 0x400A_3800

Offset address: 0x0004

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	n/a															
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	Capture Result of 16-bit Counter (ITL00) is Stored.	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 CAPEN 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.

18.2.3 ITLCTL0 : Interval Timer Control Register

Base address: TML32 = 0x400A_3800

Offset address: 0x0006

Bit position:	7	6	5	4	3	2	1	0
Bit field:	MD[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
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	MD[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 (channels 0 to 1 are used together, and channels 2 to 3 are used together). 1 0: The interval timer operates in 32-bit counter mode (channels 0 to 3 are used together). 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 MD[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.

MD[1:0] bits (Selection of 8-bit, 16-bit, or 32-bit Counter Mode)

Table 18.2 lists the target counters that can be enabled in the MD[1:0] bits and EN0 to EN3 bit settings.

Table 18.2 Target counter setting

Mode	MD[1:0]	EN3	EN2	EN1	EN0	Target counter
8-bit mode	00b	—	—	—	✓	ITL000
		—	—	✓	—	ITL001
		—	✓	—	—	ITL012
		✓	—	—	—	ITL013
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.

18.2.4 ITLCSEL0 : Interval Timer Clock Select Register 0

Base address: TML32 = 0x400A_3800

Offset address: 0x0007

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: HOCO 0 1 0: MOCO 0 1 1: MOSC 1 0 0: FSXP 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: HOCO 0 1 0: MOCO 0 1 1: MOSC 1 0 0: FSXP 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 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.

18.2.5 ITLFDIV00 : Interval Timer Frequency Division Register 0

Base address: TML32 = 0x400A_3800

Offset address: 0x0008

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.

18.2.6 ITLFDIV01 : Interval Timer Frequency Division Register 1

Base address: TML32 = 0x400A_3800

Offset address: 0x0009

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.

18.2.7 ITLCC0 : Interval Timer Capture Control Register 0

Base address: TML32 = 0x400A_3800

Offset address: 0x000A

Bit position:	7	6	5	4	3	2	1	0
Bit field:	CAPEN	CAPFCR	CAPF	CAPR	CAPCCR	—	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 1 0: FSXP (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
3	CAPCCR	Selection of Capture Counter Clearing After Capturing*3 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	CAPR	Software Capture Trigger*2 *4 0: Trigger operation does not proceed. 1: A software trigger for capturing is generated.	R/W
5	CAPF	Capture Completion Flag 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 CAPFCR bit clears this flag to 0. 0: Capturing has not been completed. 1: Capturing has been completed.	R
6	CAPFCR	Capture Completion Flag Clear*5 0: The value of the capture completion flag CAPF is held. 1: The value of the capture completion flag CAPF is cleared.	R/W
7	CAPEN	Capture Enable*6 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 0.
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. Be sure to write to the CAPCCR bit only while the ITLCTL0.EN3 to ITLCTL0.EN0 bits are all 0.
Note 4. The CAPR bit is always read as 0.
Note 5. The CAPFCR bit is always read as 0.
Note 6. Be sure to write to the CAPEN bit only while the ITLCTL0.EN3 to ITLCTL0.EN0 bits are all 0.

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.

18.2.8 ITLS0 : Interval Timer Status Register

Base address: TML32 = 0x400A_3800

Offset address: 0x000B

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
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_ITL_OR) 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 CAPEN 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_ITL_OR interrupt signal. Table 18.3 shows the conditions for setting the status flags in each timer mode selected by the ITLCTL0.MD[1:0] bits.

Table 18.3 Conditions for setting the status flags in each timer mode

Mode	ITLCTL0.MD[1:0]	ITLCC0.CAPEN	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_L and ITL000 values
		x	ITF01	The next rising edge of the counter clock following a match between the ITLCMP00_H and ITL001 values
		x	ITF02	The next rising edge of the counter clock following a match between the ITLCMP01_L and ITL012 values
		x	ITF03	The next rising edge of the counter clock following a match between the ITLCMP01_H 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

18.2.9 ITLMKF0 : Interval Timer Match Detection Mask Register

Base address: TML32 = 0x400A_3800

Offset address: 0x000C

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. Setting all functional bits to 1 for masking prevents setting of the corresponding bits in the ITLS0 register. 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 CAPF 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_ITL_OR).

18.3 Operation

18.3.1 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 18.4 to Table 18.6 show the registers and settings for use in 8-bit counter mode, 16-bit counter mode, and 32-bit counter mode, respectively.

Table 18.4 Registers and settings used in 8-bit counter mode (1 of 2)

Register name (symbol)	Bit	Setting
Interval timer compare registers 000 (ITLCMP00)	Bits 7 to 0	Specify 8-bit compare values for channels 0.
Interval timer compare registers 001 (ITLCMP00)	Bits 7 to 0	Specify 8-bit compare values for channels 1.
Interval timer compare registers 012 (ITLCMP01)	Bits 7 to 0	Specify 8-bit compare values for channels 2.
Interval timer compare registers 013 (ITLCMP01)	Bits 7 to 0	Specify 8-bit compare values for channels 3.

Table 18.4 Registers and settings used in 8-bit counter mode (2 of 2)

Register name (symbol)	Bit	Setting
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.
	MD[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.
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 18.5 Registers and settings used in 16-bit counter mode

Register name (symbol)	Bit	Setting
Interval timer compare registers 00 (ITLCMP00)	Bits 15 to 0	Specify 16-bit compare values for channels 0 and 1
Interval timer compare registers 01 (ITLCMP01)	Bits 15 to 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.
	MD[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 18.6 Registers and settings used in 32-bit counter mode (1 of 2)

Register name (symbol)	Bit	Setting
Interval timer compare registers 00 (ITLCMP00)	Bits 15 to 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)	Bits 15 to 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).

Table 18.6 Registers and settings used in 32-bit counter mode (2 of 2)

Register name (symbol)	Bit	Setting
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.
	MD[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.

18.3.2 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 18.7 show the registers and settings for use in 16-bit capture mode.

Table 18.7 Registers and settings used in 16-bit capture mode

Register name (symbol)	Bit	Setting
Interval timer compare register 00 (ITLCMP00)	Bits 15 to 0	Specify 16-bit compare values for channels 0 and 1.
Interval timer compare register 01 (ITLCMP01)*1	Bits 15 to 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.
	MD[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)	CAPEN	Set to 1.
	CAPCCR	Specify whether to clear or hold the counter value in channels 0 and 1 after the completion of capturing.
	CTRS[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.

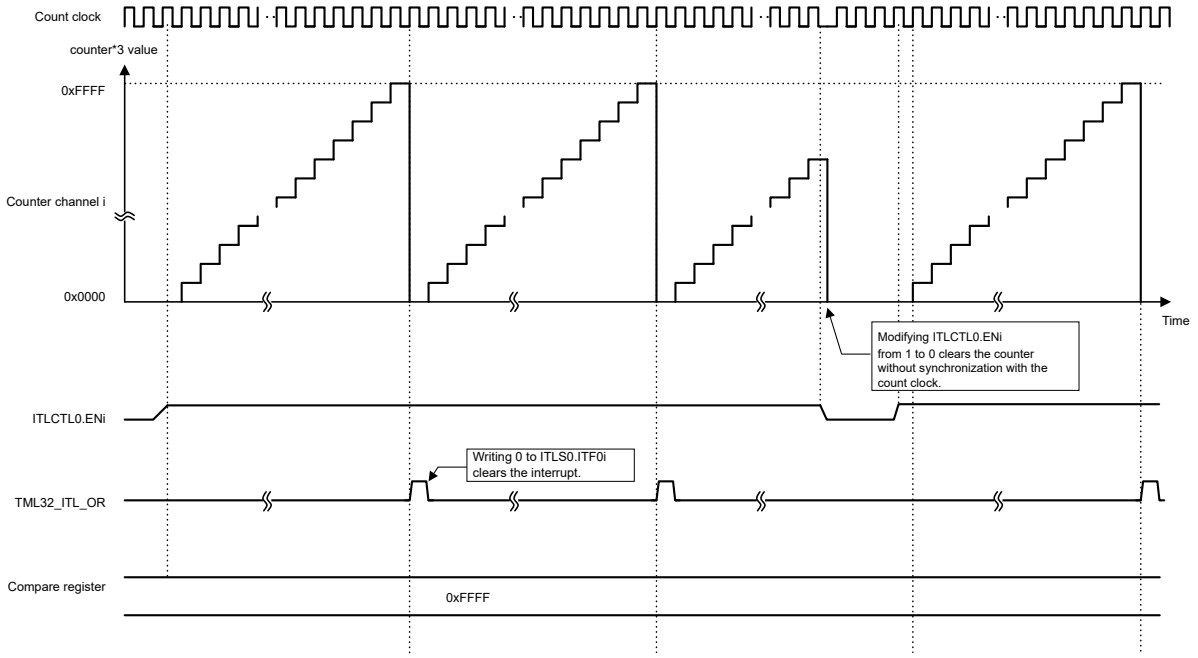
18.3.3 Timer Operation

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_ITL_OR) 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_ITL_OR) remains high until the value of the ITLS0 register becomes 0x00.

While the interrupt request signal (TML32_ITL_OR) is high, the generation of an additional interrupt request (TML32_ITL_OR) 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 18.6 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 18.6 Example of timer operation

18.3.4 Capture Operation

When the setting of the CAPEN bit in the interval timer capture control register 0 (ITLCC0) is 1, the values in the 16-bit counters (ITL000 and 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, FSXP, an event input from the ELC, and a software trigger (setting the ITLCC0.CAPR 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 FSXP, an event input from the ELC, or a software trigger (setting the ITLCC0.CAPR 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_ITL_OR) is output, the capture completion flag (ITLCC0.CAPF) and capture detection flag (ITLS0.ITF0C) are set to 1, and the flag values are retained until they are explicitly cleared *1. The ITLCC0.CAPF flag can be cleared by setting the ITLCC0.CAPFCR 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.CAPF 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 (ITF0C) is set to 1 because the interrupt request signal (TML32_ITL_OR) is kept at the high level.

Figure 18.7 shows an example of capture operation.

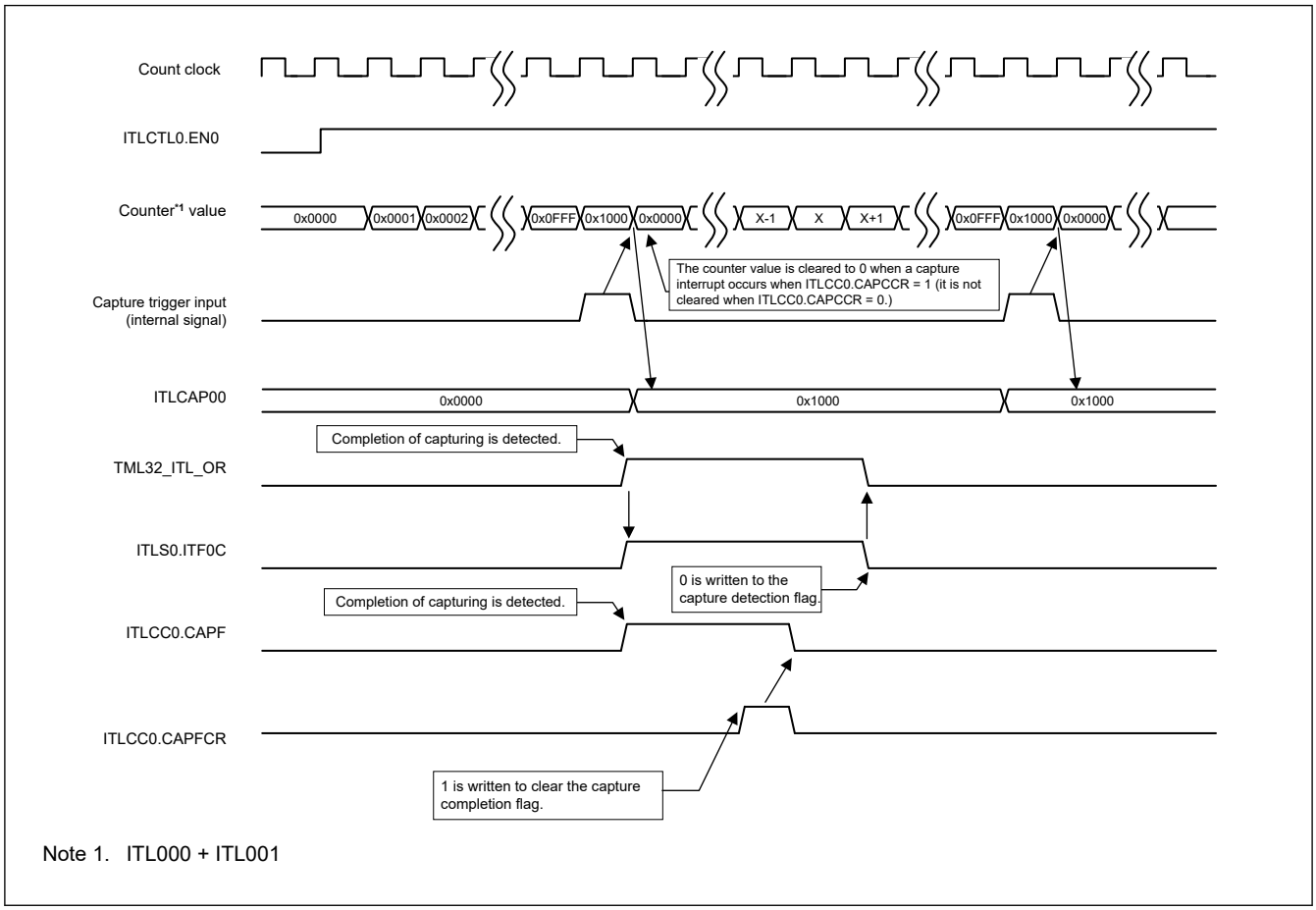


Figure 18.7 Example of capture operation

When the counter value matches the comparison value while the ITLCC0.CAPCCR 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.CAPCCR bit is set to 0 (mode where the capture counter retains its value after the completion of capture).

18.3.5 Interrupt

Table 18.8 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_ITL_OR signal.

Table 18.8 Interrupt sources in 8-bit, 16-bit, and 32-bit counter modes (1 of 2)

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

Table 18.8 Interrupt sources in 8-bit, 16-bit, and 32-bit counter modes (2 of 2)

Interrupt source	Interrupt condition in 8-bit counter mode	Interrupt condition in 16-bit counter mode	Interrupt condition in 32-bit counter mode
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_ITL_OR) is kept at the high level. Accordingly, the generation of an additional interrupt request (TML32_ITL_OR) 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 by an 8-bit memory manipulation instruction, a low-level pulse signal is output on the TML32_ITL_OR pin is set to 1. 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 18.8 shows the relationship between clearing of the detection flags and the interval detection interrupt signal.

The following describes the operation shown in Figure 18.8.

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_ITL_OR) is driven high. While the interval detection interrupt signal (TML32_ITL_OR) is kept at the high level, the generation of an additional interrupt request (TML32_ITL_OR) 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_ITL_OR pin 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_ITL_OR) 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 18.8, the ITLS0.ITF01 and ITF00 flags being set to 1 can be confirmed.

<3> Clear the ITLS0.ITF01 and ITF00 flags detected in step 2 and 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_ITL_OR) 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 18.8, the ITLS0.ITF01 flag being set to 1 can be confirmed.

<7> Clear the ITLS0.ITF01 flag detected in step 6 and 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_ITL_OR signal is temporarily driven low.

<9> The processing in response to the ITLS0.ITF00 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 18.8, the ITLS0.ITF00 flag being set to 1 can be confirmed.

<11> Clear the ITLS0.ITF00 flag detected in step 11 and write 00011101b 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.

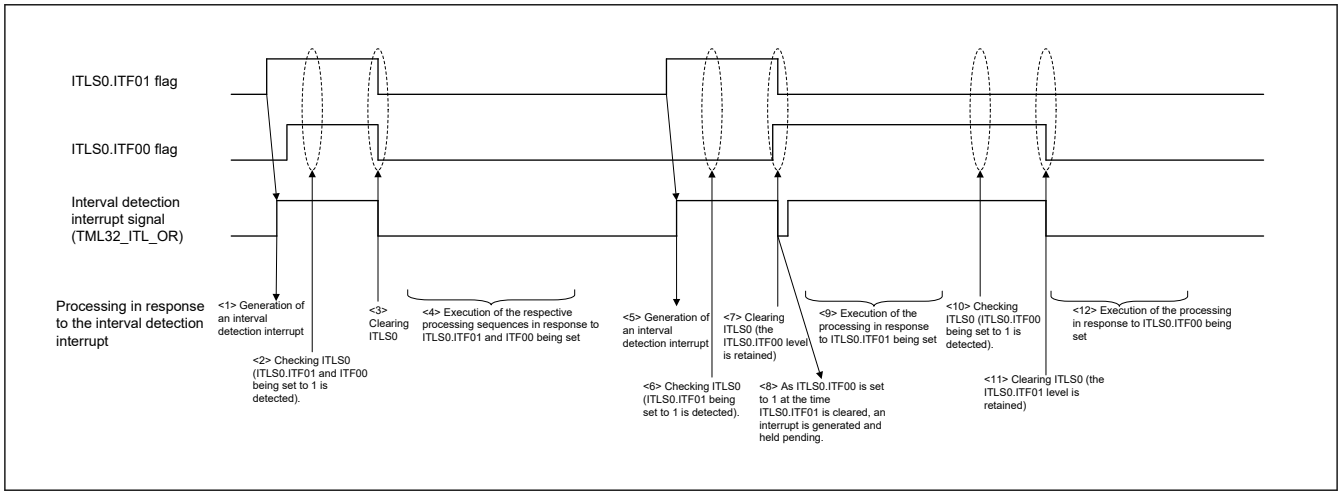


Figure 18.8 Example of clearing the detected flags

18.3.6 Interval Timer Setting Procedures

Table 18.9 shows the procedure for setting up the 32-bit interval timer.

Table 18.9 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.MD[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.CAPEN bit. Clear the ITLCC0.CAPFCR bit. Set up the ITLCSEL0.CSEL[2:0] bits. Clear the ITLCC0.CAPCCR 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.CAPFCR 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.CAPR bit to 1 when using the software capture trigger.	Set the ITLCC0.CAPR 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 18.10 shows the procedure for stopping the 32-bit interval timer.

Table 18.10 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: n = 0, 1, i = 0 to 3

Table 18.11 shows the procedure for changing the operating mode of the 32-bit interval timer.

Table 18.11 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 18.9)	Make the setting to change the operating mode.
	<7>	Completion of changing the operating mode	—

Note: i = 0 to 3

Table 18.12 shows the procedure for starting event input from the ELC.

Table 18.12 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 ELSR28[*1] register. Set the appropriate ELSR28.ELS[5:0] bits for the 32-bit interval timer to be linked.
	<3>	Set up the ELCR 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 18.9. 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 18.13 shows the procedure for stopping event input from the ELC.

Table 18.13 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 18.10.
	<4>	Set up the ELSR28 register[*1] to disable the output. (Optional: set up the ELCR register to disable all event linkage).	Set the ELSR28.ELS[5:0] 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. Realtime Clock (RTC)

This is the RTC_C version of the RTC peripheral module. RTC_C is referred to as RTC in this chapter.

19.1 Overview

The realtime clock has the following features.

Table 19.1 RTC specifications

Item	Description
Count mode	Calendar count mode
Count source	• Sub-clock (SOSC) or LOCO • 128-Hz from sub-clock (SOSC/256)
Calendar functions	Year, month, date, day of week, hour, minute, and second are counted for up to 99 years
Interrupts (RTC_ALM_OR_PRD)	The following two interrupts are the source of the realtime clock interrupt signal (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

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.

Figure 19.1 shows a block diagram of a realtime clock.

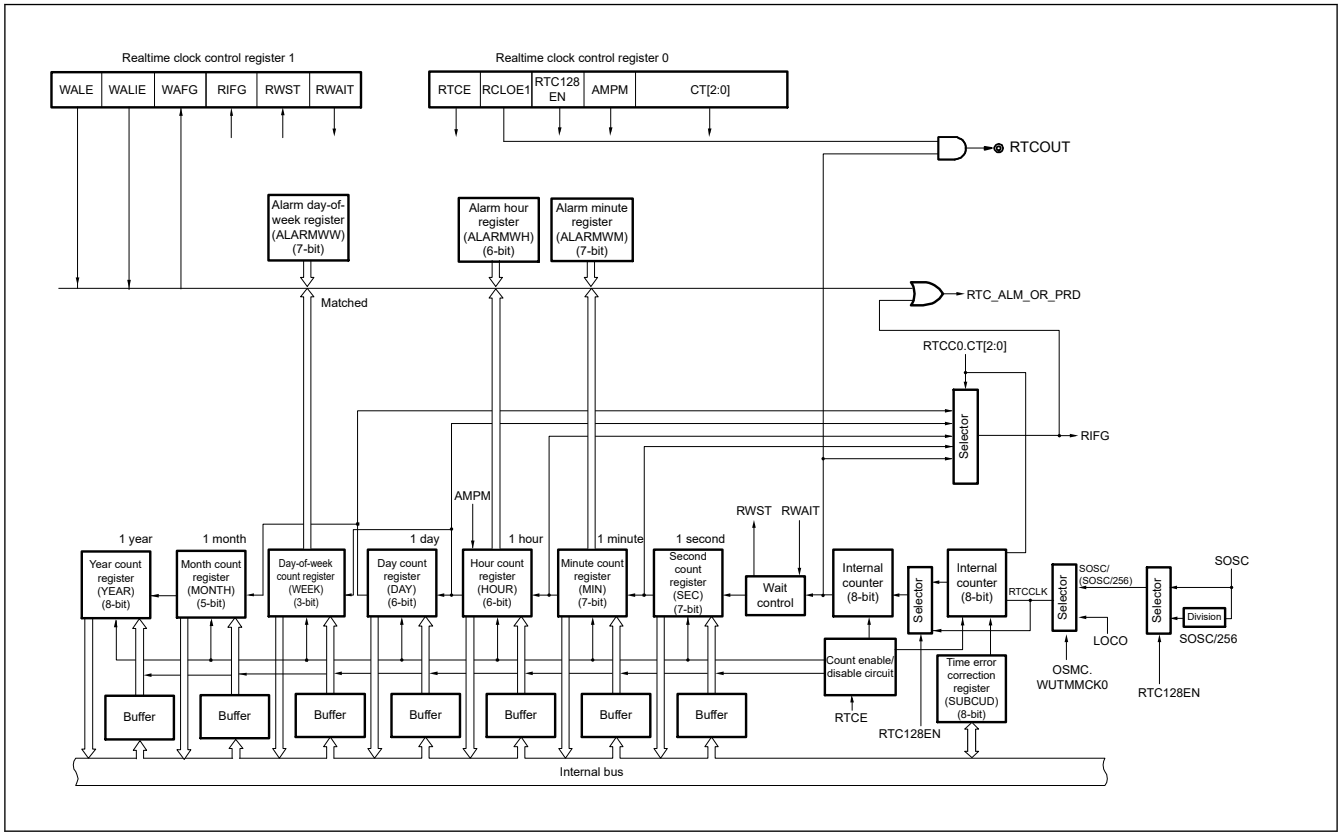


Figure 19.1 Block diagram of the realtime clock

Note: For details, see section 8, Clock Generation Circuit.

Note: The count of years, months, weeks, days, hours, minutes, and seconds can only proceed when the sub-clock oscillator (SOSC = 32.768 kHz) is selected as the operating clock of the realtime clock (RTCCLK). When the low-speed on-chip oscillator clock (LOCO = 32.768 kHz) is selected, only the fixed-cycle interrupt is available.

19.2 Register Descriptions

19.2.1 RTCC0 : Realtime Clock Control Register 0

Base address: RTC_C = 0x400A_2C00

Offset address: 0x000B

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 (synchronously as the seconds count is updated) 0 1 0: Once per 1 s (at the same time as the seconds count is updated) 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 (RTCCLK) 0: SOSC (32.768 kHz) 1: SOSC/256 (128 Hz)	R/W
5	RCLOE1	RTCOUT pin output control 0: Disables output of the RTCOUT pin (1 Hz) 1: Enables output of the RTCOUT pin (1 Hz)	R/W
6	—	This bit is read as 0. The write value should be 0.	R/W
7	RTCE*1	Realtime clock operation control 0: Stops counter operation 1: Starts counter operation	R/W

Note 1. To shift to Software Standby mode immediately after setting the RTCE bit to 1, follow the procedure in Figure 19.3.

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.

The RTCC0 is an 8-bit register that is used to start or stop the realtime clock operation, control the RTCOUT 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 by using the interrupt mask flag register. 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 19.2 shows the time (hour) digits indicated according to the setting of this bit.

RTC128EN bit (Selection of the operating clock for the realtime clock (RTCCLK))

- 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 OSMC register should be set to 0 when setting this bit to 1. For details, see section 8, Clock Generation Circuit.

RCLOE1 bit (RTCOUT pin output control)

This bit is used for RTCOUT pin output control.

RTCE bit (Realtime clock operation control)

This bit is used for realtime clock operation control.

19.2.2 RTCC1 : Realtime Clock Control Register 1

Base address: RTC_C = 0x400A_2C00

Offset address: 0x000C

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. However, if the RIFG and WAFG flags are not in use and a change to the value does not matter, using a bit manipulation instruction for writing to the RTCC1 register does not create a problem.

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 status 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.

Note: This flag is read-only.

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 is a status flag that 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))

WALE bit (Alarm operation control)

When setting a value to the WALE bit while counting is in progress (RTCC0.RTCE = 1) and WALIE is 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.

19.2.3 SEC : Second Count Register

Base address: RTC_C = 0x400A_2C00

Offset address: 0x0000

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

Bit	Symbol	Function	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 19.3.3. 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 is an 8-bit register that 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.

19.2.4 MIN : Minute Count Register

Base address: RTC_C = 0x400A_2C00

Offset address: 0x0001

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 19.3.3. 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.

19.2.5 HOUR : Hour Count Register

Base address: RTC_C = 0x400A_2C00

Offset address: 0x0002

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: Bit 5 (HOUR10[1]) of the HOUR register 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 19.3.3. 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 19.2 shows the relationship between the setting value of the RTCC0.AMPM bit, the hour count register (HOUR) value, and time.

Table 19.2 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.

19.2.6 DAY : Day Count Register

Base address: RTC_C = 0x400A_2C00

Offset address: 0x0004

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 19.3.3. 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.

19.2.7 WEEK : Day-of-Week Count Register

Base address: RTC_C = 0x400A_2C00

Offset address: 0x0003

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 19.3.3. 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.

19.2.8 MONTH : Month Count Register

Base address: RTC_C = 0x400A_2C00

Offset address: 0x0005

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 19.3.3. 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.

19.2.9 YEAR : Year Count Register

Base address: RTC_C = 0x400A_2C00

Offset address: 0x0006

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 19.3.3. 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 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.

19.2.10 SUBCUD : Time Error Correction Register

Base address: RTC_C = 0x400A_2C00

Offset address: 0x0007

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 These bits specify the adjustment value from the prescaler.	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
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 19.3.

Table 19.3 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.

19.2.11 ALARMWM : Alarm Minute Register

Base address: RTC_C = 0x400A_2C00

Offset address: 0x0008

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.

19.2.12 ALARMWH : Alarm Hour Register

Base address: RTC_C = 0x400A_2C00

Offset address: 0x0009

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.

19.2.13 ALARMWW : Alarm Day-of-Week Register

Base address: RTC_C = 0x400A_2C00

Offset address: 0x000A

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
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 19.4 shows an example of setting the alarm.

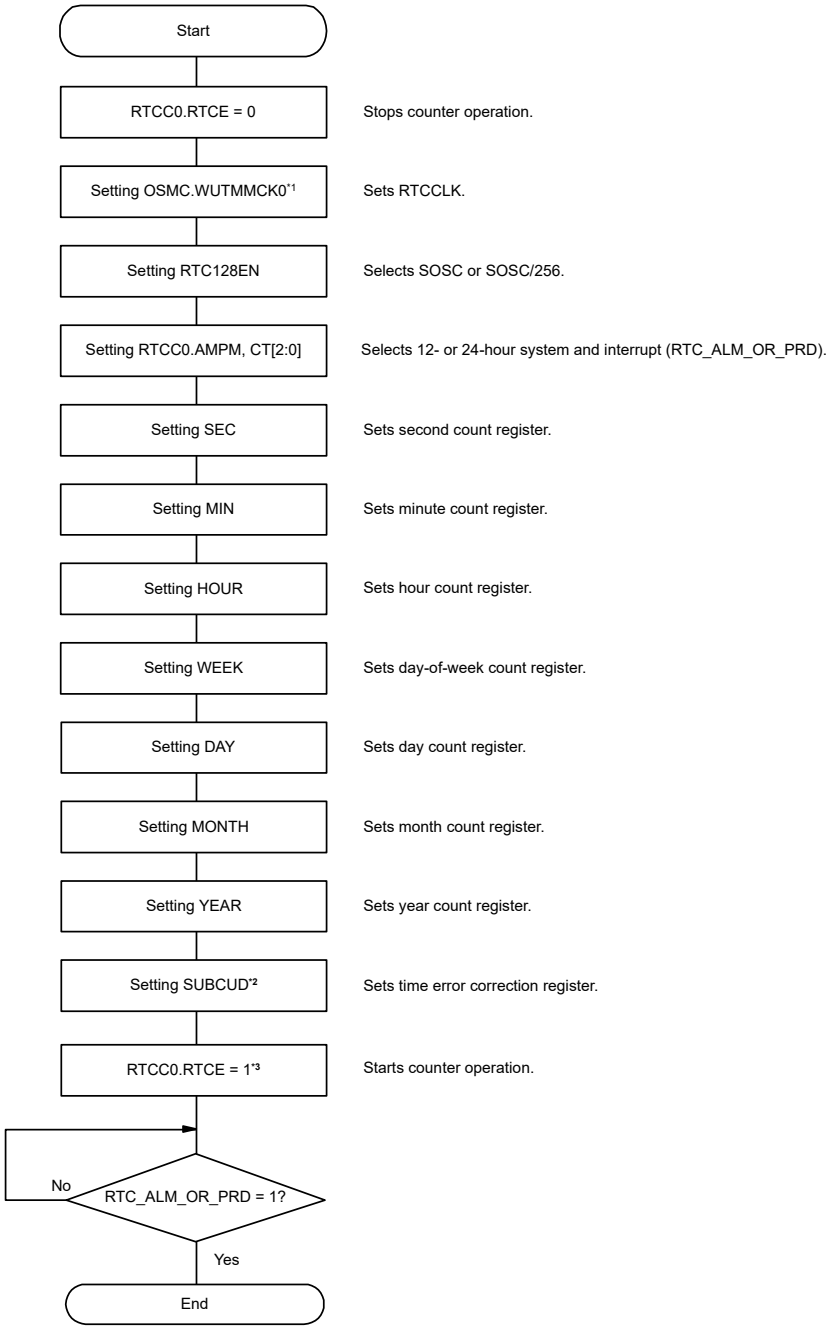
Table 19.4 Example of setting the alarm

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

19.3 Operation

19.3.1 Starting the Realtime Clock Operation

Figure 19.2 shows the procedure for starting realtime clock operation.



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 19.3.6. 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 19.3.2. 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 19.2 Procedure for starting the realtime clock operation

19.3.2 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 19.3, 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 19.3, Example 2).

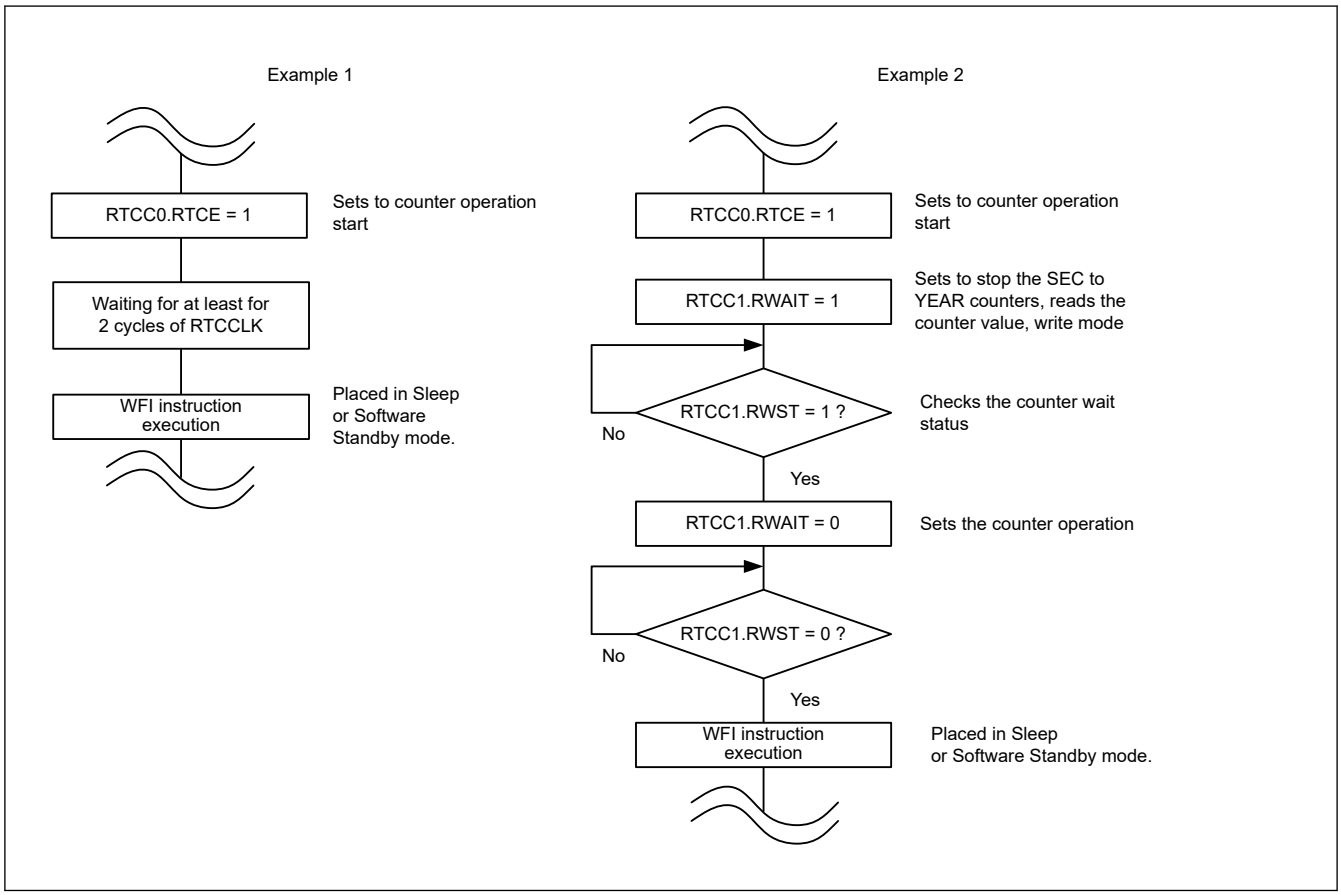


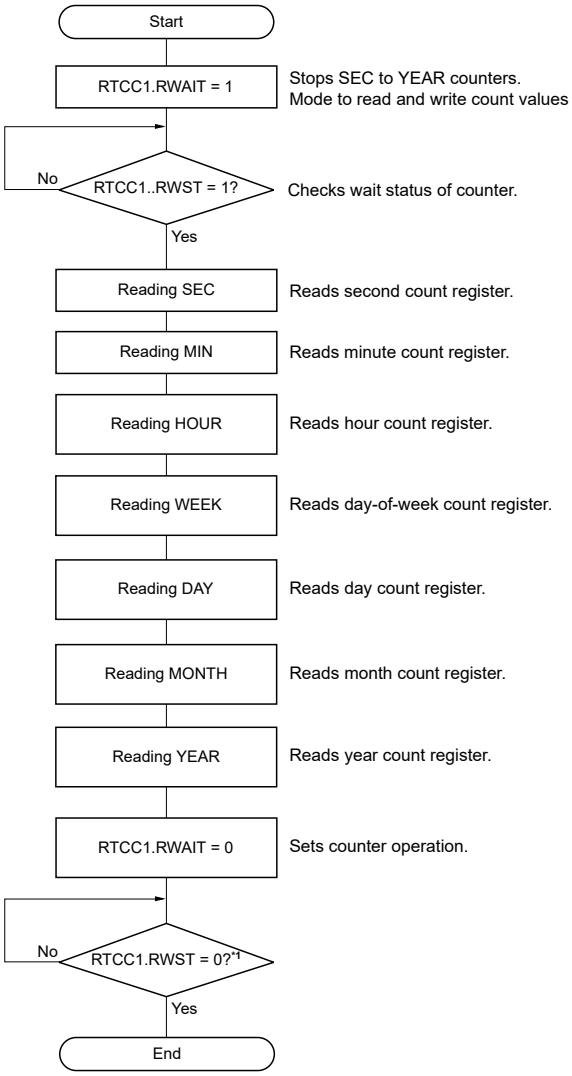
Figure 19.3 Procedure for shifting to Sleep or Software Standby mode after setting RTCC0.RTCE bit to 1

19.3.3 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 19.4 shows the procedure for reading realtime clock.



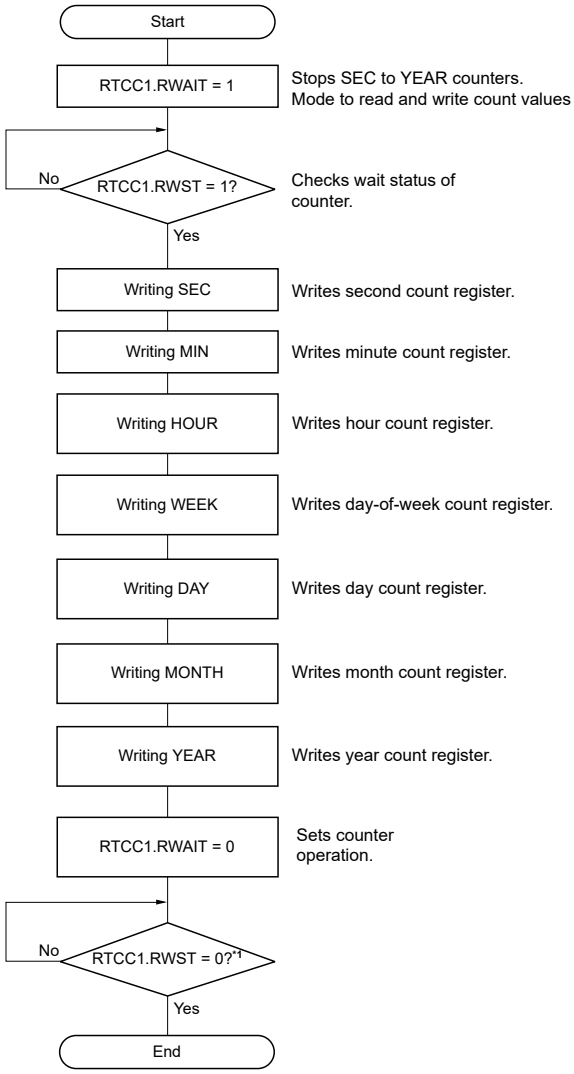
Note 1. Be sure to confirm that RTCC1.RWST = 0 before setting Software Standby mode.

Figure 19.4 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.

Figure 19.5 shows the procedure for writing realtime clock.



Note 1. Be sure to confirm that RTCC1.RWST = 0 before setting Software Standby mode.

Figure 19.5 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.

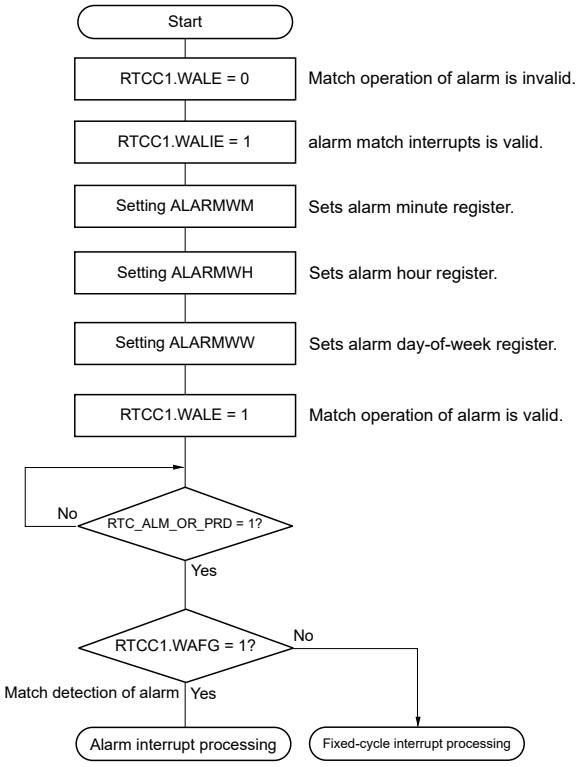
Note: When changing the values of the SEC, MIN, HOUR, WEEK, DAY, MONTH, and YEAR register while the counting is in progress (RTCC0.RTCE = 1), rewrite the values of the MIN register after disabling interrupt processing of RTC_ALM_OR_PRD by using the interrupt mask flag register. Furthermore, clear the RTCC1.WAFG and RTCC1.RIFG flags after rewriting the MIN 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.

19.3.4 Setting Alarm by the Realtime Clock

Set time of alarm after setting 0 to the RTCC1.WALE bit (alarm operation invalid.) first.

Figure 19.6 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 19.6 Alarm processing procedure

19.3.5 1 Hz Output by the Realtime Clock

Table 19.5 shows the 1 Hz output setting procedure.

Table 19.5 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 RTCOUT pin.	RTCC0.RCLOE1 bit to 1.
	<5>	Start counter operation.	Set RTCC0.RTCE bit to 1.
	<6>	Output start from RTCOUT pin.	—

Note: First set the RTCWEN bit to 1, while oscillation of the count clock (RTCCLK) is stable.

19.3.6 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 bit 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 RTCOUT 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 1

Example of correcting from 32772.3 Hz to 32768 Hz (32772.3 Hz – 131.2 ppm)

[Measuring the oscillation frequency]

To measure the oscillation frequency[*1] of each product, a signal at about 32.768 kHz can be output from the PCLBUZ0 pin when the clock error correction register (SUBCUD) is set to its initial value (0x00).

Note 1. See section 19.3.5. 1 Hz Output by the Realtime Clock for the setting procedure of the RTCOUT output, and see section 8.5.6. External Pin Output Clock (CLKOUT) for the setting procedure for output of about 32 kHz from the PCLBUZ0 pin.

[Calculating the correction value]

When the output frequency from the PCLBUZ0 pin is 32772.3 Hz:

Given that the target frequency is 32768 Hz (32772.3 Hz -131.2 ppm) and the extent of correction is -131.2 ppm (not in the range below -63.1 ppm), set the SUBCUD.DEV to 0. Accordingly, the expression for calculating the correction value when the SUBCUD.DEV is 0 is applicable.

Correction value

= Error for correction of counting of 1 minute ÷ 3

= (Oscillation frequency ÷ target frequency – 1) × 32768 × 60 ÷ 3

= (32772.3 ÷ 32768 – 1) × 32768 × 60 ÷ 3

= 86

[Calculating the values to be set to F6 bit and F[5:0] bits of the SUBCUD register]

When the correction value is 86:

If the correction value is 0 or larger (the clock is running slow), set the SUBCUD.F6 bit to 0. Calculate the SUBCUD.F[5:0] bits from the correction value.

{F[5:0] – 1} × 2 = 86

F[5:0] = 44

F[5:0] = 101100b

Consequently, when correcting from 32772.3 Hz to 32768 Hz (32772.3 Hz – 131.2 ppm), setting the correction register such that the SUBCUD.DEV is 0 and the correction value is 86 ((SUBCUD.F6, F[5:0]) = 0101100b) results in the desired frequency of 32768 Hz (error of 0 ppm).

Figure 19.7 shows the operation for correction when the value of (SUBCUD.DEV, F6, F[5:0]) is 00101100b.

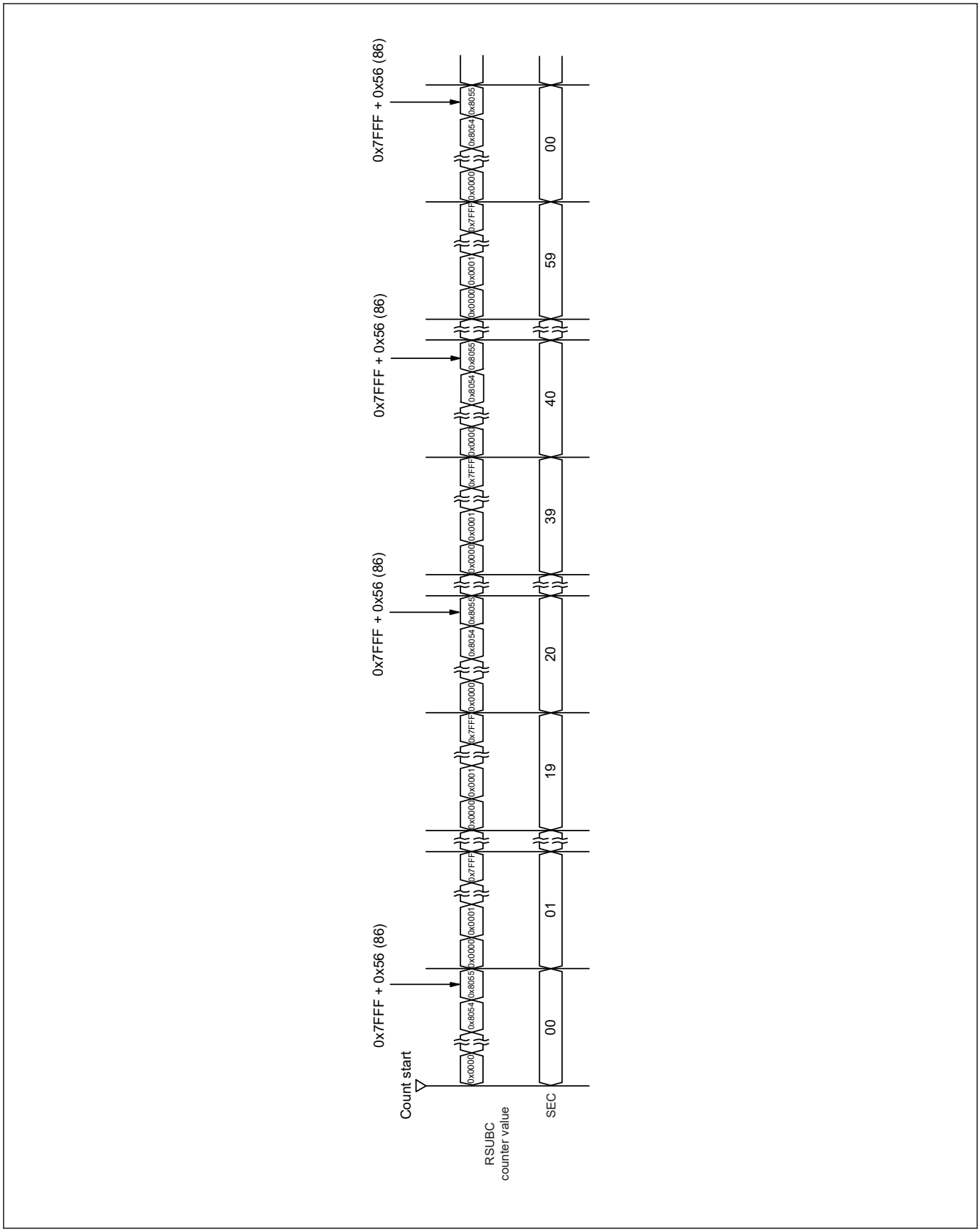


Figure 19.7 Operation for correction when the value of (SUBCUD.DEV, F6, F[5:0]) = 00101100b

(3) Correction example 2

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 RTCOUT pin when the clock error correction register (SUBCUD) is set to its initial value (0x00).

Note 1. See section 19.3.5. 1 Hz Output by the Realtime Clock for the setting procedure for output of about 1 Hz from the RTCOUT pin.

[Calculating the correction value]

When the output frequency from the RTCOUT 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 to 0. Accordingly, the expression for calculating the correction value when the SUBCUD.DEV 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 and 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 to 0. Calculate the SUBCUD.F[5:0] bits 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 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 19.8 shows the operation for correction when the value of (SUBCUD.DEV, F6, F[5:0]) is 11101110b.

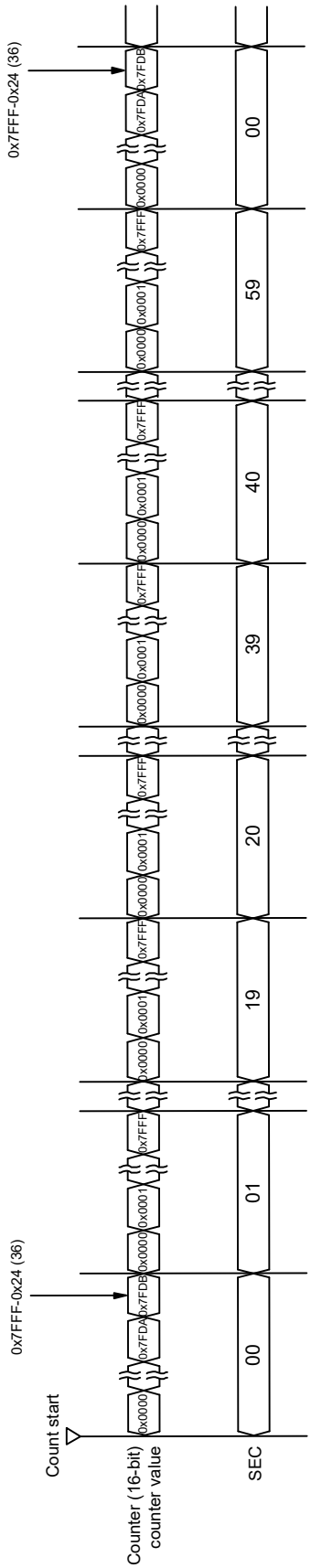


Figure 19.8 Operation for correction when the value of (SUBCUD.DEV, F6, F[5:0]) = 11101110b

20. Independent Watchdog Timer (IWDT)

20.1 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 LOCO, 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.

Table 20.1 lists the IWDT specifications and Figure 20.1 shows a block diagram.

Table 20.1 IWDT specifications

Parameter	Description
Count source	IWDT clock (IWDTCLK) = The divided LOCO by 2
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
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).

The bus interface and registers operate with PCLKB, and the 14-bit counter and control circuits operate with IWDTCLK.

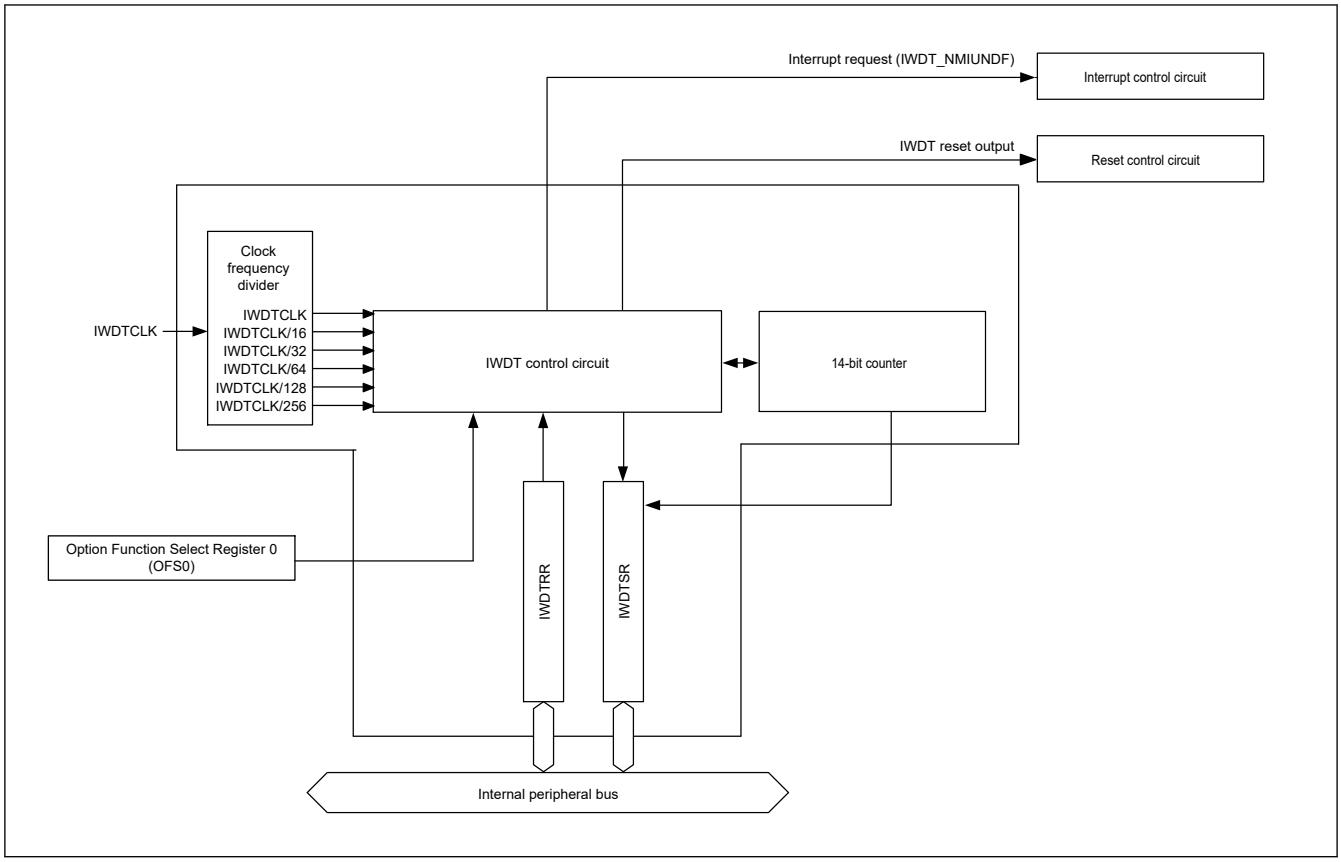


Figure 20.1 IWDT block diagram

20.2 Register Descriptions

20.2.1 IWDTRR : IWDT Refresh Register

Base address: IWDT = 0x4004_4400

Offset address: 0x0000

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	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 20.3.2. Refresh Operation.

20.2.2 IWDTSR : IWDT Status Register

Base address: IWDT = 0x4004_4400

Offset address: 0x0004

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.

20.2.3 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 20.2 lists the relationship between the IWDTCKS[3:0] and IWDTTOPS[1:0] bit settings, the timeout period, and the number of IWDTCLK cycles.

Table 20.2 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 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 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)

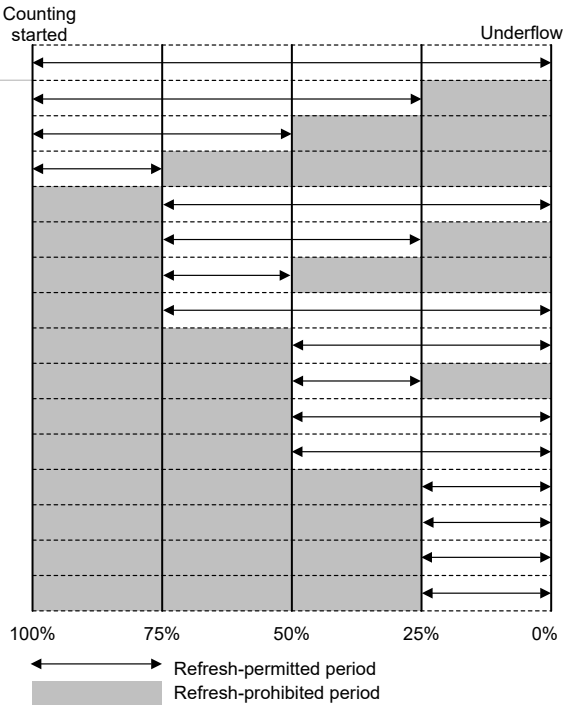
The IWDTRPSS[1:0] bits specify the window start position that indicates the refresh-permitted period. 100%, 75%, 50%, or 25% of the timeout period can be selected for the window start position. Set the window start position to a value greater than the window end position. If the window start position 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 20.3 lists the counter values for the window start and end positions, and Figure 20.2 shows the refresh-permitted period set in the IWDTRPSS[1:0], IWDTRPES[1:0], and IWDTTOPS[1:0] bits.

Table 20.3 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	
b11	b10	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 20.2 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.

20.3 Operation

20.3.1 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)

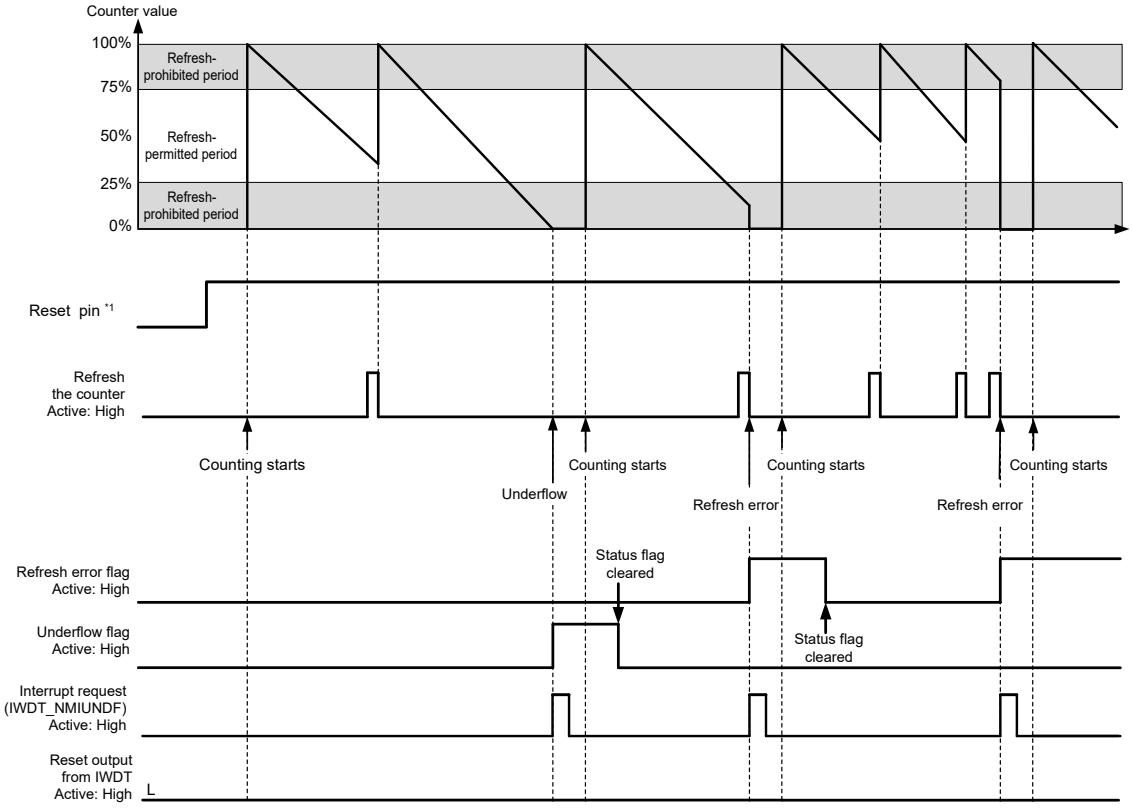
When the reset state is released, the counter automatically starts counting down from the value selected in the IWDT Timeout Period Select bits (OFS0.IWDTTOPS[1:0]).

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 20.3 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 20.3 Operation example in auto start mode

20.3.2 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 Clock Frequency Division Ratio Select bits (OFS0.IWDTCKS[3:0]) to determine how many cycles of the IWDT 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 20.4 shows the IWDT refresh-operation waveforms when PCLKB > IWDTCLK and the clock division ratio is IWDTCLK.

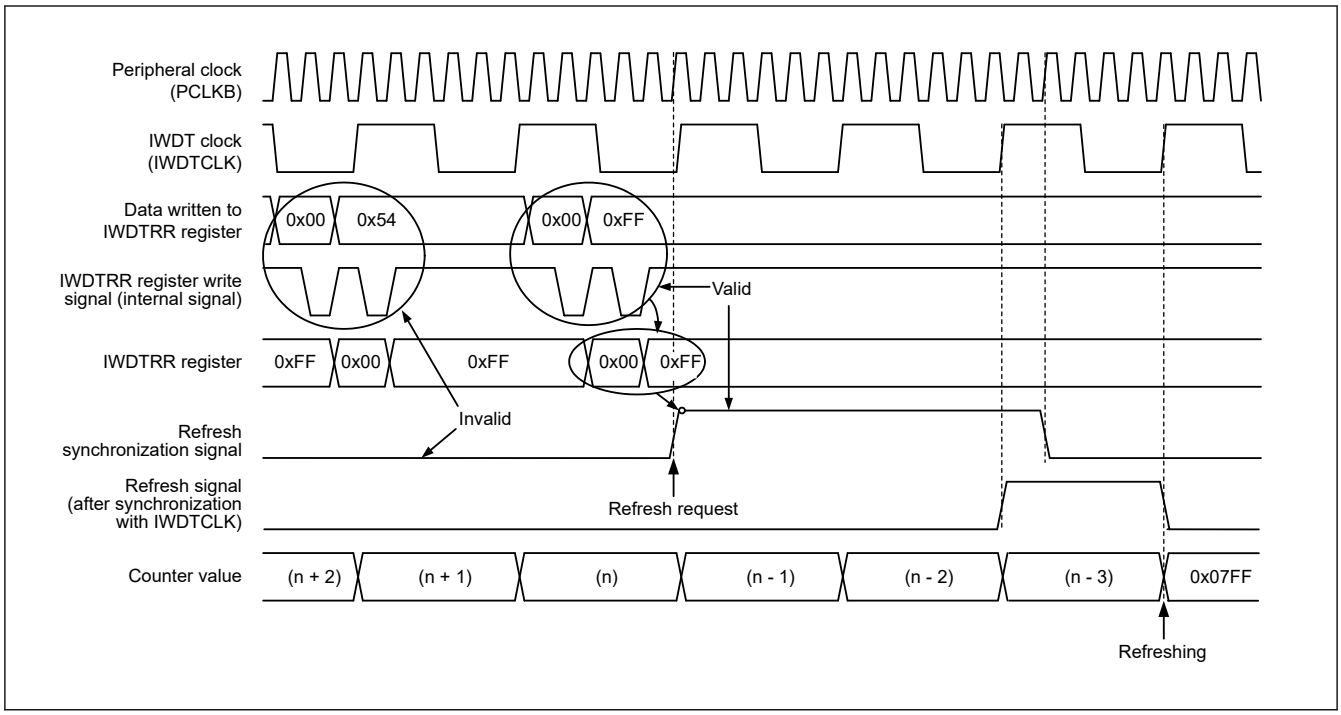


Figure 20.4 IWDT refresh operation waveforms when OFS0.IWDTCKS[3:0] = 0000b, OFS0.IWDTTOPS[1:0] = 11b

20.3.3 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 20.2.2. IWDTSR : IWDT Status Register.

20.3.4 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.

20.3.5 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 11, Interrupt Controller Unit (ICU).

Table 20.4 IWDT interrupt source

Name	Interrupt source	Interrupt to CPU	Start DTC
IWDT_NMIUNDF	• Down-counter underflow • Refresh error	Possible	Not possible

20.3.6 Reading the Down-Counter Value

As the counter is a IWDT 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 20.5 shows the processing for reading the IWDT counter value when PCLKB > IWDTCLK and the clock division ratio is IWDTCLK.

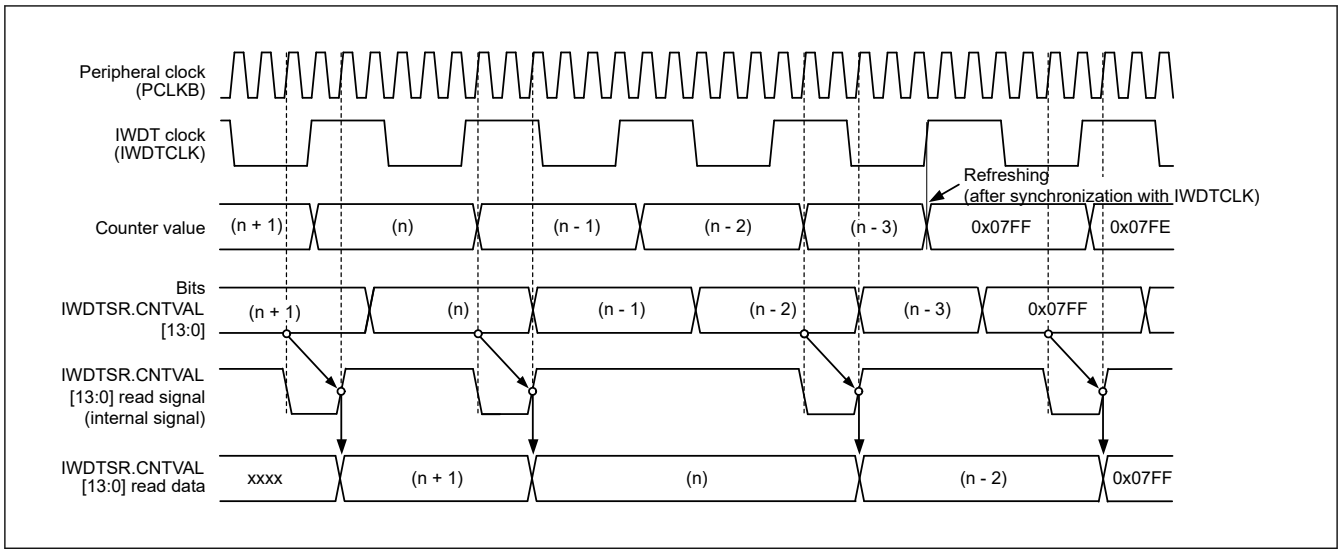


Figure 20.5 Processing for reading IWDT counter value when OFS0.IWDTCKS[3:0] = 0000b, OFS0.IWDTTOPS[1:0] = 11b

20.4 Usage Notes

20.4.1 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.

20.4.2 Clock Division Ratio Setting

Satisfy the frequency of the peripheral module clock (PCLKB) ≥ 4 × (the frequency of the count clock source after division).

21. Serial Array Unit (SAU)

21.1 Overview

A Serial Array Unit (SAU) has two units. Unit 0 has four channels and Unit 1 has two channels. Each channel can operate as 3-wire serial (simplified SPI), UART, or simplified IIC. However, these functions cannot be assigned to the same channel simultaneously.

Function assignment of each channel supported by the MCU is as shown in Table 21.1, Table 21.2 and Table 21.3.

Table 21.1 Function assignment for 32-pin products

Unit	Channel	Used as simplified SPI	Used as UART	Used as simplified I^2C
0	0	SPI00	UART0	IIC00
	1	—		IIC01
	2	—	UART1	—
	3	SPI11		IIC11
1	0	SPI20	UART2 (supporting LINbus)	IIC20
	1	—		—

Table 21.2 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 LINbus)	IIC20
	1	—		IIC21

Table 21.3 Function assignment for 64-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 LINbus)	IIC20
	1	SPI21		IIC21

Note: Most of the following descriptions in this section use the units and channels of the 64-pin products as an example.

21.1.1 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 21.5. 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. $f_{MCK}/6$*2

[Interrupt function]

- Transfer end interrupt or buffer empty interrupt (SAU0_SPI_TXRXI00/SAU0_SPI_TXRXI01/SAU0_SPI_TXRXI10/SAU0_SPI_TXRXI11/SAU1_SPI_TXRXI20/SAU1_SPI_TXRXI21)

[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

Note 1. Set up the transfer rate within a range satisfying the SCK cycle time (tKCY). For details, see section 31, Electrical Characteristics.
Note 2. f_{MCK} is a clock divided by PCLKB with a prescaler.

Note: Use a general-purpose port pin to send a chip select signal when required.

21.1.2 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 (IRQ0).

For details about the settings, see section 21.6. 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 (SAU0_UART_TXI0/SAU0_UART_RXI0/SAU0_UART_TXI1/SAU0_UART_RXI1/SAU1_UART_TXI2/SAU1_UART_RXI2)
- Error interrupt in case of framing error, parity error, or overrun error (SAU0_UART_ERRI0/SAU0_UART_ERRI1/SAU1_UART_ERRI2)

[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

The LIN-bus is accepted in UART2 (channels 0 and 1 of unit 1).

[LIN-bus function]

- Wakeup signal detection
- Break field (BF) detection
- Sync field measurement, baud rate calculation

Note 1. Only UART0 and UART2 support the 9-bit data length.

21.1.3 Simplified I^2C

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 I^2C 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 21.8. Operation of Simplified I^2C 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 (SAU0_IIC_TXRXI00/SAU0_IIC_TXRXI01/SAU0_IIC_TXRXI10/SAU0_IIC_TXRXI11/ SAU1_IIC_TXRXI20/SAU1_IIC_TXRXI21)

[Error detection flag]

- ACK error, or overrun error

[Functions not supported by simplified I^2C]

- 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 I^2C bus of full function, see section 22, I^2C Bus Interface (IICA).

Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

21.2 Configuration of Serial Array Unit

The serial array unit includes the registers, and input and output pins shown in Table 21.4.

Table 21.4 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)
Chip select input	SSI00 pin (for SPI00)
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 0 (SSC0) ● 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)
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 21.1 shows the block diagram of serial array unit 0.

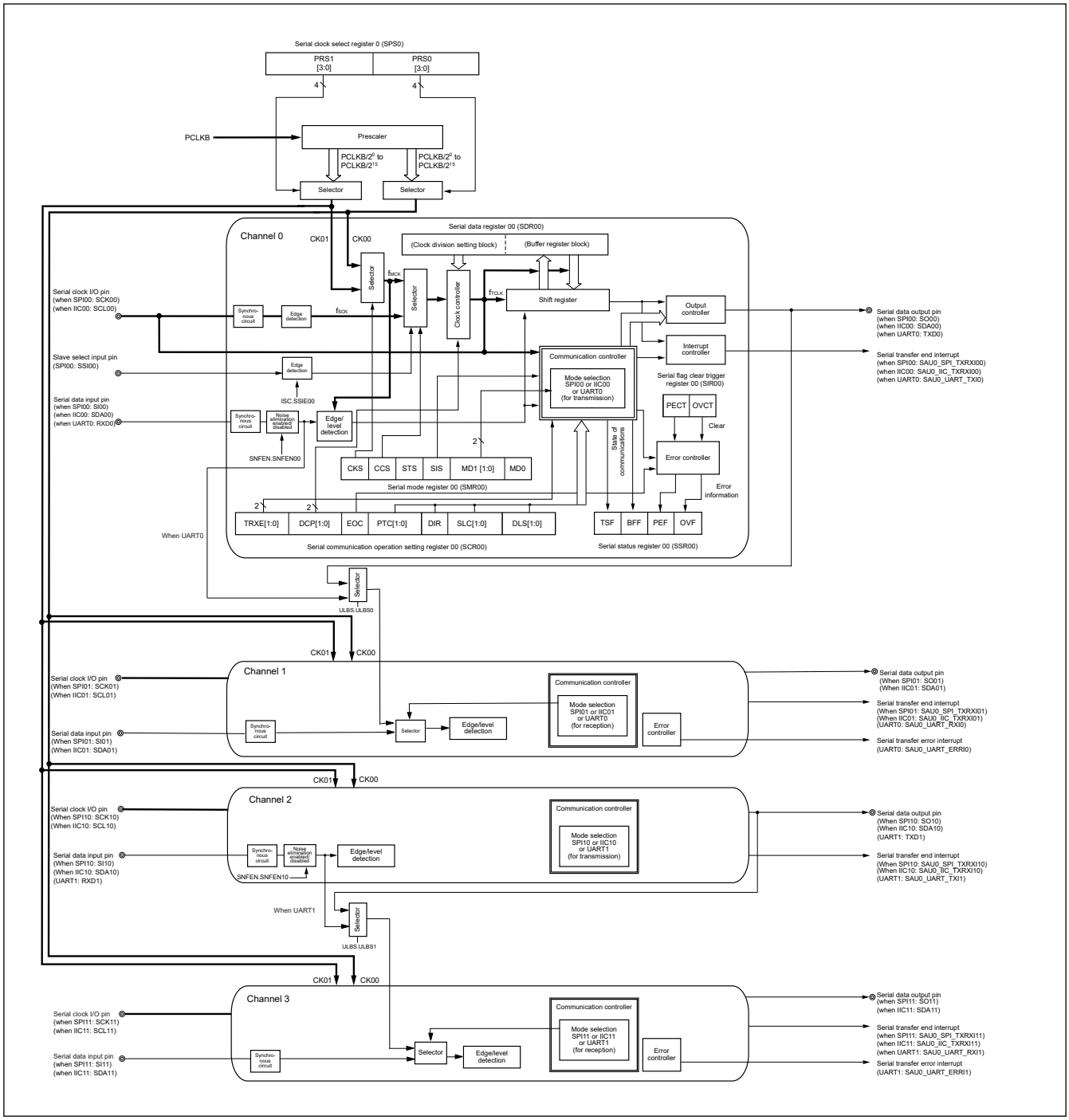


Figure 21.1 Block diagram of serial array unit 0

Figure 21.2 shows the block diagram of serial array unit 1.

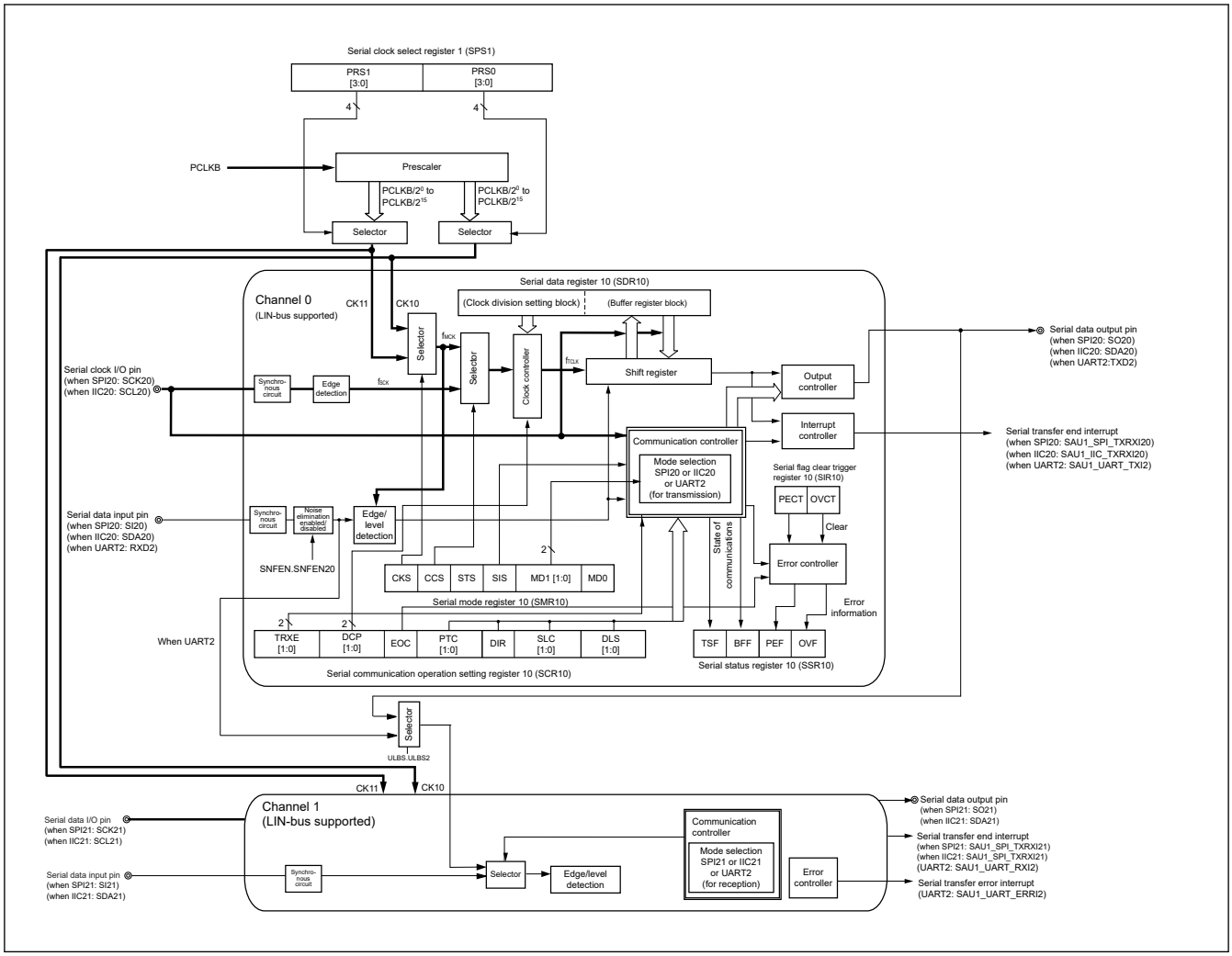


Figure 21.2 Block diagram of serial array unit 1

21.3 Register Descriptions

21.3.1 SPSm : Serial Clock Select Register m (m = 0, 1)

Base address: SAUm = 0x400A_2000 + 0x200 × m (m = 0, 1)

Offset address: 0x0126

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 is a 16-bit register that 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 21.5.

Table 21.5 Selection of operation clock (PRSk[3:0](k = 0, 1)) (1 of 2)

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
0x0	PCLKB	2 MHz	5 MHz	10 MHz	20 MHz	32 MHz
0x1	PCLKB/2	1 MHz	2.5 MHz	5 MHz	10 MHz	16 MHz
0x2	PCLKB/2^2	500 kHz	1.25 MHz	2.5 MHz	5 MHz	8 MHz
0x3	PCLKB/2^3	250 kHz	625 kHz	1.25 MHz	2.5 MHz	4 MHz
0x4	PCLKB/2^4	125 kHz	313 kHz	625 kHz	1.25 MHz	2 MHz
0x5	PCLKB/2^5	62.5 kHz	156 kHz	313 kHz	625 kHz	1 MHz
0x6	PCLKB/2^6	31.3 kHz	78.1 kHz	156 kHz	313 kHz	500 kHz

Table 21.5 Selection of operation clock (PRSk[3:0](k = 0, 1)) (2 of 2)

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
0x7	PCLKB/2^7	15.6 kHz	39.1 kHz	78.1 kHz	156 kHz	250 kHz
0x8	PCLKB/2^8	7.81 kHz	19.5 kHz	39.1 kHz	78.1 kHz	125 kHz
0x9	PCLKB/2^9	3.91 kHz	9.77 kHz	19.5 kHz	39.1 kHz	62.5 kHz
0xA	PCLKB/2^{10}	1.95 kHz	4.88 kHz	9.77 kHz	19.5 kHz	31.3 kHz
0xB	PCLKB/2^{11}	977 Hz	2.44 kHz	4.88 kHz	9.77 kHz	15.6 kHz
0xC	PCLKB/2^{12}	488 Hz	1.22 kHz	2.44 kHz	4.88 kHz	7.81 kHz
0xD	PCLKB/2^{13}	244 Hz	610 Hz	1.22 kHz	2.44 kHz	3.91 kHz
0xE	PCLKB/2^{14}	122 Hz	305 Hz	610 Hz	1.22 kHz	1.95 kHz
0xF	PCLKB/2^{15}	61.0 Hz	153 Hz	305 Hz	610 Hz	977 Hz

Note: 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).

21.3.2 SMRmn : Serial Mode Register mn (mn = 00, 02, 10)

Base address: SAUm = 0x400A_2000 + 0x200 × m

Offset address: 0x0110 + 0x2 × 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
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 higher 7 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 higher 7 bits of the SDRmn register, a transfer clock (f_{TCLK}) is generated.

21.3.3 SMRmn : Serial Mode Register mn (mn = 01, 03, 11)

Base address: SAUm = 0x400A_2000 + 0x200 × m

Offset address: 0x0110 + 0x2 × 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
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

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.

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 a valid edge on the software trigger or the RxDq pin is detected 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 higher 7 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 higher 7 bits of the SDRmn register, a transfer clock (f_{TCLK}) is generated.

21.3.4 SCRm0 : Serial Communication Operation Setting Register m0 (m = 0, 1)

Base address: SAUm = 0x400A_2000 + 0x200 × m (m = 0, 1)

Offset address: 0x0118

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 bits 0 to 8 of the SDRm0 register) (settable in UART mode only) 1 0: 7-bit data length (stored in bits 0 to 6 of the SDRm0 register) 1 1: 8-bit data length (stored in bits 0 to 7 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

Bit	Symbol	Function	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 is a communication operation setting register of channel 0. It 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 or 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)

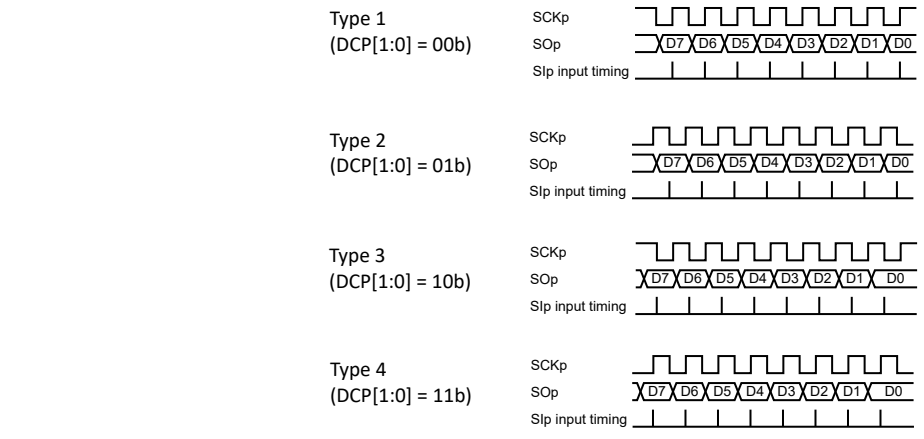
Be sure to set PTC[1:0] = 00b in the simplified SPI mode and simplified I^2C mode

DCP[1:0] bits (Selection of Data and Clock Phase in Simplified SPI Mode)

See Figure 21.3.

Be sure to set DCP[1:0] = 00b in the UART mode and simplified I^2C mode.

Figure 21.3 shows the data and clock phase in simplified SPI mode.



Note: p: SPI number (p = 00, 01, 10, 11, 20, 21)

Figure 21.3 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.

21.3.5 SCRm1 : Serial Communication Operation Setting Register m1 (m = 0, 1)

Base address: SAUm = 0x400A_2000 + 0x200 × m (m = 0, 1)

Offset address: 0x011A

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	—	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 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

Bit	Symbol	Function	R/W
10	EOC	Mask Control of Error Interrupt Signal SAU0_UART_ERRI0 (m = 0), SAU1_UART_ERRI2 (m = 1) 0: Disables generation of error interrupt SAU0_UART_ERRI0 (m = 0), SAU1_UART_ERRI2 (m = 1) (SAUm_UART_RXIq is generated) 1: Enables generation of error interrupt SAU0_UART_ERRI0 (m = 0), SAU1_UART_ERRI2 (m = 1) (SAUm_UART_RXIq 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 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 is a communication operation setting register of channel 1. It 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] 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 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 or 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_UART_ERRI0 (m = 0), SAU1_UART_ERRI2 (m = 1))

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 21.3.

Be sure to set DCP[1:0] = 00b in the UART mode and simplified I^2C mode.

TRXE[1:0] bits (Setting of Channel 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_UART_ERRI0 may be generated.
When using SPI21 not with SCR11.EOC = 0, error interrupt SAU1_UART_ERRI2 may be generated.

21.3.6 SCR02 : Serial Communication Operation Setting Register 02

Base address: SAU0 = 0x400A_2000

Offset address: 0x011C

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 is a communication operation setting register of channel 2. It 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 or 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 21.3.

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.

21.3.7 SCR03 : Serial Communication Operation Setting Register 03

Base address: SAU0 = 0x400A_2000

Offset address: 0x011E

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
10	EOC	Mask Control of Error Interrupt Signal SAU0_UART_ERRI1 0: Disables generation of error interrupt SAU0_UART_ERRI1 (SAU0_UART_RXI1 is generated) 1: Enables generation of error interrupt SAU0_UART_ERRI1 (SAU0_UART_RXI1 is not generated if an error occurs)	R/W
11	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	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 is a communication operation setting register of channel 3. It 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 or 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_UART_ERRI1)

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 21.3.

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_UART_ERRI1 may be generated.

21.3.8 SDRmn : Serial Data Register mn (mn = 00, 01, 02, 03, 10, 11)

Base address: SAUm = 0x400A_2000 + 0x200 × m

Offset address: 0x0000 + 0x2 × 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]	Transfer Clock Setting by Dividing the Operation Clock 0x00: f_{MCK} /2 (Setting prohibited for UART and simplified I^2C) 0x01: f_{MCK} /4 (Setting prohibited for UART) 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] and SDR03.DAT[8] bits are reserved bits. These bits are read as 0. The write value should be 0.

The SDRmn 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*1 and SDR03*1, 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 0000000b. 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 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.

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

Note 1. Only the following UARTs support the 9-bit data length.
- UART0 and UART2

Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

21.3.9 SIRmn : Serial Flag Clear Trigger Register mn (mn = 00, 02, 10)

Base address: SAUm = 0x400A_2000 + 0x200 × m

Offset address: 0x0108 + 0x2 × 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 bit 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 bit 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 is a trigger register that is used to clear each error flag of channel n.

When each bit (PECT, OVCT) of this register is set to 1, the corresponding bit (PEF, OVF) of serial status register mn is cleared to 0. Because the SIRmn 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.

21.3.10 SIRmn : Serial Flag Clear Trigger Register mn (mn = 01, 03, 11)

Base address: SAUm = 0x400A_2000 + 0x200 × m

Offset address: 0x0108 + 0x2 × 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 bit 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 bit 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 bit 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 is a trigger register that is used to clear each error flag of channel n.

When each bit (FECT, PECT, OVCT) of this register is set to 1, the corresponding bit (FEF, PEF, OVF) of serial status register mn is cleared to 0. Because the SIRmn 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.

21.3.11 SSRmn : Serial Status Register mn (mn = 00, 02, 10)

Base address: SAUm = 0x400A_2000 + 0x200 × m

Offset address: 0x0100 + 0x2 × 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

Bit	Symbol	Function	R/W
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 (SSC0.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 bit (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 bit (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 bit (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 bit (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.

21.3.12 SSRmn : Serial Status Register mn (mn = 01, 03, 11)

Base address: SAUm = 0x400A_2000 + 0x200 × m

Offset address: 0x0100 + 0x2 × 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: When the simplified SPI is handling reception in the Snooze mode (SSC0.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.

OVF bit (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 bit (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 bit (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 bit (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 bit (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.

21.3.13 SS0 : Serial Channel Start Register 0

Base address: SAU0 = 0x400A_2000

Offset address: 0x0122

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.	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.

The SS0 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 a bit (SS[n]) of this register, 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.

21.3.14 SS1 : Serial Channel Start Register 1

Base address: SAU1 = 0x400A_2200

Offset address: 0x0122

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.	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.

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 a bit (SS[n]) of this register, 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.

21.3.15 ST0 : Serial Channel Stop Register 0

Base address: SAU0 = 0x400A_2000

Offset address: 0x0124

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.	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.

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 a bit (ST[n]) of this register, 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.

21.3.16 ST1 : Serial Channel Stop Register 1

Base address: SAU1 = 0x400A_2200

Offset address: 0x0124

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.	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.

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 a bit (ST[n]) of this register, 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.

21.3.17 SE0 : Serial Channel Enable Status Register 0

Base address: SAU0 = 0x400A_2000

Offset address: 0x0120

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.

21.3.18 SE1 : Serial Channel Enable Status Register 1

Base address: SAU1 = 0x400A_2200

Offset address: 0x0120

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.

21.3.19 SOE0 : Serial Output Enable Register 0

Base address: SAU0 = 0x400A_2000

Offset address: 0x012A

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.	R/W

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.

21.3.20 SOE1 : Serial Output Enable Register 1

Base address: SAU1 = 0x400A_2200

Offset address: 0x012A

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.	R/W

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.

21.3.21 SO0 : Serial Output Register 0

Base address: SAU0 = 0x400A_2000

Offset address: 0x0128

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.	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.	R/W

The SO0 is a buffer register for serial output of each channel of serial array unit 0.

The value of the SO[n] bit of this register is output from the serial data output pin of channel n.

The value of the CKO[n] bit of this register is output from the serial clock output pin of channel n.

The SO[n] bit of this register 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 of this register 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.

21.3.22 SO1 : Serial Output Register 1

Base address: SAU1 = 0x400A_2200

Offset address: 0x0128

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	0	0	1	1	0	0	0	0	0	0	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
7:2	—	These bits are read as 0. The write value should be 0.	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
15:10	—	These bits are read as 0. The write value should be 0.	R/W

The SO1 is a buffer register for serial output of each channel of serial array unit 1.

The value of the SO[n] bit of this register is output from the serial data output pin of channel n.

The value of the CKO[n] bit of this register is output from the serial clock output pin of channel n.

The SO[n] bit of this register 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 of this register 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.

21.3.23 SOL0 : Serial Output Level Register 0

Base address: SAU0 = 0x400A_2000

Offset address: 0x0134

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.	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.	R/W

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 channel n is in operation (when SE0.SE[n] = 1).

Figure 21.4 shows examples in which the level of transmit data is reversed during UART transmission.

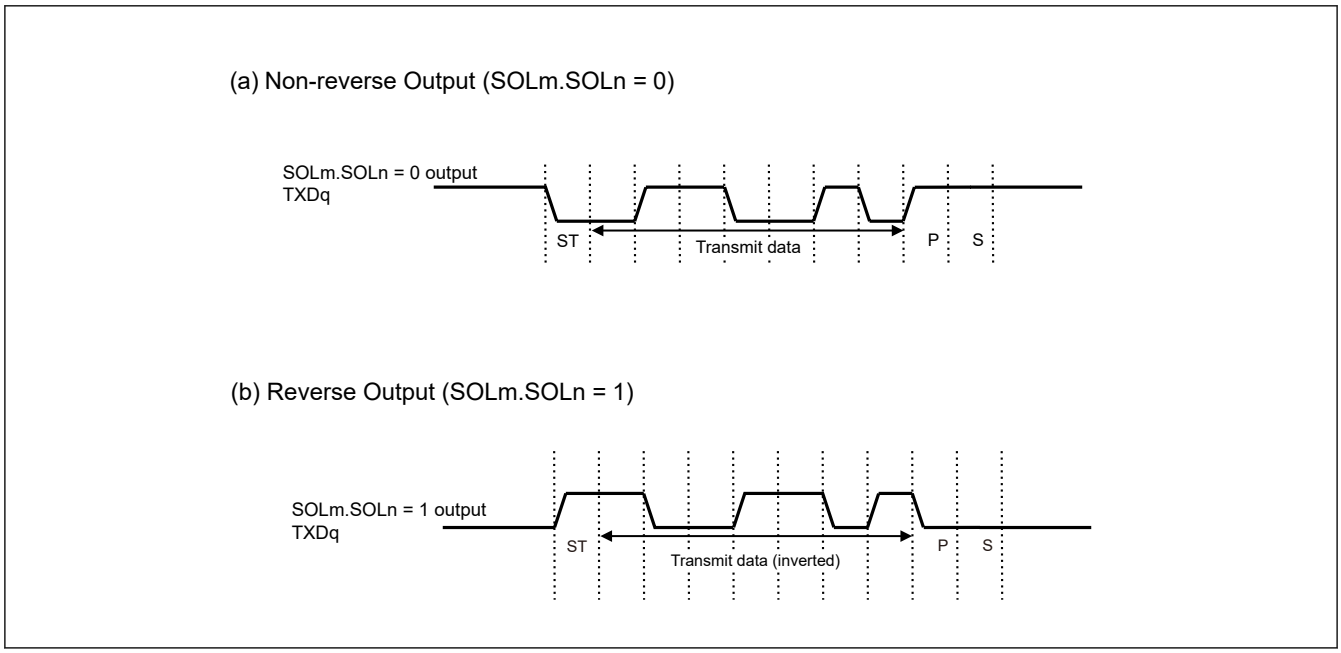


Figure 21.4 Examples of reverse transmit data

Note: m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), q: UART number (q = 0 to 2)

21.3.24 SOL1 : Serial Output Level Register 1

Base address: SAU1 = 0x400A_2200

Offset address: 0x0134

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.	R/W

The SOL1 register is used to set inversion of the data output level of each channel 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[n] = 1).

When serial output is disabled (SOE1.SOE[n] = 0), the value of the SO1.SO[n] bit is output as is.

Rewriting the SOL1 register is prohibited when the channel n is in operation (when SE1.SE[n] = 1).

Figure 21.4 shows examples in which the level of transmit data is reversed during UART transmission.

21.3.25 SSC0 : Serial Standby Control Register 0

Base address: SAU0 = 0x400A_2000

Offset address: 0x0138

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 SAU0_UART_ERRI0 1: Disable the generation of error interrupts SAU0_UART_ERRI0	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.

Note: The maximum transfer rate in the Snooze mode is as follows.
- When using SPI00: Up to 1 Mbps
- When using UART0: Up to 115.2 kbps (when setting the SBYCR.FWKUP = 1, PCLKB = HOCO (32 MHz))

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 or medium-speed on-chip oscillator clock is selected for the CPU and peripheral hardware 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, after returning from software standby mode to normal operation mode, be sure to set the SWC bit to 0.

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 21.6 shows the interrupt in UART reception operation in Snooze mode.

Table 21.6 Interrupt in UART reception operation in Snooze mode

SCRmn.EOC bit	SSEC bit	Reception ended successfully	Reception ended in an error
0	0	SAU0_UART_RXI0 is generated.	SAU0_UART_RXI0 is generated.
0	1	SAU0_UART_RXI0 is generated.	SAU0_UART_RXI0 is generated.
1	0	SAU0_UART_RXI0 is generated.	SAU0_UART_ERRI0 is generated.
1	1	SAU0_UART_RXI0 is generated.	No interrupt is generated.

21.3.26 ISC : Input Switch Control Register

Base address: PORGA = 0x400A_1000

Offset address: 0x0003

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	SSIE00	ISC1	ISC0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ISC0	Switching External Interrupt (IRQ0) Input 0: Uses the input signal of the IRQ0 pin as an external interrupt (normal operation) 1: Uses the input signal of the RXD2 pin as an external interrupt (wakeup signal detection)	R/W
1	ISC1	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	SSIE00	Setting of the SSI00 Input of Channel 0 in the Communications Through SPI00 in the Slave Mode 0: The SSI00 input is disabled. 1: The SSI00 input is enabled.	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: When the LIN-bus communication function is used, select the input signal of the RXD2 pin by setting ISC1 to 1.

The ISC1 and ISC0 bits of the ISC register are used to implement LIN-bus communication operation by using TAU channel 7 in association with the serial array unit. When bit 0 is set to 1, the input signal of the serial data input (RXD2) pin is selected as an external interrupt (IRQ0) that can be used to detect a wakeup signal. When bit 1 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, and the low width of the break field and the pulse width of the sync field can be measured by the timer.

The SSIE00 bit is used to control the SSI00 input of channel 0 in the communications through SPI00 in the slave mode.

Reception and transmission do not proceed even if the serial clock is input while the SSI00 pin is being driven high.

Reception and transmission proceed in response to an input of the serial clock according to the mode setting while the SSI00 pin is being driven low.

21.3.27 SNFEN : SAU Noise Filter Enable Register

Base address: PORGA = 0x400A_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

Bit	Symbol	Function	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].

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.

21.3.28 ULBS : UART Loopback Select Register

Base address: PORGA = 0x400A_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

Bit	Symbol	Function	R/W
4	ULBS4	Selection of the UARTA0 Loopback Function 0: Inputs the states of the RXDA0 pin of serial array unit 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 array unit 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.

21.4 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 21.7 to Table 21.10 show each register setting when stopping the operation by channels.

(a) Serial channel stop register m (STm)

The STm is a trigger register that is used to enable stopping communication or count by each channel.

Table 21.7 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 21.8 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 21.9 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 is a buffer register for serial output of each channel.

Table 21.10 Setting of serial output register m (SOm) when sopping 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] bit 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] bit to 1. 1: Serial clock output value is 1

Note: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

21.5 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 (SAU0_SPI_TXRXI00/SAU0_SPI_TXRXI01/SAU0_SPI_TXRXI10/SAU0_SPI_TXRXI11/SAU1_SPI_TXRXI20/SAU1_SPI_TXRXI21)

[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, which support asynchronous reception.

Note 1. Set up the transfer rate within a range satisfying the SCK cycle time (tKCY). For details, see section 31, 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 and 3 of SAU0 and channel 0 of SAU1. See Table 21.1 and Table 21.2.

Simplified SPI performs the following seven types of communication operations.

- Master transmission (See section 21.5.1. Master Transmission.)
- Master reception (See section 21.5.2. Master Reception.)
- Master transmission and reception (See section 21.5.3. Master Transmission and Reception.)
- Slave transmission (See section 21.5.4. Slave Transmission.)
- Slave reception (See section 21.5.5. Slave Reception.)
- Slave transmission and reception (See section 21.5.6. Slave Transmission and Reception.)
- Snooze mode function (See section 21.5.7. Snooze Mode Function.)

21.5.1 Master Transmission

Master transmission is when a microcontroller outputs a transfer clock and transmits data to another device.

Table 21.11 shows the specification of master transmission of Simplified SPI.

Table 21.11 Specification of master 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 0 of SAU1
Pins used	SCK00, SO00	SCK01, SO01	SCK10, SO10	SCK11, SO11	SCK20, SO20	SCK21, SO21
Interrupt	SAU0_SPI_TXRXI00	SAU0_SPI_TXRXI01	SAU0_SPI_TXRXI10	SAU0_SPI_TXRXI11	SAU1_SPI_TXRXI20	SAU1_SPI_TXRXI21
	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 × 2^{15} × 128) [Hz] PCLKB: System clock frequency					
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: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)
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 31, Electrical Characteristics .

(1) Register setting

Table 21.12 to Table 21.17 show examples of the register contents for master transmission of simplified SPI.

(a) Serial mode register mn (SMRmn)

Table 21.12 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 21.13 Example of serial communication operation setting register mn (SCRmn) contents for master 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 For details about the setting, see section 21.3. 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 21.14 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 21.15 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 21.16 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 21.17 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 21.18 shows the procedure for initial setting of master transmission.

Table 21.18 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	Set a Peripheral Select Register, a Pmn Output Data Register and a Pmn Direction Register (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] bit = 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.

Table 21.19 shows the procedure for stopping master transmission.

Table 21.19 Procedure for stopping master transmission

Step		Process	Detail
Procedure for stopping master transmission	<1>	Starting setting to stop	—
	<2>	Wait until SSRmn.TSF 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)

Table 21.20 shows the procedure for resuming master transmission.

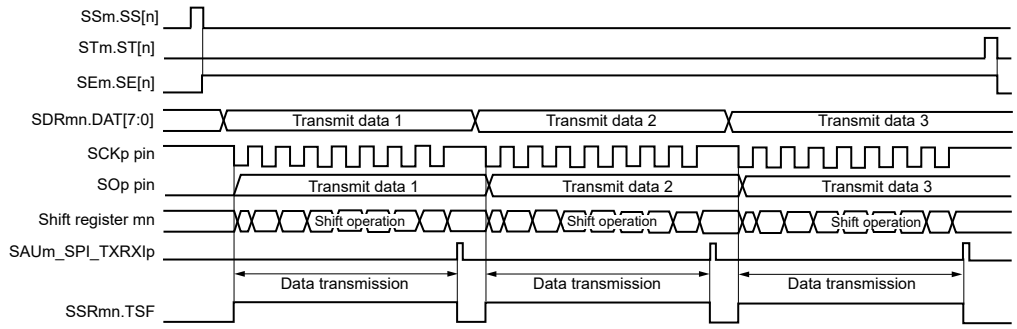
Table 21.20 Procedure for resuming master transmission

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)

(3) Processing flow (in single transmission mode)

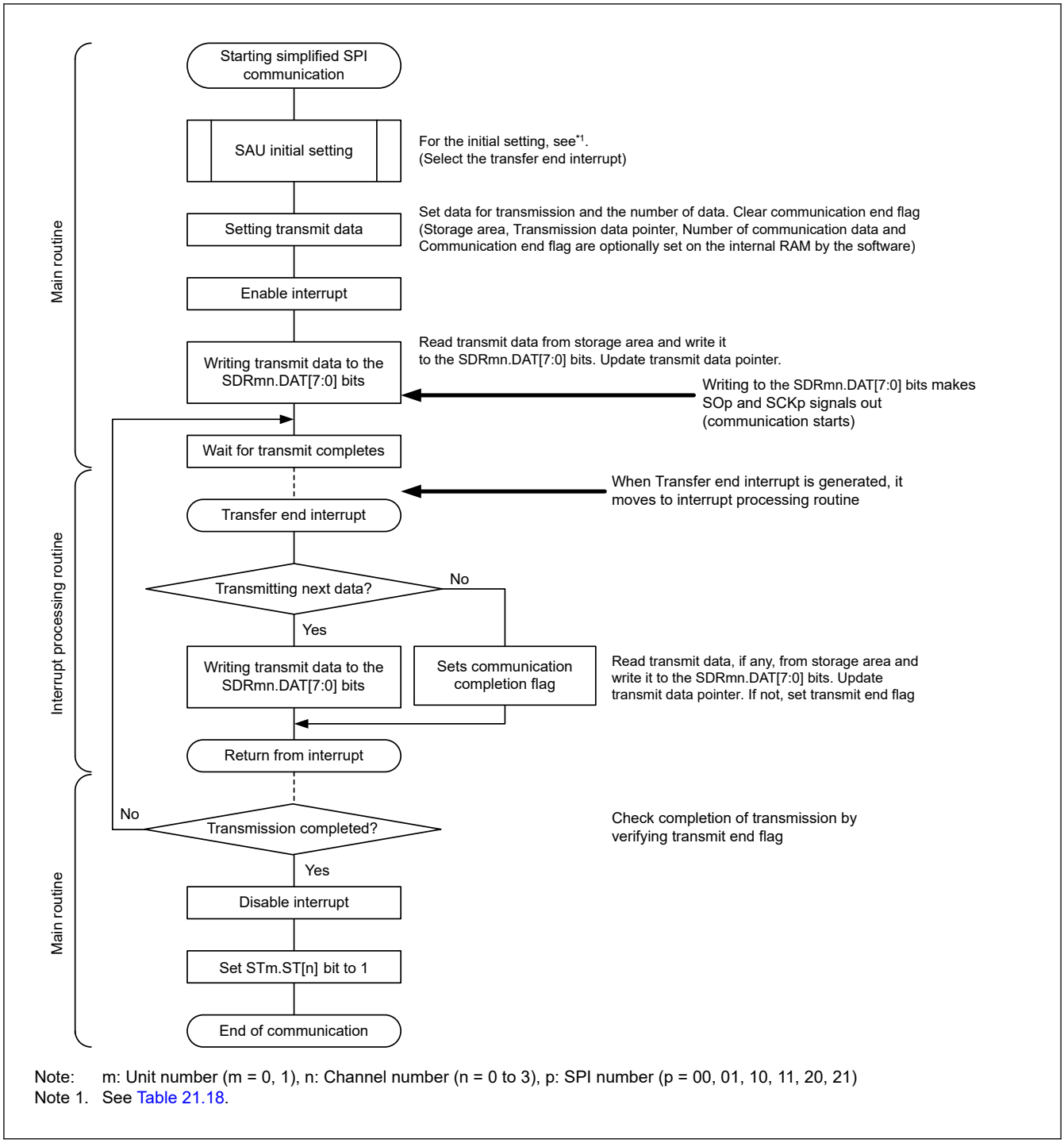
Figure 21.5 shows the timing of master transmission (in single transmission mode) (Type 1: SCRmn.DCP[1:0] = 00b).



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 21.5 Timing of master transmission (in single transmission mode) (type 1: SCRmn.DCP[1:0] = 00b)

Figure 21.6 shows the flowchart of master transmission (in single transmission mode).

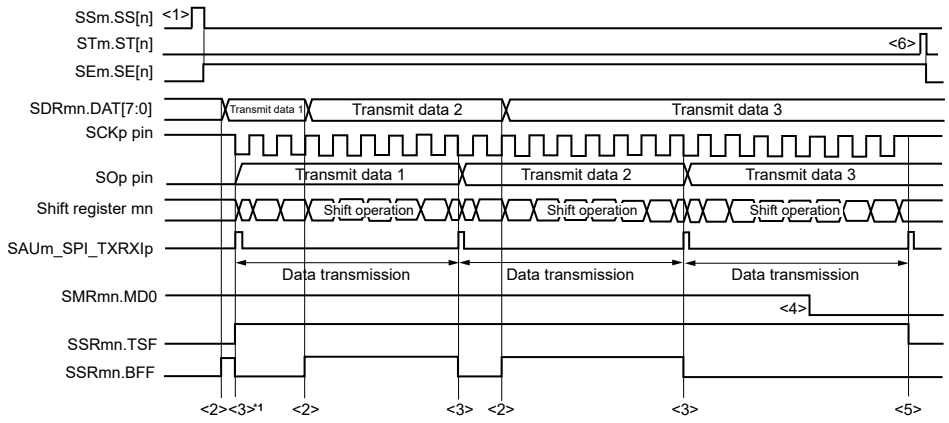


Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: SPI number (p = 00, 01, 10, 11, 20, 21)
Note 1. See Table 21.18.

Figure 21.6 Flowchart of master transmission (in single transmission mode)

(4) Processing flow (in continuous transmission mode)

Figure 21.7 shows the timing of master transmission (in continuous transmission mode) (Type 1: SCRmn.DCP[1:0] = 00b).



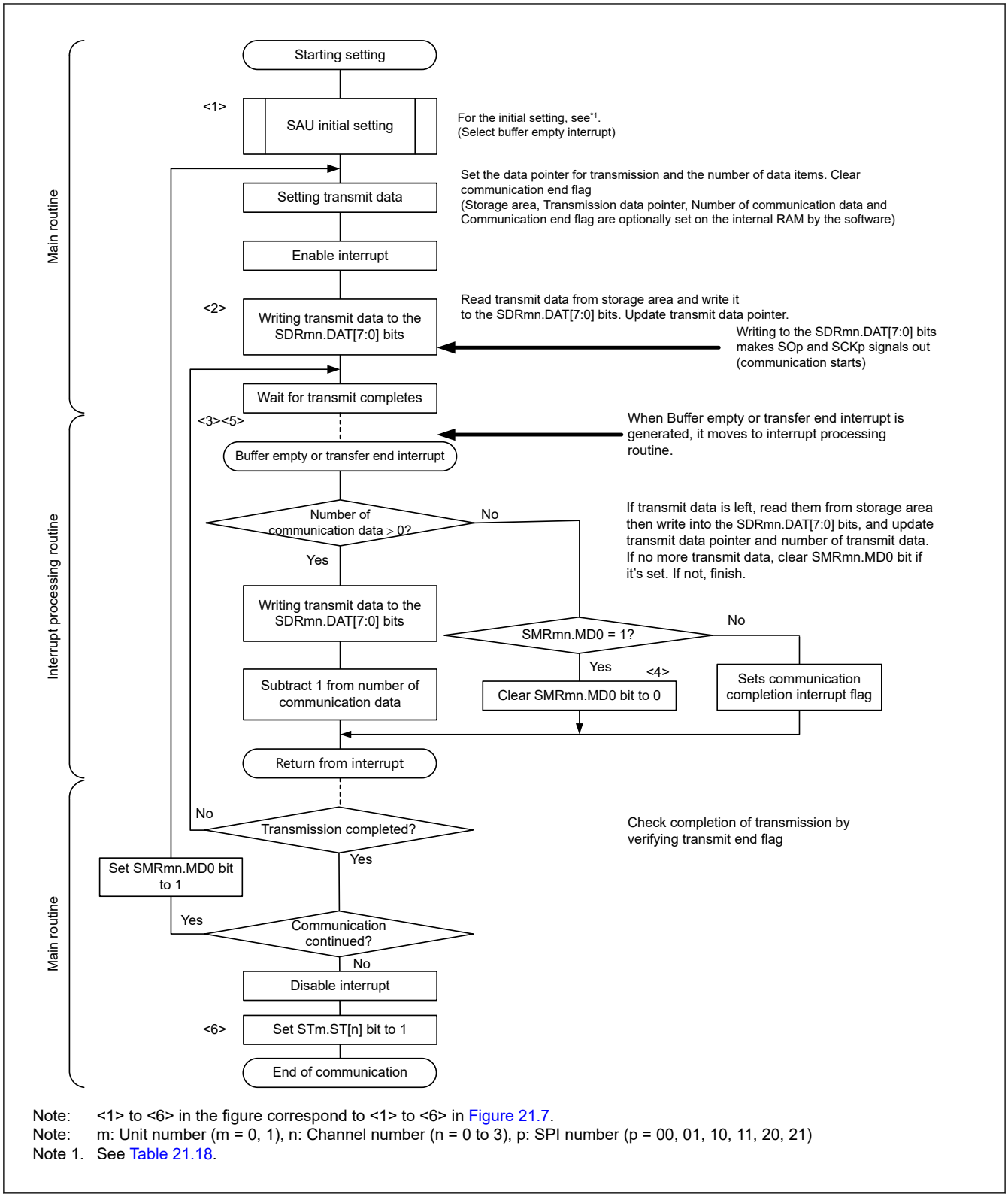
Note: The MD0 bit of serial mode register mn (SMRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last transmit data.

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 bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

Figure 21.7 Timing of master transmission (in continuous transmission mode) (type 1: SCRmn.DCP[1:0] = 00b)

Figure 21.8 shows the flowchart of master transmission (in continuous transmission mode).



Note: <1> to <6> in the figure correspond to <1> to <6> in Figure 21.7.
Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: SPI number (p = 00, 01, 10, 11, 20, 21)
Note 1. See Table 21.18.

Figure 21.8 Flowchart of master transmission (in continuous transmission mode)

21.5.2 Master Reception

Master reception is when a microcontroller outputs a transfer clock and receives data from another device.

Table 21.21 shows the specification of master reception of Simplified SPI.

Table 21.21 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_SPI_TXRXI00	SAU0_SPI_TXRXI01	SAU0_SPI_TXRXI10	SAU0_SPI_TXRXI11	SAU1_SPI_TXRXI20	SAU1_SPI_TXRXI21
	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] PCLKB: System clock frequency					
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. See section 31, Electrical Characteristics.

(1) Register setting

Table 21.22 to Table 21.27 show examples of the register contents for master reception of Simplified SPI.

(a) Serial mode register mn (SMRmn)

Table 21.22 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 21.23 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 21.3. 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 21.24 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 21.25 Example of serial output register m (SOm) contents for master 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 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 21.26 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 21.27 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: ×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Table 21.28 shows the procedure for initial setting of master reception.

Table 21.28 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)

Table 21.29 shows the procedure for stopping master reception.

Table 21.29 Procedure for stopping master reception

Step		Process	Detail
Procedure for stopping master reception	<1>	Starting setting to stop	—
	<2>	Wait until SSRmn.TSF 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)

Table 21.30 shows the procedure for resuming master reception.

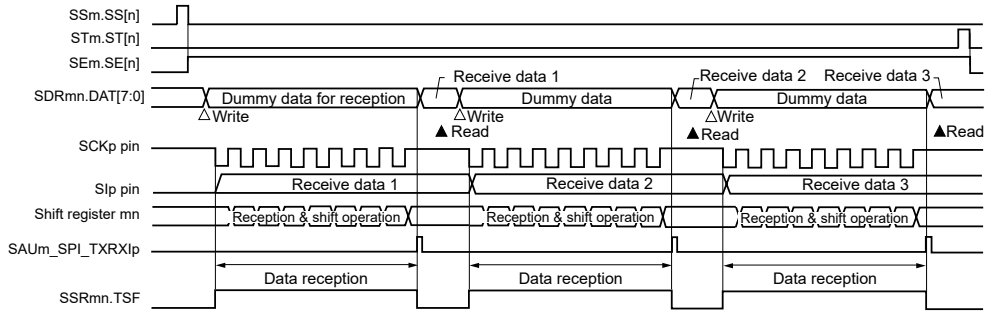
Table 21.30 Procedure for resuming master reception

Step		Process	Detail
Procedure for resuming master transmission	<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)

(3) Processing flow (in single reception mode)

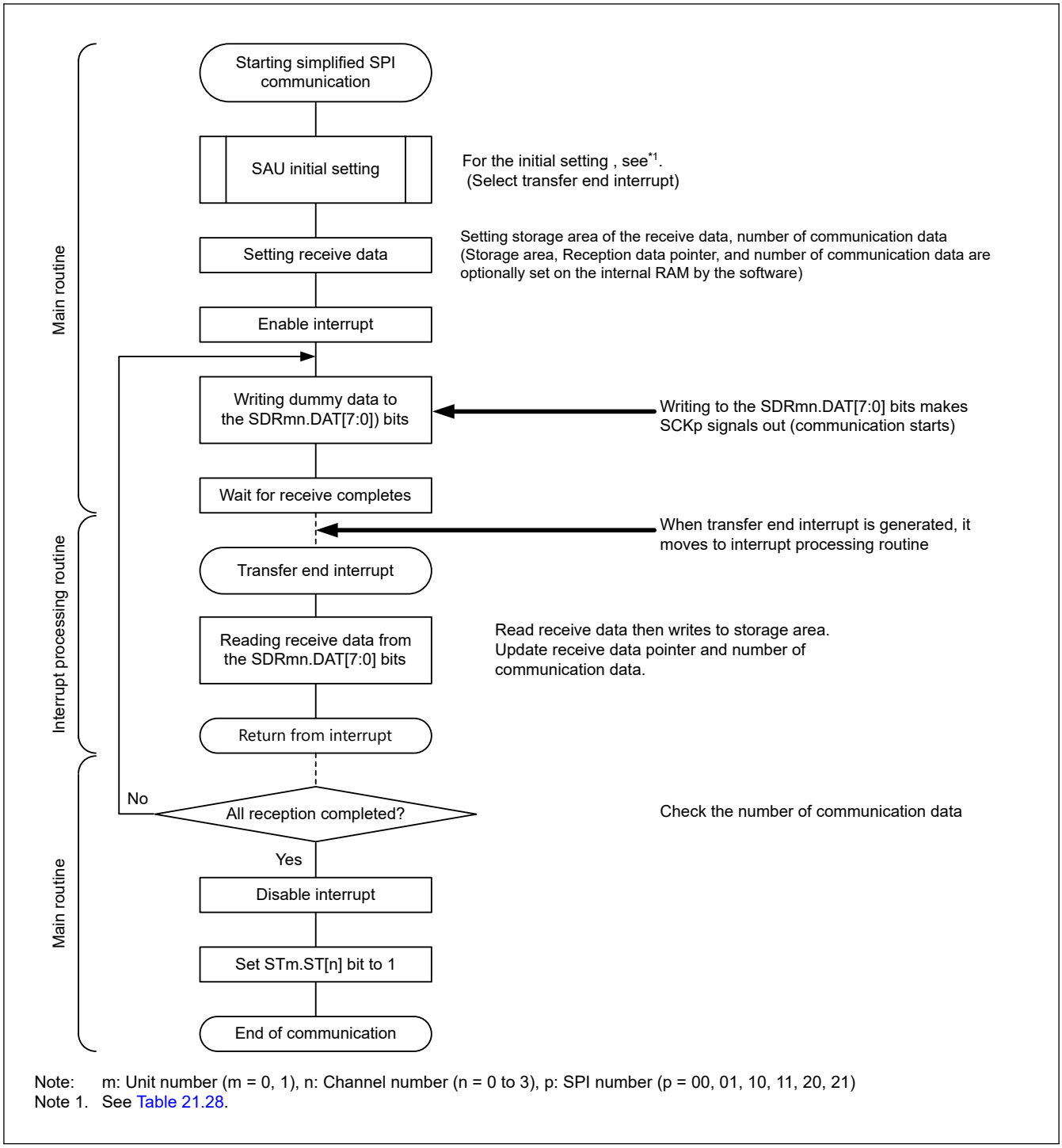
Figure 21.9 shows the timing of master reception (in single reception mode) (Type 1: SCRmn.DCP[1:0] = 00b).



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 21.9 Timing of master reception (in single reception mode) (type 1: SCRmn.DCP[1:0] = 00b)

Figure 21.10 shows the flowchart of master reception (in single reception mode).

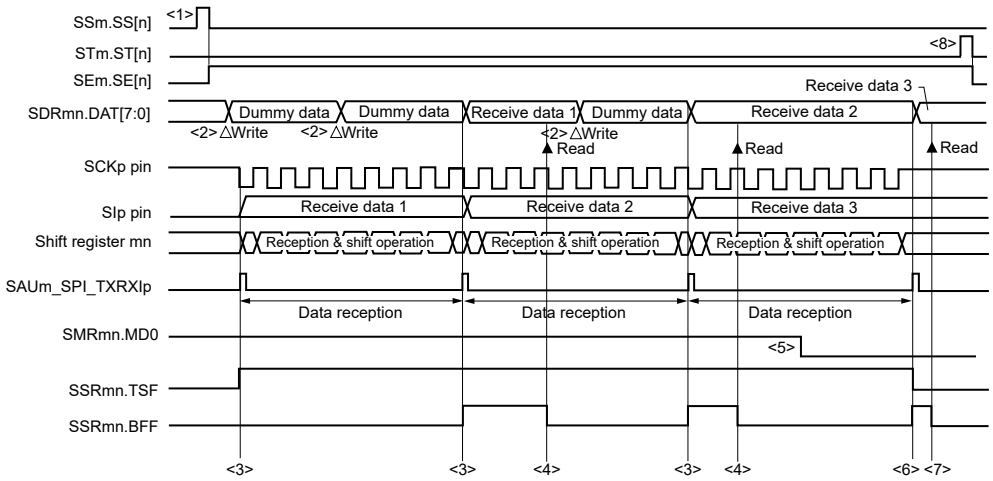


Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: SPI number (p = 00, 01, 10, 11, 20, 21)
Note 1. See Table 21.28.

Figure 21.10 Flowchart of master reception (in single reception mode)

(4) Processing flow (in continuous reception mode)

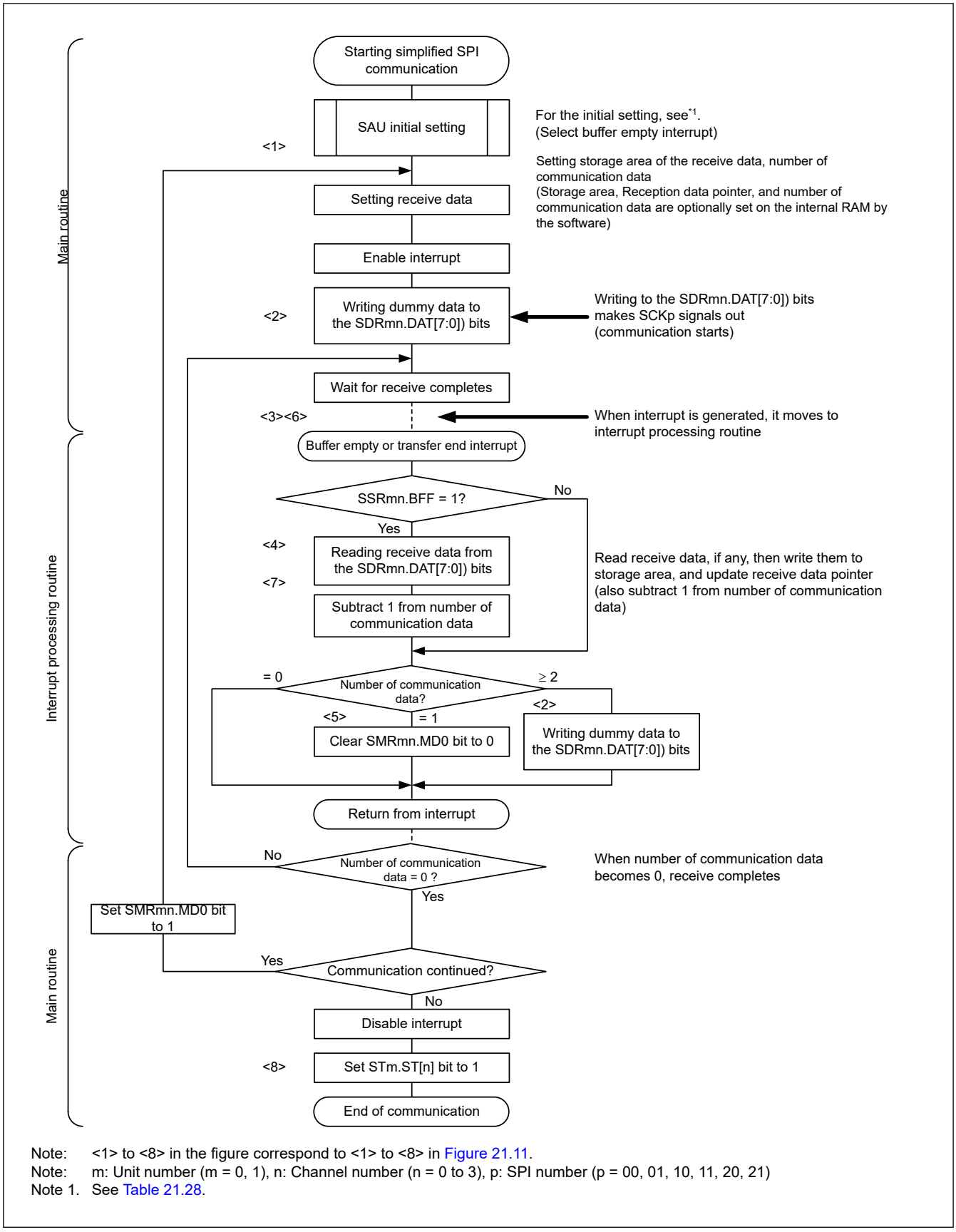
Figure 21.11 shows the timing of master reception (in continuous reception mode) (Type 1: SCRmn.DCP[1:0] = 00b).



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 21.12.
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 21.11 Timing of master reception (in continuous reception mode) (type 1: DCPmn[1:0] = 00b)

Figure 21.12 shows the flowchart of master reception (in continuous reception mode).



Note: <1> to <8> in the figure correspond to <1> to <8> in Figure 21.11.
Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: SPI number (p = 00, 01, 10, 11, 20, 21)
Note 1. See Table 21.28.

Figure 21.12 Flowchart of master reception (in continuous reception mode)

21.5.3 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 21.31 shows the specification for master transmission and reception of Simplified SPI.

Table 21.31 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_SPI_TXRXI00	SAU0_SPI_TXRXI01	SAU0_SPI_TXRXI10	SAU0_SPI_TXRXI11	SAU0_SPI_TXRXI20	SAU0_SPI_TXRXI21
	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] PCLKB: System clock frequency					
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 31, Electrical Characteristics .

(1) Register setting

Table 21.32 to Table 21.37 show examples of the register contents for master transmission and reception of simplified SPI.

(a) Serial mode register mn (SMRmn)

Table 21.32 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 21.32 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 21.33 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 21.3. 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 21.34 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 receive data read)
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 21.35 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 (n = 0 to 3) 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 21.36 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 21.37 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 21.38 shows the procedure for initial setting of master transmission and reception.

Table 21.38 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 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)

Table 21.39 shows the procedure for stopping master transmission and reception.

Table 21.39 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 SSRmn.TSF 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)

Table 21.40 shows the procedure for resuming master transmission and reception.

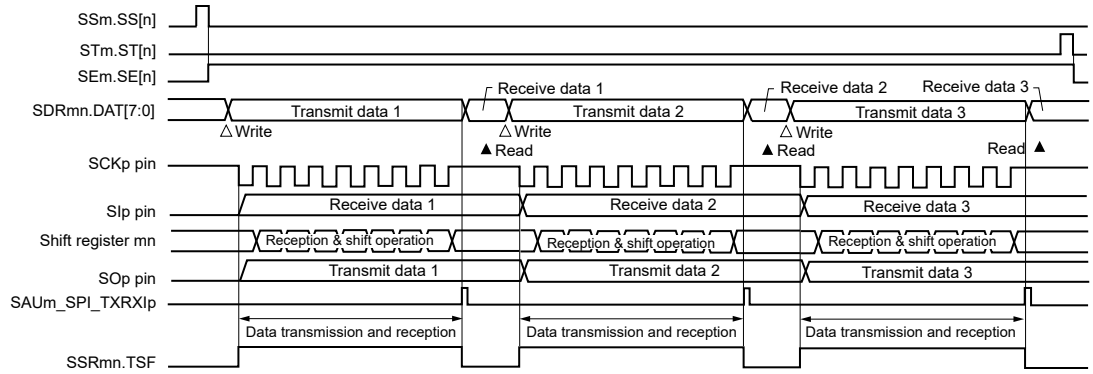
Table 21.40 Procedure for resuming master transmission and reception

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)

(3) Processing flow (in single transmission and reception mode)

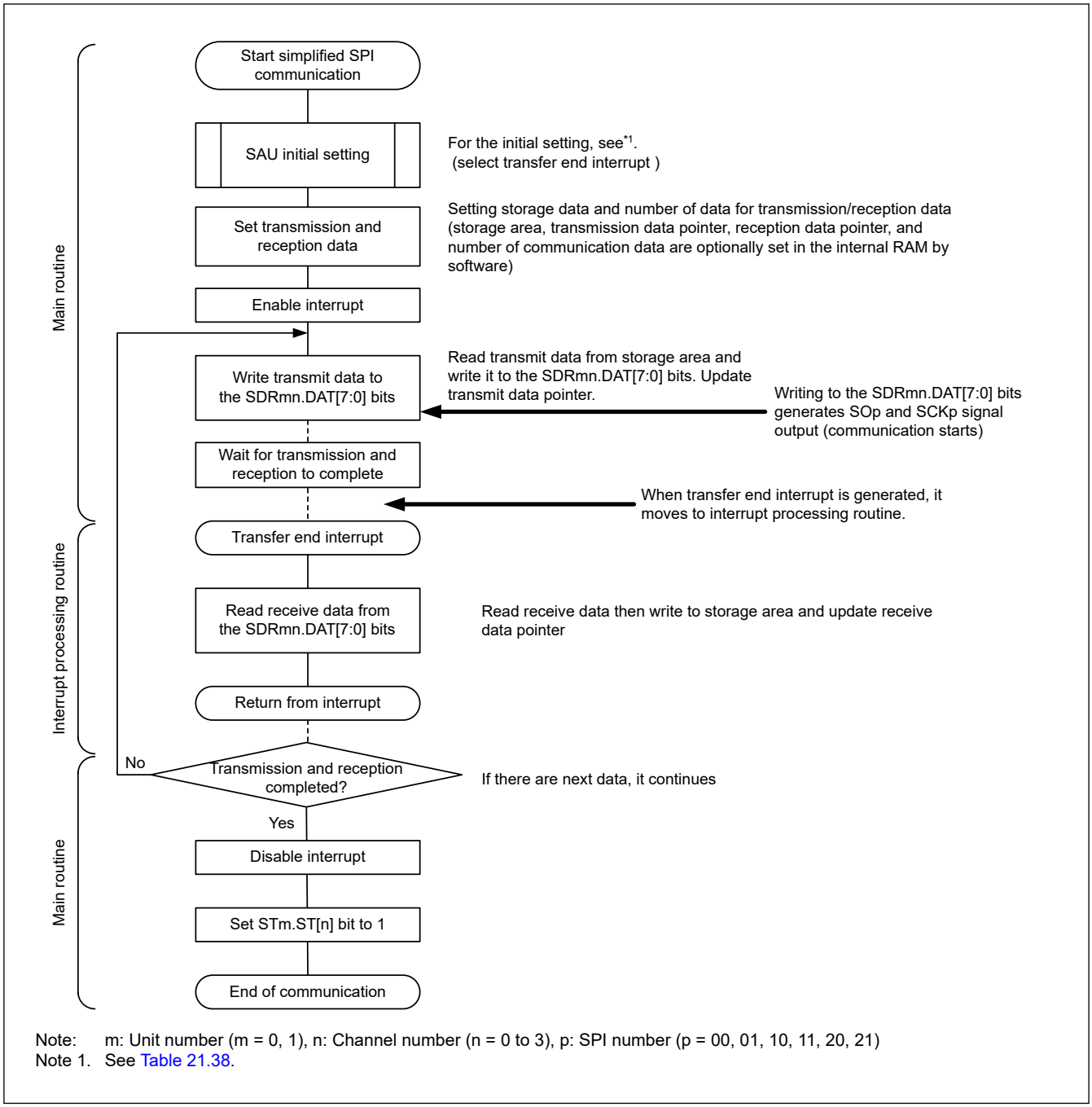
Figure 21.13 shows the timing of master transmission and reception (in single transmission and reception mode) (type 1: SCRmn.DCP[1:0] = 00b).



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 21.13 Timing of master transmission and reception (in single transmission and reception mode) (type 1: SCRmn.DCP[1:0] = 00b)

Figure 21.14 shows the flowchart of master transmission and reception (in single transmission and reception mode).

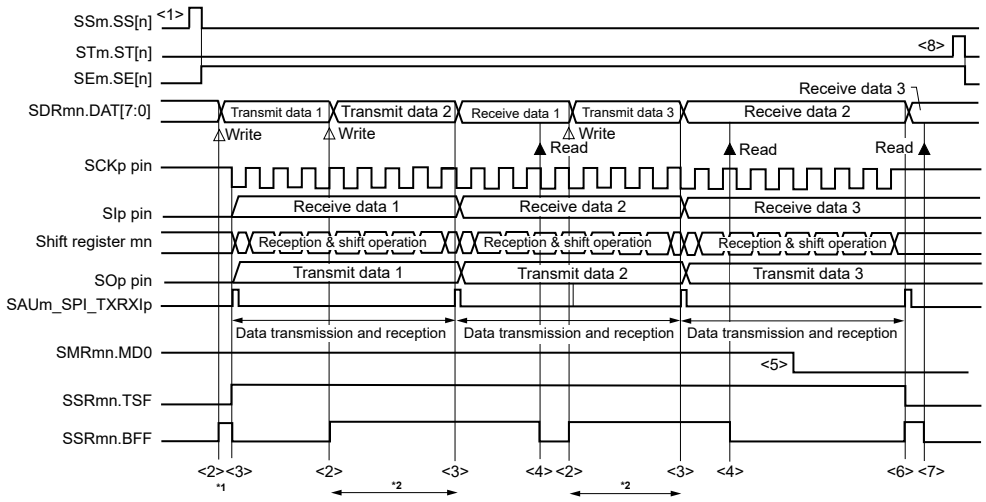


Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: SPI number (p = 00, 01, 10, 11, 20, 21)
Note 1. See Table 21.38.

Figure 21.14 Flowchart of master transmission and reception (in single transmission and reception mode)

(4) Processing flow (in continuous transmission and reception mode)

Figure 21.15 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 21.16.

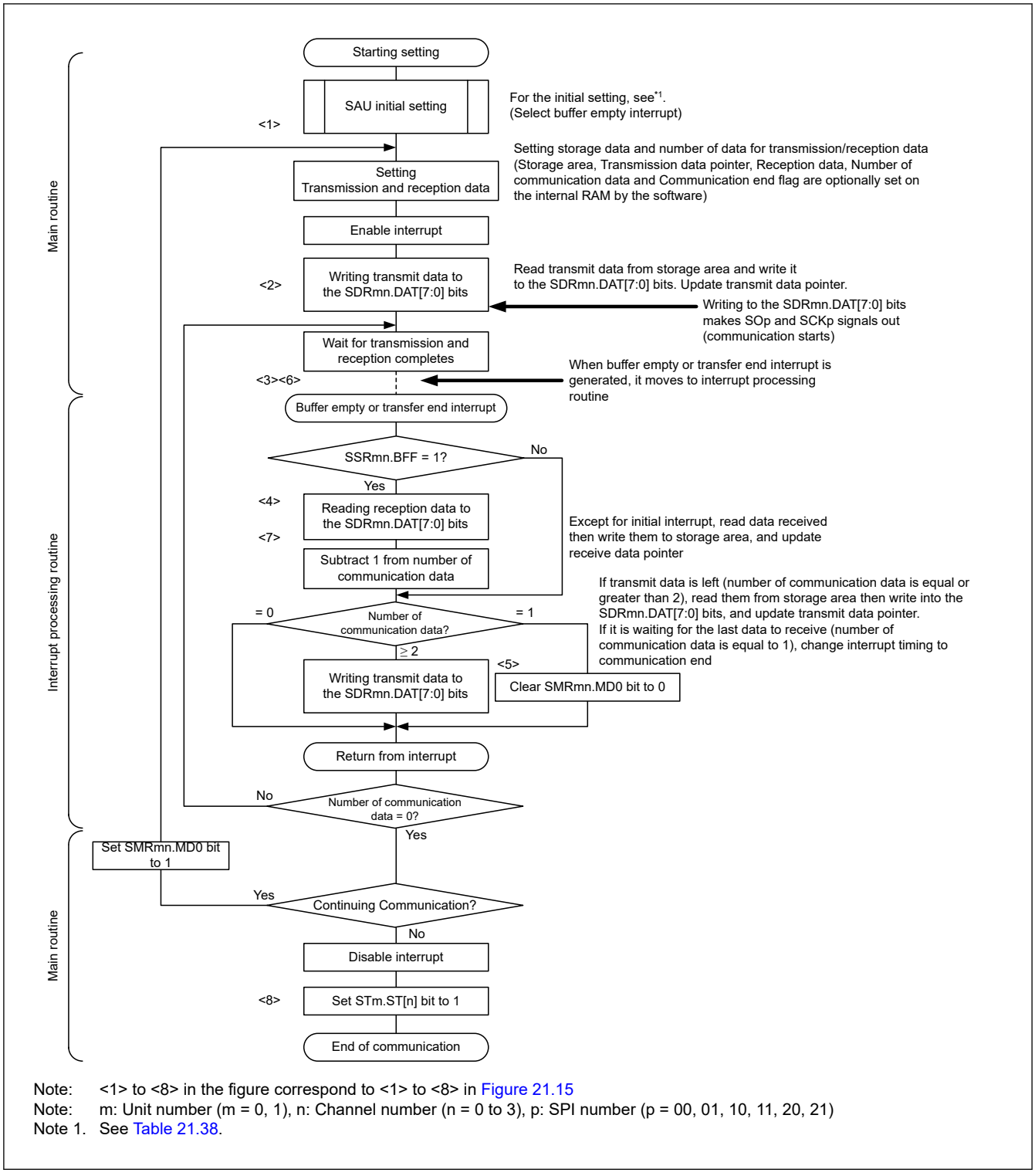
Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: SPI number (p = 00, 01, 10, 11, 20, 21) mn = 00 to 03, 10 to 11

Note 1. If transmit data is written to the SDRmn register while the BFF bit 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 21.15 Timing of master transmission and reception (in continuous transmission and reception mode) (type 1: SCRmn.DCP[1:0] = 00b)

Figure 21.16 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 21.15
Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: SPI number (p = 00, 01, 10, 11, 20, 21)
Note 1. See Table 21.38.

Figure 21.16 Flowchart of master transmission and reception (in continuous transmission and reception mode

21.5.4 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 21.41 shows the specification of slave transmission of simplified SPI.

Table 21.41 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_SPI_TXRXI00	SAU0_SPI_TXRXI01	SAU0_SPI_TXRXI10	SAU0_SPI_TXRXI11	SAU1_SPI_TXRXI20	SAU1_SPI_TXRXI21
	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, SCK11, and SCK20 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 31, Electrical Characteristics .

(1) Register setting

Table 21.42 to Table 21.47 show examples of the register contents for slave transmission of simplified SPI.

(a) Serial mode register mn (SMRmn)

Table 21.42 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 21.42 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 21.43 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 21.3. 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 21.44 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 21.45 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 21.46 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 21.47 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), mn = 00 to 03, 10 to 11

Note: ×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Table 21.48 shows the procedure for initial setting of slave transmission.

Table 21.48 Initial setting procedure for slave transmission

Step		Process	Detail
Procedure for initial setting of slave transmission	<1>	Starting initial setting	—
	<3>	Setting the SPSm register	Set the operation clock.
	<4>	Setting the SMRmn register	Set an operation mode.
	<5>	Setting the SCRmn register	Set a communication format.
	<6>	Setting the SDRmn register	Set the SDRmn.STCLK[6:0] bits to 0x00 for baud rate setting.
	<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 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 and wait for a clock from the master

Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

Table 21.49 shows the procedure for stopping slave transmission.

Table 21.49 Procedure for stopping slave transmission

Step		Process	Detail
Procedure for stopping slave transmission	<1>	Starting setting to stop	—
	<2>	Wait until the SSRmn.TSF bit 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)

Table 21.50 shows the procedure for resuming slave transmission.

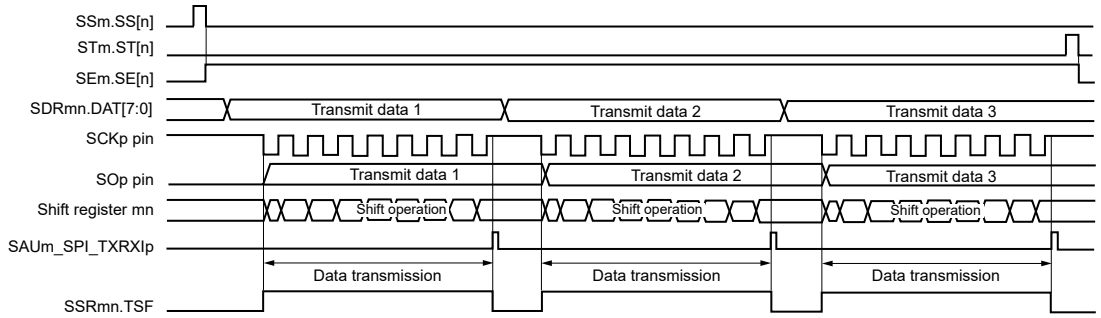
Table 21.50 Procedure for resuming slave transmission

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 SEm.SE[n] to 1 to enable operation.
	<14>	Starting communication	Sets transmit data to set the SEm.SE[n] bit 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)

(3) Processing flow (in single transmission mode)

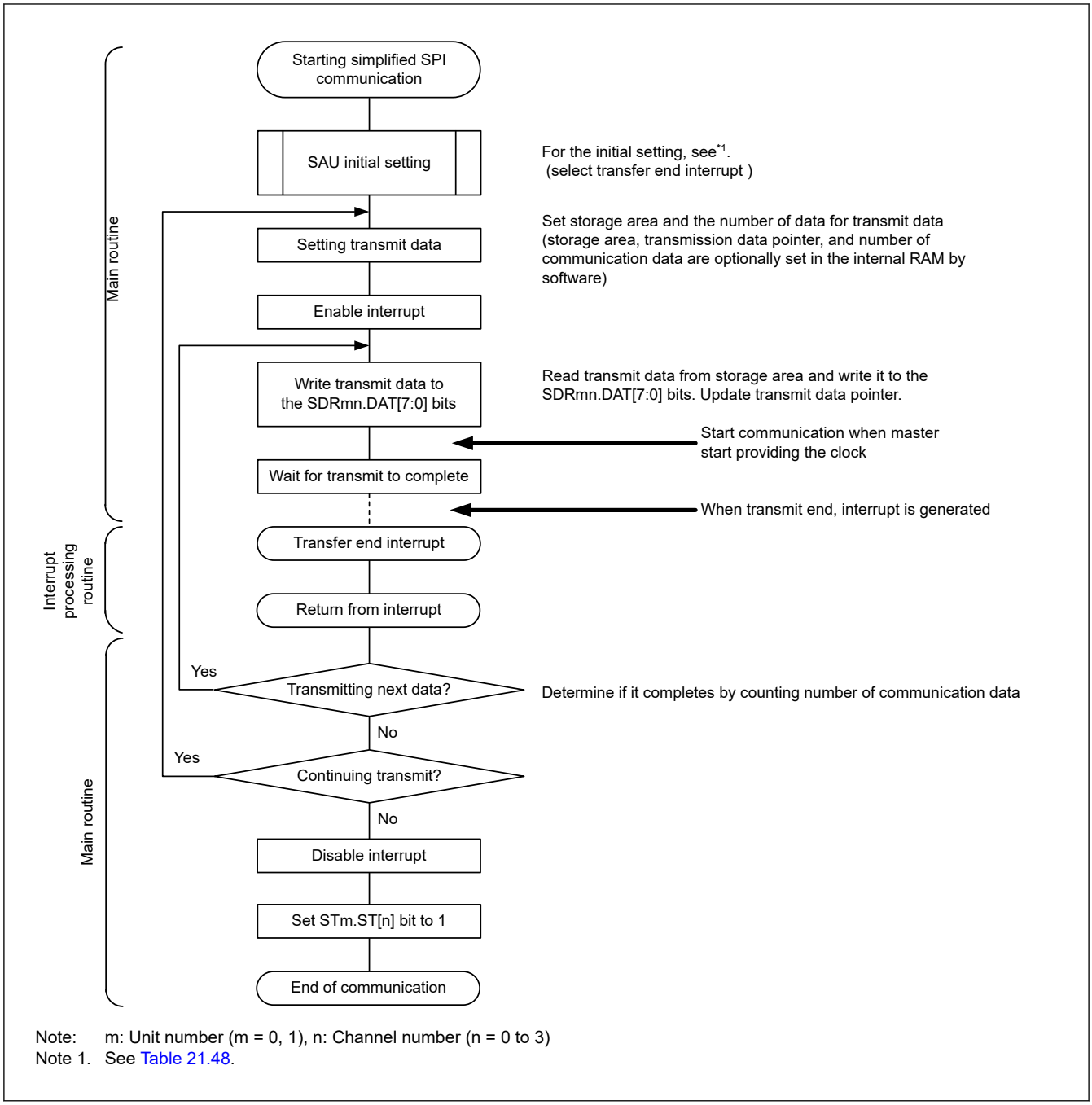
Figure 21.17 shows the timing of slave transmission (in single transmission mode) (type 1: SCRmn.DCP[1:0] = 00b).



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 21.17 Timing of slave transmission (in single transmission mode) (type 1: SCRmn.DCP[1:0] = 00b)

Figure 21.18 shows the flowchart of slave transmission (in single transmission mode).

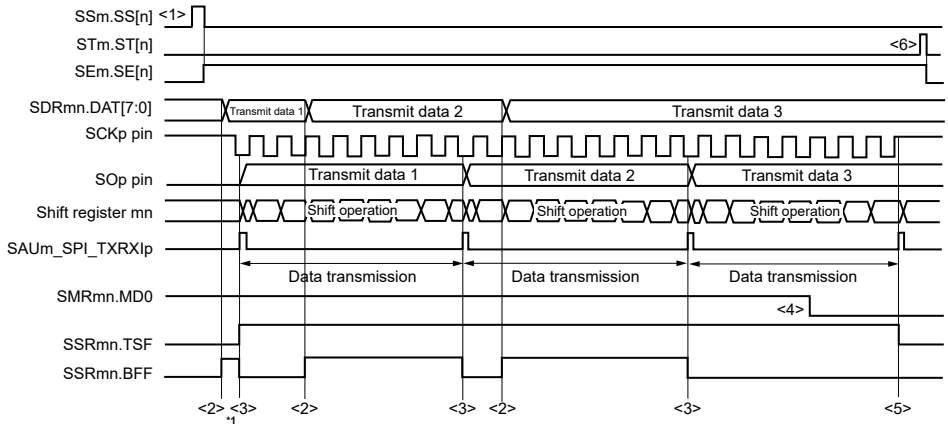


Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)
Note 1. See Table 21.48.

Figure 21.18 Flowchart of slave transmission (in single transmission mode)

(4) Processing flow (in continuous transmission mode)

Figure 21.19 shows the timing of slave transmission (in continuous transmission mode) (type 1: SCRmn.DCP[1:0] = 00b).



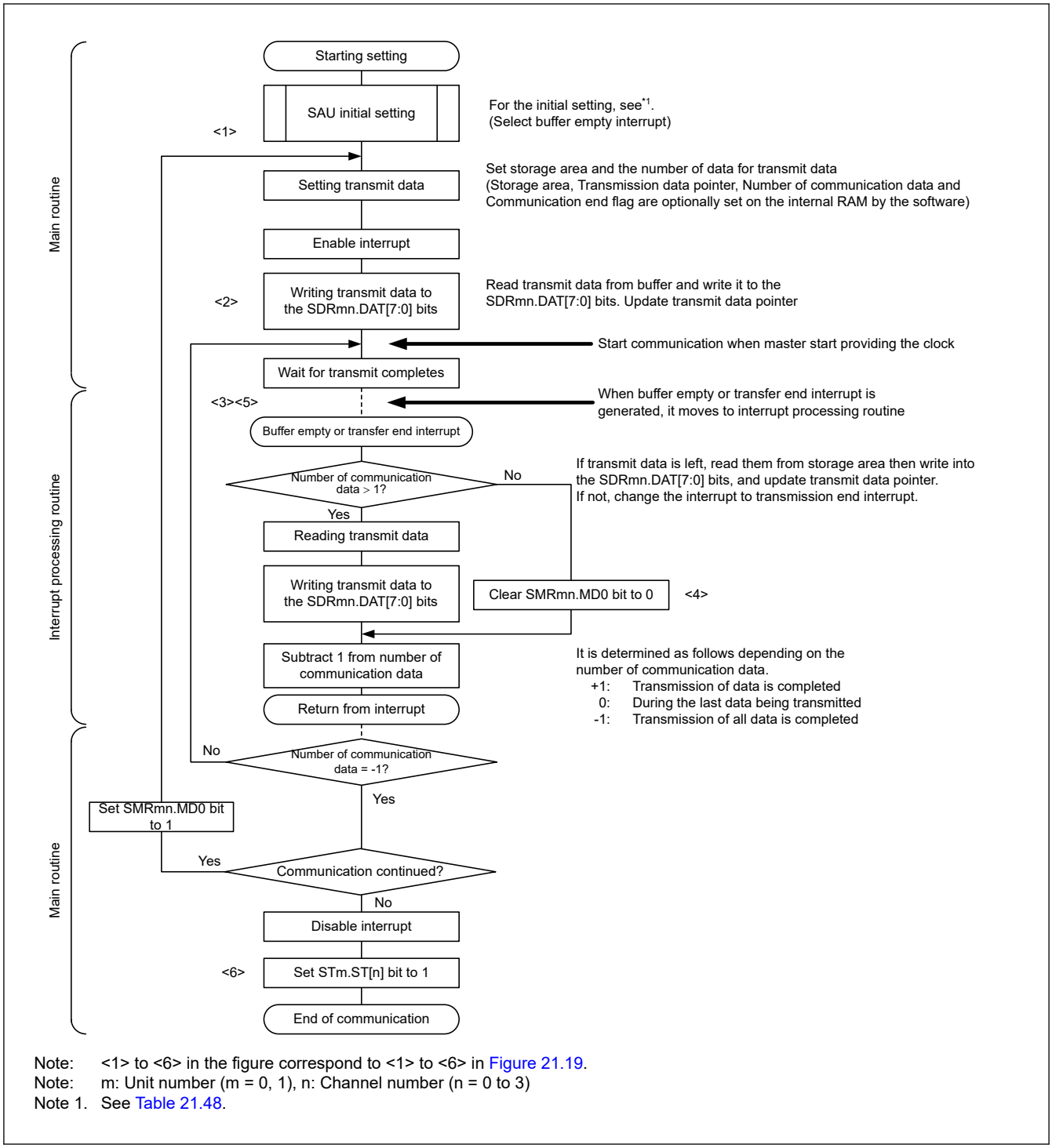
Note: The MD0 bit of serial mode register mn (SMRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started.

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 bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

Figure 21.19 Timing of slave transmission (in continuous transmission mode) (type 1: SCRmn.DCP[1:0] = 00b)

Figure 21.20 shows the flowchart of slave transmission (in continuous transmission mode).



Note: <1> to <6> in the figure correspond to <1> to <6> in Figure 21.19.
Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)
Note 1. See Table 21.48.

Figure 21.20 Flowchart of slave transmission (in continuous transmission mode)

21.5.5 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 21.51 shows the specification of slave reception of simplified SPI.

Table 21.51 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_SPI_TXRXI00	SAU0_SPI_TXRXI01	SAU0_SPI_TXRXI10	SAU0_SPI_TXRXI11	SAU1_SPI_TXRXI20	SAU1_SPI_TXRXI21
	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, SCK11, and SCK20 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 31, Electrical Characteristics .

(1) Register setting

Table 21.52 to Table 21.57 show examples of the register contents for slave reception of simplified SPI.

(a) Serial mode register mn (SMRmn)

Table 21.52 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 21.53 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 21.3. 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 21.54 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 21.55 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 21.56 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 21.57 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), mn = 00 to 03, 10 to 11

Note: ×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Table 21.58 shows the procedure for initial setting of slave reception.

Table 21.58 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 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)

Table 21.59 shows the procedure for stopping slave reception.

Table 21.59 Procedure for stopping slave reception

Step		Process	Detail
Procedure for stopping slave reception	<1>	Start setting to stop	—
	<2>	Wait until the SSRmn.TSF bit 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)

Table 21.60 shows the procedure for resuming slave reception.

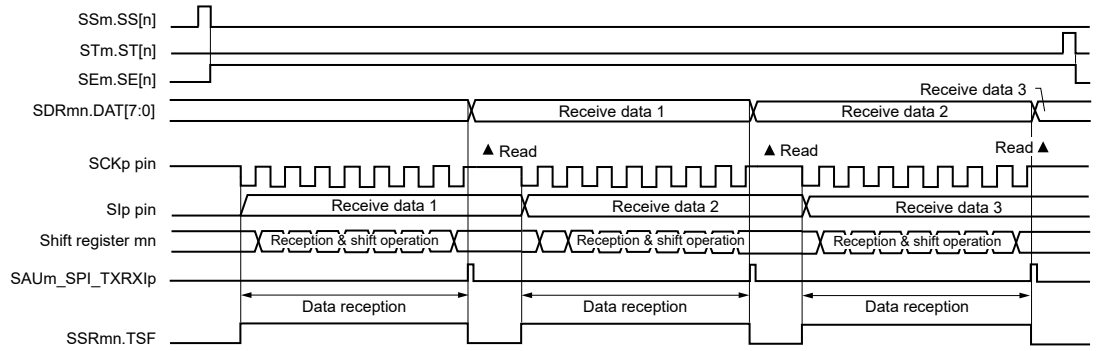
Table 21.60 Procedure for resuming slave transmission

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 output and clock output 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 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] bit = 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)

(3) Processing flow (in single reception mode)

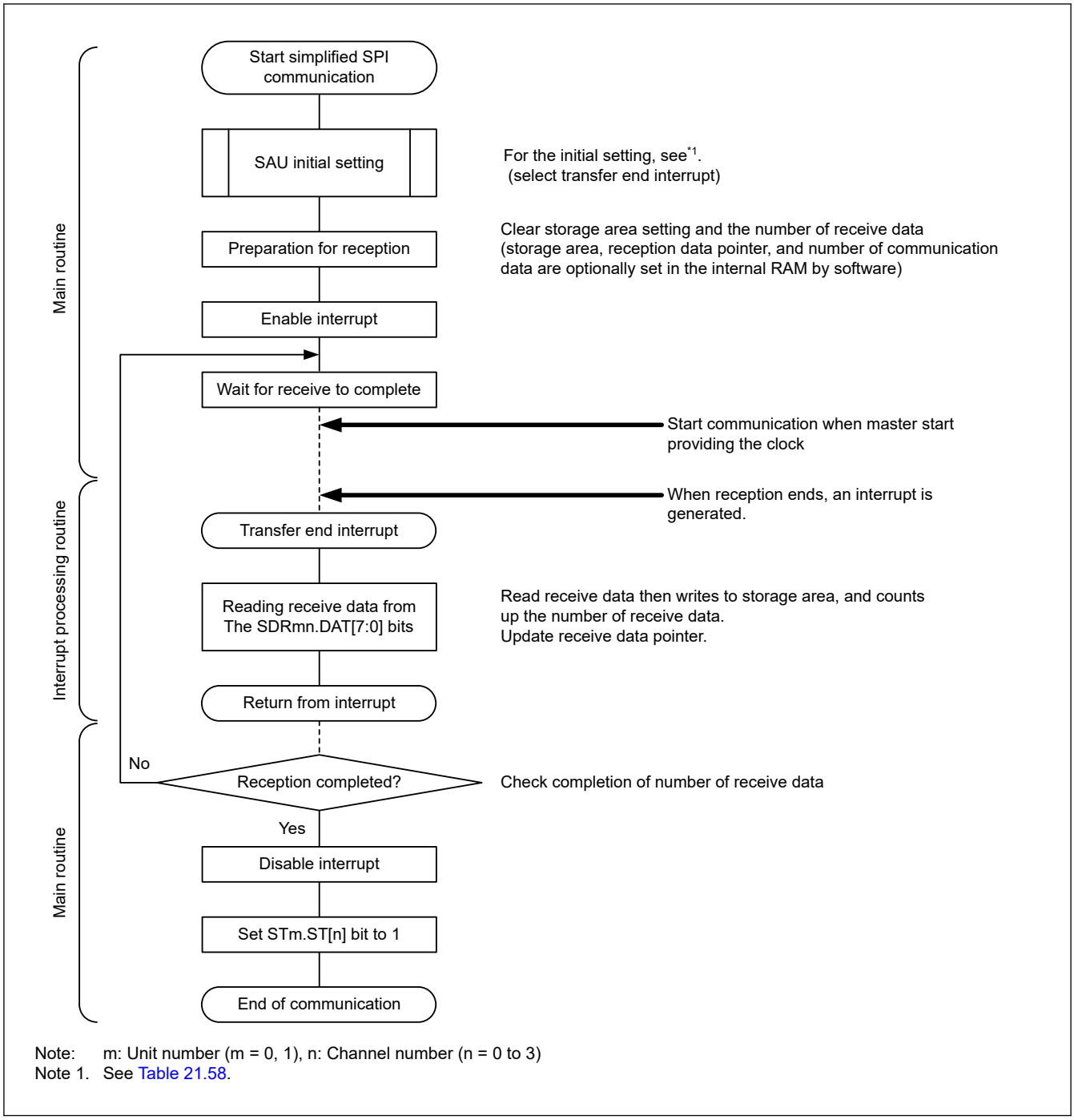
Figure 21.21 shows the timing of slave reception (in single reception mode) (type 1: SCRmn.DCP[1:0] = 00b).



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 21.21 Timing of slave reception (in single reception mode) (type 1: SCRmn.DCP[1:0] = 00b)

Figure 21.22 shows the flowchart of slave reception (in single reception mode).



Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)
Note 1. See Table 21.58.

Figure 21.22 Flowchart of slave reception (in single reception mode)

21.5.6 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 21.61 shows the specification of slave transmission and reception of simplified SPI.

Table 21.61 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 21.61 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_SPI_TXRXI00	SAU0_SPI_TXRXI01	SAU0_SPI_TXRXI10	SAU0_SPI_TXRXI11	SAU1_SPI_TXRXI20	SAU1_SPI_TXRXI21
	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 31, Electrical Characteristics .

(1) Register setting

Table 21.62 to Table 21.67 show examples of the register contents for slave transmission and reception of simplified SPI.

(a) Serial mode register mn (SMRmn)

Table 21.62 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 21.63 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 21.3. 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 21.64 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 21.65 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 n (n = 0 to 3) 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 21.66 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 21.67 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 SDRmn.DAT[7:0] bits 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: ×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Table 21.68 shows the procedure for initial setting of slave transmission and reception.

Table 21.68 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] bit = 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 SDRmn.DAT[7:0] bits before the clock from the master is started.
Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

Table 21.69 shows the procedure for stopping slave transmission and reception.

Table 21.69 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 SSRmn.TSF 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)

Table 21.70 shows the procedure for resuming slave transmission and reception.

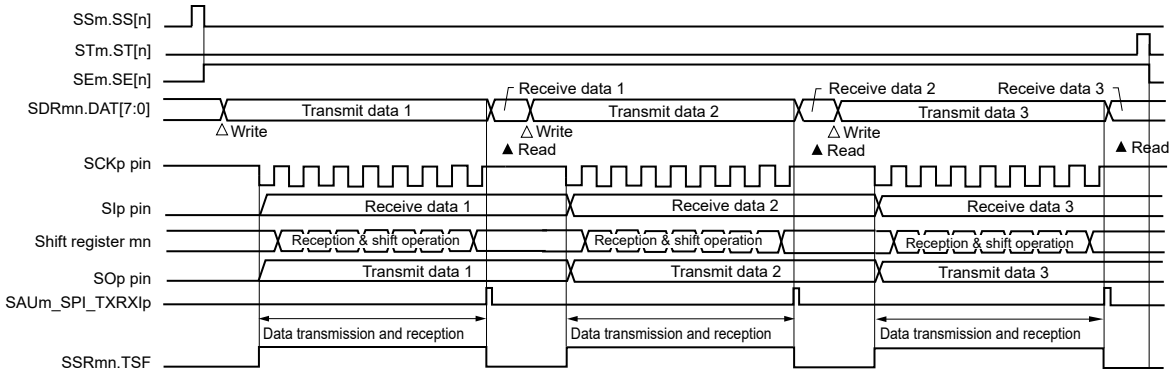
Table 21.70 Procedure for resuming slave transmission and reception

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 bit 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 SDRmn.DAT[7:0] bits before the clock from the master is started.
Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

(3) Processing flow (in single transmission and reception mode)

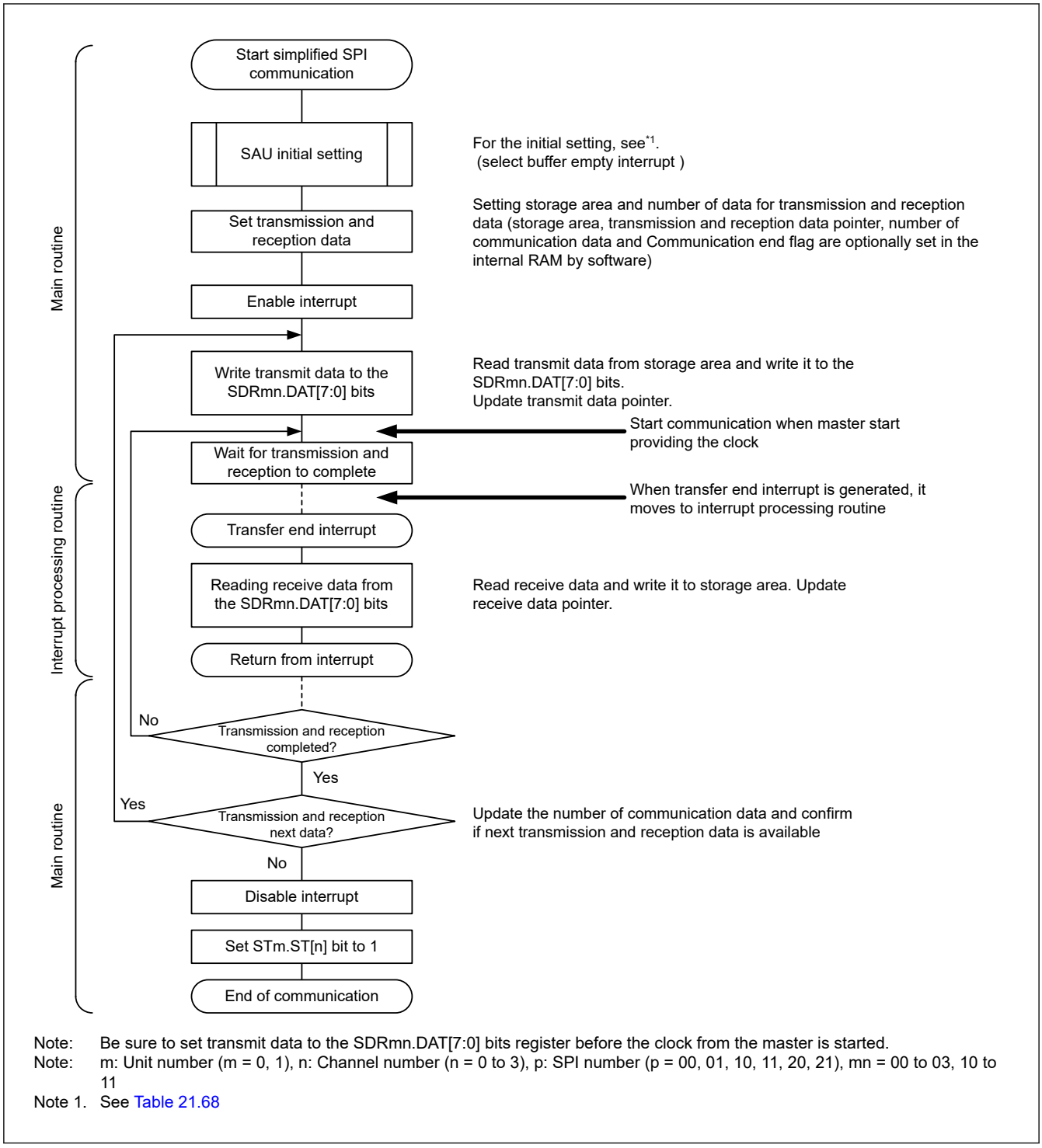
Figure 21.23 shows the timing of slave transmission and reception (in single transmission and reception mode) (type 1: SCRmn.DCP[1:0] = 00b).



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 21.23 Timing of slave transmission and reception (in single transmission and reception mode)

Figure 21.24 shows the flowchart of slave transmission and reception (in single transmission and reception mode).

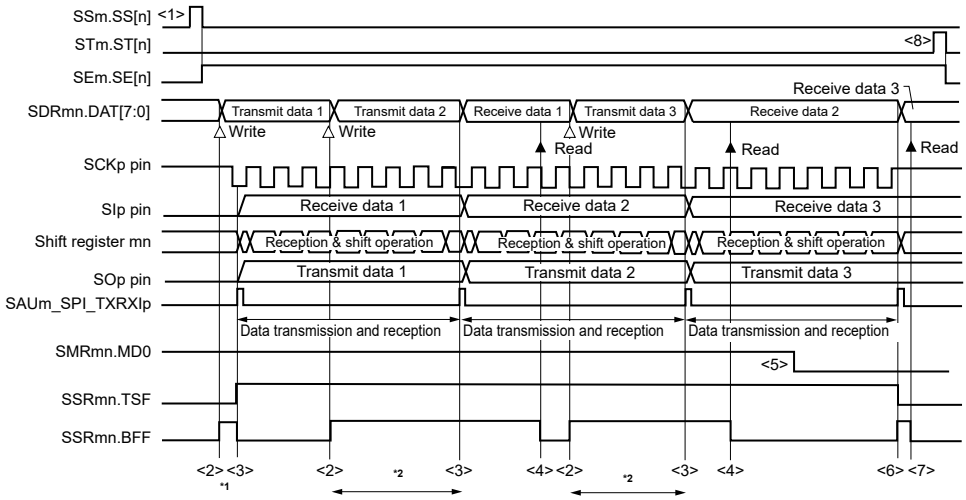


Note: Be sure to set transmit data to the SDRmn.DAT[7:0] bits register before the clock from the master is started.
Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: SPI number (p = 00, 01, 10, 11, 20, 21), mn = 00 to 03, 10 to 11
Note 1. See Table 21.68

Figure 21.24 Flowchart of slave transmission and reception (in single transmission and reception mode)

(4) Processing flow (in continuous transmission and reception mode)

Figure 21.25 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 21.26.

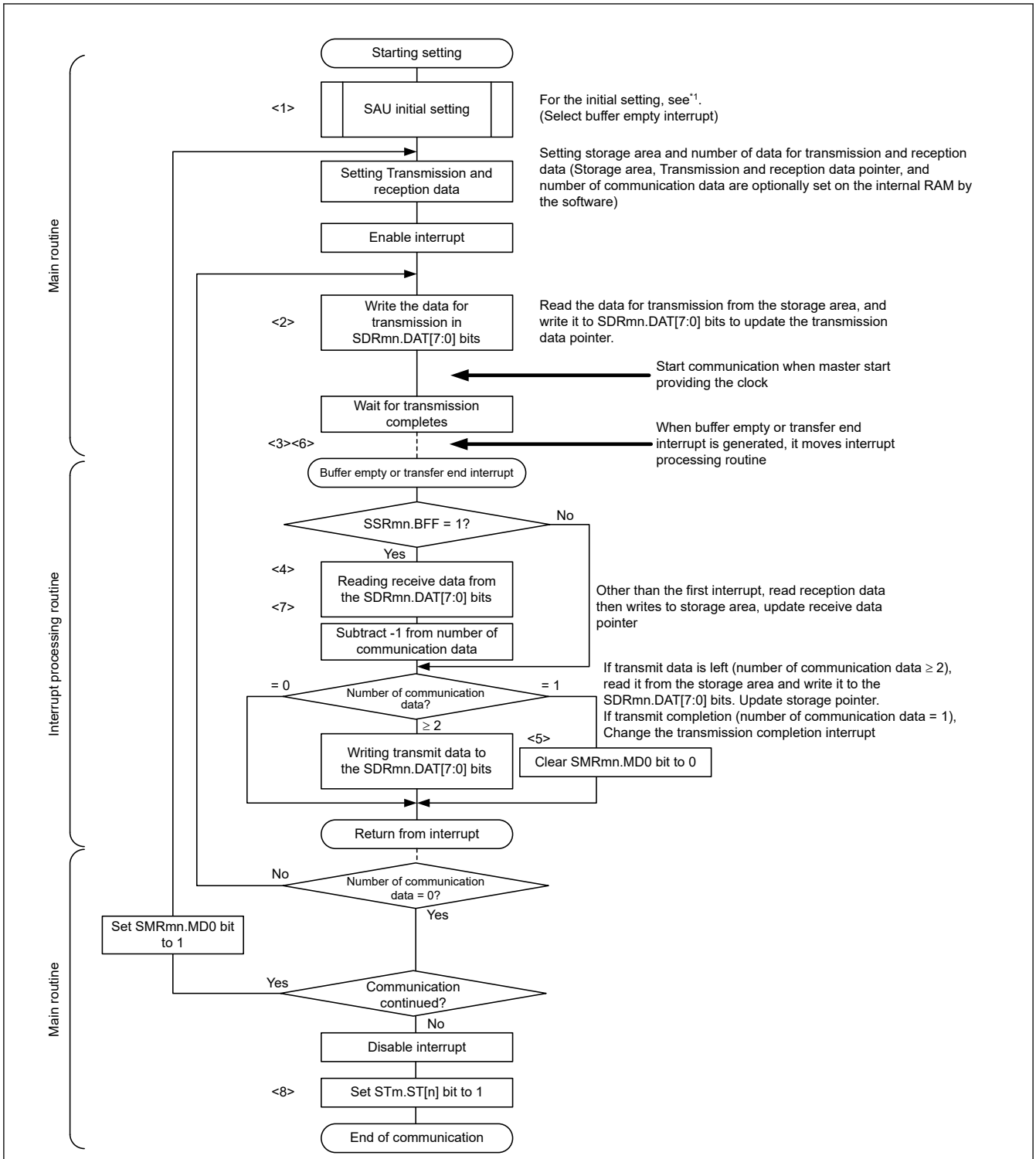
Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: SPI number (p = 00, 01, 10, 11, 20, 21) mn = 00 to 03, 10 to 11

Note 1. If transmit data is written to the SDRmn register while the BFF bit 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 21.25 Timing of slave transmission and reception (in continuous transmission and reception mode) (type 1: SCRmn.DCP[1:0] = 00b)

Figure 21.26 shows the flowchart of slave transmission and reception (in continuous transmission and reception mode).



Note: Be sure to set transmit data to the SDRmn.DAT[7:0] bits before the clock from the master is started.
Note: <1> to <8> in the figure correspond to <1> to <8> in Figure 21.25
Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)
Note 1. See Table 21.68.

Figure 21.26 Flowchart of slave transmission and reception (in continuous transmission and reception mode)

21.5.7 Snooze Mode Function

The Snooze mode makes the simplified SPI perform reception operations on SCK00 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 SCK00 pin input. Only SPI00 channel 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 21.28 and Figure 21.30.)

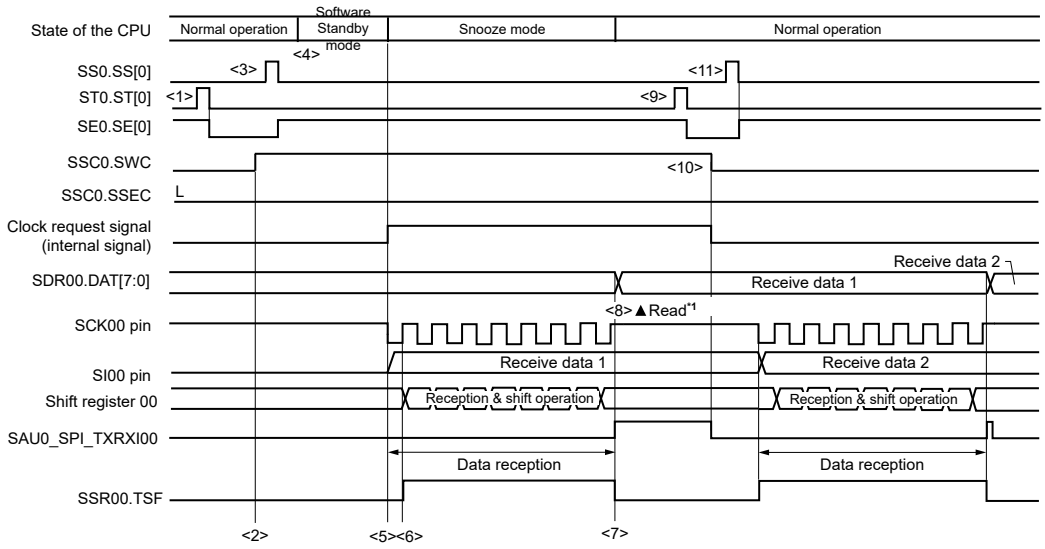
- When using the Snooze mode function, set the SWC bit of serial standby control register 0 (SSC0) to 1 just before switching to the Software Standby mode. After the initial setting has been completed, set the SS[0] bit of serial channel start register 0 (SS0) to 1.
- The CPU shifts to the Snooze mode on detecting the valid edge of the SCK00 signal following a transition to the Software Standby mode.
  An SPI00 starts reception on detecting input of the serial clock on the SCK00 pin.

Note: The Snooze mode can only be specified when the high-speed on-chip oscillator clock or middle-speed on-chip oscillator clock is selected for PCLKB.

Note: The maximum transfer rate when using SPI00 in the Snooze mode is 1 Mbps.

(1) Snooze mode operation (on startup)

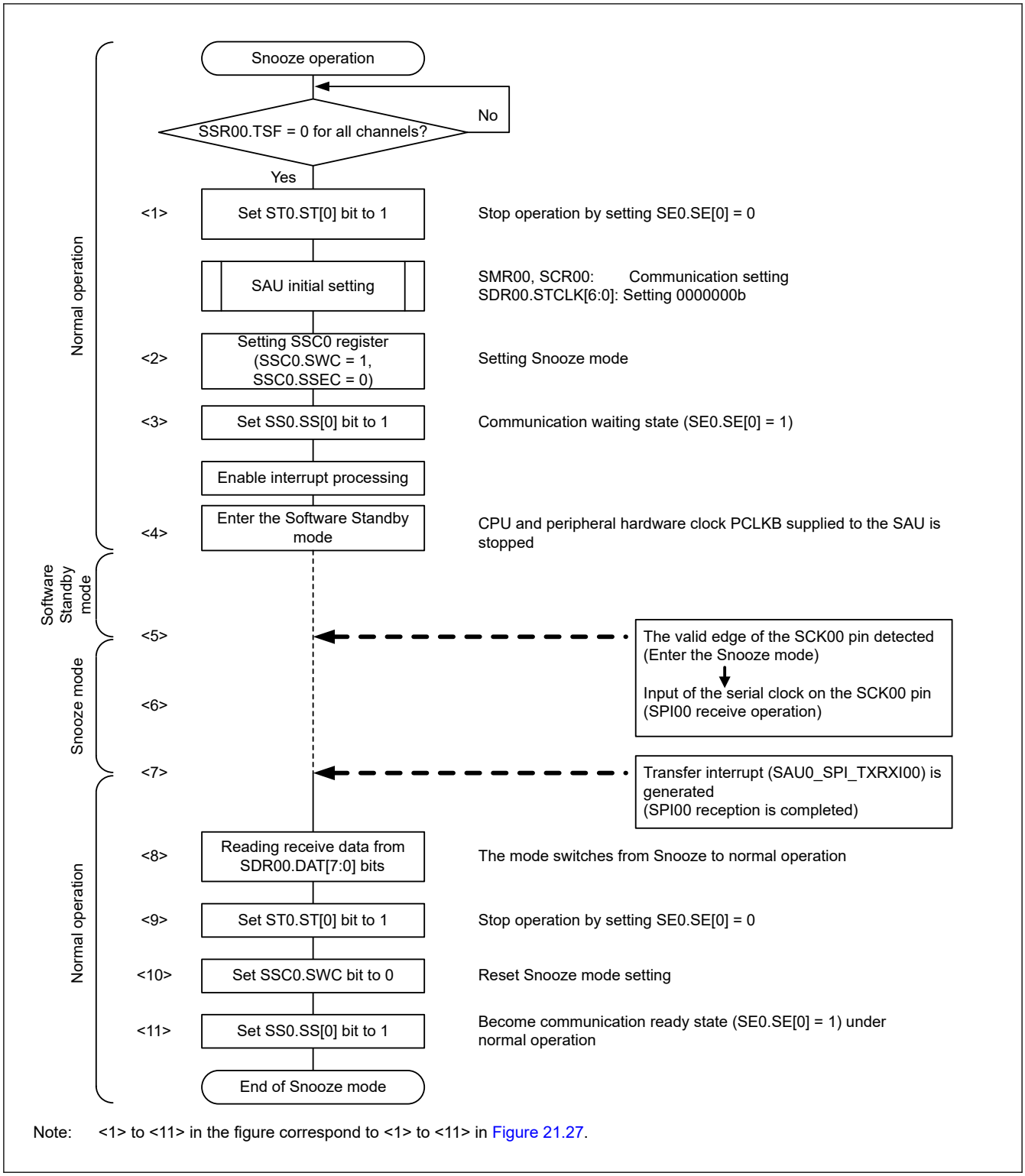
Figure 21.27 shows the timing of Snooze mode operation (on startup) (Type 1: SCR00.DCP[1:0] = 00b).



Note: Before switching to the Snooze mode or after reception operation in the Snooze mode finishes, set the ST0.ST[0] bit to 1 (the SE0.SE[0] bit is cleared and the operation stops). After the receive operation completes, clear the SSC0.SWC bit to 0 (Snooze mode release).
Note: When SSC0.SWC = 1, the SSR00.BFF and OVF flags do not change.
Note: <1> to <11> in the figure correspond to <1> to <11> in Figure 21.28.
Note 1. Only read received data while SSC0.SWC = 1 and before the next valid edge of the SCK00 pin input is detected.

Figure 21.27 Timing of Snooze mode operation (on startup) (type 1: SCR00.DCP[1:0] = 00b)

Figure 21.28 shows the flowchart of Snooze mode operation (on startup).

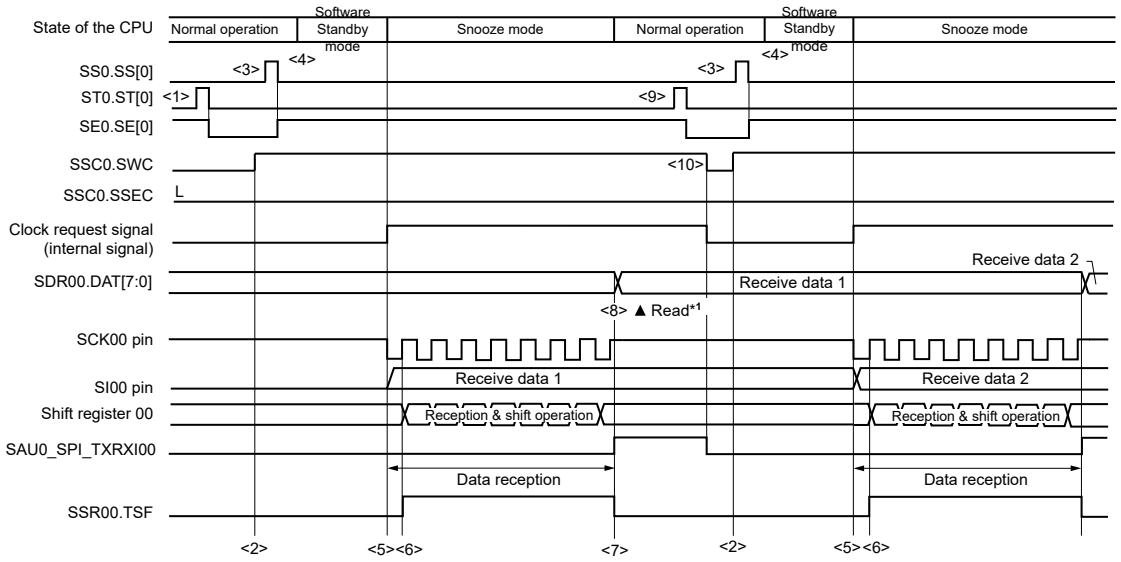


Note: <1> to <11> in the figure correspond to <1> to <11> in Figure 21.27.

Figure 21.28 Flowchart of Snooze mode operation (on startup)

(2) Snooze mode operation (continuous startup)

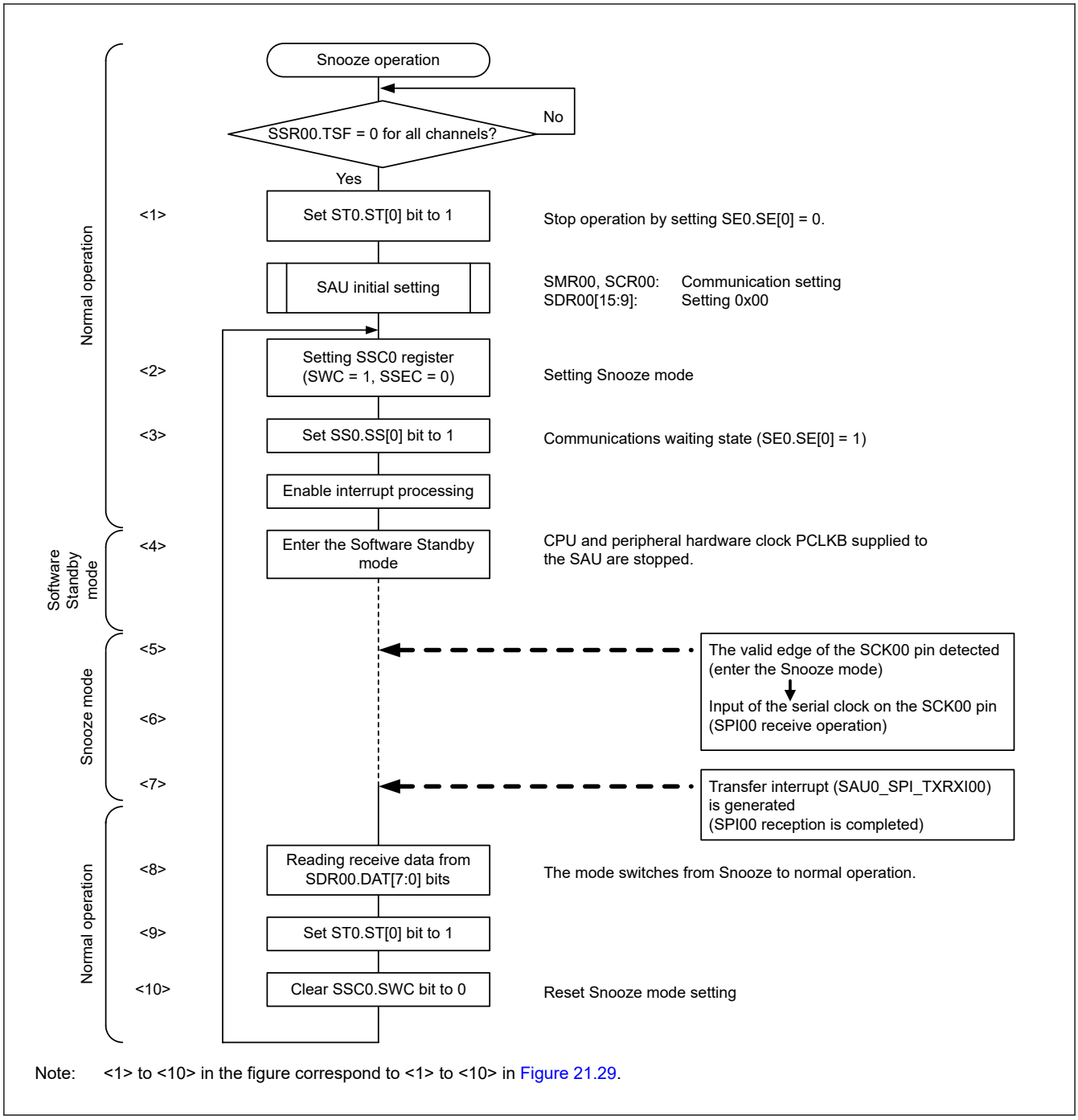
Figure 21.29 shows the timing of Snooze mode operation (continuous startup) (type 1: SCR00.DCP[1:0] = 00b).



Note: Before switching to the Snooze mode or after reception operation in the Snooze mode finishes, set the ST0.ST[0] bit to 1 (the SE0.SE[0] bit is cleared and the operation stops). After the receive operation completes, clear the SSC0.SWC bit to 0 (Snooze mode release).
Note: When SSC0.SWC = 1, the SSR00.BFF and OVF flags do not change.
Note: <1> to <10> in the figure correspond to <1> to <10> in Figure 21.30.
Note 1. Only read received data while SSC0.SWC = 1 and before the next valid edge of the SCK00 pin input is detected.

Figure 21.29 Timing of Snooze mode operation (continuous startup) (type 1: SCR00.DCP[1:0] = 00b)

Figure 21.30 shows the flowchart of Snooze mode operation (continuous startup).



Note: <1> to <10> in the figure correspond to <1> to <10> in Figure 21.29.

Figure 21.30 Flowchart of Snooze mode operation (continuous startup)

21.5.8 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 21.71.

Table 21.71 Selection of operation clock for simplified SPI, UART and simplified I²C

SMRmn register	SPSm register		Operation clock (f_{MCK}) *1	
CKS	PRS1[3:0]*2	PRS0[3:0]*2	PCLKB/2^n	PCLKB = 32 MHz
0	Don't care	0x0	PCLKB	32 MHz
		0x1	PCLKB/2	16 MHz
		0x2	PCLKB/2^2	8 MHz
		0x3	PCLKB/2^3	4 MHz
		0x4	PCLKB/2^4	2 MHz
		0x5	PCLKB/2^5	1 MHz
		0x6	PCLKB/2^6	500 kHz
		0x7	PCLKB/2^7	250 kHz
		0x8	PCLKB/2^8	125 kHz
		0x9	PCLKB/2^9	62.5 kHz
		0xA	PCLKB/2^10	31.25 kHz
		0xB	PCLKB/2^11	15.63 kHz
		0xC	PCLKB/2^12	7.81 kHz
		0xD	PCLKB/2^13	3.91 kHz
		0xE	PCLKB/2^14	1.95 kHz
		0xF	PCLKB/2^15	977 Hz
1	0x0	Don't care	PCLKB	32 MHz
	0x1		PCLKB/2	16 MHz
	0x2		PCLKB/2^2	8 MHz
	0x3		PCLKB/2^3	4 MHz
	0x4		PCLKB/2^4	2 MHz
	0x5		PCLKB/2^5	1 MHz
	0x6		PCLKB/2^6	500 kHz
	0x7		PCLKB/2^7	250 kHz
	0x8		PCLKB/2^8	125 kHz
	0x9		PCLKB/2^9	62.5 kHz
	0xA		PCLKB/2^10	31.25 kHz
	0xB		PCLKB/2^11	15.63 kHz
	0xC		PCLKB/2^12	7.81 kHz
	0xD		PCLKB/2^13	3.91 kHz
	0xE		PCLKB/2^14	1.95 kHz
	0xF		PCLKB/2^15	977 kHz

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 I²C mode, setting the value greater than 0xB is prohibited.

21.5.9 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 21.72.

Table 21.72 Processing procedure in case of overrun error

Step	Software Manipulation		State of the Hardware	Remark
<1>	Reads serial data register mn (SDRmn).	→	The BFF bit 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

21.6 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 communication 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 0 (channel 7), and an external interrupt (IRQ0).

[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 and buffer empty interrupt (SAU0_UART_TXI0/SAU0_UART_RXI0/SAU0_UART_TXI1/ SAU0_UART_RXI1/SAU1_UART_TXI2/SAU1_UART_RXI2)
- Error interrupt in case of framing error, parity error, or overrun error (SAU0_UART_ERRI0/SAU0_UART_ERRI1/ SAU1_UART_ERRI2)

[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, 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]

LIN-bus functions are achieved using the external interrupt (IRQ0) and timer array unit 0 (channel 7).

- Wakeup signal detection
- Break field (BF) detection
- Sync field measurement, baud rate calculation

Note 1. Only UART0 and UART2 support the 9-bit data length.

When the medium-speed on-chip oscillator clock (MOCO) or low-speed on-chip oscillator clock (LOCO) is selected for PCLKB, use the medium-speed on-chip oscillator trimming register (MIOTRM) and low-speed on-chip oscillator trimming register (LIOTRM) to correct oscillation frequency accuracy.

- 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 21.1 and Table 21.2.

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 21.6.1. UART Transmission.)
- UART reception (See section 21.6.2. UART Reception.)
- LIN transmission (UART2 only) (See section 21.7.1. LIN Transmission .)
- LIN reception (UART2 only) (See section 21.7.2. LIN Reception .)

21.6.1 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 21.73 shows the specification of UART transmission.

Table 21.73 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_UART_TXI0	SAU0_UART_TXI1	SAU1_UART_TXI2
	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 UART0 and UART2 support the 9-bit data length.

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 31, Electrical Characteristics .

(1) Register setting

Table 21.74 to Table 21.80 show examples of the register contents for UART transmission.

(a) Serial mode register mn (SMRmn)

Table 21.74 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_0100b	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 21.75 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 21.75 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 21.76 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 UART0 performs 9-bit communication, SDRm0.DAT[8:0] are used as the transmission data specification area. Only UART0 and UART2 support the 9-bit data length.

(d) Serial output level register m (SOLm)

Set only the bit of the target channel.

Table 21.77 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 21.78 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 21.79 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 21.80 Table 15.84 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), mn = 00, 02, 10

Note: ×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Table 21.81 shows the procedure for initial setting for UART transmission.

Table 21.81 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 set SEm.SE[n] = 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 21.82 shows the procedure for stopping master transmission

Table 21.82 Procedure for stopping UART transmission

Step		Process	Detail
Procedure for stopping UART transmission	<1>	Starting setting to stop	—
	<2>	Wait until SSRmn.TSF 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 21.83 shows the procedure for resuming UART transmission.

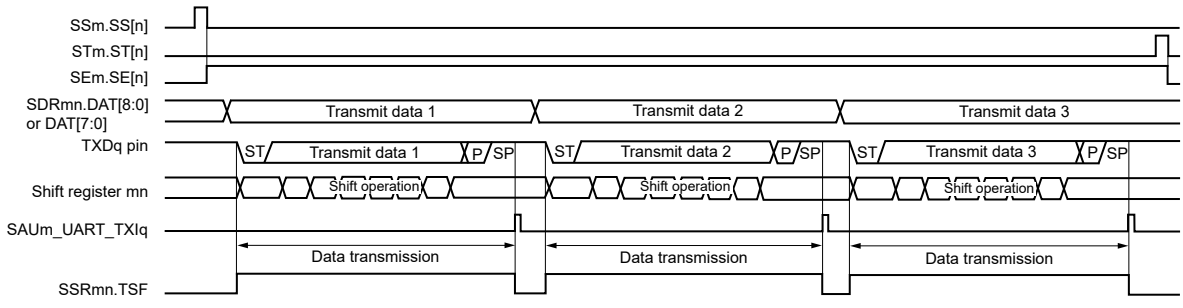
Table 21.83 Procedure for resuming UART transmission

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)

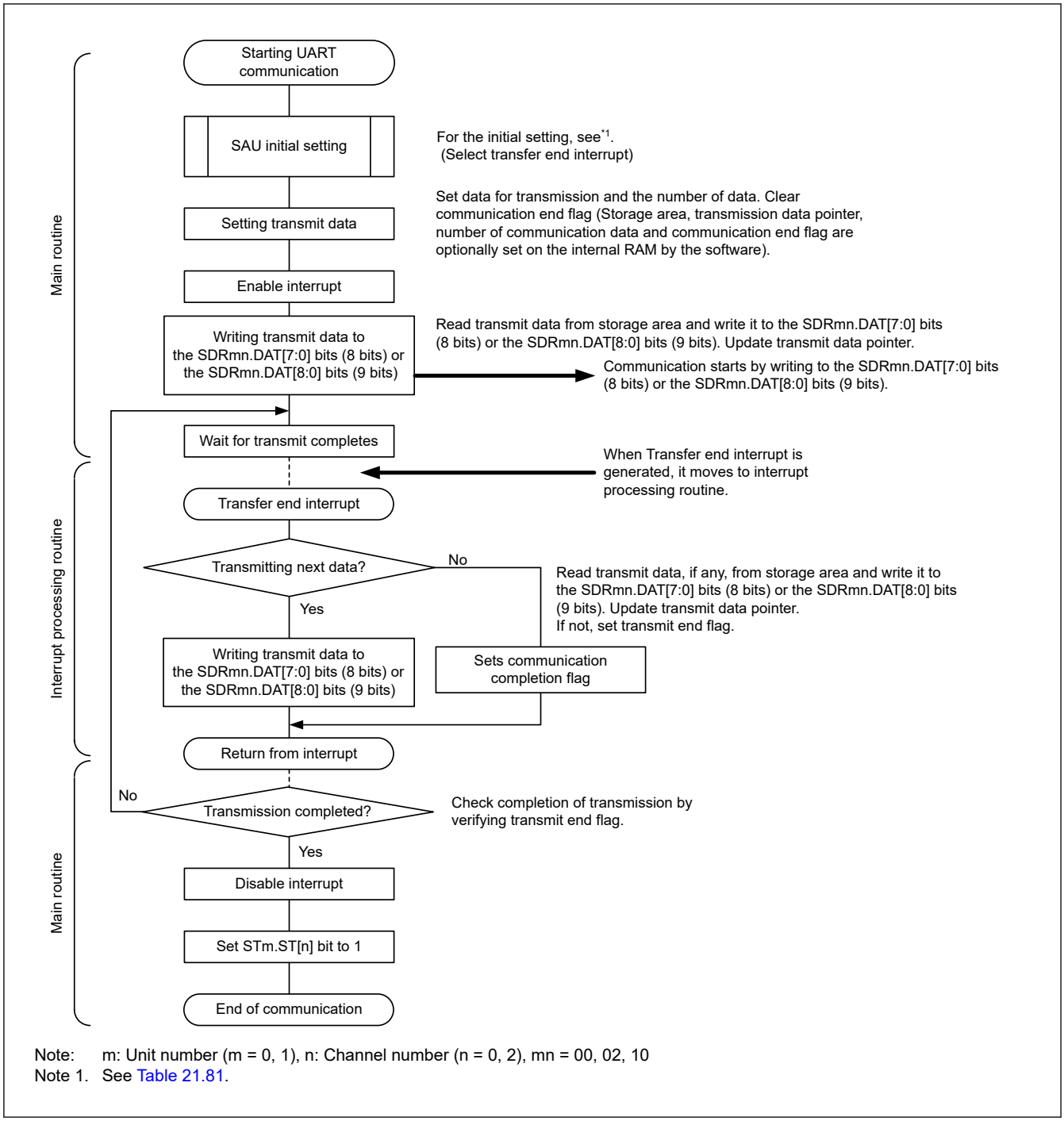
Figure 21.31 shows the timing of UART transmission (in single transmission mode).



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 21.31 Timing of UART transmission (in single transmission mode)

Figure 21.32 shows the flowchart of UART transmission (in single transmission mode).

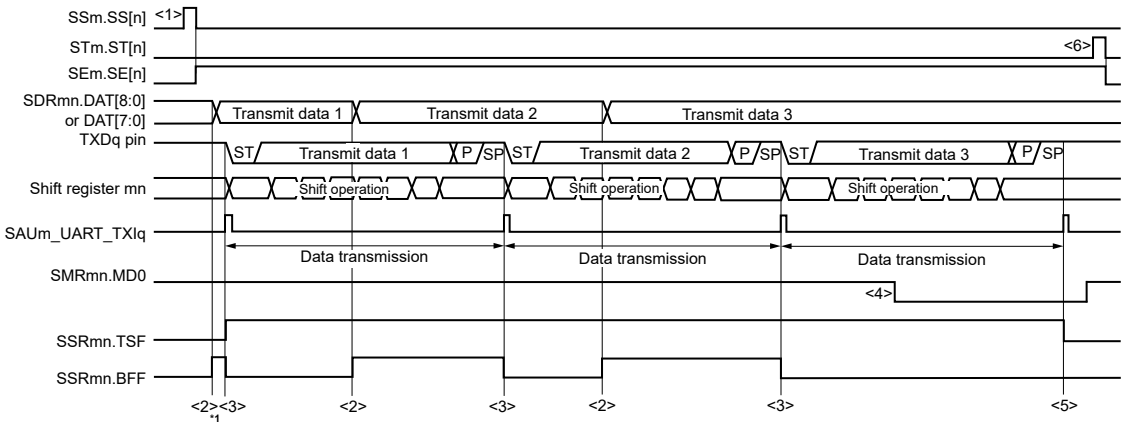


Note: m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 10
Note 1. See Table 21.81.

Figure 21.32 Flowchart of UART transmission (in single transmission mode)

(4) Processing flow (in continuous transmission mode)

Figure 21.33 shows the timing of UART transmission (in continuous transmission mode).



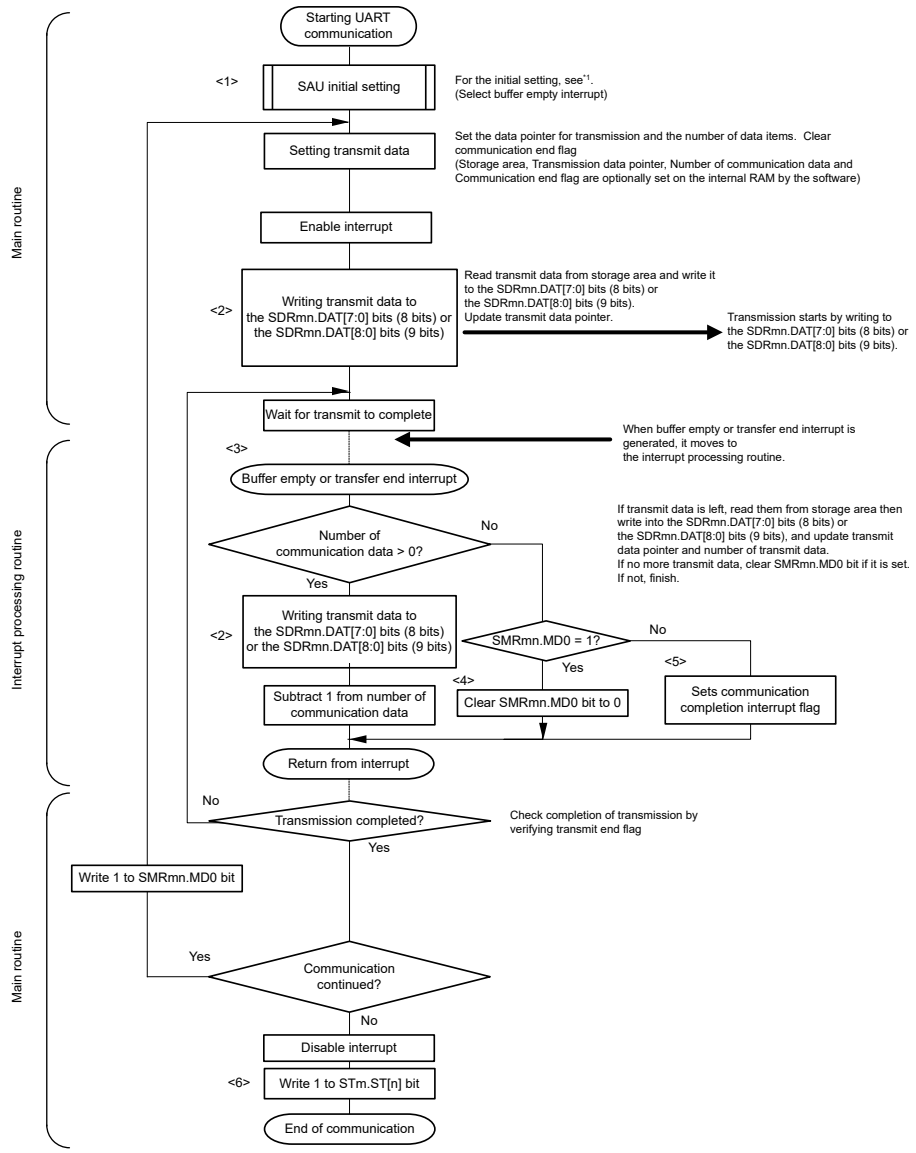
Note: The MD0 bit of serial mode register mn (SMRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last transmit data.

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 bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

Figure 21.33 Timing of UART transmission (in continuous transmission mode)

Figure 21.34 shows the flowchart of UART transmission (in continuous transmission mode).



Note: <1> to <6> in the figure correspond to <1> to <6> in Figure 21.33.
Note: m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 10
Note 1. See Table 21.81.

Figure 21.34 Flowchart of UART transmission (in continuous transmission mode)

21.6.2 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 21.84 shows the specification of UART reception.

Table 21.84 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 21.84 Specification of UART reception (2 of 2)

UART	UART0	UART1	UART2
Interrupt	SAU0_UART_RXI0	SAU0_UART_RXI1	SAU1_UART_RXI2
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)		
Error interrupt	SAU0_UART_ERRI0	SAU0_UART_ERRI1	SAU1_UART_ERRI2
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 UART0 and UART2 support the 9-bit data length.
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 31, Electrical Characteristics .

(1) Register setting

Table 21.85 to Table 21.91 show examples of the register contents for UART reception.

(a) Serial mode register mn (SMRmn)

Table 21.85 Example of serial mode register mn (SMRmn) contents for UART reception

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
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 21.86 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 21.87 Example of serial communication operation setting register mn (SCRmn) contents for UART reception

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
10	EOC	0/1	Mask control of error interrupt signal SAUm_UART_ERRIq 0: Disables generation of error interrupt SAUm_UART_ERRIq (SAUm_UART_RXIq is generated). 1: Enables generation of error interrupt SAUm_UART_ERRIq (SAUm_UART_RXIq 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 21.88 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, bits 0 to 8 of the SDRm1 register are used as the reception data specification area. Only UART0 and UART2 support the 9-bit data length.

(e) Serial output register m (SOm)

This register is not used in this mode.

Table 21.89 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 21.90 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 21.91 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), q: UART number (q = m × 2 + n/2)

Note: ×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Table 21.92 shows the procedure for initial setting for UART Reception.

Table 21.92 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 TRXEmn[0] bit of SCRmn register to 1, and then be sure to set SSm.SS[n] to 1 after at least 4 f_{MCK} clock cycles have elapsed.
Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

Table 21.93 shows the procedure for stopping UART reception.

Table 21.93 Procedure for stopping UART reception

Step		Process	Detail
Procedure for stopping UART transmission	<1>	Starting setting to stop	—
	<2>	Wait until SSRmn.TSF 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 = 0 to 3)

Table 21.94 shows the procedure for resuming UART Reception.

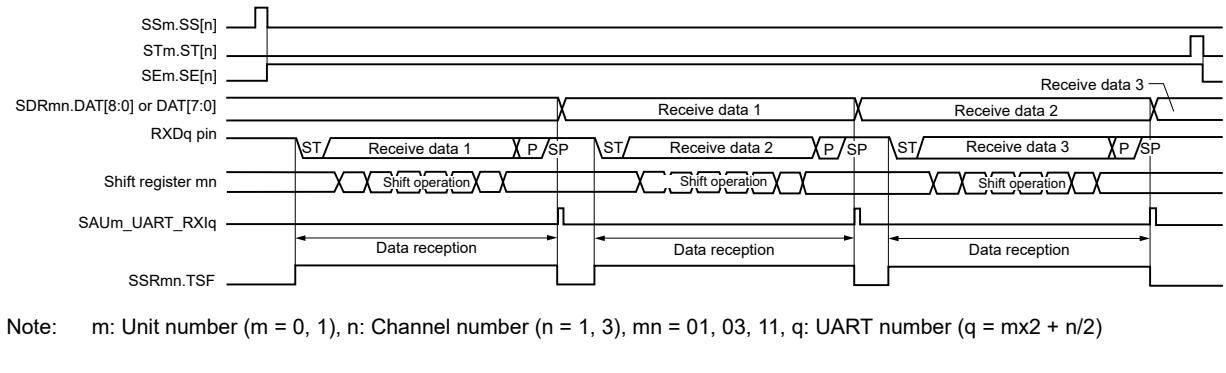
Table 21.94 Procedure for resuming UART reception

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 SSm.SS[n] to 1 after at least 4 f_{MCK} clocks have elapsed.
Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

(3) Processing flow

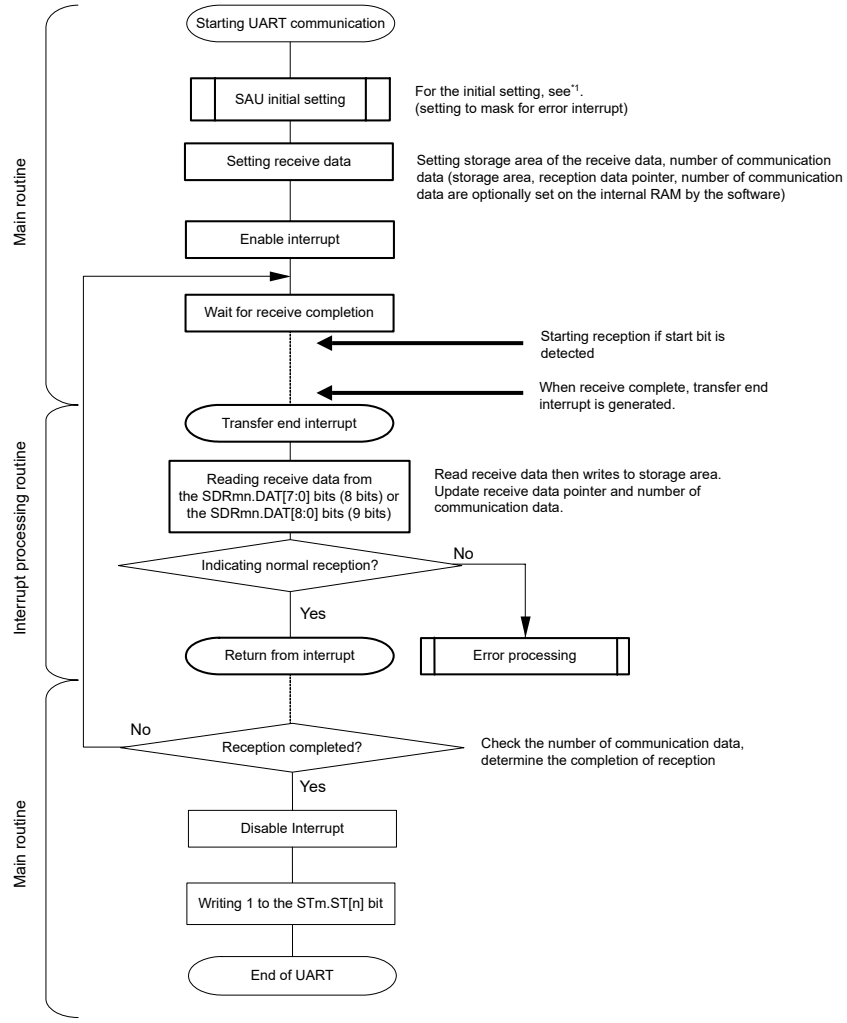
Figure 21.35 shows the timing of UART reception.



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 21.35 Timing of UART reception

Figure 21.36 shows the flowchart of UART reception.



Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)
Note 1. See Table 21.92.

Figure 21.36 Flowchart of UART reception

21.6.3 Snooze Mode Function

The Snooze mode makes the UART perform reception operations on RXD0 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 channel can be set to Snooze mode.

When using UART0 in the Snooze mode, make the following settings before entering the Software Standby mode. (See Figure 21.39 and Figure 21.41.)

- 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 SPS0 register and the SDR01.STCLK[6:0] bits with reference to Table 21.95.
- Set the SCR01.EOC and SSC0.SSEC bits. This is for enabling or stopping generation of an error interrupt (SAU0_UART_ERRI0) when a communication error occurs.
- When using the Snooze mode function, set the SWC bit of serial standby control register 0 (SSC0) 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 0 (SS0) to 1.
- A UART0 starts reception in Snooze mode on detecting input of the start bit on the RXD0 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 or medium-speed on-chip oscillator clock is selected for PCLKB.

When the medium-speed on-chip oscillator clock is selected, use the Middle-speed On-chip Oscillator Trimming Register (MIOTRM) to correct the accuracy of the oscillation frequency.

Note: The maximum transfer rate in the Snooze mode is 115.2 kbps (when setting the SBYCR.FWKUP = 1, PCLKB = HOCO (32 MHz)).

When the SBYCR.FWKUP is set to 1, PCLKB cannot be set to a value other than HOCO = 32 MHz.

Note: When SSC0.SWC = 1, UART0 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 SSC0.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 SSC0.SWC bit is returned to 0

Note: If a parity error, framing error, or overrun error occurs while the SSC0.SSEC bit is set to 1, the SSR01.PEF, FEF, or OVF flag is not set and an error interrupt (SAU0_UART_ERRI0) is not generated. Therefore, when the setting of SSC0.SSEC = 1 is made, clear the SSR01.PEF, FEF, and OVF before setting the SSC0.SWC bit to 1 and read the value in bits 7 to 0 of the SDR01 register.

Note: The CPU shifts from the Software Standby mode to the Snooze mode on detecting the valid edge of the RXD0 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 RXD0 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.

Table 21.95 shows the baud rate setting for UART reception in Snooze mode.

Table 21.95 Baud rate setting for UART reception in Snooze mode

Baud rate	High-speed on-chip oscillator (HOCO)	Operating clock (f_{MCK})	SDR01.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 high-speed on-chip oscillator 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.

Table 21.96 Baud rate setting for UART reception in Snooze mode when starting of the high-speed on-chip oscillator is at high speed (FWKUP = 1)

Baud rate	High-speed on-chip oscillator (HOCO)	Operating clock (f_{MCK})	SDR01 [15:9]	Maximum permissible value	Minimum permissible value
4800 bps	32 MHz ± 1%*1	PCLKB/2^5	106	1.45%	−1.67%
9600 bps		PCLKB/2^4	106	1.45%	−1.67%
19200 bps		PCLKB/2^3	106	1.45%	−1.67%
31250 bps		PCLKB/2^3	65	1.05%	−2.06%
38400 bps		PCLKB/2^2	106	1.45%	−1.67%
76800 bps		PCLKB/2	106	1.45%	−1.67%
115200 bps		PCLKB/2	70	1.93%	−1.21%

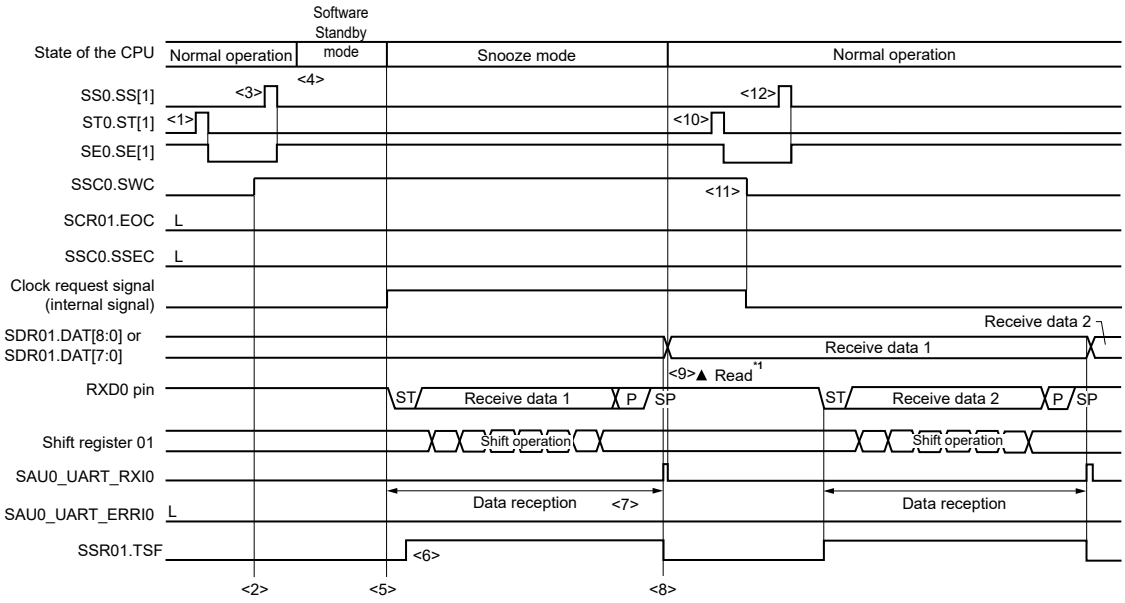
Note 1. When the accuracy of the clock frequency of the high-speed on-chip oscillator is ±1.5% or ±2.0%, the permissible range becomes smaller as shown below.
- In the case of HOCO ±1.5%, perform (Maximum permissible value – 0.5%) and (Minimum permissible value + 0.5%) to the values in the above table.
- 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 (SCR01.EOC = 0, SSC0.SSEC = 0/1)

Because of the setting of SCR01.EOC = 0, even though a communication error occurs, an error interrupt (SAU0_UART_ERRI0) is not generated, regardless of the setting of the SSC0.SSEC bit. However, a transfer end interrupt (SAU0_UART_RXI0) is generated.

Figure 21.37 shows the timing of Snooze mode operation (SCR01.EOC = 0, SSC0.SSEC = 0/1).



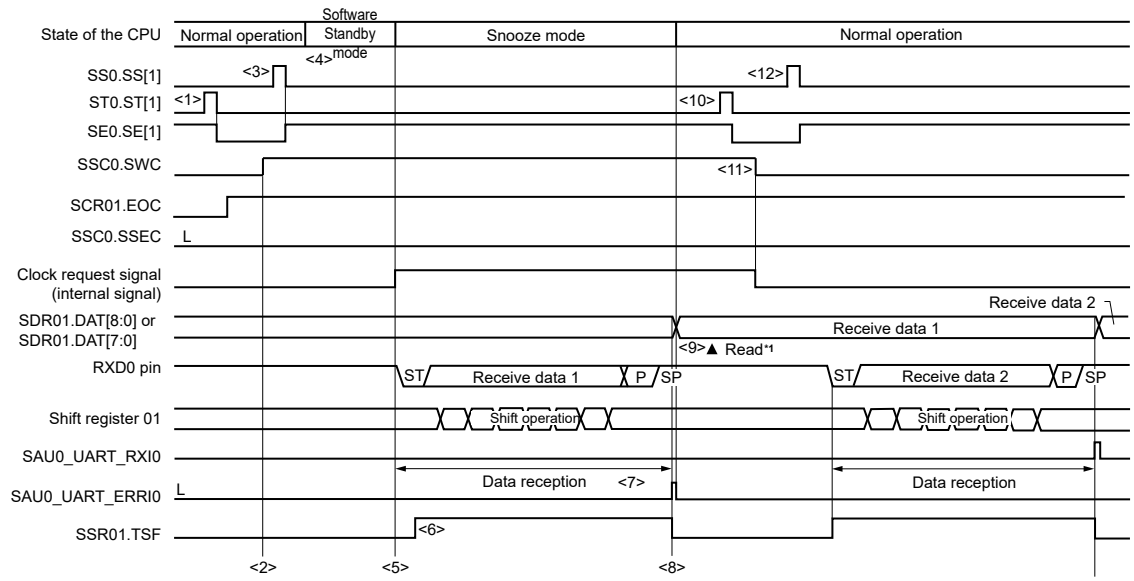
Note: Before switching to the Snooze mode or after reception operation in the Snooze mode finishes, set the ST0.ST[1] bit to 1 (the SE0.SE[1] bit is cleared and the operation stops).
After the receive operation completes, also clear the SSC0.SWC bit to 0 (Snooze mode release).
Note: <1> to <12> in the figure correspond to <1> to <12> in Figure 21.39.
Note 1. Read the received data when SSC0.SWC = 1.

Figure 21.37 Timing of Snooze mode operation (SCR01.EOC = 0, SSC0.SSEC = 0/1)

(2) Snooze mode operation (SCR01.EOC = 1, SSC0.SSEC = 0: Error interrupt (SAU0_UART_ERRI0) generation is enabled)

Because SCR01.EOC = 1 and SSC0.SSEC = 0, an error interrupt (SAU0_UART_ERRI0) is generated when a communication error occurs.

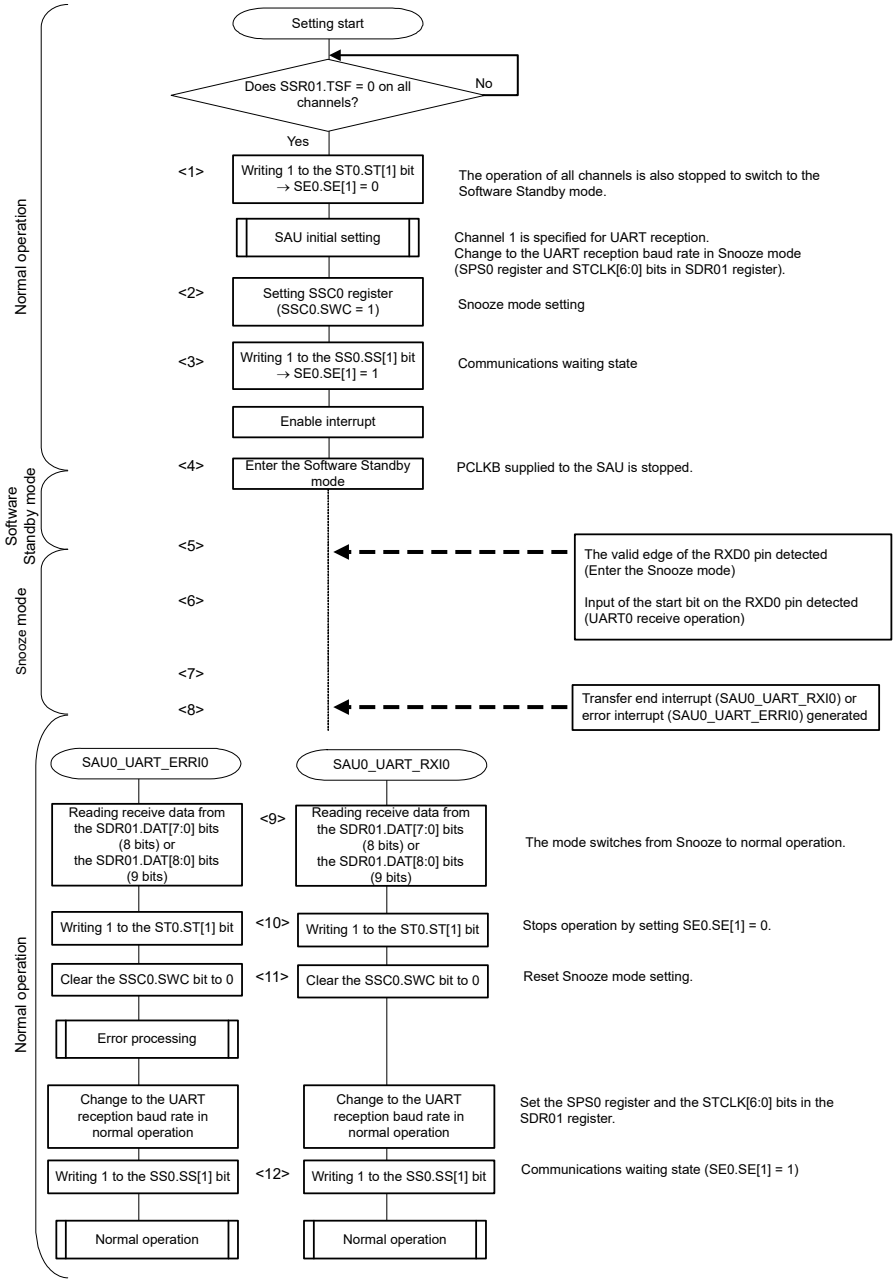
Figure 21.38 shows the timing of Snooze mode operation (SCR01.EOC = 1, SSC0.SSEC = 0).



Note: Before switching to the Snooze mode or after reception operation in the Snooze mode finishes, set the ST0.ST[1] bit to 1 (the SE0.SE[1] bit is cleared and the operation stops).
After the receive operation completes, also clear the SSC0.SWC bit to 0 (Snooze mode release).
Note: <1> to <12> in the figure correspond to <1> to <12> in Figure 21.39.
Note 1. Read the received data when SSC0.SWC = 1.

Figure 21.38 Timing of Snooze mode operation (SCR01.EOC = 1, SSC0.SSEC = 0)

Figure 21.39 shows the flowchart of Snooze mode operation (SCR01.EOC = 0, SSC0.SSEC = 0/1 or SCR01.EOC = 1, SSC0.SSEC = 0).



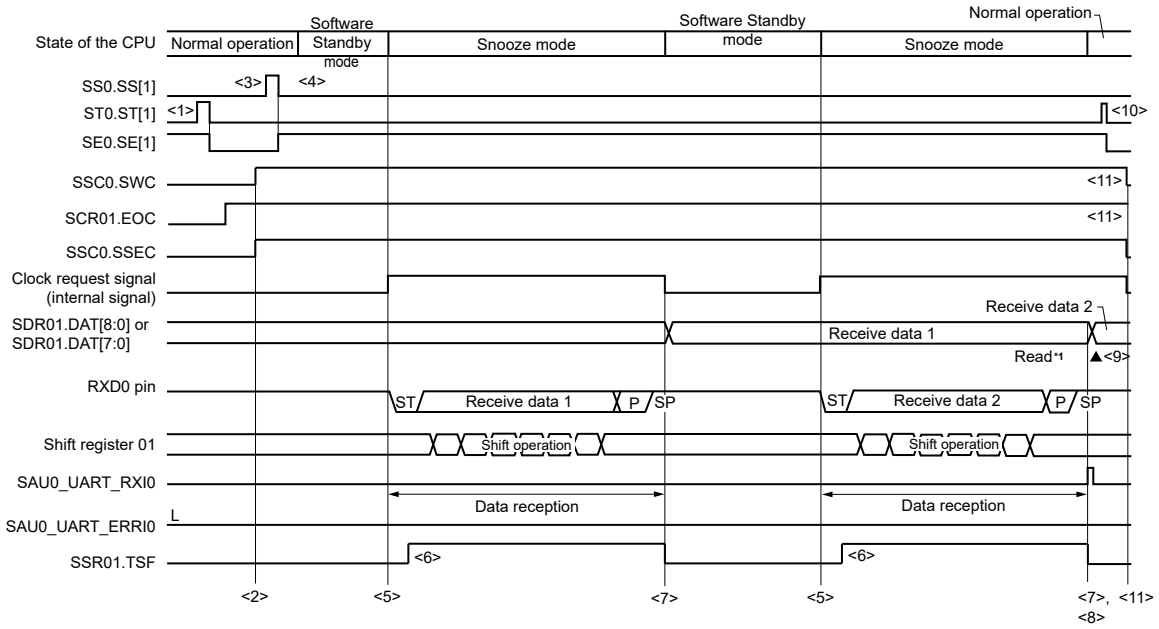
Note: <1> to <12> in the figure correspond to <1> to <12> in Figure 21.37 and Figure 21.38.

Figure 21.39 Flowchart of Snooze mode operation (SCR01.EOC = 0, SSC0.SSEC = 0/1 or SCR01.EOC = 1, SSC0.SSEC = 0)

(3) Snooze mode operation (SCR01.EOC = 1, SSC0.SSEC = 1: Error interrupt (SAU0_UART_ERRI0) generation is stopped)

Because SCR01.EOC = 1 and SSC0.SSEC = 1, an error interrupt (SAU0_UART_ERRI0) is not generated when a communication error occurs.

Figure 21.40 shows the timing of Snooze mode operation (SCR01.EOC = 1, SSC0.SSEC = 1).



Note: Before switching to the Snooze mode or after reception operation in the Snooze mode finishes, set the ST0.ST[1] bit to 1 (the SE0.SE[1] bit is cleared and the operation stops).
After the receive operation completes, also clear the SSC0.SWC bit to 0 (Snooze mode release).

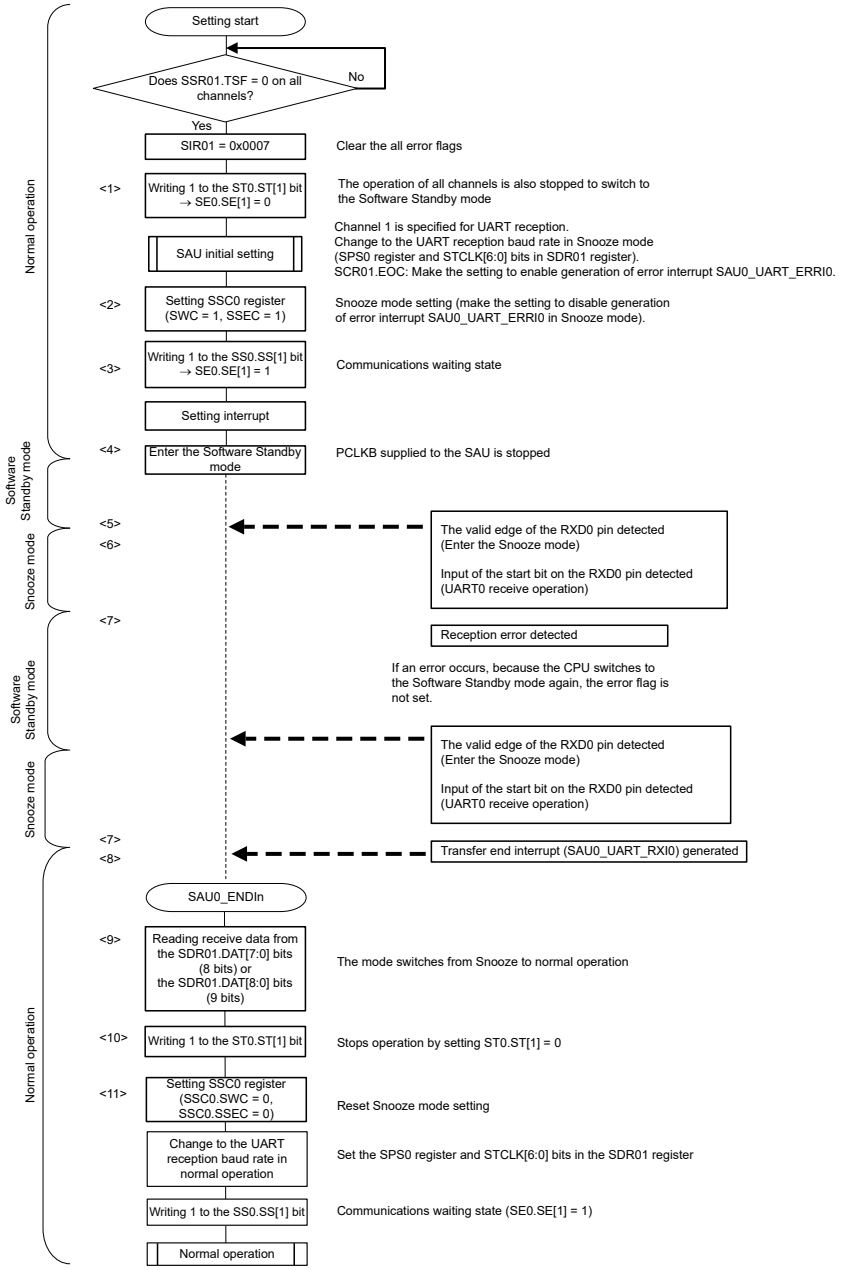
Note: If a parity error, framing error, or overrun error occurs while the SSC0.SSEC bit is set to 1, the SSR01.PEF, FEF, or OVF flag is not set and an error interrupt (SAU0_UART_ERRI0) is not generated. Therefore, when the setting of SSC0.SSEC = 1 is made, clear the SSR01.PEF, FEF, and OVF flags before setting the SSC0.SWC bit to 1 and read the value in SDR01.DAT[7:0] (8 bits) or SDR01.DAT[8:0] (9 bits).

Note: <1> to <11> in the figure correspond to <1> to <11> in Figure 21.41.

Note 1. Read the received data when SSC0.SWC = 1.

Figure 21.40 Timing of Snooze mode operation (SCR01.EOC = 1, SSC0.SSEC = 1)

Figure 21.41 shows the flowchart of Snooze mode operation (SCR01.EOC = 1, SSC0.SSEC = 1).



Note: If a parity error, framing error, or overrun error occurs while the SSC0.SSEC bit is set to 1, the SSR01.PEF, FEF, or OVF flag is not set and an error interrupt (SAU0_UART_ERRI0) is not generated. Therefore, when the setting of SSC0.SSEC = 1 is made, clear the SSR01.PEF, FEF, and OVF flags before setting the SSC0.SWC bit to 1 and read the value in SDR01.DAT[7:0] (8 bits) or SDR01.DAT[8:0] (9 bits).

Note: <1> to <11> in the figure correspond to <1> to <11> in Figure 21.40.

Figure 21.41 Flowchart of Snooze mode operation (SCR01.EOC = 1, SSC0.SSEC = 1)

21.6.4 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 21.71.

(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 21.97 shows an example of setting a UART baud rate at PCLKB = 32 MHz.

Table 21.97 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 21.42 shows the permissible baud rate range for reception (1 Data Frame Length = 11 Bits).

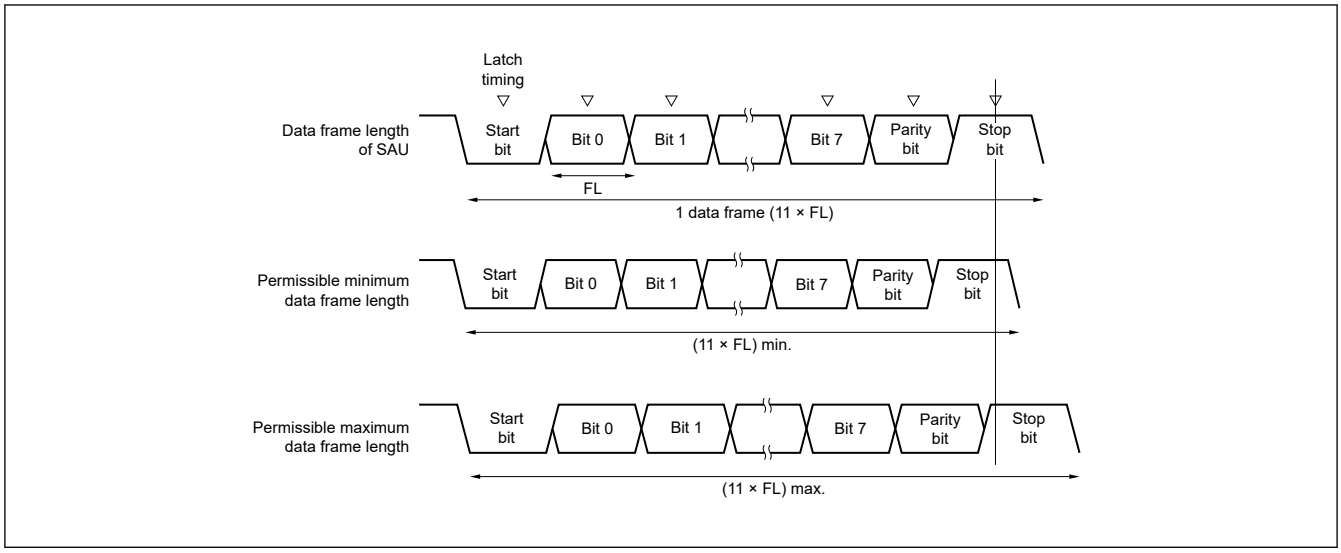


Figure 21.42 Permissible baud rate range for reception (1 data frame length = 11 bits)

As shown in Figure 21.42, the timing of latching receive data is determined by the division ratio set by 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.

21.6.5 Procedure for Processing Errors that Occurred During UART Communication

The procedure for processing errors that occurred during UART communication is described in Table 21.98 and Table 21.99.

Table 21.98 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 bit 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 21.99 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 bit 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 (ST[n]) to 1.	→	The SE[n] bit of serial channel enable status register m (SEm) is set to 0 and channel n stops operation.	—

Table 21.99 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 SSm.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

21.7 Operation of LIN Communication

21.7.1 LIN Transmission

UART2 supports LIN communication.

Channel 0 of unit 1 is used for LIN transmission.

Table 21.100 shows the specification of LIN transmission.

Table 21.100 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_UART_TXI2
	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 31, Electrical Characteristics . In general, 2.4, 9.6, or 19.2 kbps is often used in LIN communication.

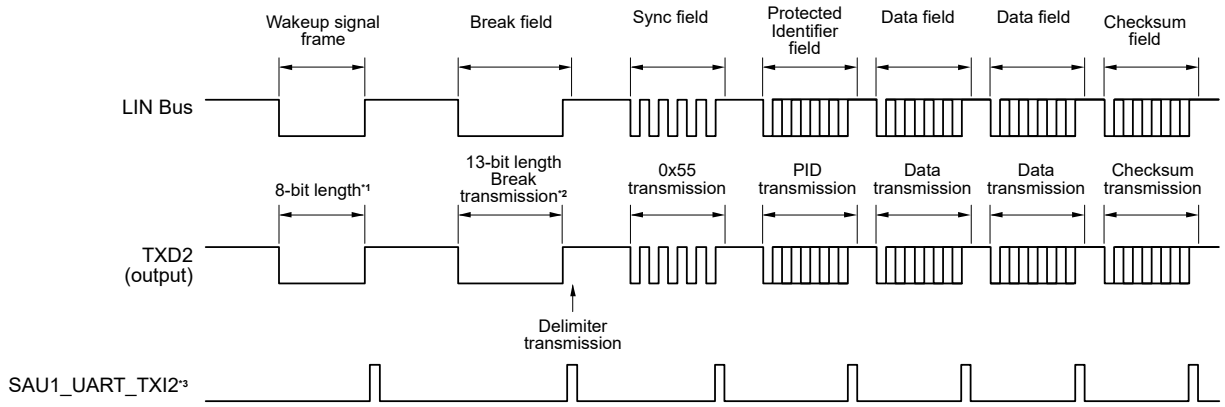
LIN stands for Local Interconnect Network and is a low-speed (1 to 20 kbps) serial communication protocol designed to reduce the cost of an automobile network.

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 through 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 through 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 21.43 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.

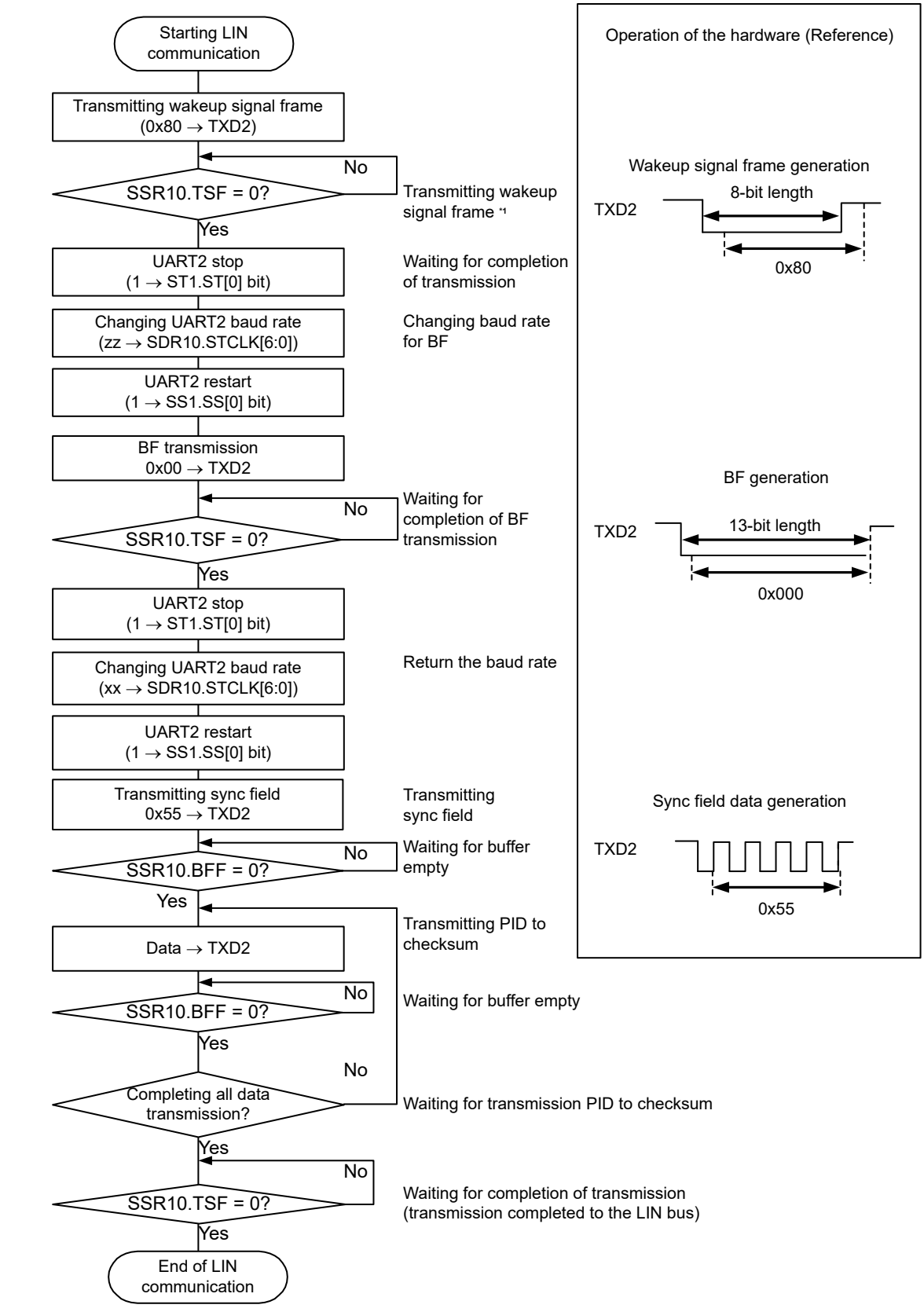
$$(\text{Baud rate of break field}) = \frac{9}{13} \times N$$

By transmitting data of 0x00 at this baud rate, a break field is generated.

Note 3. SAU1_UART_TXI2 is output on completion of transmission. SAU1_UART_TXI2 is also output at BF transmission.

Figure 21.43 Transmission operation of LIN

Figure 21.44 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 21.44 Flowchart for LIN transmission

21.7.2 LIN Reception

Of UART reception, UART2 supports LIN communication.

For LIN reception, channel 1 of unit 1 is used.

Table 21.101 shows the specification of LIN reception.

Table 21.101 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_UART_RXI2
	Transfer end interrupt only (setting the buffer empty interrupt is prohibited)		
Error interrupt	—	—	SAU1_UART_ERRI2
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 31, Electrical Characteristics .

Figure 21.45 shows a reception operation of LIN.

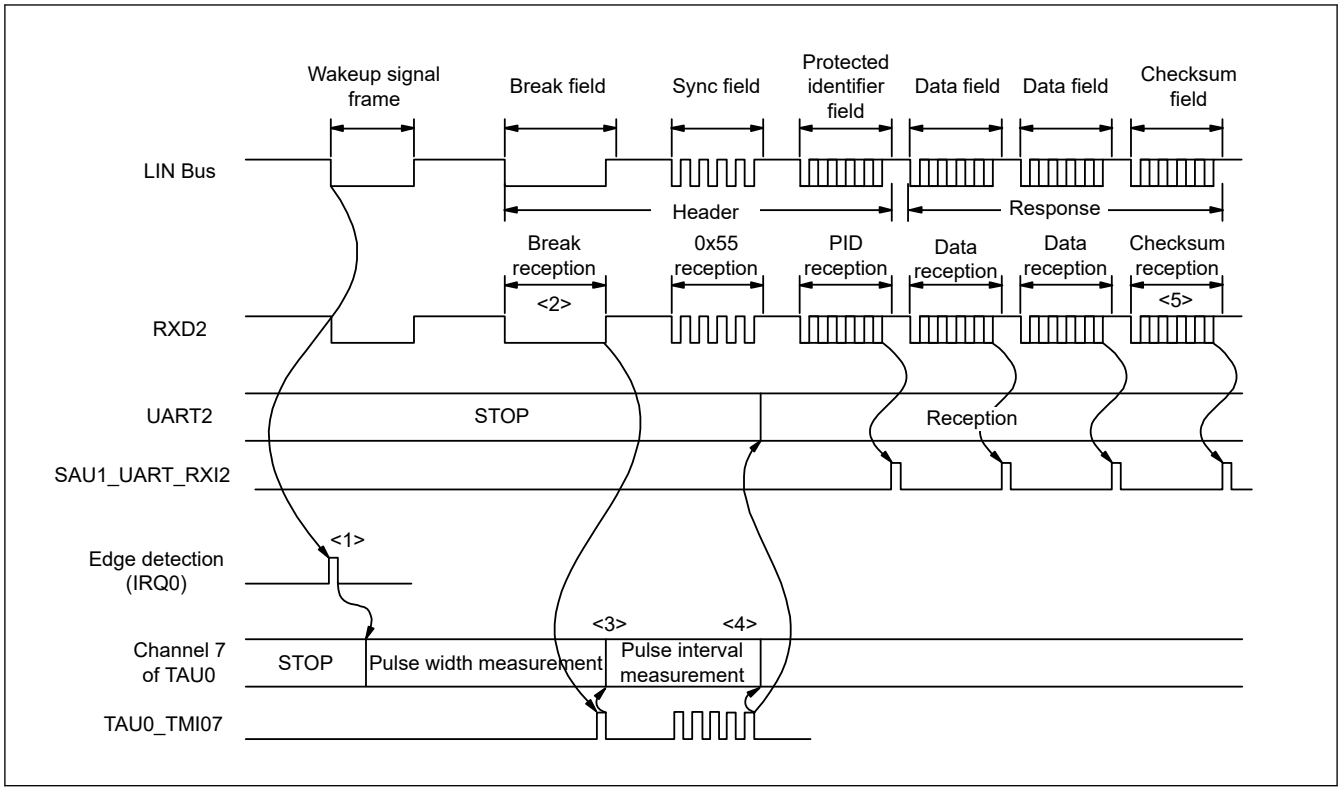


Figure 21.45 Reception operation of LIN

The flow of reception processing is as follows.

<1> The wakeup signal is detected by using an edge on the external interrupt pin (IRQ0). When the wakeup signal is detected, set channel 7 of TAU0 to the pulse width measurement function to measure the low-level width of the BF signal. Then wait for BF signal reception.

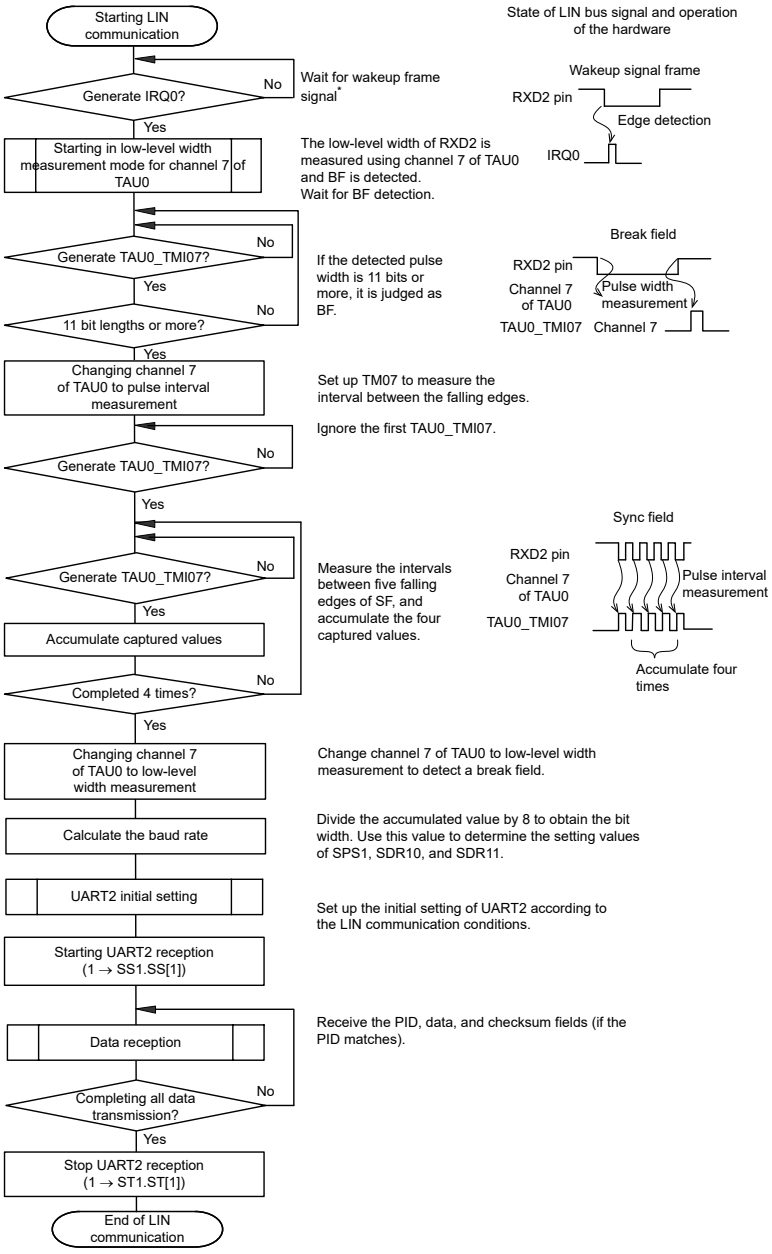
<2> Channel 7 of TAU0 starts measuring the low-level width on detection of the falling edge of the BF signal, and then captures the data on detection of the rising edge of the BF signal. The captured data is used to determine whether it is the BF signal.

<3> When the BF signal has been received normally, change channel 7 of TAU0 to pulse interval measurement and measure the interval between the falling edges of the RXD2 signal in the Sync field four times. (See section 17.7.4. 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 21.46 shows the flowchart of LIN reception.



Note: This is only required if the LIN bus is in Sleep mode.

Figure 21.46 Flowchart of LIN reception

Figure 21.47 shows the configuration of ports used for LIN reception.

The wakeup signal transmitted from the master of LIN is received by detecting an edge of an external interrupt (IRQ0). 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 0 to calculate a baud-rate error.

By using the port input switching control (the ISC.ISC0 and ISC.ISC1 bits), the signal input to the reception port (RXD2) can be used as an external interrupt (IRQ0) or sent to the timer array unit without additional external connections.

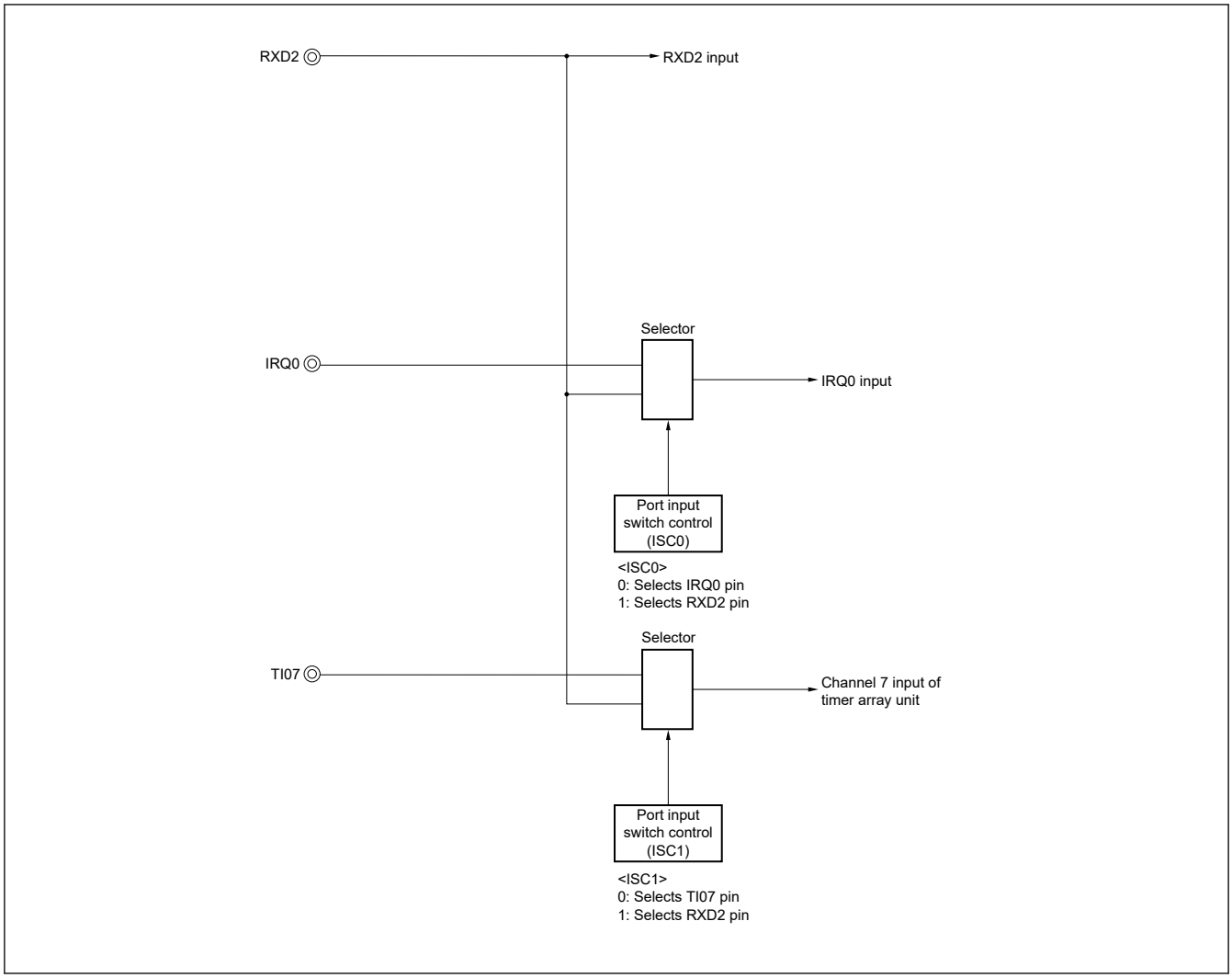


Figure 21.47 Port configuration for LIN reception

The peripheral functions used for the LIN communication operation are as follows.

<Peripheral functions used>

- External interrupt (IRQ0), 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)

21.8 Operation of Simplified I^2C 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 I^2C 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 I^2C 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 (SAU0_IIC_TXRXI00/SAU0_IIC_TXRXI01/SAU0_IIC_TXRXI10/SAU0_IIC_TXRXI11/ SAU1_IIC_TXRXI20/SAU1_IIC_TXRXI21)

[Error detection flag]

- Overrun error
- ACK error

[Functions not supported by simplified I^2C]

- 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)

The channel supporting simplified I^2C is channels 0 to 3 of SAU0 and channel 0 and 1 of SAU1. section 21, Serial Array Unit (SAU) to section 21, Serial Array Unit (SAU) show the channels supporting simplified I^2C for each product.

Simplified I^2C performs the following four types of communication operations:

- Address field transmission (see section 21.8.1. Address Field Transmission)
- Data transmission (see section 21.8.2. Data Transmission)
- Data reception (see section 21.8.3. Data Reception)
- Stop condition generation (see section 21.8.4. Stop Condition Generation.)

21.8.1 Address Field Transmission

Address field transmission is a transmission operation that first executes in I^2C 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 21.102 shows the specification of address field transmission of Simplified I^2C.

Table 21.102 Specification of address field transmission of simplified I^2C (1 of 2)

Simplified I^2C	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_IIC_TXRXI00	SAU0_IIC_TXRXI01	SAU0_IIC_TXRXI10	SAU0_IIC_TXRXI11	SAU1_IIC_TXRXI20	SAU1_IIC_TXRXI21
	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)					

Table 21.102 Specification of address field transmission of simplified I²C (2 of 2)

Simplified I²C	IIC00	IIC01	IIC10	IIC11	IIC20	IIC21
Transfer rate*2	Max. f_{MCK}/4 [Hz] (SDRmn.STCLK[6:0] = 1 or more) f_{MCK}: Operation clock frequency of target channel. 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)
Note 1. To perform communication using simplified I²C, set the NMOS open drain output mode with the Port mn Pin Function Select Register (PmnPFS_A). 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 31, Electrical Characteristics .

(1) Register setting

Table 21.103 to Table 21.108 show examples of the register contents for address field transmission of simplified I²C.

(a) Serial mode register mn (SMRmn)

Table 21.103 Example of serial mode register mn (SMRmn) contents for address field transmission of simplified I²C

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
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 21.104 Example of serial communication operation setting register mn (SCRmn) contents for address field transmission of simplified I²C (1 of 2)

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)

Table 21.104 Example of serial communication operation setting register mn (SCRmn) contents for address field transmission of simplified I²C (2 of 2)

Bit	Symbol	Set value	Function
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 I²C mode because it is for simplified SPI and UART modes.
9:8	PTC[1:0]	00b	This bit is fixed in simplified I²C mode because it is for UART mode.
10	EOC	0	This bit is fixed in simplified I²C 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 I²C mode because it is for simplified SPI mode.
15:14	TRXE[1:0]	10b	Setting TRXE[1:0] = 10b is fixed in the simplified I²C address field transmission

(c) Serial data register mn (SDRmn)

Table 21.105 Example of serial data register mn (SDRmn) contents for address field transmission of simplified I²C

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]	0x00 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 21.106 Example of serial output register m (SOm) contents for address field transmission of simplified I²C

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 21.107 Example of serial output enable register m (SOEm) contents for address field transmission 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)

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 21.108 Example of serial channel start register m (SSm) contents for address field transmission 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)

Note: 0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Table 21.109 shows the procedure for initial setting of simplified I²C address field transmission.

Table 21.109 Initial setting procedure for simplified I²C address field transmission

Step		Process	Detail
Procedure for initial setting of I²C 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)

(3) Processing flow

Figure 21.48 shows the timing of address field transmission.

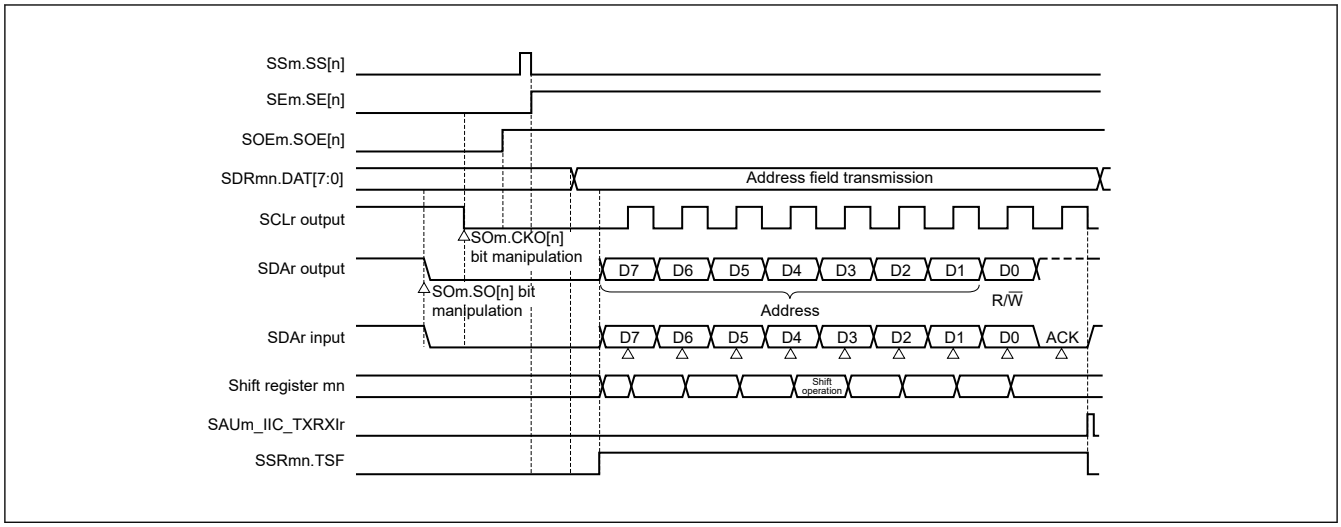


Figure 21.48 Timing of address field transmission

Table 21.110 shows the procedure for simplified I^{2}C address field transmission.

Table 21.110 Procedure for simplified I^{2}C address field transmission

Step		Process	Detail
Procedure for simplified I^{2}C address field transmission	<1>	Transmitting address field	—
	<2>	Default setting	For the initial setting, see Table 21.109.
	<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 complete. Clear the interrupt request flag.
	<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 SSRmn.PEF bit. 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)

21.8.2 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 21.111 shows the specification of data transmission of simplified I^{2}C.

Table 21.111 Specification of data transmission of simplified I^{2}C (1 of 2)

Simplified I^{2}C	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

Table 21.111 Specification of data transmission of simplified I²C (2 of 2)

Simplified I²C	IIC00	IIC01	IIC10	IIC11	IIC20	IIC21
Interrupt	SAU0_IIC_TXRXI00	SAU0_IIC_TXRXI01	SAU0_IIC_TXRXI10	SAU0_IIC_TXRXI11	SAU1_IIC_TXRXI20	SAU1_IIC_TXRXI21
	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[15:9] = 1 or more) f_{MCK}: Operation clock frequency of target channel. 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
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 mn Pin Function Select Register (PmnPFS_A). 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 31, Electrical Characteristics.

(1) Register setting

Table 21.112 to Table 21.117 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 21.112 Example of serial mode register mn (SMRmn) contents for data transmission of simplified I²C

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
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 SCRmn.TRXE[1:0] bits, during data transmission and reception.

Table 21.113 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 21.114 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]	0000000b to 1111111b	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 21.115 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 21.116 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 21.117 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)

Note: 0/1: Set to 0 or 1 depending on the usage of the user.

(2) Processing flow

Figure 21.49 shows the timing of data transmission.

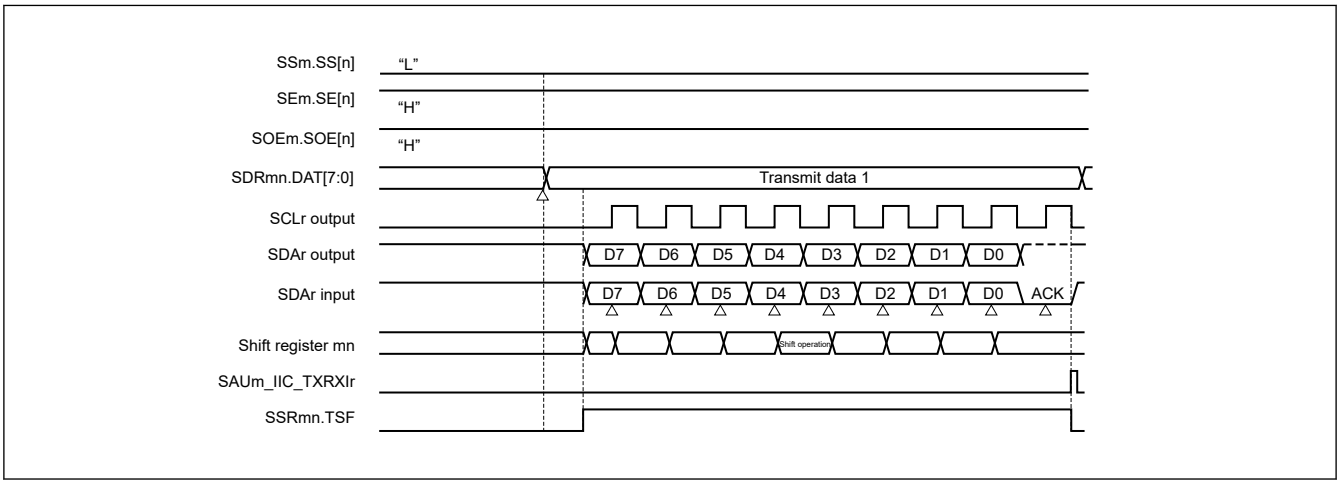


Figure 21.49 Timing of data transmission

Table 21.118 shows the procedure for simplified I^2C data transmission.

Table 21.118 Procedure for simplified I^2C data transmission

Step		Process	Detail
Procedure for simplified I^2C 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 complete. Clear the interrupt request flag.
	<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)

21.8.3 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 21.119 shows the specification of data reception of simplified I²C.

Table 21.119 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_IIC_TXRXI00	SAU0_IIC_TXRXI01	SAU0_IIC_TXRXI10	SAU0_IIC_TXRXI11	SAU1_IIC_TXRXI20	SAU1_IIC_TXRXI21
	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) f_{MCK}: Operation clock frequency of target channel. 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)
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 mn Pin Function Select Register (PmnPFS_A). 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 31, Electrical Characteristics.

(1) Register setting

Table 21.120 to Table 21.125 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 21.120 Example of serial mode register mn (SMRmn) contents for data reception 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)

Table 21.120 Example of serial mode register mn (SMRmn) contents for data reception of simplified I^2C (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 CKSmn 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 TRXEmn [1:0] bits, during data transmission and reception.

Table 21.121 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 21.122 Example of serial data register mn (SDRmn) contents for data reception of simplified I^2C

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]	0x00 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 21.123 Example of serial output register m (SOm) contents for data reception 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 21.124 Example of serial output enable register m (SOEm) contents for data reception 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)

Do not manipulate this register during data transmission and reception.

Table 21.125 Example of serial channel start register m (SSm) contents for data reception 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.

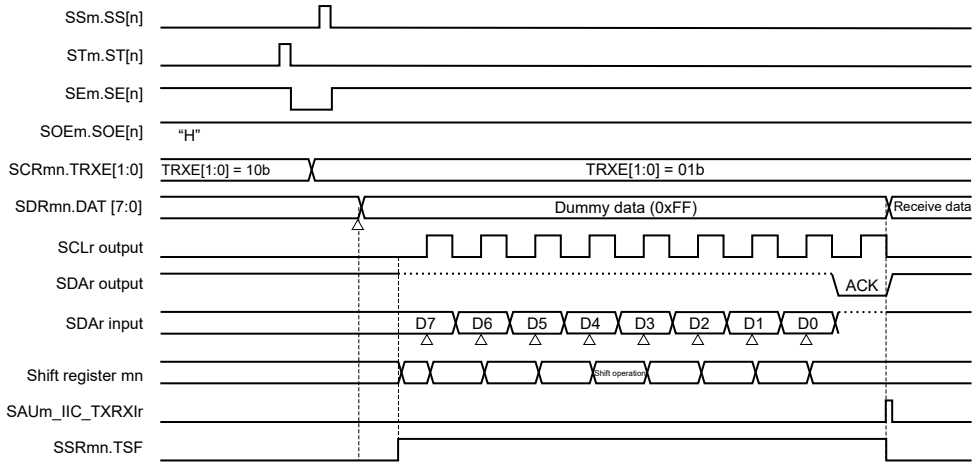
Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

Note: 0/1: Set to 0 or 1 depending on the usage of the user

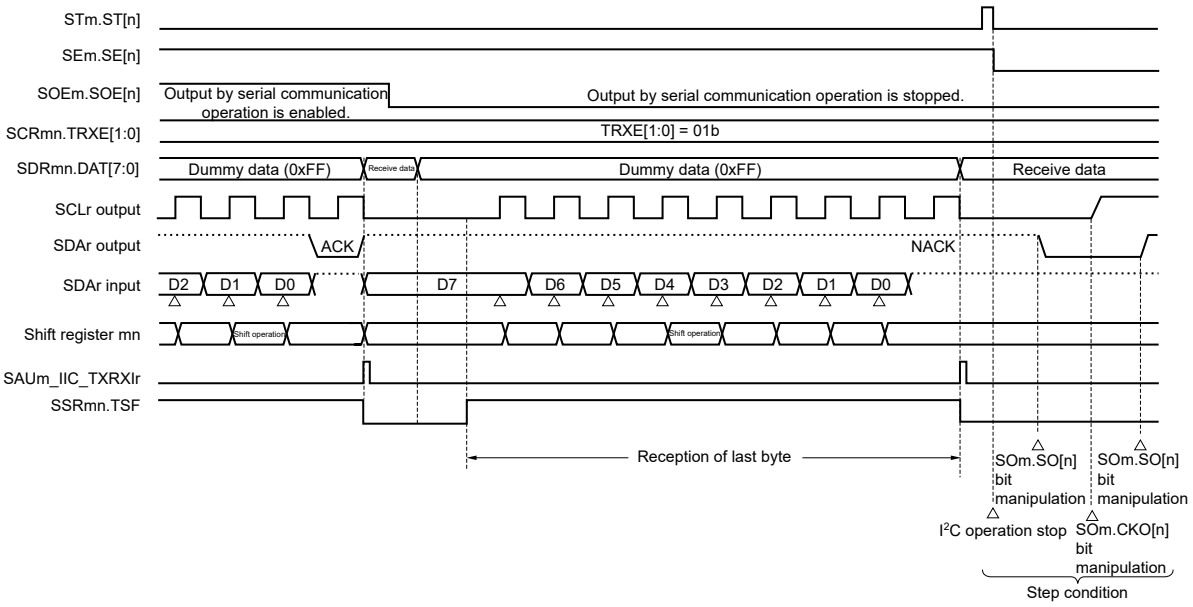
(2) Processing flow

Figure 21.50 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)

Figure 21.50 Timing of data reception

Table 21.126 shows the procedure for data reception.

Table 21.126 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 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 the completion of reception. Clear the interrupt request flag.
	<10>	Reading SDRmn.DAT[7:0] bits	Reading receive data, perform processing (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	—

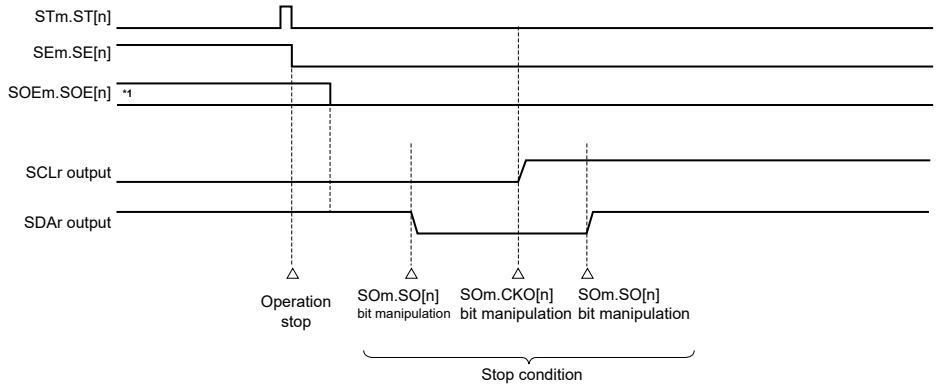
Note: ACK is not output when the last data is received (NACK). Communication is then completed by setting 1 in the ST[n] bit of serial channel stop register m (STm) to stop operation and generating a stop condition.
Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

21.8.4 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 21.51 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)
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 21.51 Timing of stop condition generation

Table 21.127 shows the procedure for stop condition generation.

Table 21.127 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)

21.8.5 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] value satisfies the I^2C bus specifications.

Note: m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

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 21.71.

Table 21.128 shows an example of setting an I^2C transfer rate where f_{MCK} = PCLKB = 32 MHz.

Table 21.128 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%.

21.8.6 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 21.129 and Table 21.130.

Table 21.129 Processing procedure for overrun error

Step	Software manipulation		State of the hardware	Remark
<1>	Read serial data register mn (SDRmn).	→	The BFF bit 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 21.130 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)

22. I2C Bus Interface (IICA)

22.1 Overview

The I2C bus interface has two channels and supports the following three modes:

- Operation stop mode
- I2C bus mode (multimaster supported)
- Wakeup mode

Table 22.1 shows specifications of the I2C bus interface.

Table 22.1 I2C specifications

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	• If multiple master devices generate start conditions simultaneously, the master with the greater number of clock cycles before the data changes will control the communication. • When arbitration lost occurs, both the SCLA0 and SDAA0 lines of that master device become high impedance and the bus is released. • 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 (IICAn_TXRXI)	• 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
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

Note: n = 0, 1

Figure 22.1 shows a block diagram of the I2C bus interface.

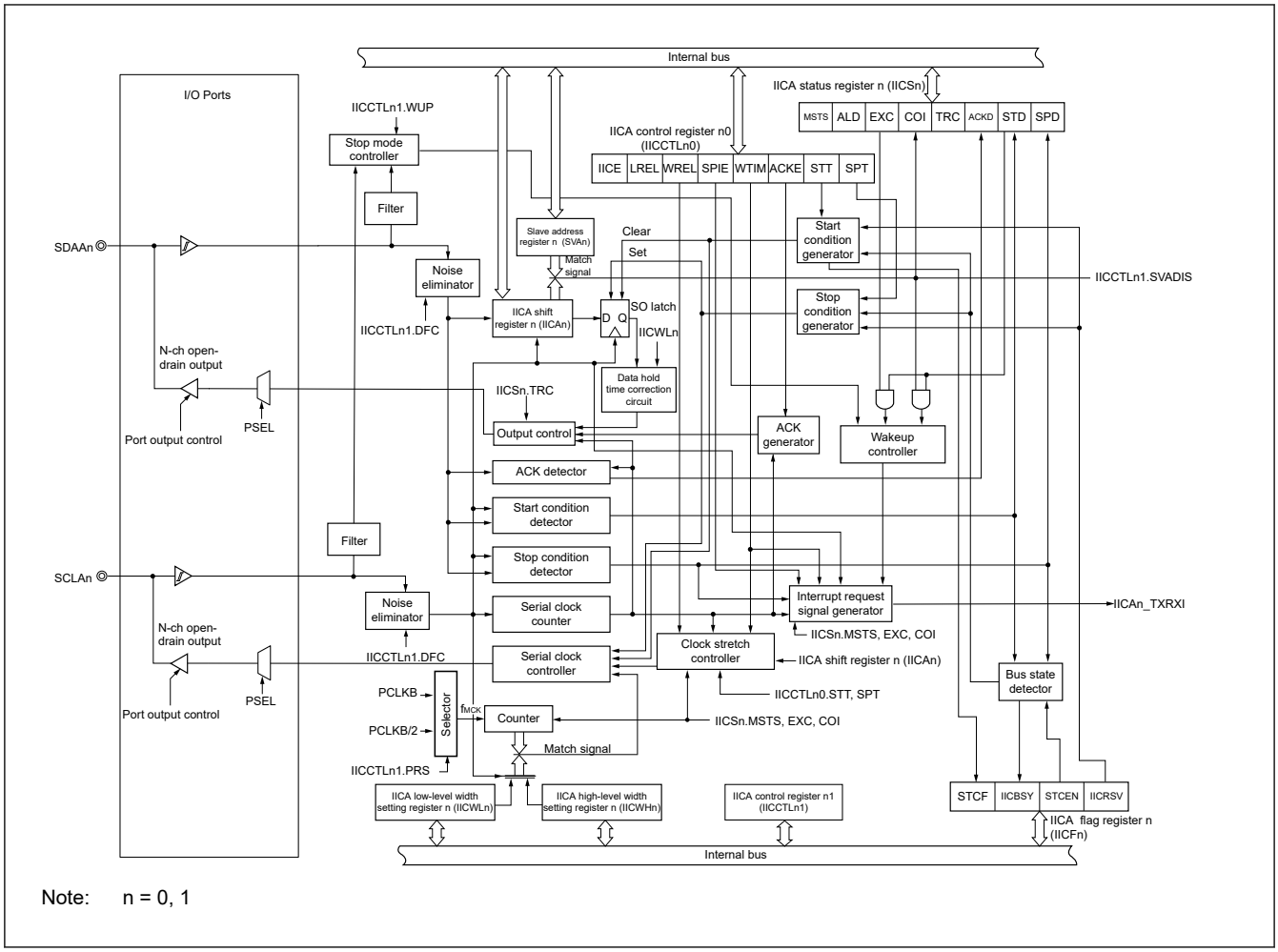


Figure 22.1 Block diagram of I2C bus interface

(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 (multimaster 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 I2C bus format. In master mode, 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 mode automatically detects these states and data by hardware. This function can simplify the part of application program that controls the I2C bus.

Since the SCLAn and SDAAn pins are used for open drain outputs, I2C bus interface (IICA) 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 (IICAn_TXRXI) 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 IICCTLn1 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 (IICAn_TXRXI) 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 (IICAn_TXRXI). An I^2C interrupt request is generated by the following two triggers:

- Falling edge of the 8th or 9th clock cycle 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 generates the hold time for data after the falling edge of the serial clock.

(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 bit = 1) and the bus is busy (IICFn.IICBSY bit = 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 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 22.2 shows an example of the serial bus configuration using the I^2C bus.

Note: n = 0, 1

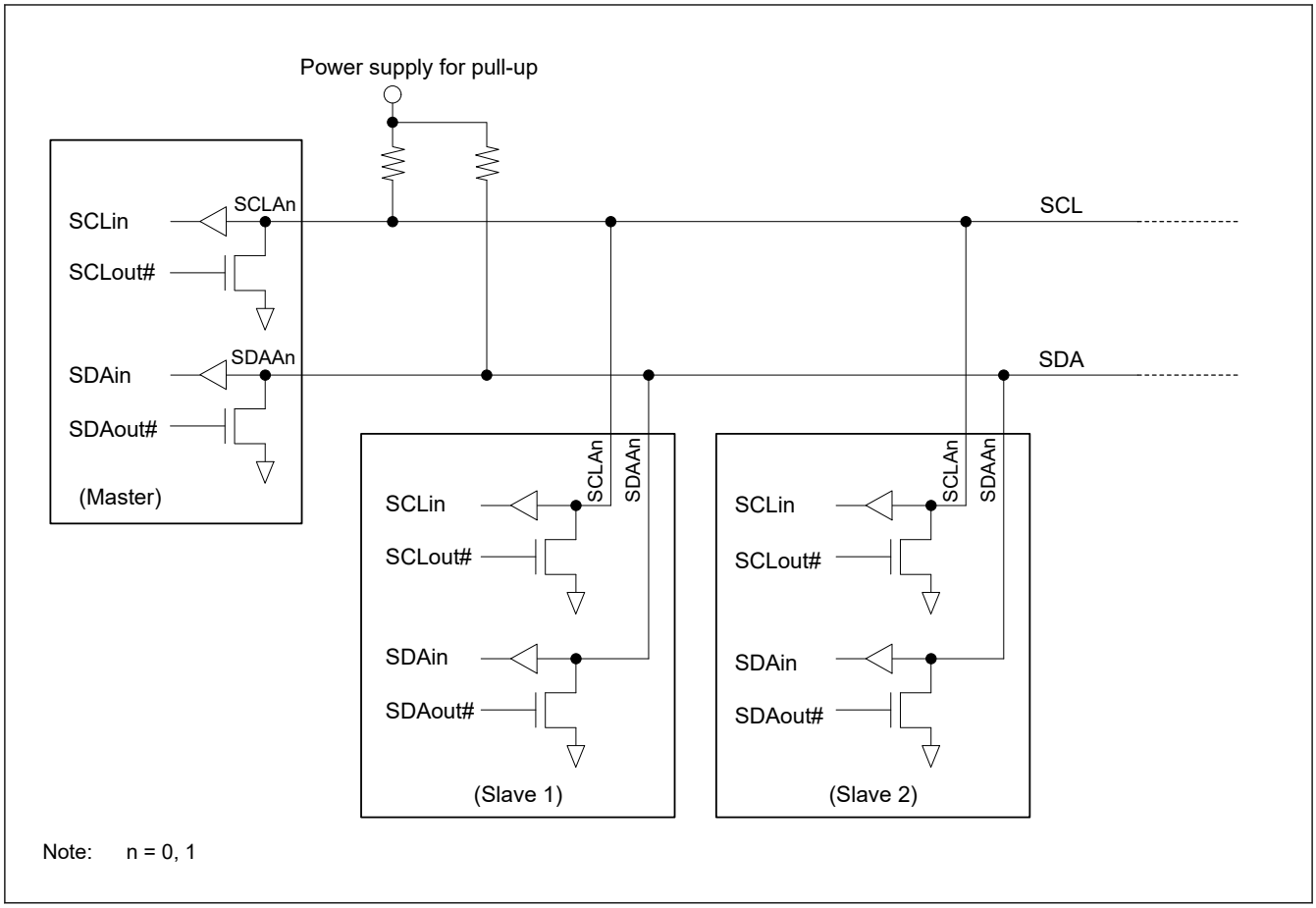


Figure 22.2 Example of the serial bus configuration using the I²C bus

22.2 Register Descriptions

22.2.1 IICAn : IICA Shift Register n (n = 0, 1)

Base address: IICA = 0x400A_3000

Offset address: 0x0000 + 0x8 × 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

Note: n = 0, 1

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 I²C bus interface (IICA) 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 mode, 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.

Note: n = 0, 1

22.2.2 SVAn : Slave Address Register n (n = 0, 1)

Base address: IICA = 0x400A_3000

Offset address: 0x0104 + 0x8 × 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 seven bits (A[6:0]) of the local address when in slave mode.

Rewriting to this register is prohibited while IICSn.STD = 1 (while the start condition is detected).

Note: n = 0, 1

22.2.3 IICCTLn0 : IICA Control Register n0 (n = 0, 1)

Base address: IICA = 0x400A_3000

Offset address: 0x0100 + 0x8 × 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 (End of transfer as master device)	R/W
1	STT*2 *3	Start Condition Trigger 0: Do not generate a start condition 1: When bus is released (in communication standby status when IICFn.IICBSY = 0): If this bit is set to 1, a start condition is generated (startup as the master mode). When a third party is communicating: ● When communication reservation function is enabled (IICFn.IICRSV = 0) 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
2	ACKE*4 *5	Acknowledgment Control 0: Disable acknowledgment 1: Enable acknowledgment. During the 9th clock cycle, the SDAAn line is set to low level.	R/W

Bit	Symbol	Function	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 communication standby status. 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 • IICCTLn0.SPT • IICSn.MSTS • IICSn.EXC • IICSn.COI • IICSn.TRC • IICSn.ACKD • IICSn.STD	R/W
7	IICE	I2C Operation Enable 0: Stop operation. Reset the IICA status register n (IICSn)*8. Stop internal operation. 1: Enable operation.	R/W

Note: n = 0, 1
Note 1. The SPT bit is always read as 0.
Note 2. The signal of this bit is invalid while IICE is 0.
Note 3. The STT bit is always read as 0.
Note 4. The signal of this bit is invalid while IICE 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 mode 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 IICE 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 I2C operations, set the timing of clock stretching, and set other I2C operations.

Note that bits SPIE, WTIM, and ACKE must be set while the setting of IICE 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 cycle 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 (IICAn).

If the I2C 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 (DFC bit of IICCTLn1 register = 1), a start condition is inadvertently detected immediately. In this case, set 1 to the LREL bit after enabling the I2C operation (IICE = 1).

Note: n = 0, 1

SPT bit (Stop Condition Trigger)

Cautions concerning set timing

- For master reception: Cannot be set to 1 during transfer.
  After setting the ACKE bit to 0, telling the slave device that it is the last to receive, this bit can be set to 1 only during the clock stretch period.
- For master transmission: During the period of Acknowledge, it may not be possible to generate stop conditions correctly.
  Therefore, set it during the clock stretch period that follows output of the 9th clock cycle.
- 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 9th clock cycle 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 9th clock cycle.
- 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: The read value of the SPT bit is always 0.

STT bit (Start Condition Trigger)

Cautions concerning set timing

- For master reception: Cannot be set to 1 during transfer.
  After setting the ACKE bit to 0, telling the slave device that it is the last to receive, this bit can be set to 1 only during the clock stretch period.
- For master transmission: During the period of Acknowledge, it may not be possible to generate start conditions correctly.
  Set to 1 during the clock stretch period that follows output of the 9th clock cycle.
- 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 9th 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 9th clock cycle during address transfer. In slave mode, when a local address is received, clock stretching is inserted at the falling edge of the 9th clock cycle after an acknowledge (ACK) is issued. However, in slave mode, when an extension code is received, clock stretching is inserted at the falling edge of the 8th 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 8th 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

Note: n = 0, 1

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 9th clock cycle 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

Note: n = 0, 1

LREL bit (Exit from Communications)

The communication standby status 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

Note: n = 0, 1

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

Note: n = 0, 1

22.2.4 IICSn : IICA Status Register n (n = 0, 1)

Base address: IICA = 0x400A_3000

Offset address: 0x0001 + 0x8 × 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 9th clock cycle).	R
4	COI	Detection of Matching Addresses 0: Addresses do not match. 1: Addresses match. Or, the all address match function is enabled.	R

Bit	Symbol	Function	R/W
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 mode status or communication standby status 1: Master mode communication status	R

Note: n = 0, 1

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 IICAn_TXRXI 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 (SPIE = 1) the interrupt generated by detecting a stop condition and read the IICSn register after the interrupt has been detected.

Note: n = 0, 1

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

Note: n = 0, 1

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

Note: n = 0, 1

ACKD bit (Detection of Acknowledge (ACK))

Condition for clearing (ACKD = 0)

- When a stop condition is detected
- At the rising edge of the next byte's first clock
- 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 SCLAn line's 9th clock cycle

Note: n = 0, 1

TRC bit (Detection of Transmit and Receive Status)

Condition for clearing (TRC = 0)

<Both master mode and slave mode>

- 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
- Not participating in communication (MSTS, EXC, COI = 0)

<Master mode>

- When 1 is output to the LSB of the first byte (transfer direction specification bit)

<Slave mode>

- When a start condition is detected
- When 0 is input to the LSB of the first byte (transfer direction specification bit)

Condition for setting (TRC = 1)

<Master mode>

- 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 mode>

- When 1 (slave transmission) is input to the LSB (transfer direction specification bit) of the first byte from the master mode (during address transfer)

Note 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 cycle 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 0.

Note: n = 0, 1

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 8th clock cycle).
- When an address is received while the all address match function is enabled (IICCTLn1.SVADIS = 1) (set at the rising edge of the 8th clock cycle).

Note: n = 0, 1

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 8th clock cycle).
- When an address is received while the all address match function is enabled (IICCTLn1.SVADIS = 1) (set at the rising edge of the 8th clock cycle).

Note: n = 0, 1

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)

- Loss in arbitration

Note: n = 0, 1

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

Note: n = 0, 1

22.2.5 IICFn : IICA Flag Register n (n = 0, 1)

Base address: IICA = 0x400A_3000

Offset address: 0x0002 + 0x8 × 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

Note: n = 0, 1

This register sets the operation mode of I^2C and indicates the state of the I^2C bus.

The IICCTLn0.STT clear flag (STCF) and I^2C bus status flag (IICBSY) bits are read-only.

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.

Write to the STCEN bit only when the operation is stopped (IICCTLn0.IICE = 0).

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.

Write to the IICRSV bit only when the operation is stopped (IICCTLn0.IICE = 0).

Note: n = 0, 1

IICRSV bit (Communication Reservation Function Disable Bit)

Condition for clearing (IICRSV = 0)

- Cleared by instruction
- Reset

Condition for setting (IICRSV = 1)

- Set by instruction

STCEN bit (Initial Start Enable Trigger)

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 (I²C 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

Note: n = 0, 1

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).

Note: n = 0, 1

22.2.6 IICCTLn1 : IICA Control Register n1 (n = 0, 1)

Base address: IICA = 0x400A_3000

Offset address: 0x0101 + 0x8 × 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)	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

Bit	Symbol	Function	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

Note: n = 0, 1

This register is used to set the operation mode of I^2C and detect the states of the SCLAn and SDAAn pins.

The CLD and DAD bits are read-only.

Set the IICCTLn1 register, except the WUP bit, while I^2C operation is disabled (IICCTLn0.IICE).

The fastest operation frequency of the IICA operation clock (f_{MCK}) is 20 MHz (max.).

Set PRS bit of the IICA control register n1 (IICCTLn1) to 1 only when the PCLKB exceeds 20 MHz.

Note the minimum PCLKB operation frequency when setting the transfer clock.

The minimum PCLKB operation frequency for I^2C bus interface (IICA) 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.)

Note: n = 0, 1

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

Note: n = 0, 1

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

Note: n = 0, 1

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, IICSn.COI is set to 1, and IICSn.EXC is set to 1.

For details about extension code reception, see section 22.3.13. Extension Code.

Note: n = 0, 1

WUP bit (Control of Address Match Wakeup)

To shift to Software Standby mode when WUP = 1, execute WFI instruction while SBYCR.SSBY bit is 1 at least three cycles of f_{MCK} after setting the WUP bit to 1 (see Table 22.5).

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)

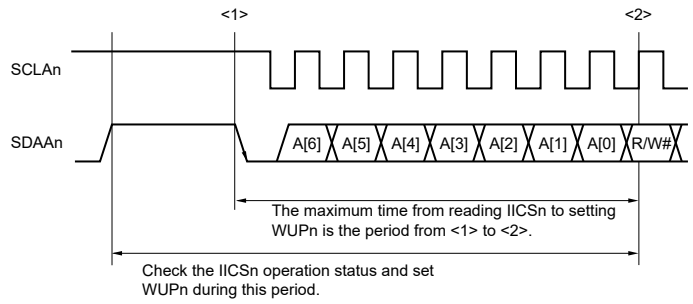
- Cleared by instruction (after address match, address reception with the all address match function enabled, or extension code reception)

Condition for setting (WUP = 1)

- Set by instruction (when the IICSn.MSTS, IICSn.EXC, and IICSn.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 22.3.

Note: n = 0, 1



Note: n = 0, 1

Figure 22.3 WUP bit setting period

22.2.7 IICWLn : IICA Low-level Width Setting Register n (n = 0, 1)

Base address: IICA = 0x400A_3000

Offset address: 0x0102 + 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	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 (IICA) and to control the SDAAn pin signal.

Set the IICWLn register while the I^2C operation is disabled (IICCTLn0.IICE).

For details about setting the IICWLn register, see section 22.3.2. Setting Transfer Clock Using IICWLn and IICWHn Registers. The data hold time is one-quarter of the time set by the IICWLn register.

Note: n = 0, 1

Note: The minimum serial clock cycle is (IICWLn + 1) + (IICWHn + 1)

22.2.8 IICWHn : IICA High-level Width Setting Register n (n = 0, 1)

Base address: IICA = 0x400A_3000

Offset address: 0x0103 + 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	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 (IICA) and to control the SDAAn pin signal.

Set the IICWHn register while the I^2C operation is disabled (IICCTLn0.IICE).

For the procedures for setting the transfer clock in master mode and the IICWL0 and IICWH0 registers in slave mode, refer to (1) Transfer clock setting in master mode and (2) Setting of IICWL0 and IICWH0 registers in slave mode, respectively.

Note: n = 0, 1

Note: The minimum serial clock cycle is (IICWLn + 1) + (IICWHn + 1)

22.2.9 Registers to Control the Port Function Multiplexed with the I2C I/O Pins

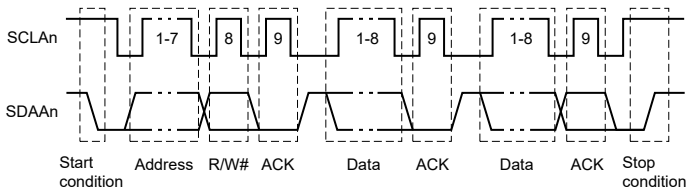
For information on how to set up the I/O ports, see section 16, I/O Ports.

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.

Note: n = 0, 1

22.3 I2C Bus Definitions and Control Methods

The following section describes the I2C bus's serial data communication format and the signals used by the I2C bus. Figure 22.4 shows the transfer timing for the "start condition", "address", "data", and "stop condition" output through the I2C bus's serial data bus.



Note: n = 0, 1

Figure 22.4 I2C bus serial data 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.

22.3.1 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 22.2 shows a serial bus configuration example using the I2C bus.

Note: n = 0, 1

22.3.2 Setting Transfer Clock Using IICWLn and IICWHn Registers

(1) Transfer clock setting in master mode

$$\text{Transfer clock} = \frac{f_{MCK}}{\text{IICWLn} + \text{IICWHn} + f_{MCK}(t_R + t_F)}$$

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
  $\mathrm{IICWLn} = \frac{0.52}{\text{Transfer clock}} \times f_{\mathrm{MCK}}$
  $\mathrm{IICWHn} = \left(\frac{0.48}{\text{Transfer clock}} - t_R - t_F\right) \times f_{\mathrm{MCK}}$
- When the normal mode
  $\mathrm{IICWLn} = \frac{0.47}{\text{Transfer clock}} \times f_{\mathrm{MCK}}$
  $\mathrm{IICWHn} = \left(\frac{0.53}{\text{Transfer clock}} - t_R - t_F\right) \times f_{\mathrm{MCK}}$
- When the fast mode plus
  $\mathrm{IICWLn} = \frac{0.50}{\text{Transfer clock}} \times f_{\mathrm{MCK}}$
  $\mathrm{IICWHn} = \left(\frac{0.50}{\text{Transfer clock}} - t_R - t_F\right) \times f_{\mathrm{MCK}}$

(2) Setting of IICWL0 and IICWH0 registers in slave mode

The fractional parts of all setting values are rounded up.

- When the fast mode
  $\mathrm{IICWLn} = 1.3\ \mu\mathrm{s} \times f_{\mathrm{MCK}}$
  $\mathrm{IICWHn} = (1.2\ \mu\mathrm{s} - t_R - t_F) \times f_{\mathrm{MCK}}$
- When the normal mode
  $\mathrm{IICWLn} = 4.7\ \mu\mathrm{s} \times f_{\mathrm{MCK}}$
  $\mathrm{IICWHn} = (5.3\ \mu\mathrm{s} - t_R - t_F) \times f_{\mathrm{MCK}}$
- When the fast mode plus
  $\mathrm{IICWLn} = 0.50\ \mu\mathrm{s} \times f_{\mathrm{MCK}}$
  $\mathrm{IICWHn} = (0.50\ \mu\mathrm{s} - t_R - t_F) \times f_{\mathrm{MCK}}$

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 0
IICWHn: IICA high-level width setting register 0
t_F: SDAAn and SCLAn signal falling times
t_R: SDAAn and SCLAn signal rising times
f_{MCK}: IICA operation clock frequency

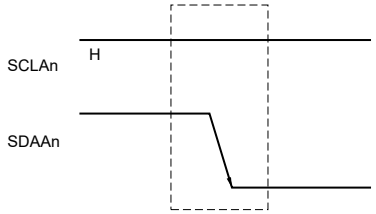
Note: n = 0, 1

Note: The minimum serial clock cycle is (IICWLn + 1) + (IICWHn + 1). Determine the values to be set in the IICWL0 and IICWH0 registers by considering the rise time (t_R) and fall time (t_F) of the SDAA0 and SCLA0 signals.

22.3.3 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 generates to the slave device when starting a serial transfer. When the device is used as a slave device, start conditions can be detected. Figure 22.5 shows the start conditions.



Note: n = 0, 1

Figure 22.5 Start conditions

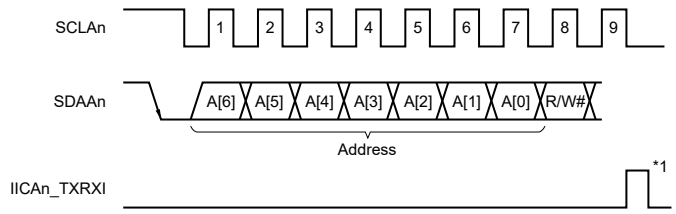
A start condition is output when STT bit of IICA control register n0 (IICCTLn0) is set to 1 after a stop condition has been detected (SPD bit of the IICA status register n (IICSn) = 1). When a start condition is detected, STD bit of the IICSn register is set to 1.

Note: n = 0, 1

22.3.4 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 through the bus lines. Therefore, each slave device connected through the bus lines must have a unique address. The slave devices include hardware that detects the start condition and checks whether 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 22.6 shows the address.



Note: n = 0, 1
Note 1. IICAn_TXRXI 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 22.6 Addresses

Addresses are output when a total of 8 bits consisting of the slave address and the transfer direction described in section 22.3.5. Transfer Direction Specification are written to the IICA shift register n (IICAn). The received addresses are written to the IICAn register.

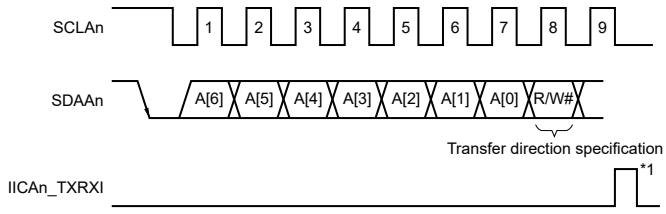
The slave address is assigned to the higher 7 bits of the IICAn register.

Note: n = 0, 1

22.3.5 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 22.7 shows the transfer direction specification.



Note: n = 0, 1
Note 1. IICAn_TXRXI 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 22.7 Transfer direction specification

22.3.6 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).

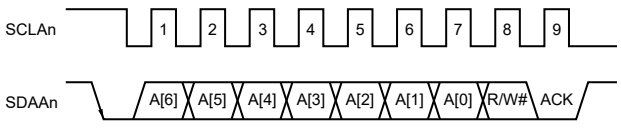
When the master device receives the last data item, it does not return ACK and instead generates a stop condition. If a slave device does not return ACK after receiving data, the master device outputs a stop condition or restart condition and stops transmission. If ACK is not returned, the possible causes 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 9th clock cycle (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 device can receive no more data during reception (IICCTLn0.TRC = 0) or does not require the next data item, then the slave device must inform the master device, by clearing the IICCTLn0.ACKE bit to 0, that it will not receive any more data.

When the master device 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 device at the transmission side that it does not require any more data (transmission will be stopped). Figure 22.8 shows the ACK.



Note: n = 0, 1

Figure 22.8 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 8th clock cycle of the SCLAn pin.

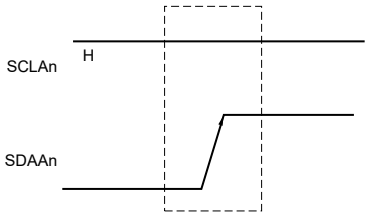
- When 9th cycle clock stretching is selected (IICCTLn0.WTIM = 1):
  ACK is generated if the IICCTLn0.ACKE bit is set to 1 in advance.

Note: n = 0, 1

22.3.7 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 device, stop conditions can be detected.

Figure 22.9 shows the stop conditions.



Note: n = 0, 1

Figure 22.9 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 IICAn_TXRXI is generated when SPIE bit of the IICCTLn0 register is set to 1.

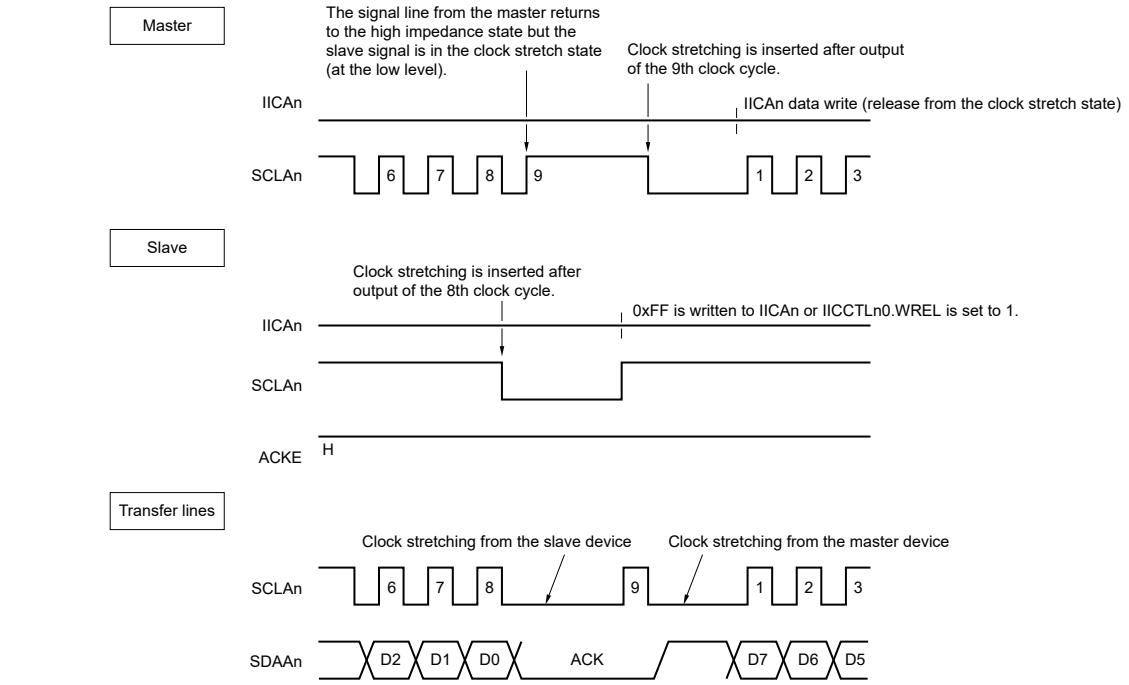
Note: n = 0, 1

22.3.8 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 (i.e., 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 22.10 shows the clock stretching.

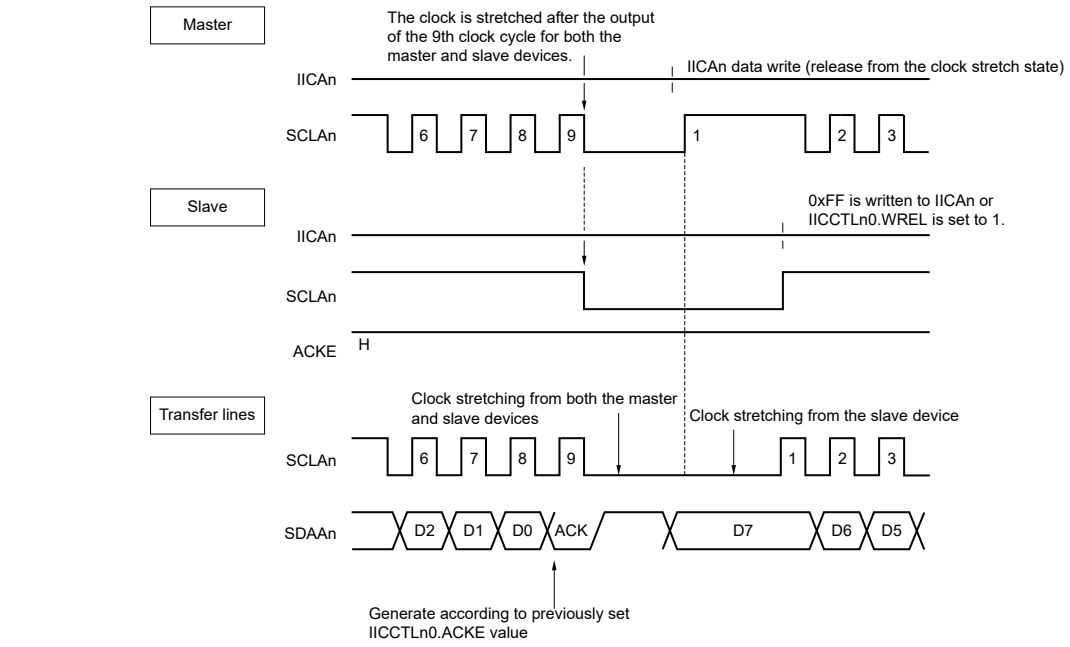
(1) When clock stretching is set for the 9th and 8th clock cycles for the master and slave devices, respectively (master: transmission, slave: reception, and IICCTLn0.ACKE = 1)



Note: n = 0, 1

Figure 22.10 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)



Note: n = 0, 1

Figure 22.11 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 by 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

Note: n = 0, 1

22.3.9 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 mode 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.

Note: n = 0, 1

22.3.10 Timing of Generation of the Interrupt Request Signal (IICAn_TXRXI) and Control of Clock Stretching

The setting of WTIM bit of IICA control register n0 (IICCTLn0) determines the timing by which IICAn_TXRXI is generated and controls clock stretching, as shown in Table 22.2.

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 22.2 IICAn_TXRXI generation timing and control of clock stretching

	During slave device operation			During master device operation		
WTIM	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 *3	9*2	9*2	9	9	9

Note: n = 0, 1

1. During address transmission/reception
   - Slave device operation: The timing of the interrupt and clock stretching depends on the conditions*1 *2 *3, regardless of the setting of the IICCTLn0.WTIM bit.
   - Master device operation: The interrupt and clock stretching occur at the falling edge of the 9th 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)*4
   - Setting SPT bit of the IICCTLn0 register (generating stop condition)*4

   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
   IICAn_TXRXI is generated when a stop condition is detected (only when IICCTLn0.SPIE = 1).

Note 1. The IICAn_TXRXI 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, IICAn_TXRXI occurs at the falling edge of the 8th clock cycle.

However, if the address does not match after restart, IICAn_TXRXI is generated at the falling edge of the 9th clock cycle, but clock stretching does not occur.

Note 2. When the Slave Address Register n (SVAn) does not match the received address, the address match function is disabled, and an extended code has not been received, neither IICA0_TXRXI nor clock stretching will occur.

Note 3. When the WTIM bit is set to 1, for a slave device that has received an extension code or an address while the all address match function is enabled, IICAn_TXRXI and clock stretching occur at the falling edge of both the eighth and ninth clock cycles.

Note 4. Master mode only.

Note: n = 0, 1

22.3.11 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 (IICAn_TXRXI) 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.

Note: n = 0, 1

22.3.12 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.

Note: n = 0, 1

22.3.13 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 (IICAn_TXRXI) is issued at the falling edge of the 8th clock cycle.
   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 device 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 IICAn_TXRXI occurs at the falling edge of the 8th clock cycle.
   - 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 mode, 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 communication standby status for the next communication operation.

Note: n = 0, 1

Table 22.3 shows the bit definitions for the major extension codes.

Table 22.3 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^2 bus specifications issued by NXP Semiconductors for details of extension codes other than those described above.

22.3.14 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 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 22.3.10. Timing of Generation of the Interrupt Request Signal (IICAn_TXRXI) and Control of Clock Stretching.

Figure 22.12 shows the arbitration timing example.

Note: n = 0, 1

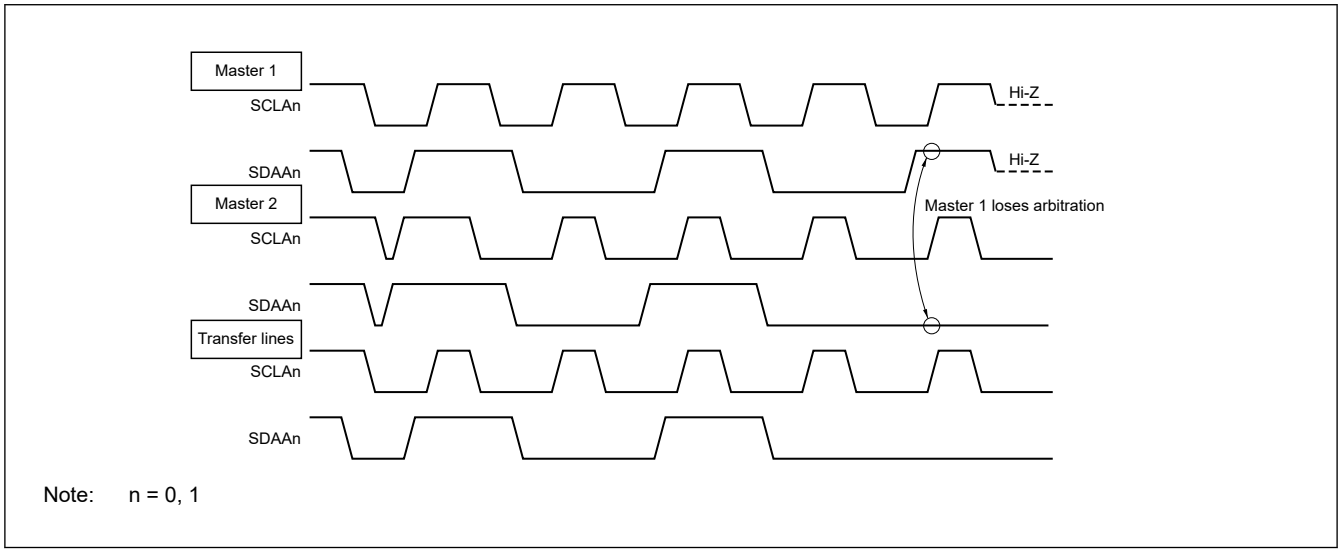


Figure 22.12 Arbitration timing example

Table 22.4 shows the status during arbitration and when interrupt requests are generated.

Table 22.4 Status during arbitration and interrupt request generation timing (1 of 2)

Status during arbitration	Interrupt request generation timing
During address transmission	At falling edge of 8th or 9th clock cycle 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	

Table 22.4 Status during arbitration and interrupt request generation timing (2 of 2)

Status during arbitration	Interrupt request generation timing
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 8th or 9th clock cycle 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 cycle following byte transfer*1
When SCLAn is at low level while attempting to generate a restart condition	

Note: n = 0, 1
Note 1. When the IICCTLn0.WTIM = 1, an interrupt request occurs at the falling edge of the 9th clock cycle. When IICCTLn0.WTIM = 0, the extension code's slave address 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 cycle.
Note 2. When there is a chance that arbitration will occur, set IICCTLn0.SPIE = 1 for master device operation.

22.3.15 Wakeup Function

The I^2C bus slave function is a function that generates an interrupt request signal (IICAn_TXRXI) 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 IICAn_TXRXI signal from occurring when addresses do not match while the all address match function is disabled.

When a start condition is detected, wakeup standby state is set. Even a master device that has generated a start condition enters the wakeup standby state while transmitting an address because the master device may become a slave device 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 (IICAn_TXRXI) 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 22.5 shows the step for setting IICCTLn1.WUP = 1 and Table 22.6 shows the step for setting IICCTLn1.WUP = 0 upon an address match (or when the all address match function is enabled).

Note: n = 0, 1

Table 22.5 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>	Stop instruction execution	—

Note: n = 0, 1

Table 22.6 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 IICAn_TXRXI = 1
	<3>	Disable operation of address match wakeup function	Clear IICCTLn1.WUP bit.
	<4>	Wait	Waits for 5 cycles of f_{MCK}
	<5>	Reading IICSn	—
	<6>	Executes next processing	Executes processing corresponding to the operation to be executed after checking the operation state of I^2C bus interface (IICA).

Note: n = 0, 1

Use the following flows to perform the processing to release the Software Standby mode other than by an interrupt request signal (IICAn_TXRXI) generated from I^2C bus interface (IICA).

- When operating next IIC communication as master device: Flow shown in Table 22.7.
- When operating next IIC communication as slave device:
  When released by IICAn_TXRXI interrupt: Same as the flow in Table 22.6.
  When released by other than IICAn_TXRXI interrupt: Wait for IICAn_TXRXI interrupt with IICCTLn1.WUP left set to 1.

Note: n = 0, 1

Table 22.7 When operating as master device after releasing Software Standby mode other than by IICAn_TXRXI

Step		Process	Detail
When operating as master device after releasing Software Standby mode other than by IICAn_TXRXI	<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>	Stop instruction execution	Software standby state
	<6>	Releasing Software Standby mode	Releases Software Standby mode by an interrupt other than IICAn_TXRXI
	<7>	Disable operation of address match wakeup function	Clear IICCTLn1.WUP bit.
	<8>	Interrupt check	Wait until IICAn_TXRXI = 1
	<9>	Reading IICSn	—
	<10>	Executes next processing	Executes processing corresponding to the operation to be executed after checking the operation state of I^2C bus interface (IICA).

Note: n = 0, 1

22.3.16 Communication Reservation

(1) When communication reservation function is enabled (IICRSV bit of IICA flag register 0 (IICF0) = 0)

To start master 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 device nor slave device operation

- While the all address match function is disabled, when an extension code is received and slave device 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)

When setting bit 1 (STT) of the IICCTLn0 register while in a non-participatory state on the bus, after the bus is released (upon detecting a stop condition), it automatically generates a start condition and enters communication standby status.

Setting the SPIE bit of the IICCTLn0 register to 1, and detecting the release of the bus upon interrupt request (IICAn_TXRXI) (detecting a stop condition), after writing the address to the IICA shift register n (IICAn), automatically initiates communication as a 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 (communication standby status) … 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.

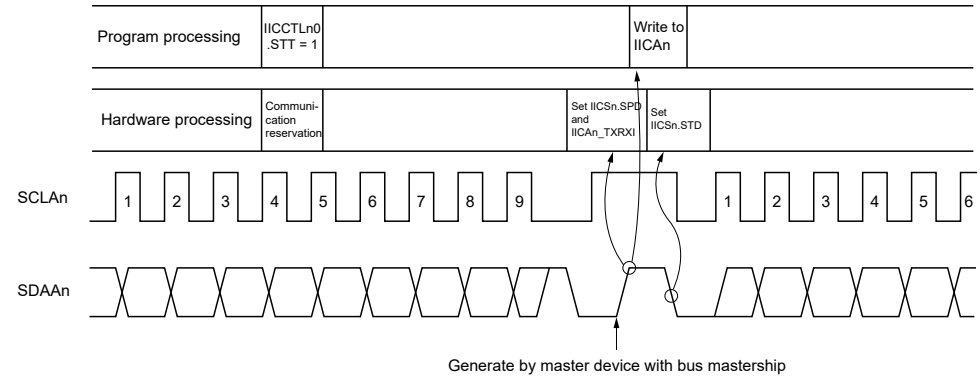
Wait time from setting IICCTLn0.STT = 1 to checking the IICSn.MSTS flag:

(IICWLn setting value + IICWHn setting value + 4) / f_{MCK} + t_F × 2

Note: IICWLn: IICA low-level width setting register 0
IICWHn: IICA high-level width setting register 0
t_F: SDAAn and SCLAn signal falling times
f_{MCK}: IICA operation clock frequency

Note: n = 0, 1

Figure 22.13 shows the communication reservation timing.

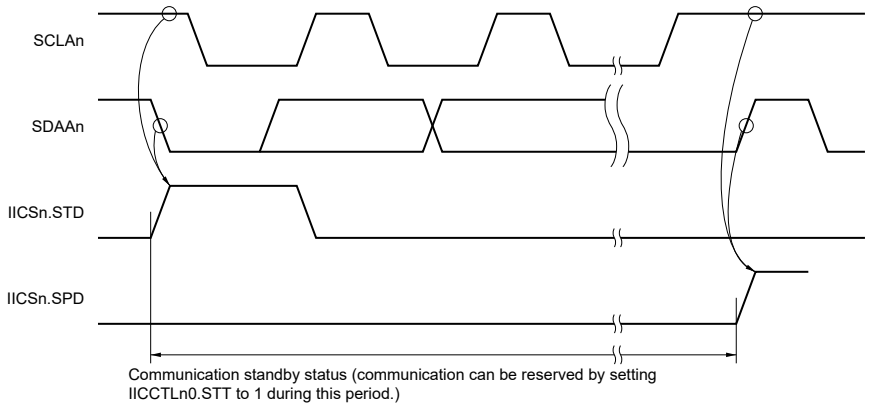


Note: n = 0, 1

Figure 22.13 Communication reservation timing

Communication reservations are accepted with the timing shown in Figure 22.14. 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.

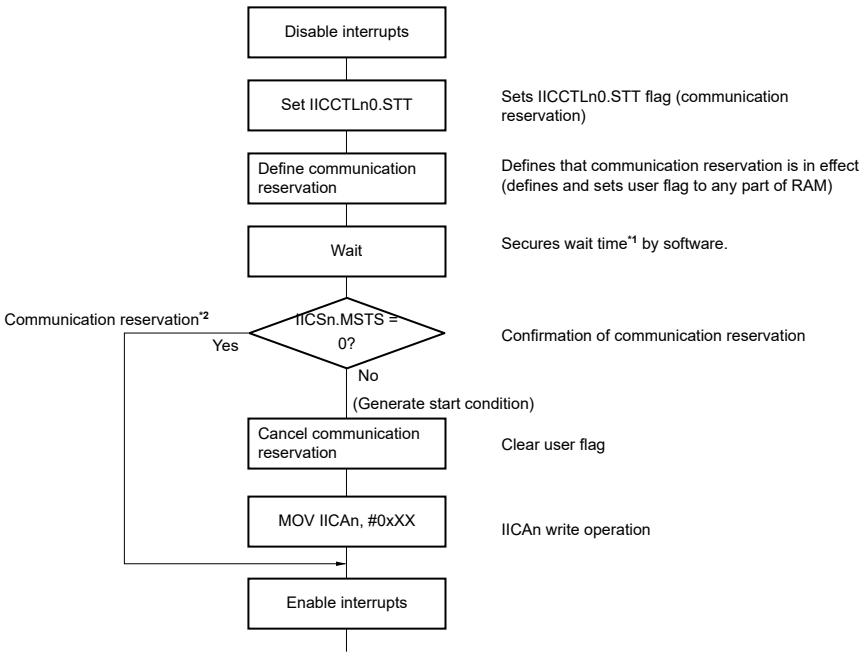
Note: n = 0, 1



Note: n = 0, 1

Figure 22.14 Timing for accepting communication reservations

Figure 22.15 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 22.15 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 in communication and is not participating in this communication, this request is rejected and a start condition is not generated. In this case, non-participation on the bus includes the following two states.

- When arbitration results in neither master device nor slave device operation

- While the all address match function is disabled, when an extension code is received and slave device 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.

Note: n = 0, 1

22.3.17 Usage Notes

1. When IICFn.STCEN = 0
   The generation of a start condition is based upon the detection of a stop condition. Therefore, changing from a mode in which no stop condition has been detected, first generate a stop condition to release the bus.
   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 communication mode, first generate a stop condition to release the bus, then perform master communication.
   When using multiple master devices, it is not possible to perform master 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
   The generation of a start condition is independent with stop condition detection, so firstly it is necessary to confirm with the bus has been released.
   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 (IICAn_TXRXI) 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.

Note: n = 0, 1

22.3.18 Communication Operations

The following shows three operation procedures with the flowchart.

1. Master device operation in single-master system
   The flowchart when using this product as the master device in a single master system is shown in Figure 22.16.

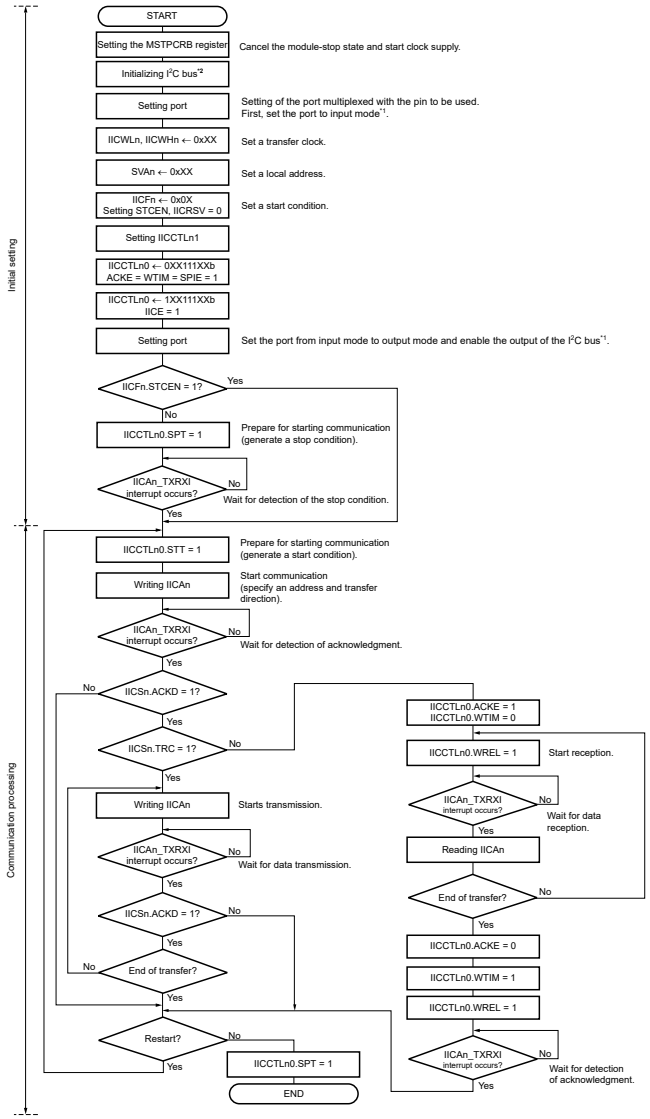
This flowchart is broadly divided into the initial settings and communication processing. Execute the initial settings at startup. If communication with the slave device is required, prepare the communication and then execute communication processing.

2. Master device 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 looses in arbitration and is specified as the slave device is omitted in Figure 22.17, and only the processing as the master device is shown. Execute the initial settings at startup to take part in a communication. Then, wait for the communication request as the master device or wait for the specification as the slave device. The actual communication is performed in the communication processing, and it supports the transmission and reception with the slave device and the arbitration with other master devices.
3. Slave device operation
   An example of when this product is used as the I^2C bus slave device is shown in Figure 22.21 and Figure 22.22.
   When used as the slave, operation is started by an interrupt. Execute the initial settings at startup, then wait for the IICAn_TXRXI interrupt occurrence (communication waiting). When an IICAn_TXRXI 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.

Note: n = 0, 1

(1) Master device 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 22.2.9. Registers to Control the Port Function Multiplexed with the I2C I/O Pins.

Note 2. Release (SCLAn and SDAAn pins = high level) the I2 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 22.16 Master device operation in single-master system

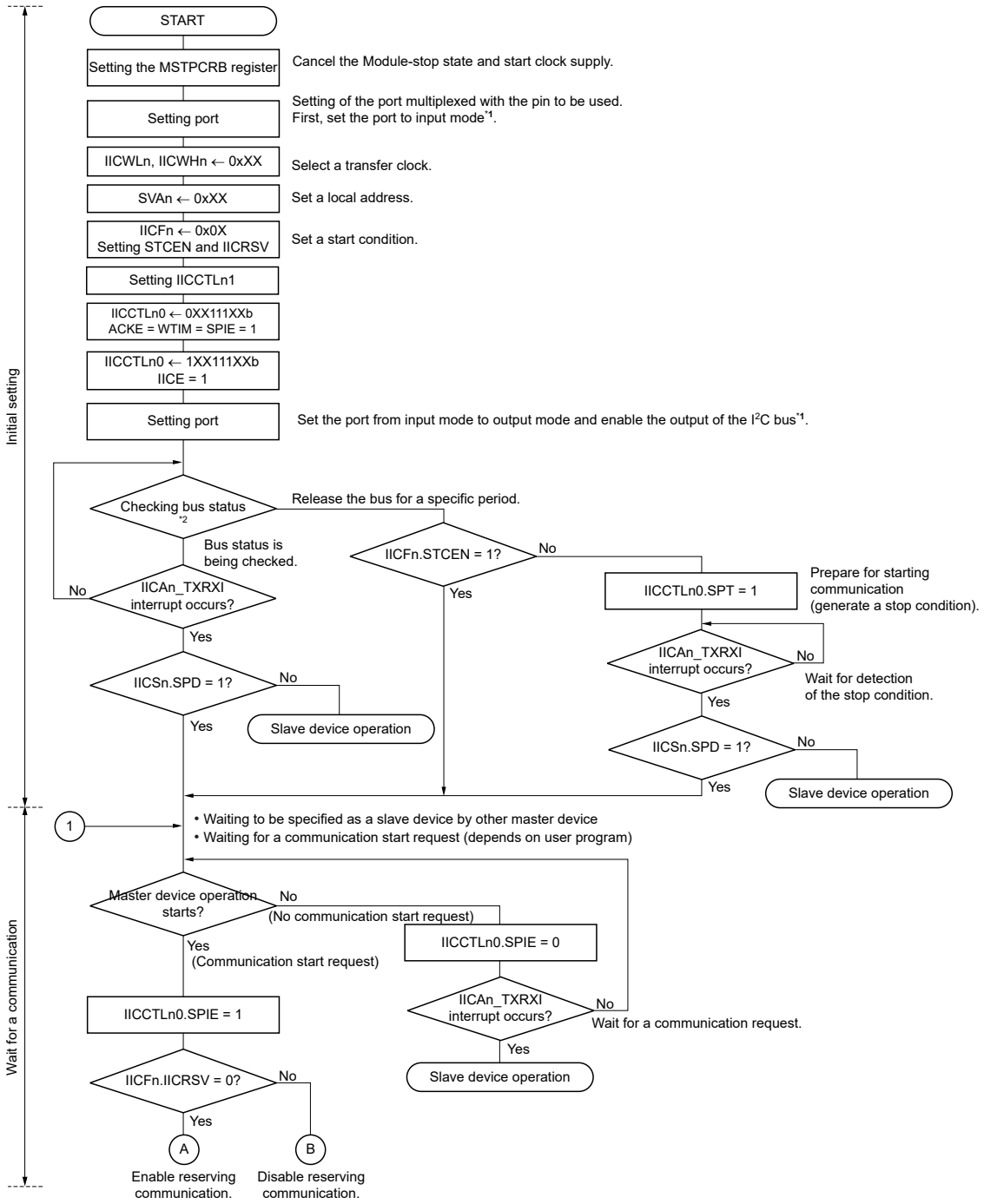
(2) Master device operation in multi-master system

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 device in a multi-master system, read the IICSn.MSTS bit each time interrupt IICAn_TXRXI occurred to check the arbitration result.

To use the device as a slave device 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 IICAn_TXRXI occurred, and determine the processing to be performed next.

Note: n = 0, 1

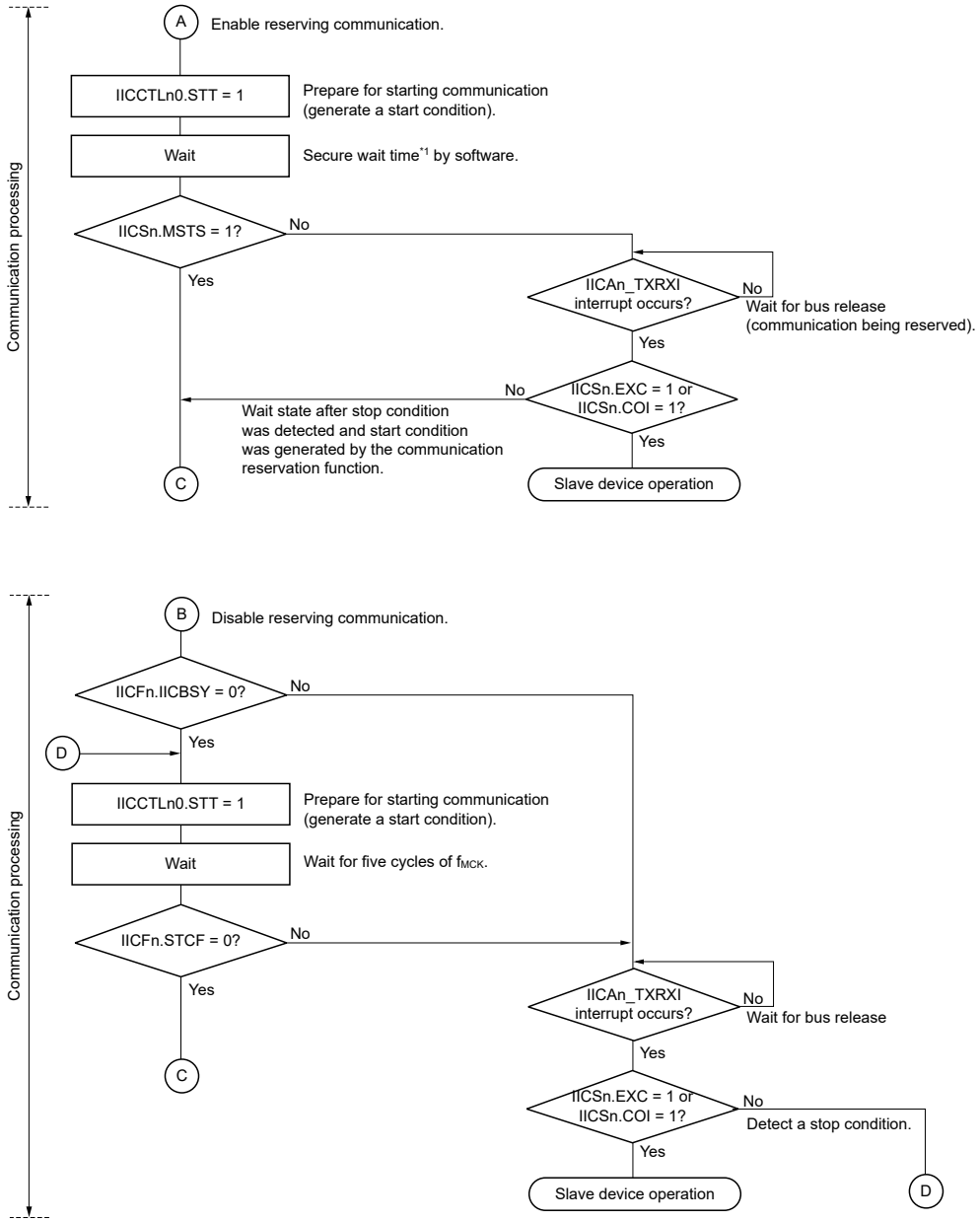


Note: n = 0, 1

Note 1. section 22.2.9. Registers to Control the Port Function Multiplexed with the I2C I/O Pins.

Note 2. Confirm that the bus is released (IICCTLn1.CLD bit = 1, IICCTLn1.DAD bit = 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 22.17 Master device operation in multi-master system (1/3)

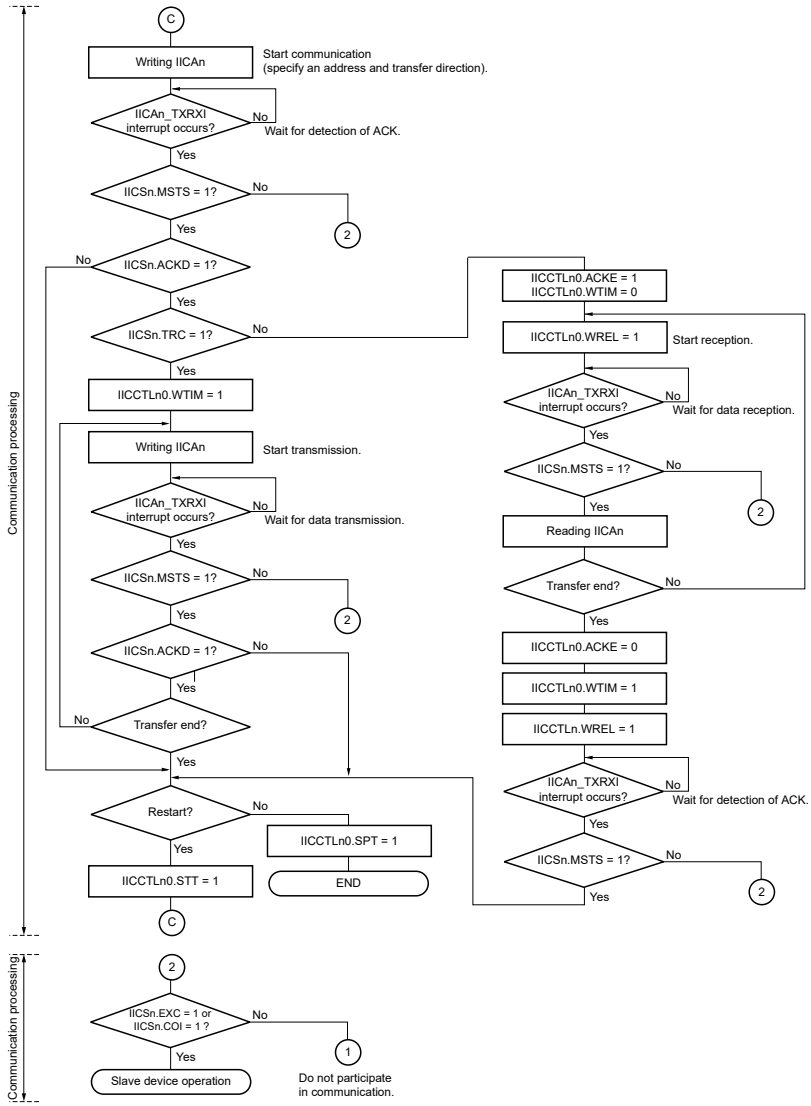


Note: n = 0, 1

Note 1. The wait time is calculated as follows.
(IICWLn setting value + IICWHn setting value + 4) / f_{MCK} + t_F × 2

Note 2. t_F: SDAAn and SCLAn signal falling times
f_{MCK}: IICA operation clock frequency

Figure 22.18 Master device operation in multi-master system (2/3)



Note: n = 0, 1

Figure 22.19 Master device operation in multi-master system (3/3)

(3) Slave device operation

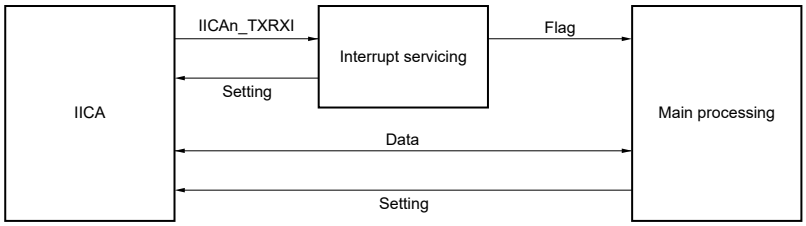
The processing procedure of the slave device operation is as follows.

Basically, the slave device operation is event-driven. Therefore, processing by the IICAn_TXRXI 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 IICAn_TXRXI interrupt processing only performs status transition processing, and that actual data communication is performed by the main processing.

Figure 22.20 shows an interface configuration with the main processor in slave device operation.

Note: n = 0, 1



Note: n = 0, 1

Figure 22.20 Interface configuration with the main processor in slave device operation

Therefore, data communication processing is performed by preparing the following three flags and passing them to the main processing instead of IICAn_TXRXI.

<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 device, address mismatch)

<2> Ready flag

This flag indicates that data communication is enabled. Its function is the same as the IICAn_TXRXI 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 IICSn.TRC bit.

The main processing of the slave device operation is explained next.

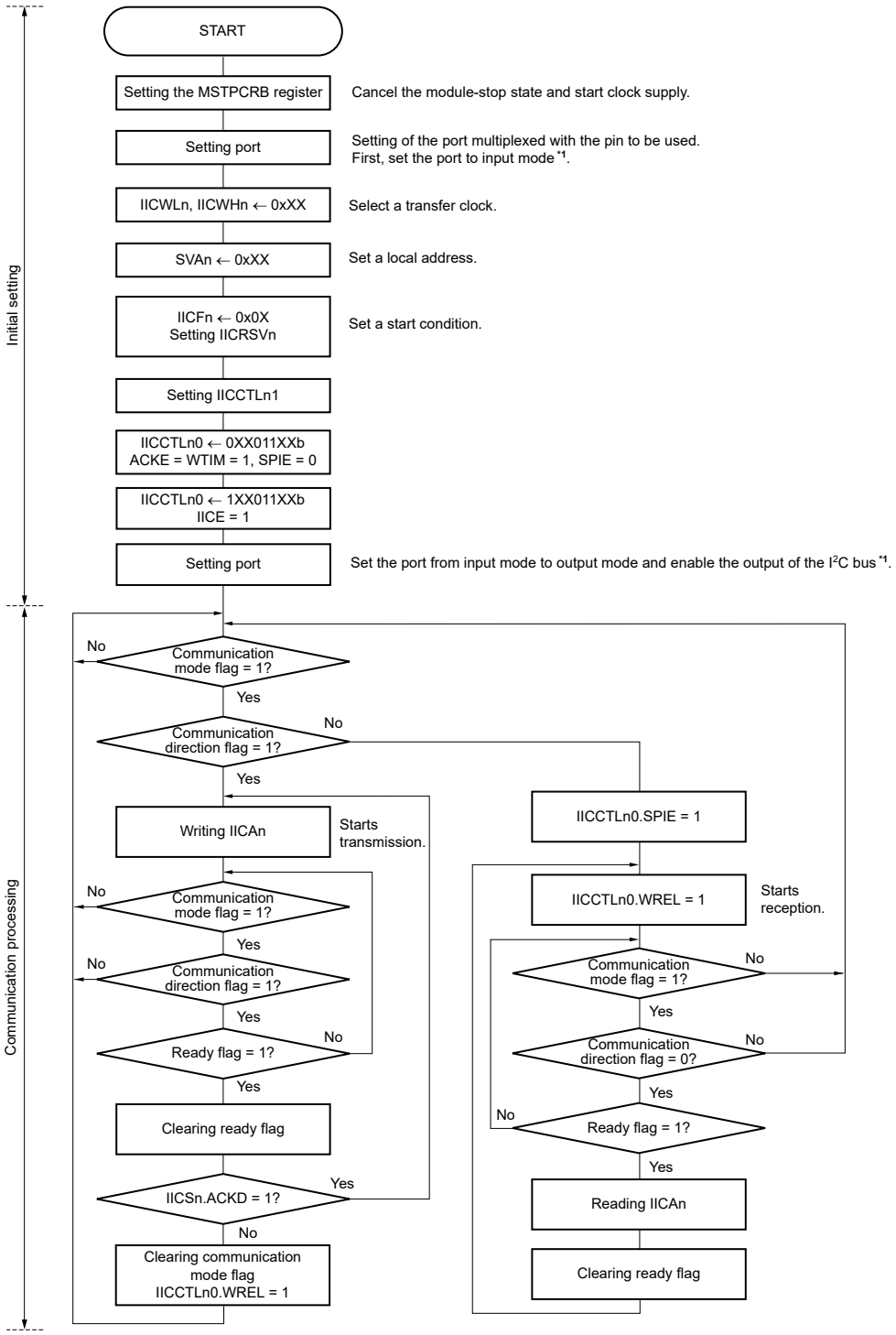
Start I^2C bus interface (IICA) and wait until communication is enabled. When communication is enabled, execute communication by using the communication mode flag and ready flag (processing of the stop condition and start condition is performed by an interrupt. Here, check the status by using the flags).

The transmission operation is repeated until the master device no longer returns ACK. If ACK is not returned from the master device, 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 device 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: n = 0, 1

Note 1. See section 22.2.9. Registers to Control the Port Function Multiplexed with the I2C I/O Pins.

Figure 22.21 Slave device operation flowchart (1)

An example of the processing procedure of the slave device operation with the IICAn_TXRXI interrupt is explained below (processing is performed assuming that the all address match function is disabled and no extension code is used). The IICAn_TXRXI 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.

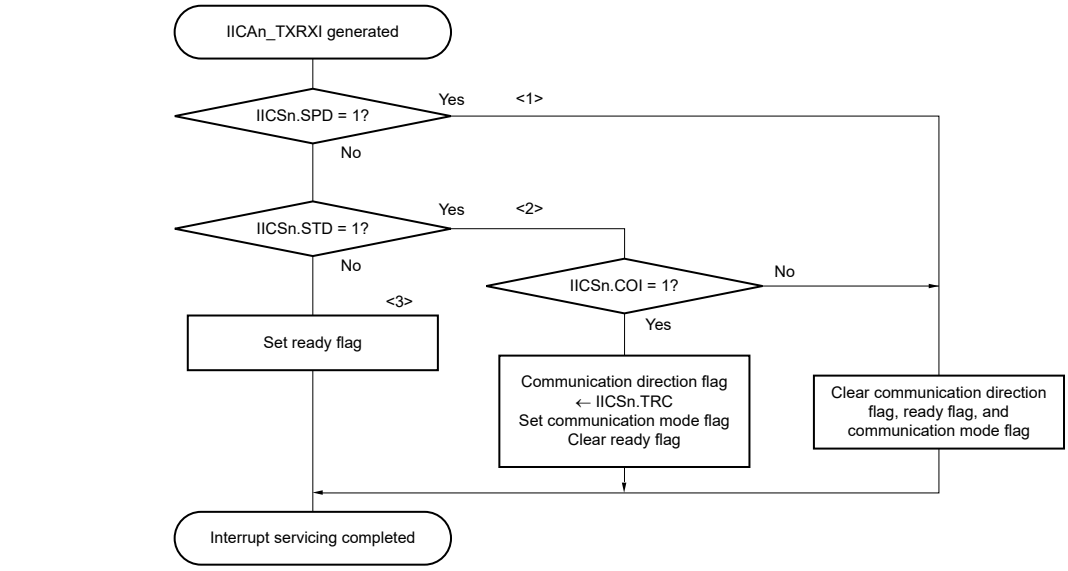
If the address matches, the communication mode is set, wait is canceled, and processing returns from the interrupt (the ready flag is cleared).

<3> For data transmit/receive, only the ready flag is set. Processing returns from the interrupt with the I^2C bus remaining in the communication standby status.

Note: n = 0, 1

Note: <1> to <3> above correspond to <1> to <3> in Figure 22.22.

Figure 22.22 shows the interrupt flowchart for slave device operation.



Note: n = 0, 1

Figure 22.22 Slave device operation flowchart (2)

22.3.19 Timing of I^2C Interrupt Request Signal (IICAn_TXRXI) Occurrence

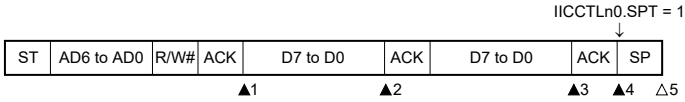
The timing of transmitting or receiving data and generation of interrupt request signal IICAn_TXRXI, and the value of the IICA status register n (IICSn) when the IICAn_TXRXI signal is generated are shown in Figure 22.23 to Figure 22.62.

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

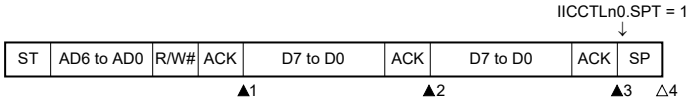


▲1: IICSn = 1000x110b
▲2: IICSn = 1000x000b
▲3: IICSn = 1000x000b (set the IICCTLn0.WTIM bit to 1)*1
▲4: IICSn = 1000xx00b (set the IICCTLn0.SPT bit to 1)
△5: IICSn = 00000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
x: Don't care
Note 1. To generate a stop condition, set the IICCTLn0.WTIM bit to 1 and change the timing for generating the IICAn_TXRXI interrupt request signal.

Figure 22.23 Master device operation reception/transmission (IICCTLn0.WTIM = 0)

2. When IICCTLn0.WTIM = 1



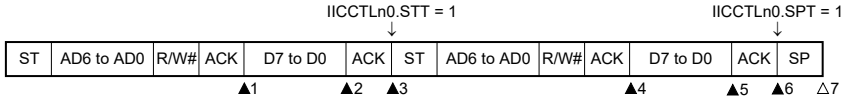
▲1: IICSn = 1000x110b
▲2: IICSn = 1000x100b
▲3: IICSn = 1000xx00b (set the IICCTLn0.SPT bit to 1)
△4: IICSn = 00000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
x: Don't care

Figure 22.24 Master device operation reception/transmission (IICCTLn0.WTIM = 1)

(b) Start → Address → Data → Start → Address → Data → Stop (restart)

1. When IICCTLn0.WTIM = 0



▲1: IICSn = 1000x110b
▲2: IICSn = 1000x000b (set the IICCTLn0.WTIM bit to 1)*1
▲3: IICSn = 1000xx00b (clear the IICCTLn0.WTIM bit to 0*2, set the IICCTLn0.STT bit to 1)
▲4: IICSn = 1000x110b
▲5: IICSn = 1000x000b (set the IICCTLn0.WTIM bit to 1)*3
▲6: IICSn = 1000xx00b (set the IICCTLn0.SPT bit to 1)
△7: IICSn = 00000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
x: Don't care
Note 1. To generate a start condition, set the IICCTLn0.WTIM bit to 1 and change the timing for generating the IICAn_TXRXI 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 IICAn_TXRXI interrupt request signal.

Figure 22.25 Master device operation restart (IICCTLn0.WTIM = 0)

2. When IICCTLn0.WTIM = 1

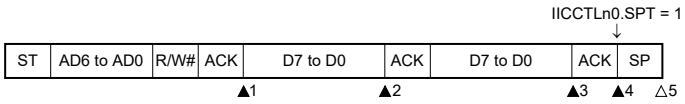
▲1: IICSn = 1000x110b
▲2: IICSn = 1000xx00b (set the IICCTLn0.STT bit to 1)
▲3: IICSn = 1000x110b
▲4: IICSn = 1000xx00b (set the IICCTLn0.SPT bit to 1)
△5: IICSn = 00000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
x: Don't care

Figure 22.26 Master device operation restart (IICCTLn0.WTIM = 1)

(c) Start → Code → Data → Data → Stop (extension code transmission)

1. When IICCTLn0.WTIM = 0

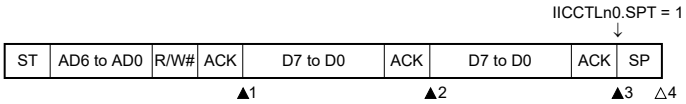


▲1: IICSn = 1010x110b
▲2: IICSn = 1010x000b
▲3: IICSn = 1010x000b (set the IICCTLn0.WTIM bit to 1)*1
▲4: IICSn = 1010xx00b (set the IICCTLn0.SPT bit to 1)
△5: IICSn = 00000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
x: Don't care
Note 1. To generate a stop condition, set the IICCTLn0.WTIM bit to 1 and change the timing for generating the IICAn_TXRXI interrupt request signal.

Figure 22.27 Master device operation extension code transmission (IICCTLn0.WTIM = 0)

2. When IICCTLn0.WTIM = 1



▲1: IICSn = 1010x110b
▲2: IICSn = 1010x100b
▲3: IICSn = 1010xx00b (set the IICCTLn0.SPT bit to 1)
△4: IICSn = 00000001b

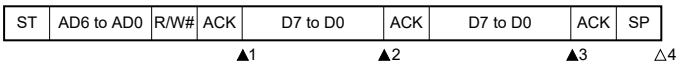
Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
x: Don't care

Figure 22.28 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

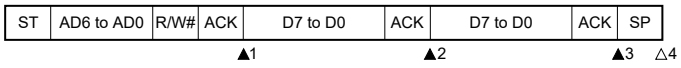


▲1: IICSn = 0001x110b
▲2: IICSn = 0001x000b
▲3: IICSn = 0001x000b
△4: IICSn = 00000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
x: Don't care

Figure 22.29 Slave device operation slave address data reception (IICCTLn0.WTIM = 0)

2. When IICCTLn0.WTIM = 1



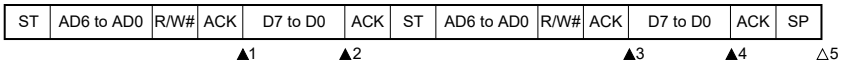
▲1: IICSn = 0001x110b
▲2: IICSn = 0001x100b
▲3: IICSn = 0001xx00b
△4: IICSn = 00000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
x: Don't care

Figure 22.30 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, matches with SVAn, the all address match function is disabled)

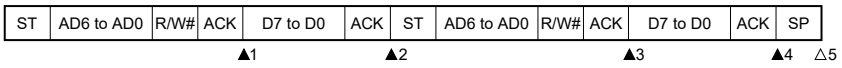


▲1: IICSn = 0001x110b
▲2: IICSn = 0001x000b
▲3: IICSn = 0001x110b
▲4: IICSn = 0001x000b
△5: IICSn = 00000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
x: Don't care

Figure 22.31 Slave device operation after normal access, matches with SVAn (IICCTLn0.WTIM = 0)

2. When IICCTLn0.WTIM = 1 (after restart, matches with SVAn, the all address match function is disabled)



▲1: IICSn = 0001x110b
▲2: IICSn = 0001xx00b
▲3: IICSn = 0001x110b
▲4: IICSn = 0001xx00b
△5: IICSn = 00000001b

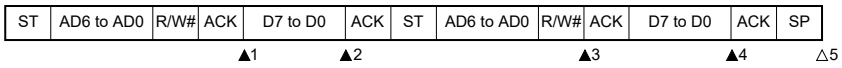
Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
x: Don't care

Figure 22.32 Slave device operation after normal access, matches with SVAn (IICCTLn0.WTIM = 1)

(c) Start → Address → Data → Start → Code → Data → Stop

1. When IICCTLn0.WTIM = 0

(after restart, does not match address (extension code, the all address match function is disabled))



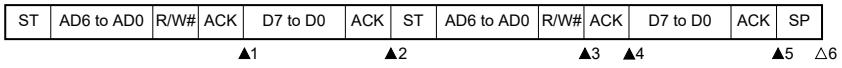
▲1: IICSn = 0001x110b
▲2: IICSn = 0001x000b
▲3: IICSn = 0010x010b
▲4: IICSn = 0010x000b
△5: IICSn = 00000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
x: Don't care

Figure 22.33 Slave device operation after normal access, matches the extension code (IICCTLn0.WTIM = 0)

2. When IICCTLn0.WTIM = 1

(after restart, does not match address (extension code, the all address match function is disabled))



▲1: IICSn = 0001x110b
▲2: IICSn = 0001xx00b
▲3: IICSn = 0010x010b
▲4: IICSn = 0010x110b
▲5: IICSn = 0010xx00b
△6: IICSn = 00000001b

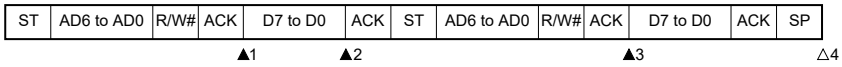
Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
x: Don't care

Figure 22.34 Slave device operation after normal access, matches the extension code (IICCTLn0.WTIM = 1)

(d) Start → Address → Data → Start → Address → Data → Stop

1. When IICCTLn0.WTIM = 0

(after restart, does not match address (not extension code, the all address match function is disabled))



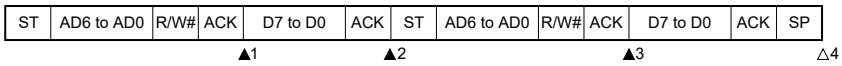
▲1: IICSn = 0001x110b
▲2: IICSn = 0001x000b
▲3: IICSn = 00000x10b
△4: IICSn = 00000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
x: Don't care

Figure 22.35 Slave device operation after normal access, does not matches (IICCTLn0.WTIM = 0)

2. When IICCTLn0.WTIM = 1

(after restart, does not match address (not extension code, the all address match function is disabled))



▲1: IICSn = 0001x110b
▲2: IICSn = 0001xx00b
▲3: IICSn = 00000x10b
△4: IICSn = 00000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
x: Don't care

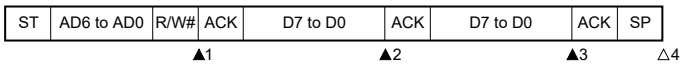
Figure 22.36 Slave device operation after normal access, does not matches (IICCTLn0.WTIM = 1)

(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

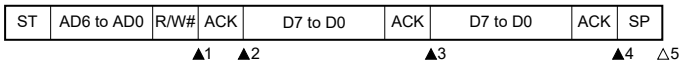


▲1: IICSn = 0010x010b
▲2: IICSn = 0010x000b
▲3: IICSn = 0010x000b
△4: IICSn = 00000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
x: Don't care

Figure 22.37 Slave device operation receiving extension code (IICCTLn0.WTIM = 0)

2. When IICCTLn0.WTIM = 1



▲1: IICSn = 0010x010b
▲2: IICSn = 0010x110b
▲3: IICSn = 0010x100b
▲4: IICSn = 0010xx00b
△5: IICSn = 00000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
x: Don't care

Figure 22.38 Slave device operation receiving extension code (IICCTLn0.WTIM = 1)

(b) Start → Code → Data → Start → Address → Data → Stop

1. When IICCTLn0.WTIM = 0 (after restart, matches with SVAn, the all address match function is disabled)

▲1: IICSn = 0010x010b
▲2: IICSn = 0010x000b
▲3: IICSn = 0001x110b
▲4: IICSn = 0001x000b
△5: IICSn = 00000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
x: Don't care

Figure 22.39 Slave device operation after code access, matches with SVAn (IICCTLn0.WTIM = 0)

2. When IICCTLn0.WTIM = 1 (after restart, matches with SVAn, the all address match function is disabled)

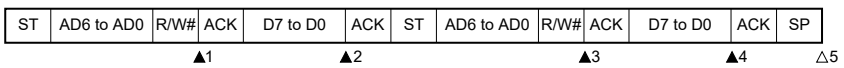
▲1: IICSn = 0010x010b
▲2: IICSn = 0010x110b
▲3: IICSn = 0010xx00b
▲4: IICSn = 0001x110b
▲5: IICSn = 0001xx00b
△6: IICSn = 00000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
x: Don't care

Figure 22.40 Slave device operation after code access, matches with SVAn (IICCTLn0.WTIM = 1)

(c) Start → Code → Data → Start → Code → Data → Stop

1. When IICCTLn0.WTIM = 0 (after restart, extension code reception, the all address match function is disabled)



▲1: IICSn = 0010x010b
▲2: IICSn = 0010x000b
▲3: IICSn = 0010x010b
▲4: IICSn = 0010x000b
△5: IICSn = 00000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
x: Don't care

Figure 22.41 Slave device operation after code access, matches the extension code (IICCTLn0.WTIM = 0)

2. When IICCTLn0.WTIM = 1 (after restart, extension code reception, the all address match function is disabled)

▲1: IICSn = 0010x010b
▲2: IICSn = 0010x110b
▲3: IICSn = 0010xx00b
▲4: IICSn = 0010x010b
▲5: IICSn = 0010x110b
▲6: IICSn = 0010xx00b
△7: IICSn = 00000001b

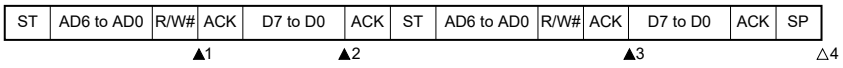
Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
x: Don't care

Figure 22.42 Slave device operation after code access, matches the extension code (IICCTLn0.WTIM = 1)

(d) Start → Code → Data → Start → Address → Data → Stop

1. When IICCTLn0.WTIM = 0

(after restart, does not match address (not extension code, the all address match function is disabled))



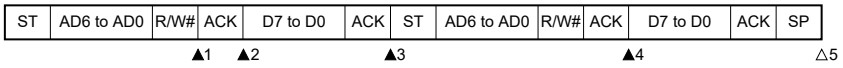
▲1: IICSn = 0010x010b
▲2: IICSn = 0010x000b
▲3: IICSn = 00000x10b
△4: IICSn = 00000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
x: Don't care

Figure 22.43 Slave device operation after code access, does not matches (IICCTLn0.WTIM = 0)

2. When IICCTLn0.WTIM = 1

(after restart, does not match address (not extension code, the all address match function is disabled))



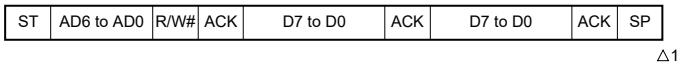
▲1: IICSn = 0010x010b
▲2: IICSn = 0010x110b
▲3: IICSn = 0010xx00b
▲4: IICSn = 00000x10b
△5: IICSn = 00000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
x: Don't care

Figure 22.44 Slave device operation after code access, does not matches (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 22.45 Operation without communication

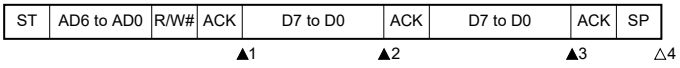
(5) Arbitration loss operation (operation as slave mode after arbitration loss)

When the device is used as a master device in a multi-master system, read the IICSn.MSTS bit each time interrupt request signal IICAn_TXRXI has occurred to check the arbitration result.

Note: n = 0, 1

(a) When arbitration loss occurs during transmission of slave address data

1. When IICCTLn0.WTIM = 0

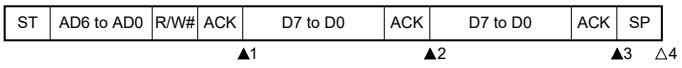


▲1: IICSn = 0101x110b
▲2: IICSn = 0001x000b
▲3: IICSn = 0001x000b
△4: IICSn = 00000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
x: Don't care

Figure 22.46 Arbitration loss when sending slave address data (IICCTLn0.WTIM = 0)

2. When IICCTLn0.WTIM = 1



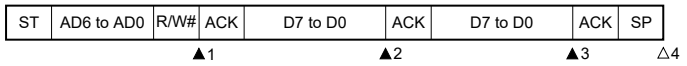
▲1: IICSn = 0101x110b
▲2: IICSn = 0001x100b
▲3: IICSn = 0001xx00b
△4: IICSn = 00000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
x: Don't care

Figure 22.47 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

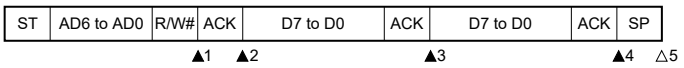


▲1: IICSn = 0110x010b
▲2: IICSn = 0010x000b
▲3: IICSn = 0010x000b
△4: IICSn = 00000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
x: Don't care

Figure 22.48 Arbitration loss when sending extension code (IICCTLn0.WTIM = 0)

2. When IICCTLn0.WTIM = 1



▲1: IICSn = 0110x010b
▲2: IICSn = 0010x110b
▲3: IICSn = 0010x100b
▲4: IICSn = 0010xx00b
△5: IICSn = 00000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
x: Don't care

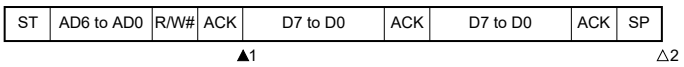
Figure 22.49 Arbitration loss when sending extension code (IICCTLn0.WTIM = 1)

(6) Operation when arbitration loss occurs (no communication after arbitration loss)

When the device is used as a master device in a multi-master system, read the IICSn.MSTS bit each time interrupt request signal IICAn_TXRXI has occurred to check the arbitration result.

Note: n = 0, 1

(a) When IICCTLn0.WTIM = 1 (arbitration loss occurs during transmission of slave address data)

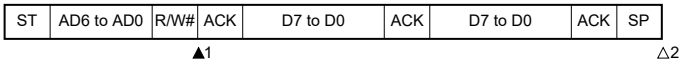


▲1: IICSn = 01000110b
△2: IICSn = 00000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1

Figure 22.50 Operation when arbitration loss occurs during slave address data transmission (IICCTLn0.WTIM = 1)

(b) When arbitration loss occurs during transmission of extension code (the all address match function is disabled)



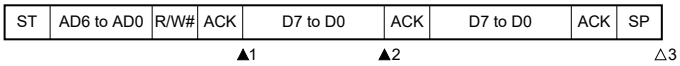
▲1: IICSn = 0110x010b
Sets IICCTLn0.LREL = 1 by software
△2: IICSn = 00000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
x: Don't care

Figure 22.51 Operation when arbitration loss occurs during extension code transmission

(c) When arbitration loss occurs during transfer of data

1. When IICCTLn0.WTIM = 0

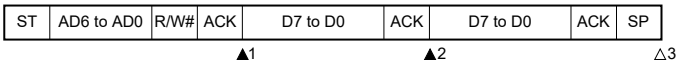


▲1: IICSn = 10001110b
▲2: IICSn = 01000000b
△3: IICSn = 00000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1

Figure 22.52 Operation when arbitration loss occurs during data transfer (IICCTLn0.WTIM = 0)

2. When IICCTLn0.WTIM = 1



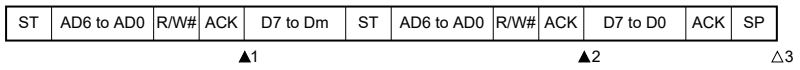
▲1: IICSn = 10001110b
▲2: IICSn = 01000100b
△3: IICSn = 00000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1

Figure 22.53 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)

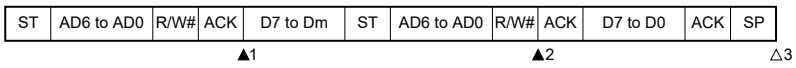


▲1: IICSn = 1000x110b
▲2: IICSn = 01000110b
△3: IICSn = 00000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
x: Don't care
m = 6 to 0

Figure 22.54 Operation when arbitration loss occurs due to restart during data transfer (not extension code)

2. Extension code (the all address match function is disabled)

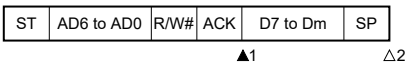


▲1: IICSn = 1000x110b
▲2: IICSn = 01100010b
Sets IICCTLn0.LREL = 1 by software
△3: IICSn = 00000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
x: Don't care
m = 6 to 0

Figure 22.55 Operation when arbitration loss occurs due to restart during data transfer (extension code)

(e) When loss occurs due to stop condition during data transfer



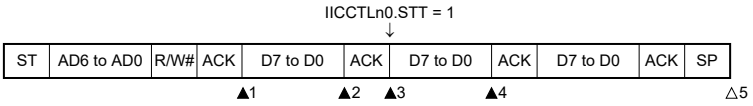
▲1: IICSn = 10000110b
△2: IICSn = 01000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
x: Don't care
m = 6 to 0

Figure 22.56 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

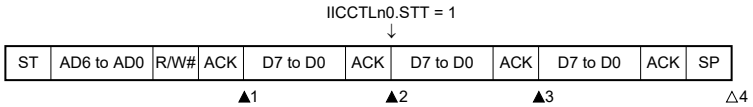


▲1: IICSn = 1000x110b
▲2: IICSn = 1000x000b (set the IICCTLn0.WTIM bit to 1)
▲3: IICSn = 1000x100b (clear the IICCTLn0.WTIM bit to 0)
▲4: IICSn = 01000000b
△5: IICSn = 00000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
x: Don't care

Figure 22.57 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



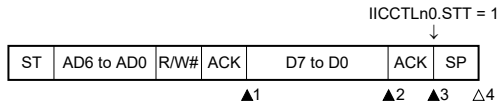
▲1: IICSn = 1000x110b
▲2: IICSn = 1000x100b (set the IICCTLn0.STT bit to 1)
▲3: IICSn = 01000100b
△4: IICSn = 00000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
x: Don't care

Figure 22.58 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 = 1000x110b
▲2: IICSn = 1000x000b (set the IICCTLn0.WTIM bit to 1)
▲3: IICSn = 1000xx00b (set the IICCTLn0.STT bit to 1)
△4: IICSn = 01000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
x: Don't care

Figure 22.59 Operations when arbitration loss occurs due to stop condition when attempting to generate a restart condition (IICCTLn0.WTIM = 0)

2. When IICCTLn0.WTIM = 1

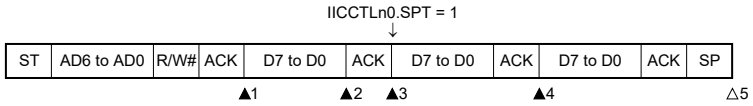
▲1: IICSn = 1000x110b
▲2: IICSn = 1000xx00b (set the IICCTLn0.STT bit to 1)
△3: IICSn = 01000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
x: Don't care

Figure 22.60 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

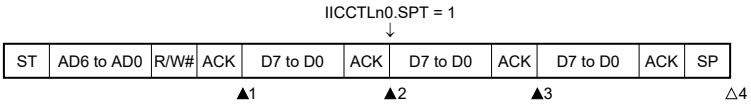


▲1: IICSn = 1000x110b
▲2: IICSn = 1000x000b (Set the IICCTLn0.WTIM bit to 1)
▲3: IICSn = 1000x100b (Clear the IICCTLn0.WTIM bit to 0)
▲4: IICSn = 01000100b
△5: IICSn = 00000001b

Note: n = 0, 1
Note: ▲: Always generated
△: Generated only when IICCTLn0.SPIE = 1
x: Don't care

Figure 22.61 Operations when arbitration loss occurs due to S low-Level data when attempting to generate a stop condition (IICCTLn0.WTIM = 0)

2. When IICCTLn0.WTIM = 1



▲1: IICSn = 1000x110b
▲2: IICSn = 1000x100b (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
x: Don't care

Figure 22.62 Operations when arbitration loss occurs due to S low-Level data when attempting to generate a stop condition (IICCTLn0.WTIM = 1)

22.4 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 TRC bit (bit 3 of the IICA status register n (IICSn)), which specifies the data transfer direction, and then starts serial communication with the slave device.

(1) Example of Master device to Slave device Communications (When the master device and the slave device insert clock stretching on the 9th cycle.) and (2) Example of Slave device to Master device Communications (8th Cycle Clock Stretching Is Selected for the Master device and 9th Cycle Clock Stretching Is Selected for the Slave device) 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.

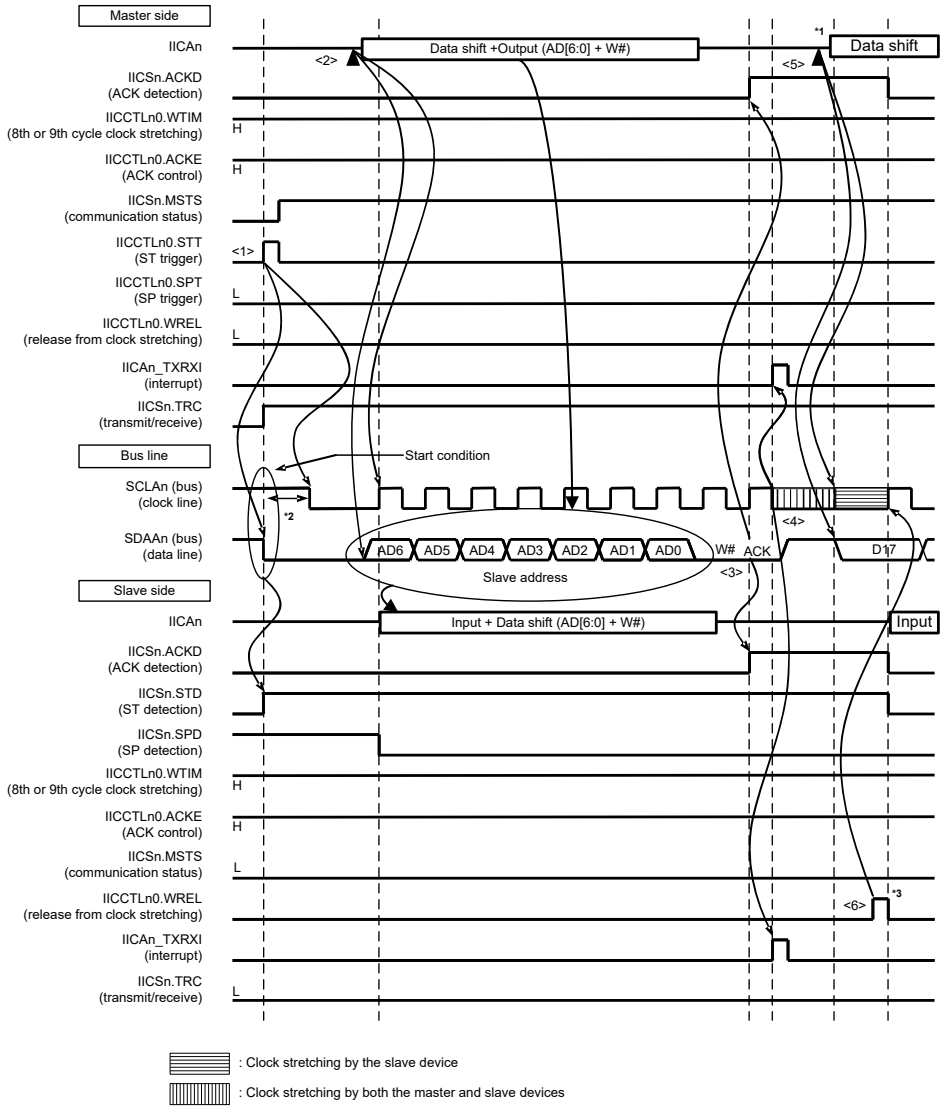
Data input through the SDAAn pin is captured into IICAn at the rising edge of SCLAn.

Note: n = 0, 1

In the timing diagrams described in this section, it is assumed that the all address match function is disabled.

(1) Example of Master device to Slave device Communications (When the master device and the slave device insert clock stretching on the 9th cycle.)

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 mode, write 0xFF to IICAn or set the IICCTLn0.WREL bit.

Figure 22.63 Example of master device to slave device communications (When the master device and the slave device insert clock stretching on the 9th cycle.) (1/4)

The meanings of <1> to <6> in Figure 22.63 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 mode 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.

Note: n = 0, 1

<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 cycle.

<4> The master device issues an interrupt (IICAn_TXRXI: end of address transmission) at the falling edge of the 9th clock cycle. The slave device with the address matching the transmitted slave address sets the clock stretch state (SCLAn = 0) and issues an interrupt (IICAn_TXRXI: address match).

Note: n = 0, 1

<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 IICAn_TXRXI interrupt (address match) and does not set the clock stretch state.

The master device, however, issues the IICAn_TXRXI interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

Note: n = 0, 1

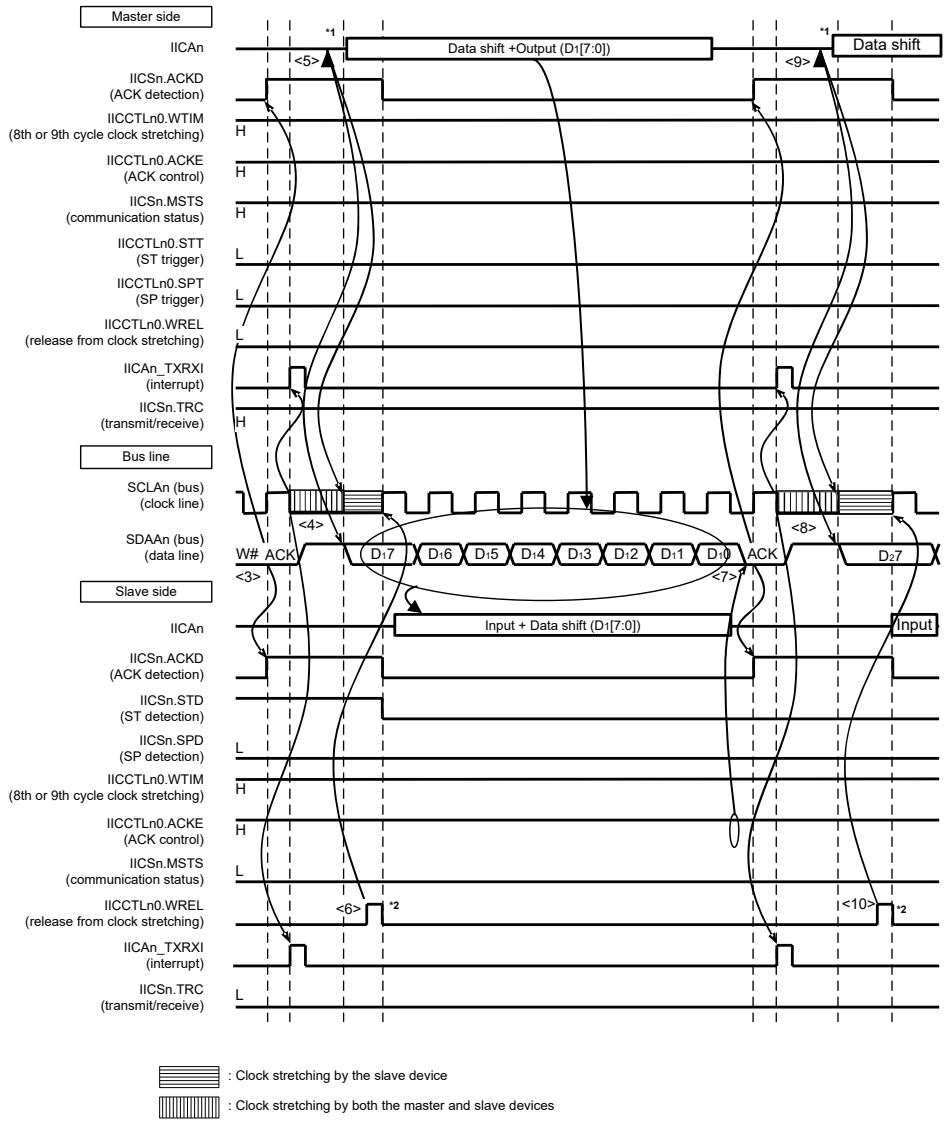
Note: <1> to <15> in (1) Example of Master device to Slave device Communications (When the master device and the slave device insert clock stretching on the 9th cycle.) represent the entire procedure for communicating data using the I^2C bus.

Figure 22.63 shows the processing from <1> to <6>,

Figure 22.64 shows the processing from <3> to <10>, and

Figure 22.65 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 mode, write 0xFF to IICAn or set the IICCTLn0.WREL bit.

Figure 22.64 Example of master device to slave device communications (When the master device and the slave device insert clock stretching on the 9th cycle.) (2/4)

The meanings of <3> to <10> in Figure 22.64 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 cycle.

<4> The master device issues an interrupt (IICAn_TXRXI: end of address transmission) at the falling edge of the 9th clock cycle. The slave device with the address matching the transmitted slave address sets the clock stretch state (SCLAn = 0) and issues an interrupt (IICAn_TXRXI: address match).

Note: n = 0, 1

<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 cycle.

<8> The master device and slave device set the clock stretch state (SCLAn = 0) at the falling edge of the 9th clock cycle, and both the master device and slave device issue an interrupt (IICAn_TXRXI: end of transfer).

Note: n = 0, 1

<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 IICAn_TXRXI interrupt (address match) and does not set the clock stretch state.

The master device, however, issues the IICAn_TXRXI interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

Note: n = 0, 1

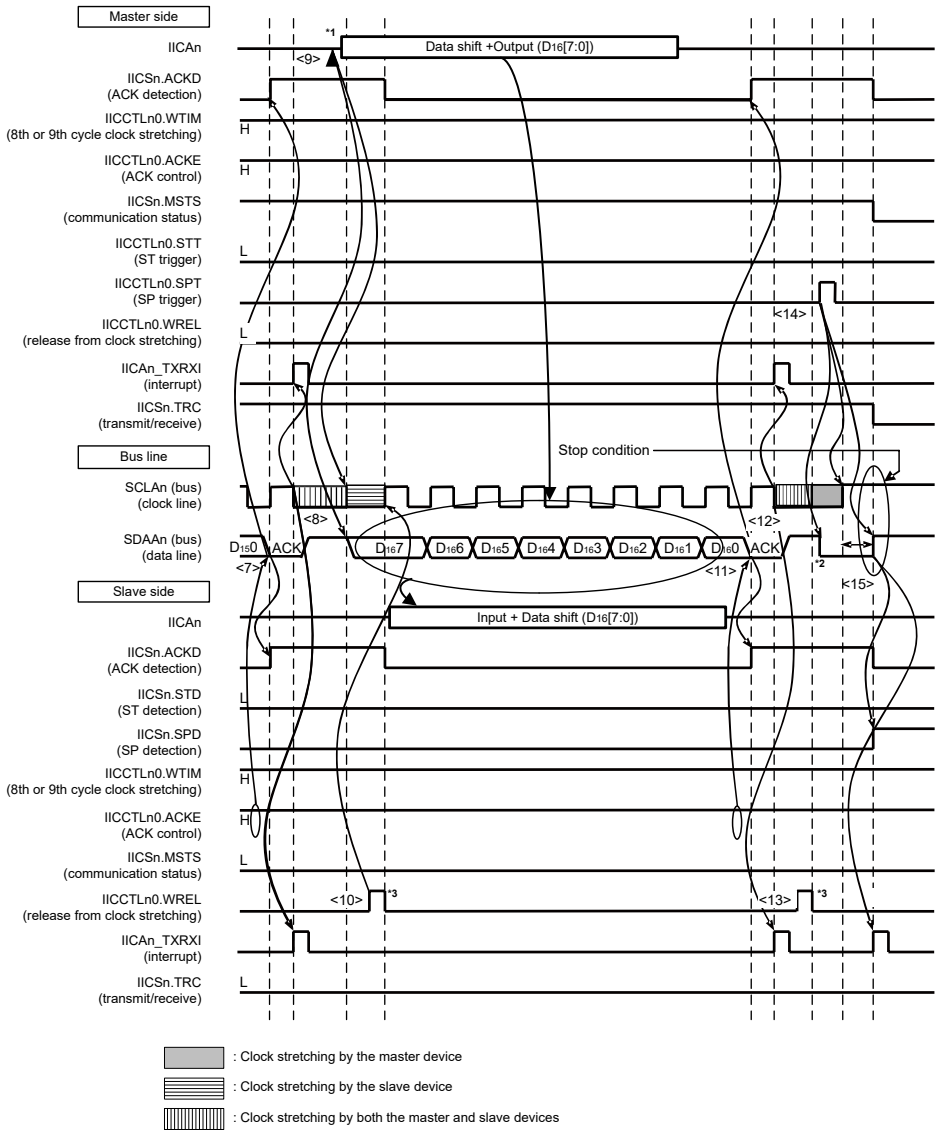
Note: <1> to <15> in (1) Example of Master device to Slave device Communications (When the master device and the slave device insert clock stretching on the 9th cycle.) represent the entire procedure for communicating data using the I^2C bus.

Figure 22.63 shows the processing from <1> to <6>,

Figure 22.64 shows the processing from <3> to <10>, and

Figure 22.65 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 mode, write 0xFF to IICAn or set the IICCTLn0.WREL bit.

Figure 22.65 Example of master device to slave device communications (When the master device and the slave device insert clock stretching on the 9th cycle.) (3/4)

The meanings of <7> to <15> in Figure 22.65 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 cycle.

<8> The master device and slave device set the clock stretch state (SCLAn = 0) at the falling edge of the 9th clock cycle, and both the master device and slave device issue an interrupt (IICAn_TXRXI: end of transfer).

Note: n = 0, 1

<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 cycle.

<12>The master device and slave device set the clock stretch state (SCLAn = 0) at the falling edge of the 9th clock cycle, and both the master device and slave device issue an interrupt (IICAn_TXRXI: 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 (IICAn_TXRXI: stop condition).

Note: n = 0, 1

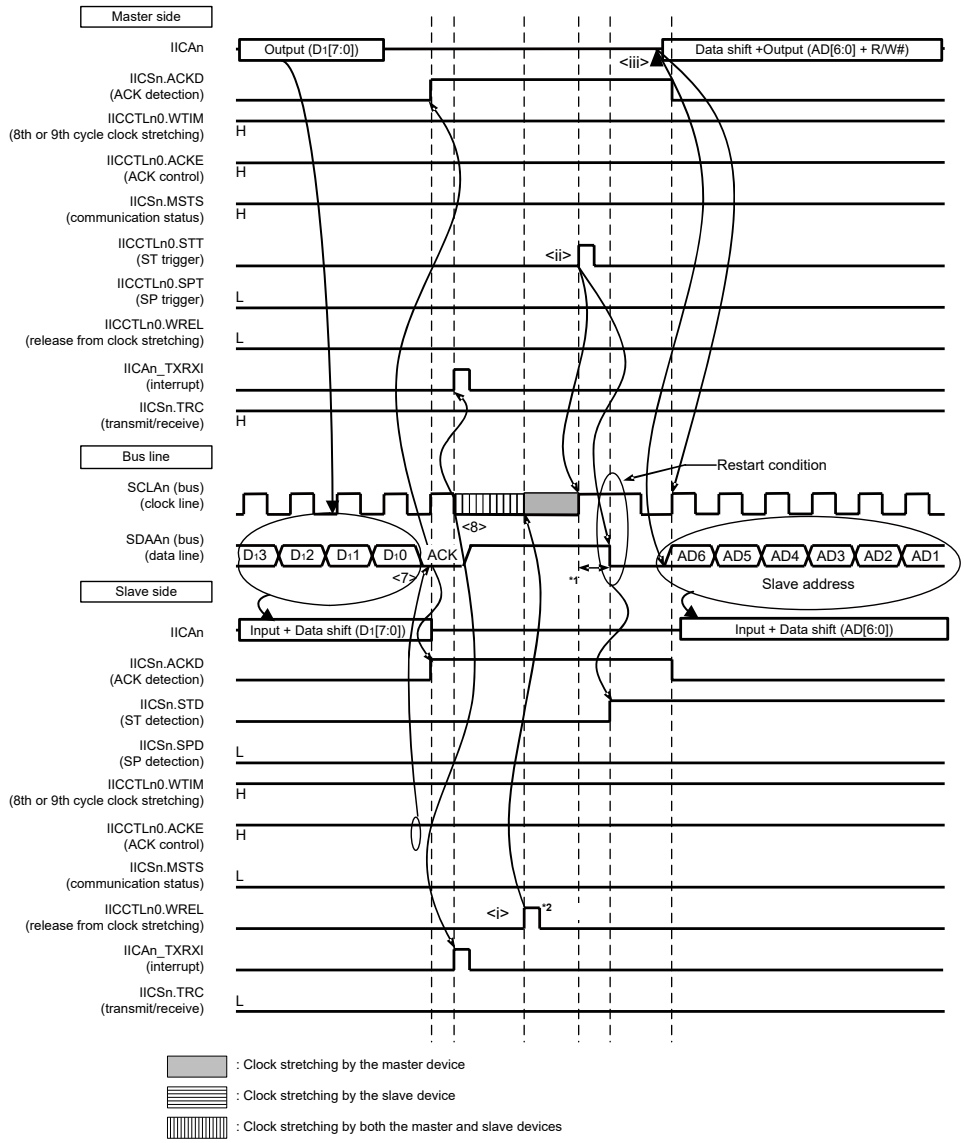
Note: <1> to <15> in (1) Example of Master device to Slave device Communications (When the master device and the slave device insert clock stretching on the 9th cycle.) represent the entire procedure for communicating data using the I^2C bus.

Figure 22.63 shows the processing from <1> to <6>,

Figure 22.64 shows the processing from <3> to <10>, and

Figure 22.65 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 mode, write 0xFF to IICAn or set the IICCTLn0.WREL bit.

Figure 22.66 Example of master device to slave device communications (When the master device and the slave device insert clock stretching on the 9th cycle.) (4/4)

The following describes the operations in Figure 22.66. 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 cycle.

<8> The master device and slave device set the clock stretch state (SCLAn = 0) at the falling edge of the 9th clock cycle, and both the master device and slave device issue an interrupt (IICAn_TXRXI: end of transfer).

<i> The slave device reads the received data and releases the clock stretch state (IICCTLn0.WREL = 1).

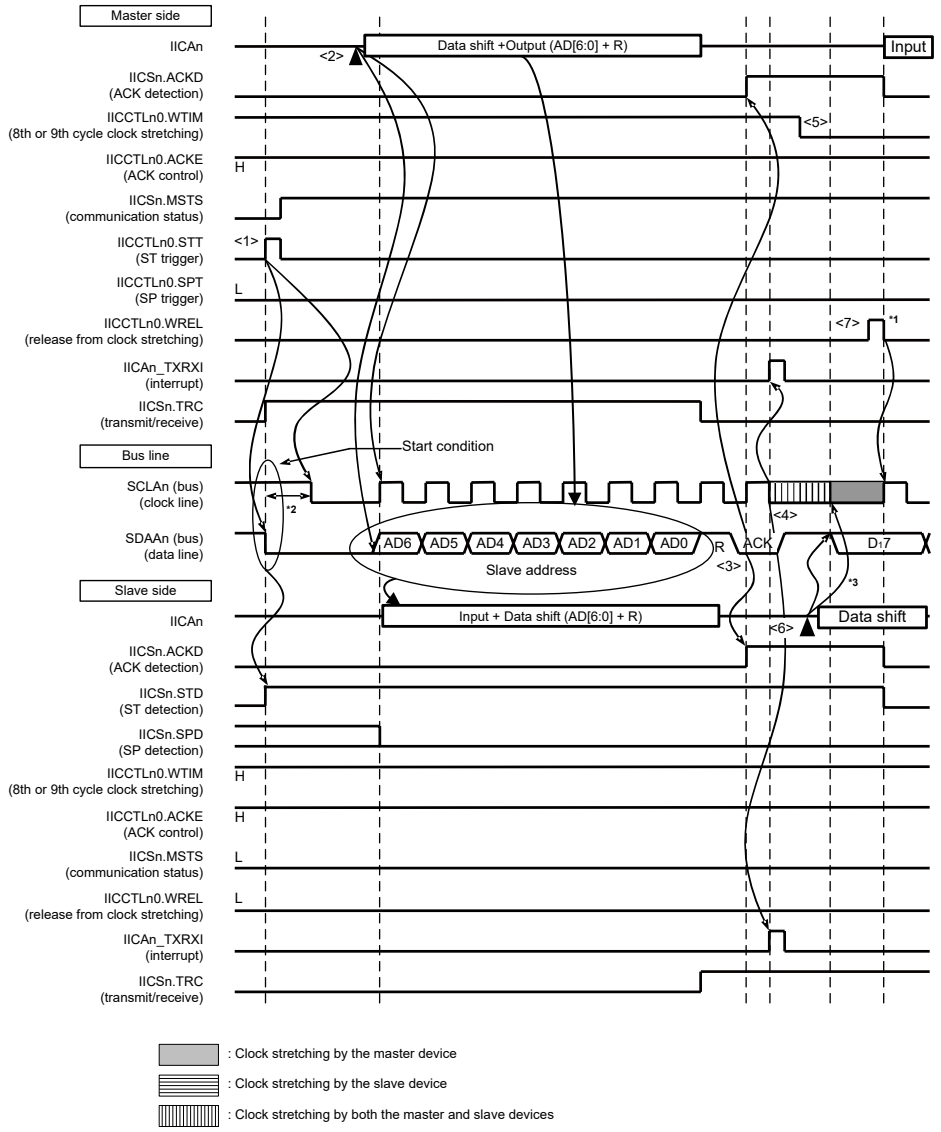
<ii> The start condition trigger is set again by the master device (IICCTLn0.STT = 1) and a start condition (SCLAn = 1 and SDAAn changes from 1 to 0) is generated once the bus clock line goes high (SCLAn = 1) and the bus data line goes low (SDAAn = 0) after the restart condition setup time has elapsed. 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.

Note: n = 0, 1

(2) Example of Slave device to Master device Communications (8th Cycle Clock Stretching Is Selected for the Master device and 9th Cycle Clock Stretching Is Selected for the Slave device)

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 mode, write data to the IICAn register instead of setting the IICCTLn0.WREL bit.

Figure 22.67 Example of slave device to master device communications (8th cycle clock stretching is selected for the master device and 9th cycle clock stretching is selected for the slave device) (1/3)

The meanings of <1> to <7> in Figure 22.67 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 mode enters the master mode 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.

Note: n = 0, 1

<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 cycle.

<4> The master device issues an interrupt (IICAn_TXRXI: end of address transmission) at the falling edge of the 9th clock cycle. The slave device with the address matching the transmitted slave address sets the clock stretch state (SCLAn = 0) and issues an interrupt (IICAn_TXRXI: address match).

Note: n = 0, 1

<5> The timing at which the master device sets the clock stretch state changes to the 8th clock cycle (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 IICAn_TXRXI interrupt (address match) and does not set the clock stretch state.

The master device, however, issues the IICAn_TXRXI interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

Note: n = 0, 1

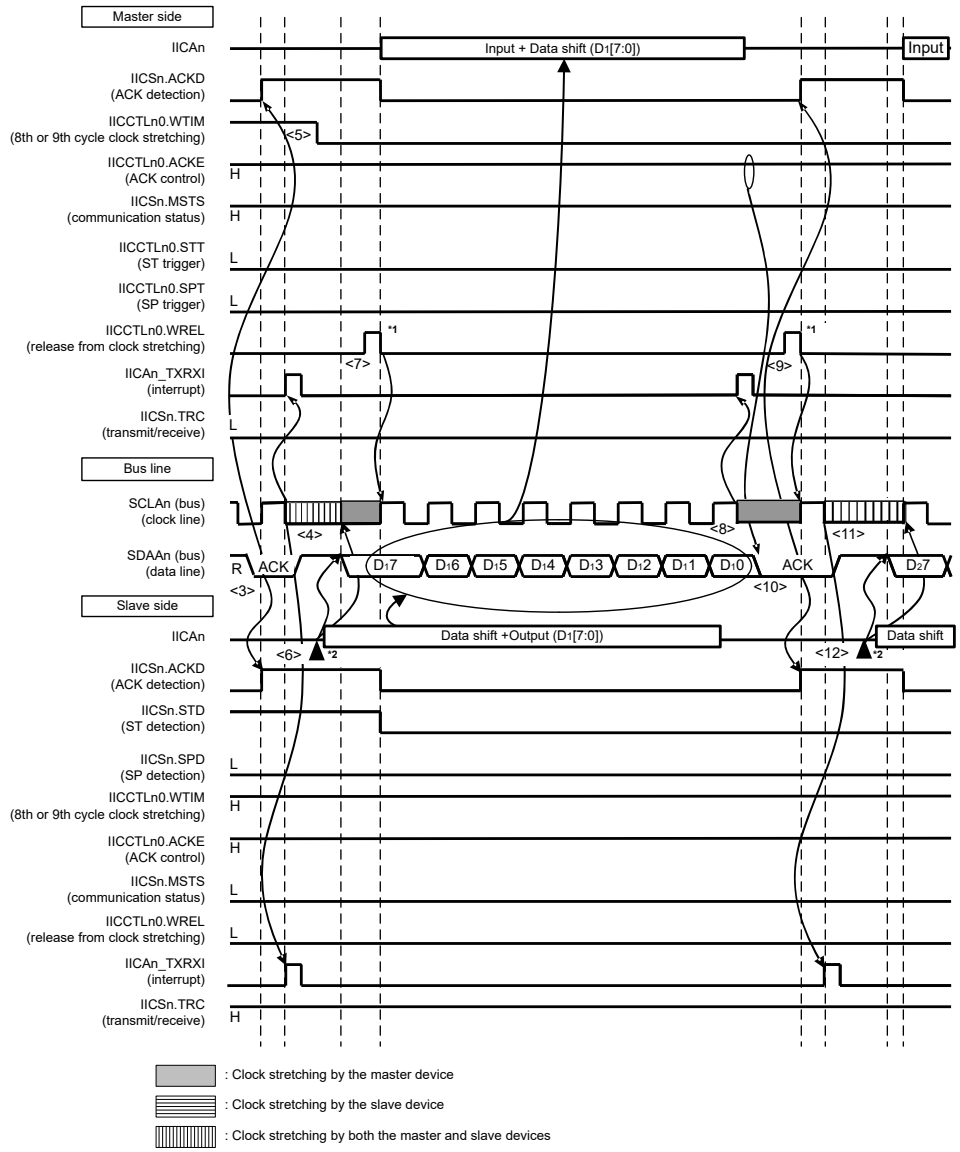
Note: <1> to <19> in (2) Example of Slave device to Master device Communications (8th Cycle Clock Stretching Is Selected for the Master device and 9th Cycle Clock Stretching Is Selected for the Slave device) represent the entire procedure for communicating data using the I^2C bus.

Figure 22.67 shows the processing from <1> to <7>,

Figure 22.68 shows the processing from <3> to <12>, and

Figure 22.69 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 mode, write data to the IICAn register instead of setting the IICCTLn0.WREL bit.

Figure 22.68 Example of slave device to master device communications (8th cycle clock stretching is selected for the master device and 9th cycle clock stretching is selected for the slave device) (2/3)

The meanings of <3> to <12> in Figure 22.68 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 cycle.

<4> The master device issues an interrupt (IICAn_TXRXI: end of address transmission) at the falling edge of the 9th clock cycle. The slave device with the address matching the transmitted slave address sets the clock stretch state (SCLAn = 0) and issues an interrupt (IICAn_TXRXI: address match).

Note: n = 0, 1

<5> The master device changes the timing of clock stretching to the 8th clock cycle (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 cycle, and issues an interrupt (IICAn_TXRXI: 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 cycle.

<11>The slave device sets the clock stretch state (SCLAn = 0) at the falling edge of the 9th clock cycle, and the slave device issue an interrupt (IICAn_TXRXI: 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 IICAn_TXRXI interrupt (address match) and does not set the clock stretch state.

The master device, however, issues the IICAn_TXRXI interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

Note: n = 0, 1

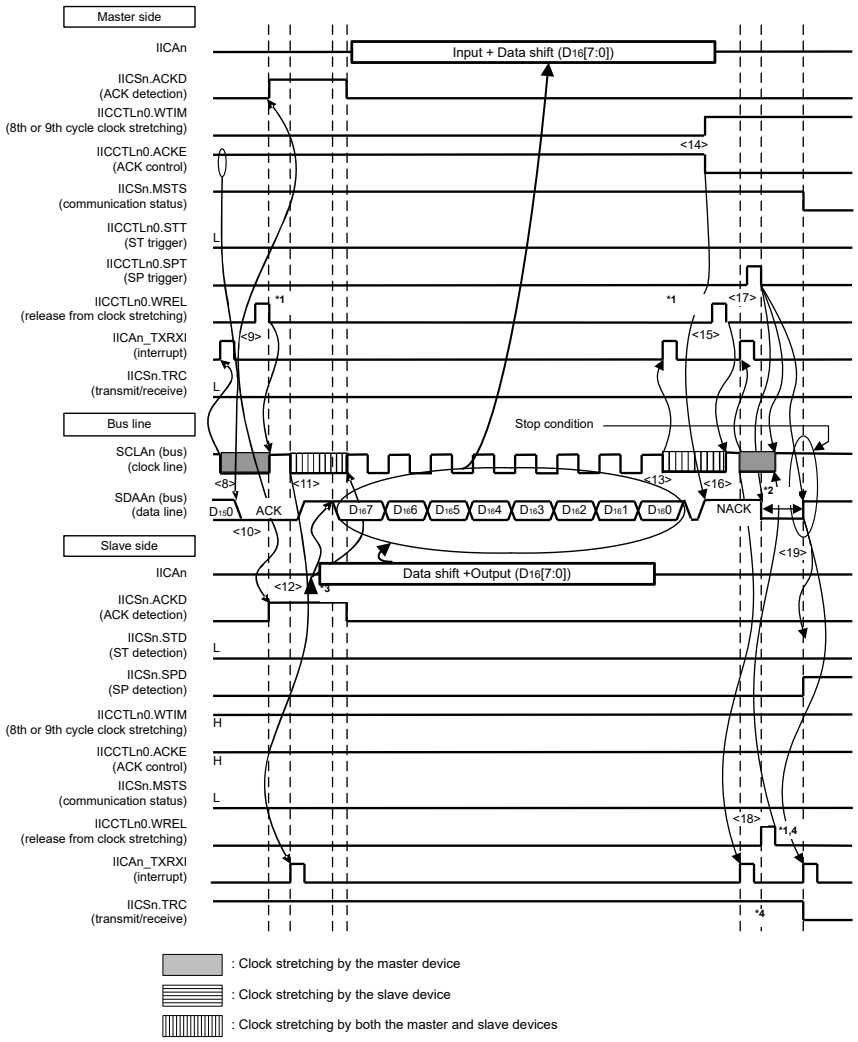
Note: <1> to <19> in (2) Example of Slave device to Master device Communications (8th Cycle Clock Stretching Is Selected for the Master device and 9th Cycle Clock Stretching Is Selected for the Slave device) represent the entire procedure for communicating data using the I^2C bus.

Figure 22.67 shows the processing from <1> to <7>,

Figure 22.68 shows the processing from <3> to <12>, and

Figure 22.69 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 mode is released by setting the IICCTLn0.WREL bit, the IICSn.TRC bit is cleared.

Figure 22.69 Example of slave device to master device communications (8th cycle clock stretching is selected for the master device and 9th cycle clock stretching is selected for the slave device) (3/3)

The meanings of <8> to <19> in Figure 22.69 are explained below.

<8> The master device sets the clock stretch state (SCLAn = 0) at the falling edge of the 8th clock cycle, and issues an interrupt (IICAn_TXRXI: 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 cycle.

<11>The slave device sets the clock stretch state (SCLAn = 0) at the falling edge of the 9th clock cycle, and the slave device issue an interrupt (IICAn_TXRXI: 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 (IICAn_TXRXI: end of transfer) at the falling edge of the 8th clock cycle, 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.

Note: n = 0, 1

<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 cycle (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 cycle.

<16>The master device and slave device set the clock stretch state (SCLAn = 0) at the falling edge of the 9th clock cycle, and both the master device and slave device issue an interrupt (IICAn_TXRXI: 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).

Note: n = 0, 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 (IICAn_TXRXI: stop condition).

Note: n = 0, 1

Note: <1> to <19> in (2) Example of Slave device to Master device Communications (8th Cycle Clock Stretching Is Selected for the Master device and 9th Cycle Clock Stretching Is Selected for the Slave device) represent the entire procedure for communicating data using the I^2C bus.

Figure 22.67 shows the processing from <1> to <7>,

Figure 22.68 shows the processing from <3> to <12>, and

Figure 22.69 shows the processing from <8> to <19>.

23. Serial Interface UARTA (UARTA)

23.1 Overview

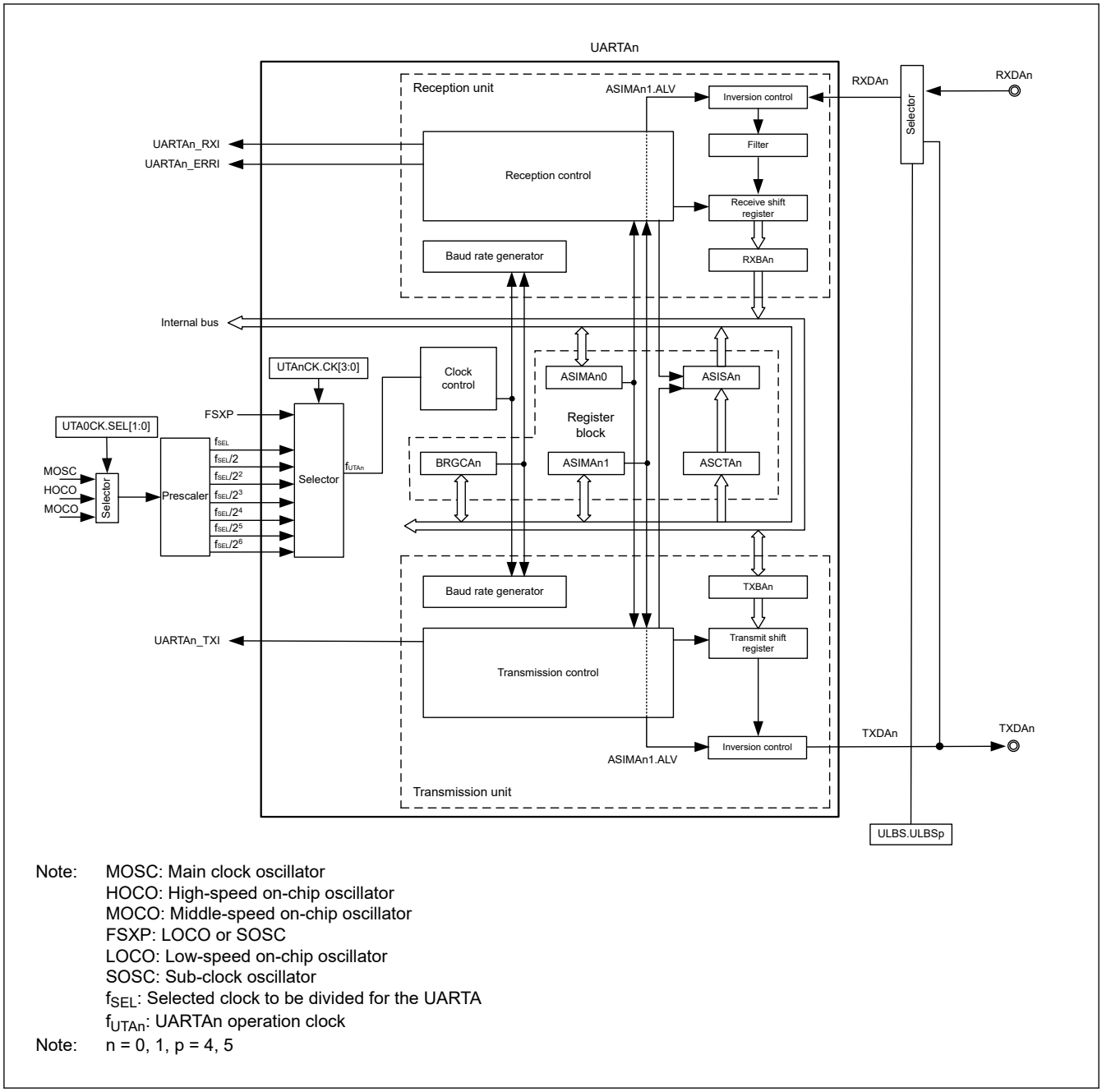
The serial interface UARTA has two channels. Table 23.1 lists specifications of the serial interface UARTA.

Table 23.1 UARTA specifications

Item	Specifications
Serial interface modes	• Operation stop mode • UART mode
Interfaces	• TXDAn: Transmit data output pin • RXDAn: Receive data input pin
Operation clock sources	Operating clock independent of the CPU/peripheral hardware clock selectable to MOSC, HOCO, MOCO and FSXP (LOCO or SOSC)*1
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 (UARTAn_TXI/UARTAn_RXI/UARTAn_ERRI)	• 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: n = 0, 1
Note 1. Selectable either LOCO or SOSC as the FSXP by setting the OSMC.WUTMMCK0 bit.

Figure 23.1 shows a block diagram of UARTAn and Table 23.2 shows the pin configuration of UARTAn.



Note: MOSC: Main clock oscillator
HOCO: High-speed on-chip oscillator
MOCO: Middle-speed on-chip oscillator
FSXP: LOCO or SOSC
LOCO: Low-speed on-chip oscillator
SOSC: Sub-clock oscillator
f_{SEL}: Selected clock to be divided for the UARTA
f_{UTAn}: UARTAn operation clock
Note: n = 0, 1, p = 4, 5

Figure 23.1 Block diagram of UARTAn

Table 23.2 UARTAn pin configuration

Name	I/O	Function
RXDAn	Input	Serial data input signal
TXDAn	Output	Serial data output signal

Note: n = 0, 1

23.2 Register Descriptions

23.2.1 TXBAn : Transmit Buffer Register n (n = 0, 1)

Base address: UARTA = 0x400A_3400

Offset address: 0x0000 + 0x8 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	n/a							
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, 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.

23.2.2 RXBAn : Receive Buffer Register n (n = 0, 1)

Base address: UARTA = 0x400A_3400

Offset address: 0x0001 + 0x8 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	n/a							
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.

23.2.3 ASIMAn0 : Operation Mode Setting Register n0 (n = 0, 1)

Base address: UARTA = 0x400A_3400

Offset address: 0x0002 + 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 UARTAn_ERRI interrupt is generated when a reception error occurs (UARTAn_RXI is not generated) 1: The UARTAn_RXI interrupt is generated when a reception error occurs (UARTAn_ERRI is not generated)	R/W
1	ISSMA	Transmit Interrupt Mode Select 0: The UARTAn_TXI interrupt is generated on completion of transmission 1: The UARTAn_TXI 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.	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*1	UART Operation Enable 0: Disables the UART operation clock (resets the internal circuits*2) 1: Enables the UART operation clock	R/W

Note 1. When EN = 0, the level being output from the TXDAn pin is determined according to the setting of the ALVn bit as described below.
- When ASIMAn1.ALV = 0, output from the TXDAn pin is high.
- When ASIMAn1.ALV = 1, output from the TXDAn pin is low.

Note 2. The ASISAn and RXBAn registers are reset by clearing the EN bit to 0.

The ASIMAn0 register is an 8-bit register that controls serial communication of the serial interface UARTAn.

Note: 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.

Note: 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.

Note: Follow the procedure below 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.

Note: 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 be initialized.

Note: 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.

Note: Clear the RXEA bit to 0 before modifying the ISRMA bit.

23.2.4 ASIMAn1 : Operation Mode Setting Register n1 (n = 0, 1)

Base address: UARTA = 0x400A_3400

Offset address: 0x0003 + 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 is an 8-bit register that controls serial communication of the serial interface UARTAn.

Note: Clear both the ASIMAn0.TXEA and RXEA bits to 0 before modifying the ASIMAn1 register.

Note: Reception is always handled as including a stop bit. The setting of the SL bit does not affect reception.

23.2.5 BRGCAn : Baud Rate Generator Control Register n (n = 0, 1)

Base address: UARTA = 0x400A_3400

Offset address: 0x0004 + 0x8 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	n/a							
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.

Note: Modify the BRGCAn register while the ASIMAn0.TXEA and RXEA bits are 0 (in the transmission and reception stopped state).

Note: The baud rate is one half the frequency of the output clock signal from the 8-bit counter.

Note: For an example of the baud rate setting, see (c) Baud rate setting example.

23.2.6 ASISAn : Status Register n (n = 0, 1)

Base address: UARTA = 0x400A_3400

Offset address: 0x0005 + 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

Bit	Symbol	Function	R/W
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 bits are initialized by clearing the ASIMAn0.EN or TXEA bit to 0.

Note: 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 become undefined.

However, the TXBFA flag need not be checked when continuous transmission is performed by using the buffer empty interrupt (ASIMAn0.ISSMA bit = 1).

Note: 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 become undefined.

Note: The operation of the PEA bit depends on the setting of the PS[1:0] bits of the ASIMAn1 register.

Note: For the receive data, only the first 1 bit of the stop bits is checked regardless of the stop bit length.

Note: When an overrun error occurs, the next receive data is not written to the RXBAn register and 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.)

23.2.7 ASCTAn : Status Clear Trigger Register n (n = 0, 1)

Base address: UARTA = 0x400A_3400

Offset address: 0x0006 + 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 bits of the ASISAn register, respectively. When writing 0, the corresponding error flags are not cleared.

Note: 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.

23.2.8 UTA0CK : UARTA Clock Select Register 0

Base address: UARTA = 0x400A_3400

Offset address: 0x0100

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	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: FSXP Others: Setting prohibited	R/W
5:4	SEL[1:0]	f_{SEL} Clock Select 0 0: Stop 0 1: MOSC 1 0: HOCO 1 1: MOCO	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W

The UTA0CK register selects the operating clock of the UARTA0. The SEL[1:0] bits select the clock source, f_{SEL}, for UARTA0 from MOSC, HOCO, and MOCO. The CK[3:0] bits select the operating clock for UARTA0 from f_{SEL} to $f_{SEL}/64$, and FSXP.

Note: This register should be read or written when the TXEA and RXEA bits are 0 (in the transmission and reception stopped state).

Note: f_{SEL}: Selected clock to be divided for the UARTA

23.2.9 UTA1CK : UARTA Clock Select Register 1

Base address: UARTA = 0x400A_3400

Offset address: 0x0101

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	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: FSXP Others: Setting prohibited	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W

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 FSXP.

Note: This register should be read or written when the TXEA and RXEA bits are 0 (in the transmission and reception stopped state).

Note: f_{SEL}: Selected clock to be divided for the UARTA

23.2.10 ULBS : UART Loopback Select Register

Base address: PORGA = 0x400A_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 array unit 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 array unit UARTA1 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
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.

23.3 Operation

UARTAn operates in the following two modes.

- Operation stop mode
- UART mode

23.3.1 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 bits [7:5] (EN, TXEA, RXEA) of the ASIMAn0 register to 0.

The bus clock is not stopped by the above setting. To completely stop operation, set bit 15 of the MSTPCRB register to 1 after the above setting.

23.3.2 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 23.3 shows the step of communication procedure.

Table 23.3 Step of communication procedure

Step		Process	Detail
Step of communication procedure	<1>	Enable clock supply	Set bit 15 of the MSTPCRB register to 0.
	<2>	Baud rate setting	Set the BRGCAn register.
	<3>	Operation mode setting 1	Set bits [6:0] (PS[1:0], CL[1:0], SL, DR, LV) of the ASIMAn1 register.
	<4>	Operation mode setting 2	Set bits [1:0] (ISSMA and ISRMA) of the ASIMAn0 register
	<5>	Enable operation	Set bit 7 (EN) of the ASIMAn0 register to 1.
	<6>	Enable communication	Set bit 6 (TXEA) of the ASIMAn0 register to 1 to enable transmission. Set bit 5 (RXEA) of the ASIMAn0 register to 1 to enable reception.
	<7>	Write transmit data	Write transmit data to the TXBAn register.
	<8>	Start of transmission	—

Note: When using the receiving function, set the port pin allocated for reception to input mode by using the Pmn direction register. When using the transmitting function, set the port pin allocated for transmission to output mode by using the Pmn direction register, and set the respective bits in the port registers to 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 23.2 shows the data format.

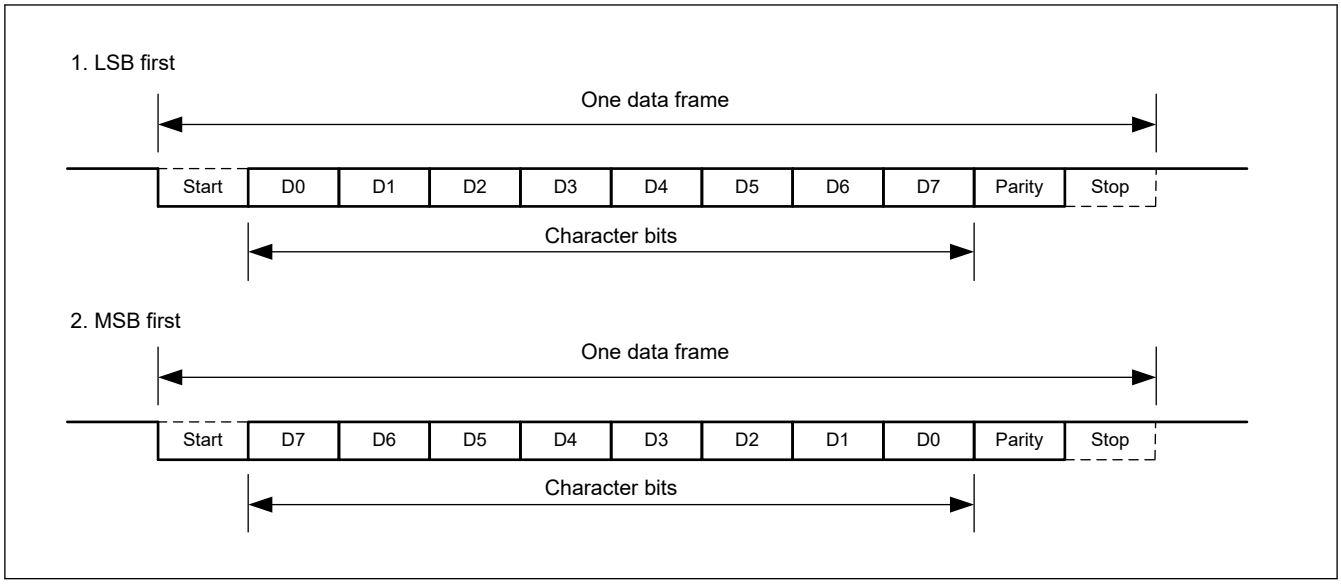


Figure 23.2 Transmit and receive data format

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.

Figure 23.3 shows the examples of transmit and receive data waveforms.

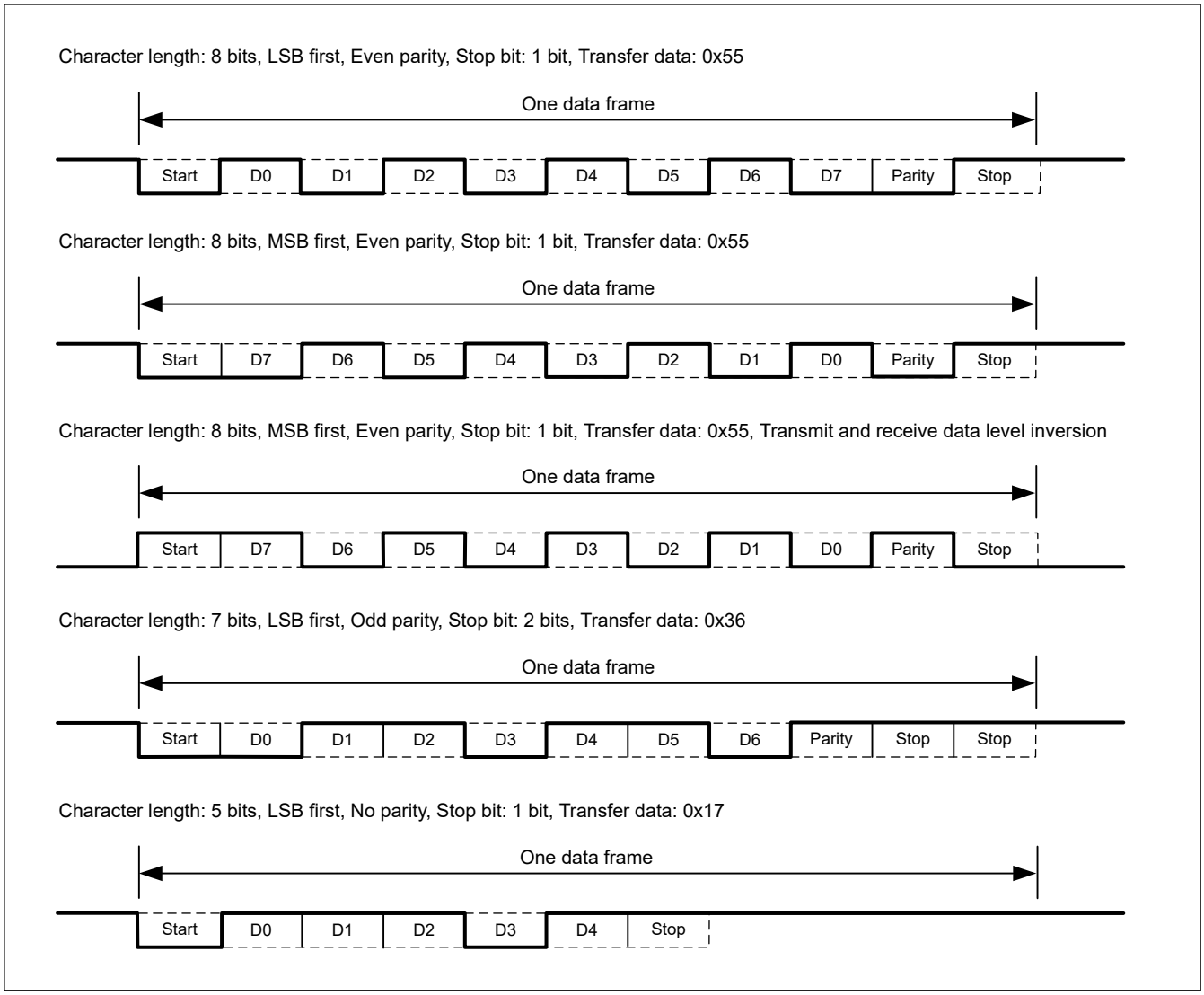


Figure 23.3 Example of transmit and receive data waveform

(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 bit 7 (EN) of the operation mode setting register n0 (ASIMAn0) to 1 and then setting bit 6 (TXEA) 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 (UARTAn_TXI) is generated.

Transmission is suspended until the next transmit data is written to the TXBAn register.

Figure 23.4 shows the timing of the transfer completion interrupt request signal (UARTAn_TXI). UARTAn_TXI is issued at the following timing.

a. When ASIMAn0.ISSMA = 0 (UARTAn_TXI functions as a transfer completion interrupt.)
   UARTAn_TXI is issued after the output of the last stop bit.
b. ASIMAn0.ISSMA = 1 (UARTAn_TXI functions as a buffer empty interrupt.)
   UARTAn_TXI is issued when the start bit is output.

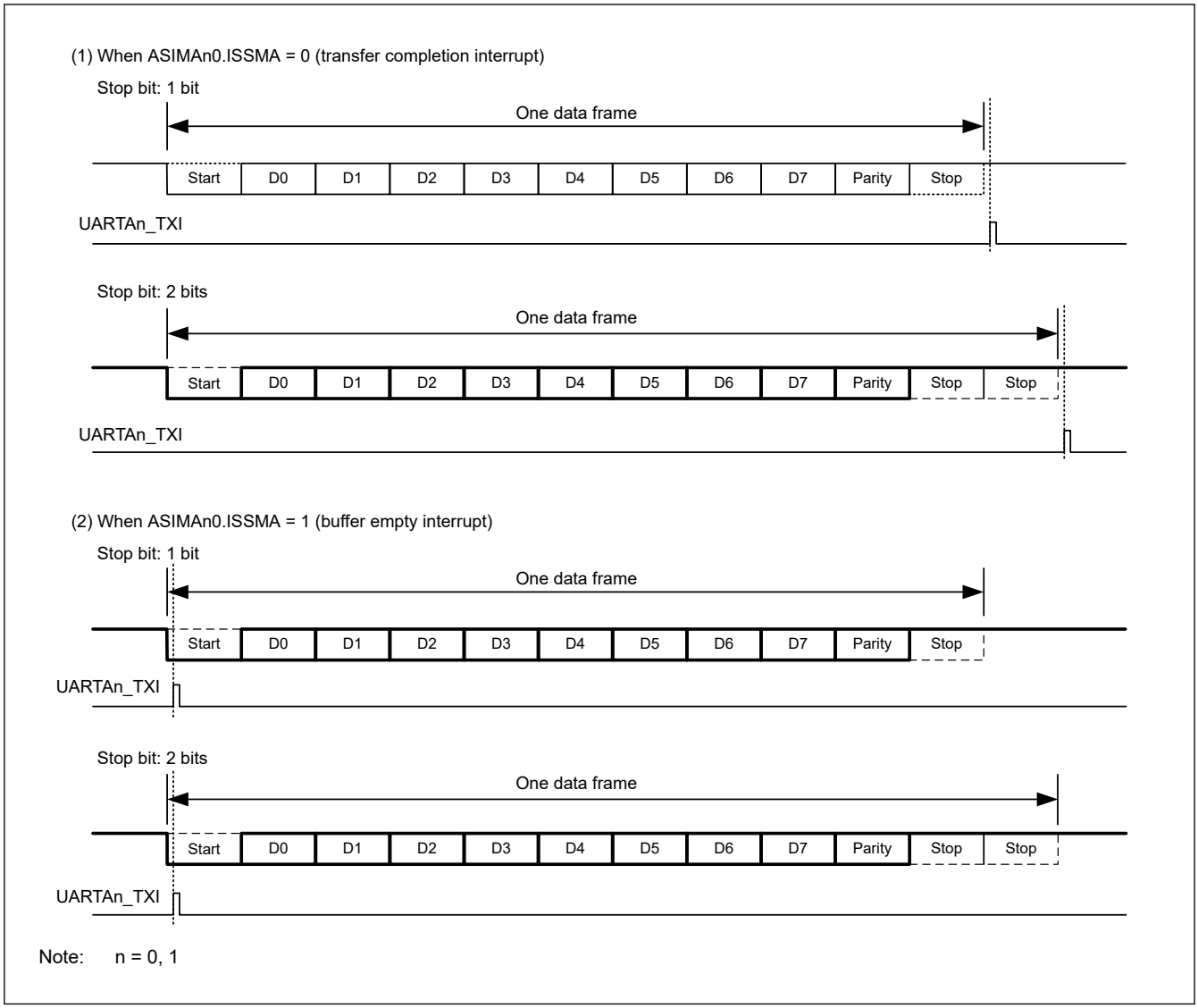


Figure 23.4 Interrupt output timing

(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 transmit buffer data flag (bit 5: TXBFA) and the transmit shift register data flag (bit 4: TXSFA) of the status register (ASISAn).

When using this method, clear bit 1 (ISSMA) of the operation mode setting register n0 (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 (TXBFA) is 0, and then write the second byte of data. Similarly, check that the TXBFA flag is 0 and then write the subsequent data to the TXBAn register.

Table 23.4 shows the determination flag indicating that writing to TXBAn is enabled or disabled at the start of continuous transmission.

Table 23.4 Determination flag indicating that writing to TXBAn is enabled or disabled at the start of continuous transmission

ASISAn.TXBFA	Determination flag indicating that writing to TXBAn is enabled or disabled at the start of continuous transmission
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.

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 23.5 shows the confirmation flag indicating whether transmission is in progress or not.

Table 23.5 Confirmation flag indicating whether transmission is in progress or not

ASISAn.TXSFA	Confirmation flag indicating whether transmission is in progress or not
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 UARTAn_TXI 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 = 0, 1

Table 23.6 shows a step example of continuous transmission processing by polling.

Table 23.6 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 TXBAn If yes, go to <5> If no, go to <3>	—
	<3>	Wait until the ASISAn.TXBFA bit is cleared.	Data is transferred to the transmit shift register.
	<4>	Write transmit data to TXBAn. Go to <2>	—
	<5>	Wait until the ASISAn.TXSFA bit 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	—

(b) Continuous transfer by using an interrupt

Continuous transmission is achieved by using the interrupt (UARTAn_TXI).

Note: n = 0, 1

An interrupt can be generated when data in the transmit buffer register (TXBAn) are transferred to the transmit shift register by setting bit 1 (ISSMA) to 1 in the operation mode setting register n0 (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.

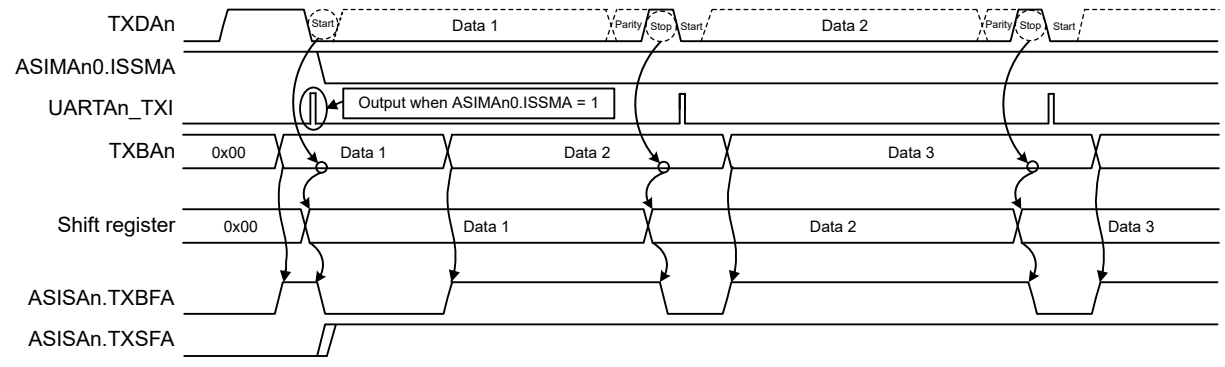
Table 23.7 shows a step example of continuous transmission using interrupt.

Table 23.7 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 TXBAn	—
	<3>	Waiting for UARTAn_TXI	Buffer empty interrupt
	<4>	Generation of UARTAn_TXI	—
	<5>	Check if the required number of transmit data are written to TXBAn. If yes, go to step <6>. If no, go to step <2>.	—
	<6>	Set ASIMAn0	ASIMAn0.ISSMA = 0
	<7>	Waiting for UARTAn_TXI	Transfer completion interrupt
	<8>	Generation of UARTAn_TXI	—
	<9>	End of transmission processing	—

Note: n = 0, 1

Figure 23.5 and Figure 23.6 show the timing when continuous transmission is started and completed, respectively.



Note: n = 0, 1
Note: When the ASISAn register is read, both the ASISAn.TXBFA and TXSFA bits are read as 1 within this period. Accordingly, use only the ASISAn.TXBFA flag to determine if writing is enabled or disabled.

Figure 23.5 Timing when continuous transmission is started

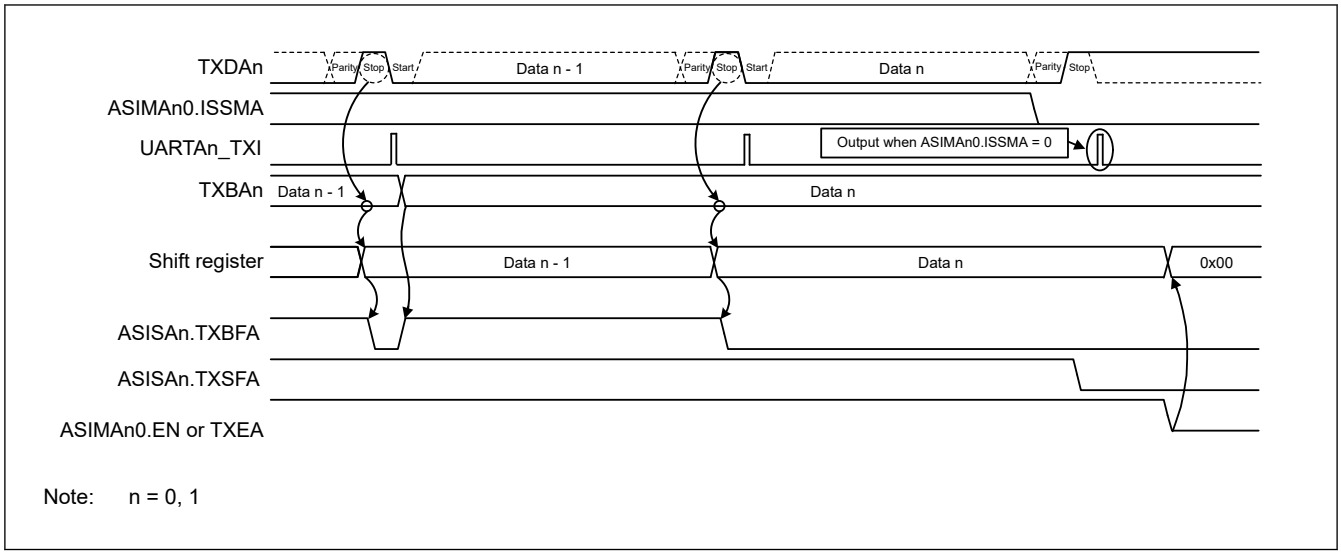


Figure 23.6 Timing when continuous transmission is completed

(6) Normal reception

When setting bit 7 (EN) of the operation mode setting register n0 (ASIMAn0) to 1 and then setting bit 5 (RXEA) 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 23.7). 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 23.7). If the RXDAn pin is high, it is regarded as a start bit.

Note: n = 0, 1

Figure 23.7 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 (UARTAn_RXI) 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 (OVEA) occurs, the data received on occurrence of the error is not written to the

RXBAn register.

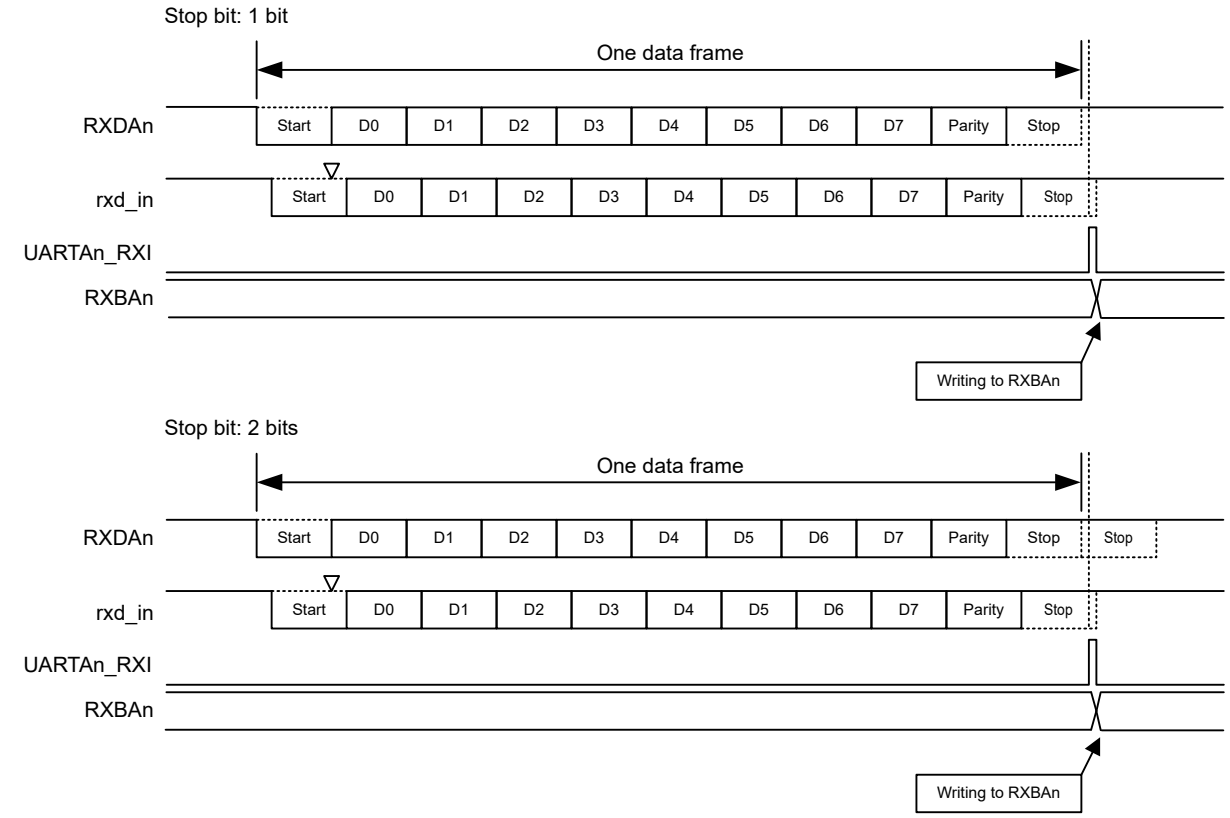
When a parity error (PEA) or a framing error (FEA) occurs during reception, reception continues until a stop bit is received. After completion of the reception, the reception error interrupt (UARTAn_RXI and UARTAn_ERRI) set in the ASIMAn0.ISRMA bit is generated.

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 = 0, 1



Note: 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 UARTAn_RXI output timing in the figure is just an example.
The timing relative to RXDAn varies according to the setting of the BRGCAn register.

Figure 23.7 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 (UARTAn_RXI or UARTAn_ERRI) is generated.

Note: n = 0, 1

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 23.8 shows the causes of the reception errors.

Table 23.8 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 bit 0 (ISRMA) of the operation mode setting register n0 (ASIMAn0) to 0 allows the reception error interrupt to be separated from UARTAn_RXI and allows it to be generated as UARTAn_ERRI.

Note: n = 0, 1

Figure 23.8 shows the interrupt output waveform which varies depending on the setting of the ASIMAn0.ISRMA bit.

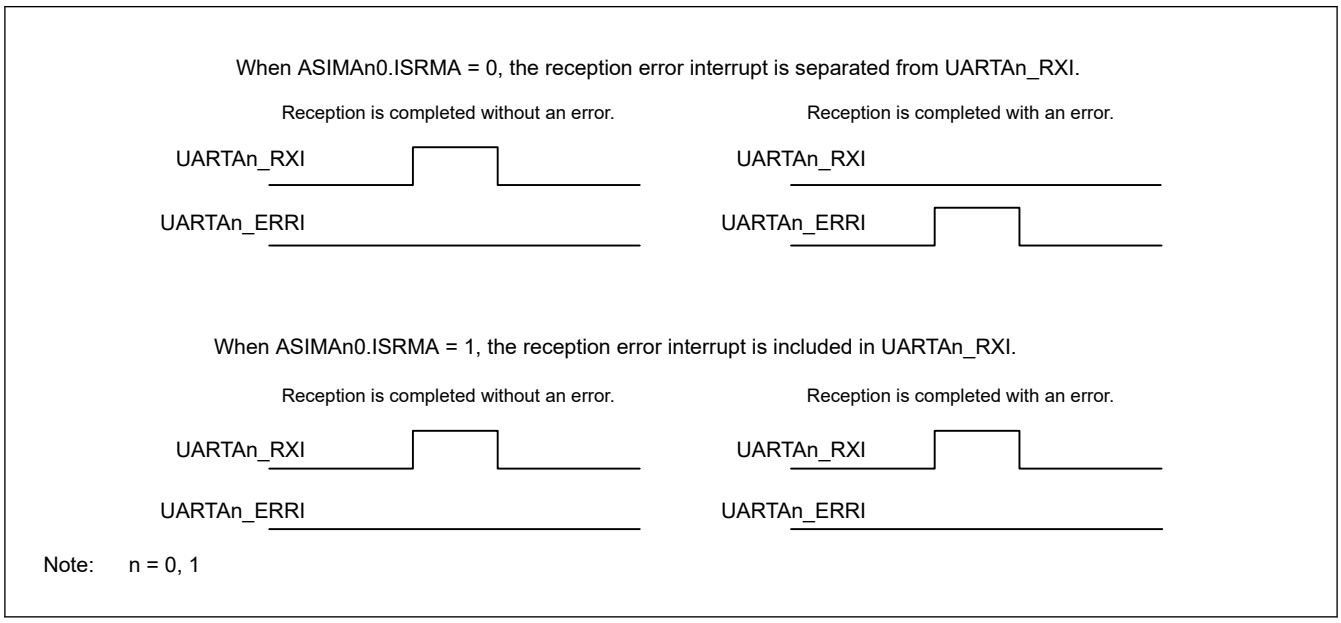


Figure 23.8 Various interrupt output waveforms depending on ASIMAn0.ISRMA setting

23.3.3 Receive Data Noise Filter

This filter samples the receive data (RXDAn), and determines the level when the same level is sampled twice.

Note: n = 0, 1

The receive data is delayed by maximum of three cycles of the operating clock because of the circuit configuration.

Figure 23.9 shows the noise filter circuit.

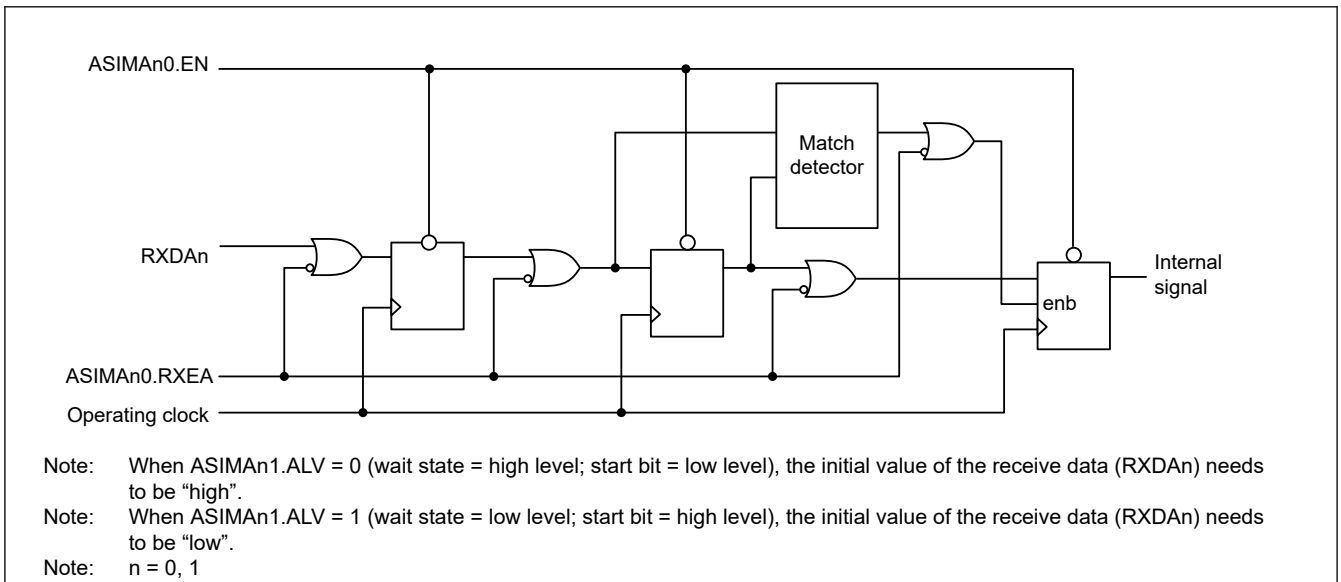


Figure 23.9 Noise filter

23.3.4 Baud Rate Generator

The baud rate generator consists of 8-bit programmable counters, and generates a serial clock for transmission and reception of UARTAn.

Note: n = 0, 1

An 8-bit counter is provided each for transmission and reception.

(1) Configuration of baud rate generator

(a) UARTAn operation clock

Note: n = 0, 1

When bit 7 (EN) = 1 in the operation mode setting register n0 (ASIMAn0), 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 bit 7 (EN) = 0 or bit 6 (TXEA) = 0 in the operation mode setting register n0 (ASIMAn0). 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 EN = 0 or TXEA = 0 in the ASIMAn0 register, the counter stops at 0x00.

(c) Reception counter

This counter is cleared to 0 and stops when bit 7 (EN) = 0 or bit 5 (RXEA) = 0 in the operation mode setting register n0 (ASIMAn0). 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 EN = 0 or RXEA = 0 in the ASIMAn0 register, the counter stops at 0x00.

Figure 23.10 shows the configuration of the baud rate generator.

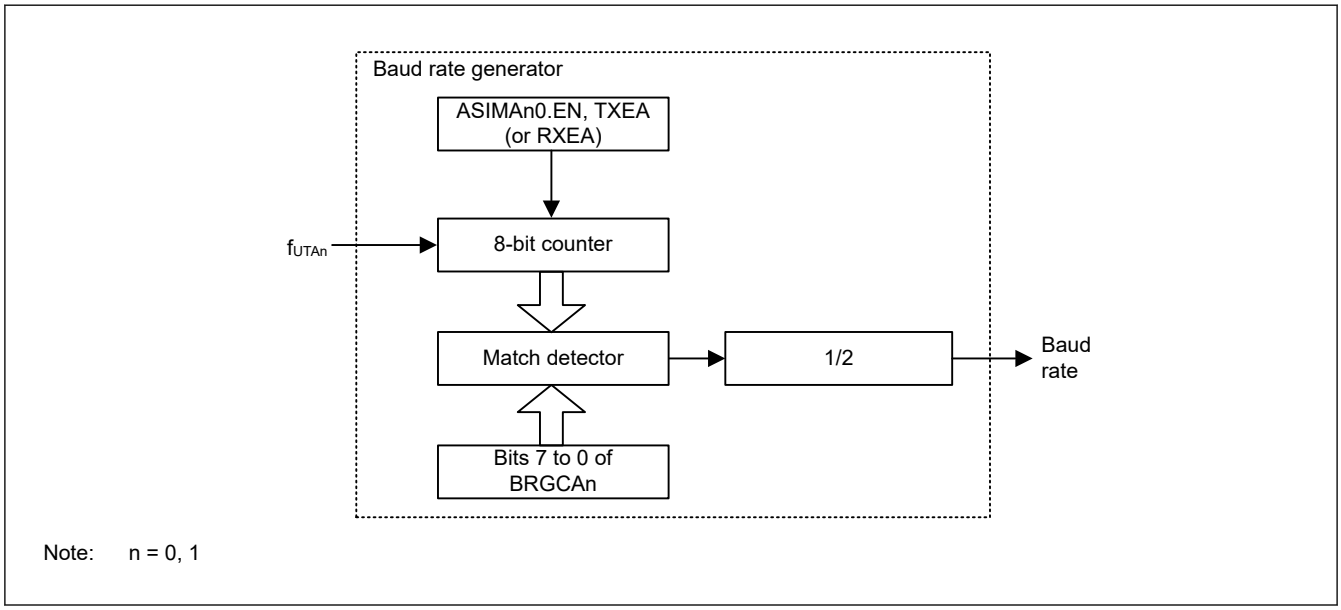


Figure 23.10 Configuration of baud rate generator

(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 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.

(c) Baud rate setting example

Table 23.9 to Table 23.12 show the set data of the baud rate generator.

Table 23.9 Set data of baud rate generator (1/4)

Desired baud rate	In operation with HOCO = 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] = 0000b)		(UTAnCK.CK [3:0] = 0001b)		(UTAnCK.CK [3:0] = 0010b)		(UTAnCK.CK [3:0] = 0011b)		(UTAnCK.CK [3:0] = 0100b)		(UTAnCK.CK [3:0] = 0101b)		(UTAnCK.CK [3:0] = 0110b)	
	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	

Note: k: Value set by the baud rate generator control register (BRGCAn) (k = 2, 3, 4, …, 255)

Table 23.10 Set data of baud rate generator (2/4)

Desired baud rate	In operation with MOCO = 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] = 0000b)		(UTAnCK.CK [3:0] = 0001b)		(UTAnCK.CK [3:0] = 0010b)		(UTAnCK.CK [3:0] = 0011b)		(UTAnCK.CK [3:0] = 0100b)		(UTAnCK.CK [3:0] = 0101b)		(UTAnCK.CK [3:0] = 0110b)	
	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	
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	

Note: k: Value set by the baud rate generator control register (BRGCAn) (k = 2, 3, 4, …, 255)

Table 23.11 Set data of baud rate generator (3/4)

Desired baud rate	In operation with MOSC = 20 MHz (UTA0CK.SEL[1:0] = 01b)													
	No division		×1/2		×1/4		×1/8		×1/16		×1/32		×1/64	
	(UTAnCK.CK [3:0] = 0000b)		(UTAnCK.CK [3:0] = 0001b)		(UTAnCK.CK [3:0] = 0010b)		(UTAnCK.CK [3:0] = 0011b)		(UTAnCK.CK [3:0] = 0100b)		(UTAnCK.CK [3:0] = 0101b)		(UTAnCK.CK [3:0] = 0110b)	
	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	

Note: k: Value set by the baud rate generator control register (BRGCAn) (k = 2, 3, 4, …, 255)

Table 23.12 Set data of baud rate generator (4/4)

Desired baud rate	In operation with FSXP = 32.768 kHz (UTAnCK.CK[3:0] = 1000b)	
	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%
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: k: Value set by the baud rate generator control register (BRGCAn) (k = 2, 3, 4, …, 255)

(d) Permissible baud rate range during reception

Figure 23.11 shows the permissible error from the baud rate on the transmitting side during reception.

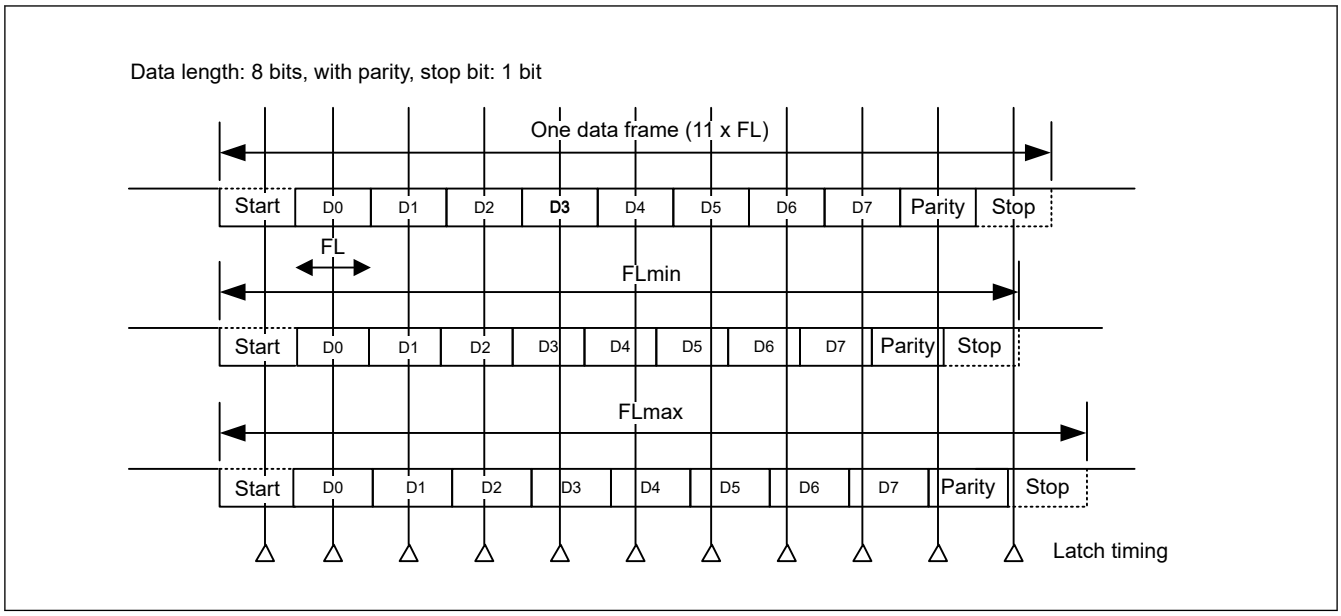


Figure 23.11 Permissible baud rate range during reception

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 BRGCAn FL: 1-bit data length

Margin of latch timing: 2 clock

- Minimum permissible data frame length (FLmin)
  $k = 3$ to 255: $\mathrm{FLmin} = 11 \times \mathrm{FL} - \frac{k-2}{2k} \times \mathrm{FL} = \frac{21k+2}{2k}\mathrm{FL}$
- Maximum permissible baud rate for reception on the transmitting side (BRmax)
  $k = 2$: $\mathrm{BRmax} = \mathrm{Brate} + \frac{1}{22k}\mathrm{Brate}$
  $k = 3$ to 255: $\mathrm{BRmax} = (\mathrm{FLmin}/11)^{-1} = \frac{22k}{21k+2}\mathrm{Brate}$
- Maximum permissible data frame length (FLmax)
  $k = 3$ to 255: $\mathrm{FLmax} = \frac{21k+2}{20k}\mathrm{FL} \times 11$
- Minimum permissible baud rate for reception on the transmitting side (BRmin)
  $k = 2$: $\mathrm{BRmin} = \mathrm{Brate} - \frac{1}{22k}\mathrm{Brate}$
  $k = 3$ to 255: $\mathrm{BRmin} = (\mathrm{FLmax}/11)^{-1} = \frac{20k}{21k-2}\mathrm{Brate}$

Table 23.13 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 23.13 Maximum and minimum permissible baud rate error

Division ratio (k)	Maximum permissible baud rate error	Minimum permissible baud rate error
2	+2.27%	−2.27%
4	+2.33%	−2.44%
8	+3.53%	−3.61%
20	+4.27%	−4.31%
50	+4.56%	−4.58%
100	+4.66%	−4.67%
255	+4.72%	−4.73%

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 BRGCAn

23.4 Usage Notes

23.4.1 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.

Note: n = 0, 1

23.4.2 Point for Caution when Selecting the UARTAn Operation Clock (f_{UTAn})

When the Middle-speed on-chip oscillator (MOCO) 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 trimming register (MIOTRM).

When the Low-speed peripheral clock (FSXP) is selected for f_{UTAn} and the Low-speed on-chip oscillator (LOCO) is selected for FSXP, 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 trimming register (LIOTRM).

Note: n = 0, 1

24. Cyclic Redundancy Check (CRC)

24.1 Overview

The Cyclic Redundancy Check (CRC) generates CRC codes to detect errors in the data. The bit order of CRC calculation results can be selected LSB-first communication. Additionally, two CRC-generation polynomials (16-bit CRC-CCITT and 32-bit CRC-32) are available.

Table 24.1 lists the CRC calculator specifications and Figure 24.1 shows a block diagram.

Table 24.1 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	[16-bit CRC] • $X^{16} + X^{12} + X^5 + 1$ (CRC-CCITT).	[32-bit CRC] • $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$ (CRC-32)
Module-stop function	Module-stop state can be set to reduce power consumption	

Note 1. This function cannot divide data used in CRC calculations. Write data in 8-bit or 32-bit units.

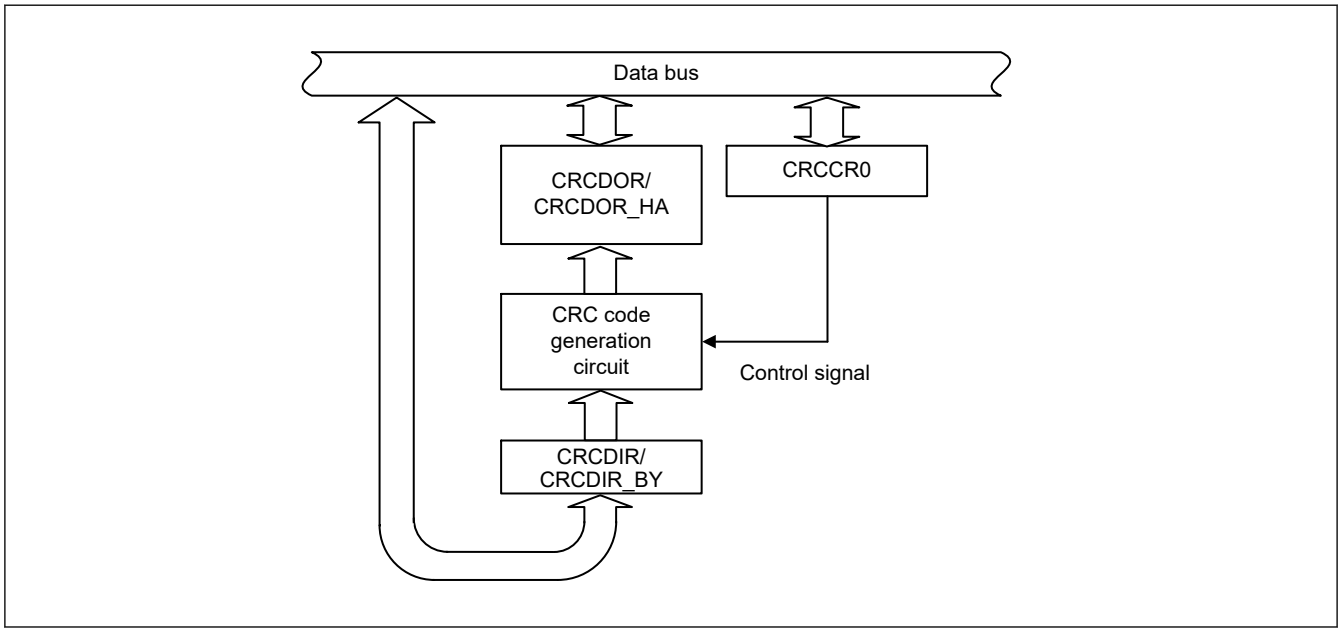


Figure 24.1 CRC calculator block diagram

24.2 Register Descriptions

24.2.1 CRCCR0 : CRC Control Register 0

Base address: CRC = 0x4007_4000

Offset address: 0x0000

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DORCLR	—	—	—	—	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 1 1: 16-bit CRC-CCITT ($X^{16} + X^{12} + X^5 + 1$) 1 0 0: 32-bit CRC-32 ($X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$) Others: No calculation is executed	R/W
6:3	—	These bits are read as 0. The write value should be 0.	R/W
7	DORCLR	CRCDOR/CRCDOR_HA Register Clear 0: No effect 1: Clear the CRCDOR/CRCDOR_HA register	W

GPS[2:0] bits (CRC Generating Polynomial Switching)

The GPS[2:0] bits select the CRC generating polynomial.

DORCLR bit (CRCDOR/CRCDOR_HA Register Clear)

Write 1 to the DORCLR bit to set the CRCDOR/CRCDOR_HA register to 0x00000000. This bit is read as 0. Only 1 can be written to it.

24.2.2 CRCDIR/CRCDIR_BY : CRC Data Input Register

Base address: CRC = 0x4007_4000

Offset address: 0x0004

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 input data The CRCDIR register is a 32-bit read/write register to write data for CRC-32 calculation. The CRCDIR_BY (CRCDIR[31:24], address: 0x4007_4004) is an 8-bit read/write register to write data for CRC-CCITT calculation.	R/W

24.2.3 CRCDOR/CRCDOR_HA : CRC Data Output Register

Base address: CRC = 0x4007_4000

Offset address: 0x0008

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 calculation. The CRCDOR_HA (CRCDOR[31:16], address: 0x4007_4008) register is a 16-bit read/write register for CRC-CCITT calculation. Because its initial value is 0x00000000, rewrite the CRCDOR/CRCDOR_HA 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 register. If the CRC code is calculated following the transferred data and the result is 0x00000000, there is no CRC error.	R/W

24.3 Operation

24.3.1 Basic Operation

The CRC calculator generates CRC codes for use in LSB-first.

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 a 32-bit CRC is in use, the valid bits of the CRC code are obtained in CRCDOR.

Figure 24.2 shows the LSB-first data transmission examples respectively. Figure 24.3 shows the LSB-first data reception examples.

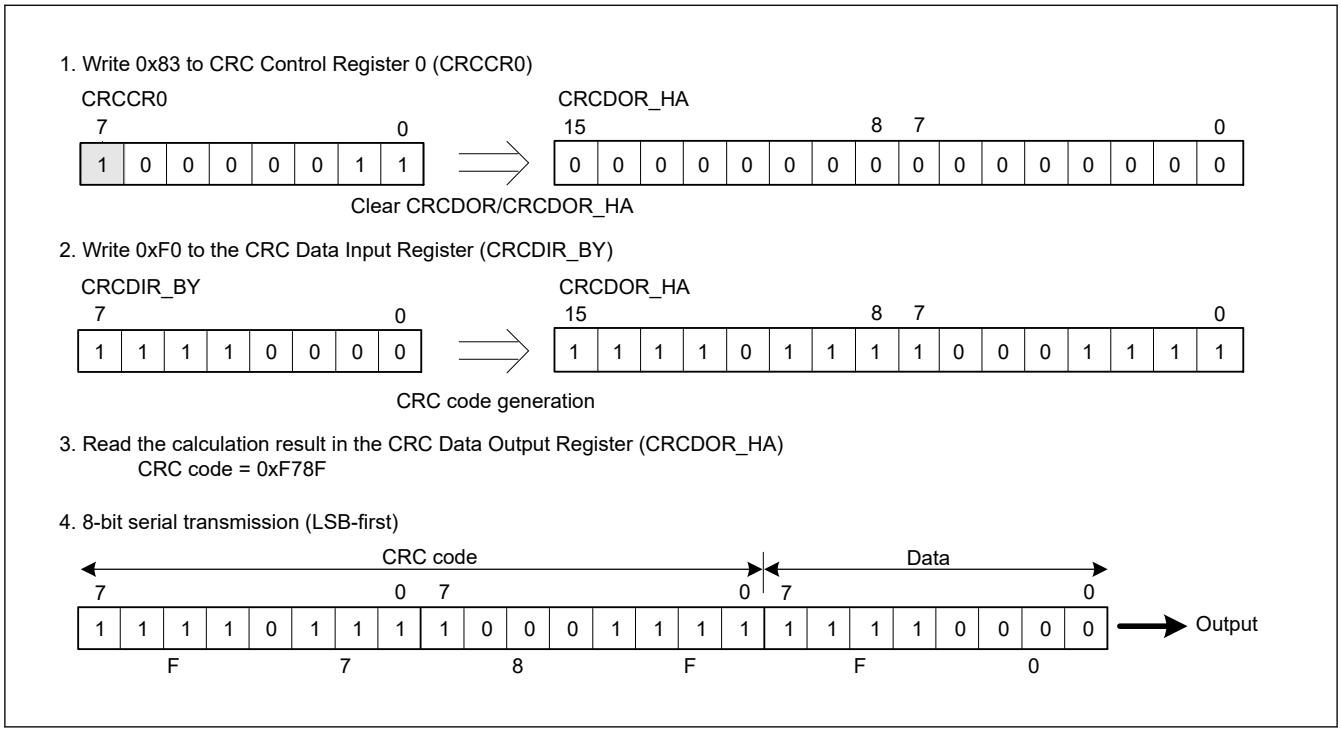


Figure 24.2 LSB-first data transmission

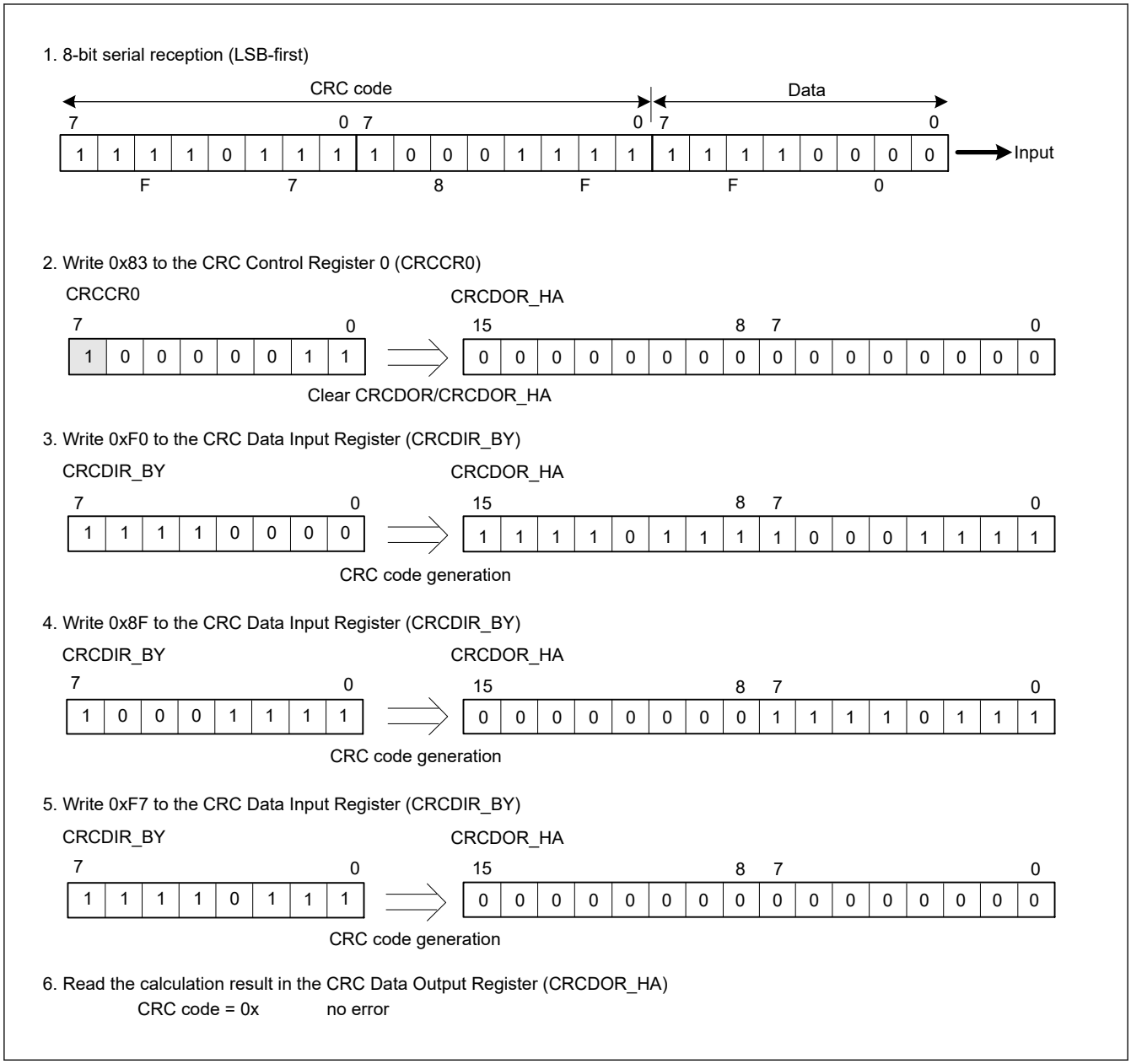


Figure 24.3 LSB-first data reception

24.4 Usage Notes

24.4.1 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 9, Low Power Modes.

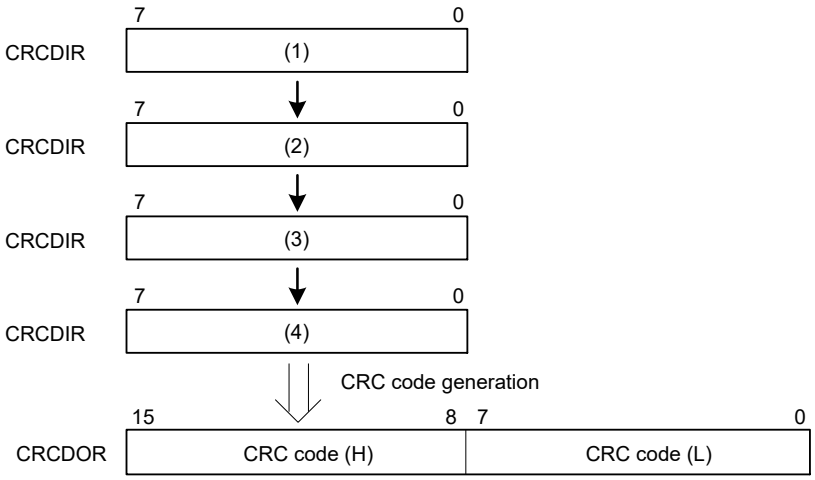
24.4.2 Note on Transmission

The transmission sequence for the CRC code differs based on whether the transmission is LSB-first. Figure 24.4 shows an LSB-first data transmission.

When transmitting 32-bit data (for operation executed on 8 bits in parallel)

1. CRC code

After specifying the method for generation calculation, write data to CRCDIR in order of (1), (2), (3), and (4).



2. Transmit data

(i) When transmission is LSB-first

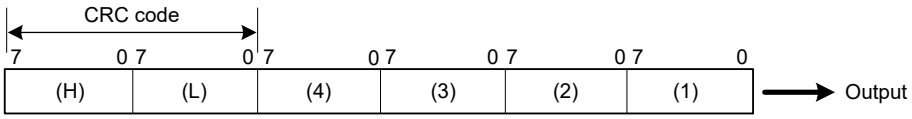


Figure 24.4 LSB-first data transmission

25. 12-bit A/D Converter (ADC12)

This is the ADC_D version of the ADC12 peripheral module. ADC_D is referred to as ADC12 in this chapter.

25.1 Overview

The A/D converter is used to convert analog input signals into digital values, and is configured to control up to 15 channels of A/D converter analog inputs (AN000 to AN012, AN021 and AN022). 12-bit, 10-bit, or 8-bit resolution can be selected by the ADTYP[1:0] bits of the A/D converter mode register 2 (ADM2). The A/D converter has the following function.

Table 25.1 lists the ADC12 specifications and Figure 25.1 shows a block diagram of ADC12.

Table 25.1 ADC12 specifications (1 of 2)

Parameter	Specifications
Number of units	One unit
Input channels	Up to 15 channels (AN000 to AN012, AN021 and AN022)*1
Extended analog function	Temperature sensor output, internal reference voltage
A/D conversion method	Successive approximation method
Resolution	12-bit/10-bit/8-bit
Conversion time	For details, see section 25.2.1. ADM0 : A/D Converter Mode Register 0 to section 25.2.1. ADM0 : A/D Converter Mode Register 0.
A/D conversion clock	PCLKB and A/D conversion clock f_{AD} can be set with the following division ratios : f_{AD} to PCLKB frequency ratio = 1:1, 1:2, 1:4, 1:8, 1:16, 1:32
Data registers	• 5 registers for 12-bit/10-bit A/D conversion • 5 registers for 8-bit A/D conversion
Operating modes	Various A/D conversion modes can be specified by using the mode combinations shown in Table 25.2 and Table 25.3.
Conditions for A/D conversion start	• Software trigger • Hardware trigger from the Event Link Controller • Hardware trigger from Timer Array Unit channel 1 count or capture end interrupt • Hardware trigger from Realtime clock interrupt • Hardware trigger from 32-bit interval timer interrupt
Functions	• 12-bit, 10-bit, or 8-bit resolution can be selected by ADTYP[1:0] bits. • Trigger mode has 4 modes, Software trigger no-wait mode, Software trigger wait mode, Hardware trigger no-wait mode, and Hardware trigger wait mode. • Channel selection mode has 2 modes, Select mode and Scan mode. • Conversion operation mode has 2 modes, One-shot conversion mode and Sequential conversion mode. • Operation voltage mode has 4 modes, Normal 1, Normal 2, Low voltage 1, and Low voltage 2. • Snooze mode function
Interrupt sources	• In Select mode, an A/D conversion end interrupt request (ADC12_ADI) can be generated on completion of single scan. • In Scan mode, an A/D conversion end interrupt request (ADC12_ADI) can be generated on completion of all the selected channel scans. • The A/D conversion results and ADUL register value are compared, and interrupt signal (ADC12_ADI) generation is controlled in the range specified by the ADRCK bit of A/D converter mode register 2 (ADM2). • The A/D conversion results and ADLL register value are compared, and interrupt signal (ADC12_ADI) generation is controlled in the range specified by the ADRCK bit of A/D converter mode register 2 (ADM2). • ADC12_ADI can activate the Data Transfer Controller (DTC).
ELC interface	• In Select mode, an event can be generated on completion of single scan. • In Scan mode, an event can be generated on completion of all the selected channel scans. • The A/D conversion results and ADUL register value are compared, and event generation is controlled in the range specified by the ADRCK bit of A/D converter mode register 2 (ADM2). • The A/D conversion results and ADLL register value are compared, and event generation is controlled in the range specified by the ADRCK bit of A/D converter mode register 2 (ADM2). • Conversion can be started by a trigger from the ELC.

Table 25.1 ADC12 specifications (2 of 2)

Parameter	Specifications
Reference voltage	• VREFH0, VCC, or internal reference voltage (BGR) (external reference voltage or output voltage from reference voltage generation circuit) can be selected as the analog reference voltage. • VREFL0 or VSS can be selected as the analog reference ground.
Module-stop function	Module-stop state can be set to reduce power consumption.*2

Note 1. AN000 to AN012, AN021, AN022 for LFQFP 64-pin
AN000 to AN010, AN021, AN022 for LFQFP/HWQFN 48-pin
AN000 to AN007, AN021, AN022 for LQFP/HWQFN 32-pin
Note 2. For details, see section 9, Low Power Modes.

Table 25.2 A/D conversion mode

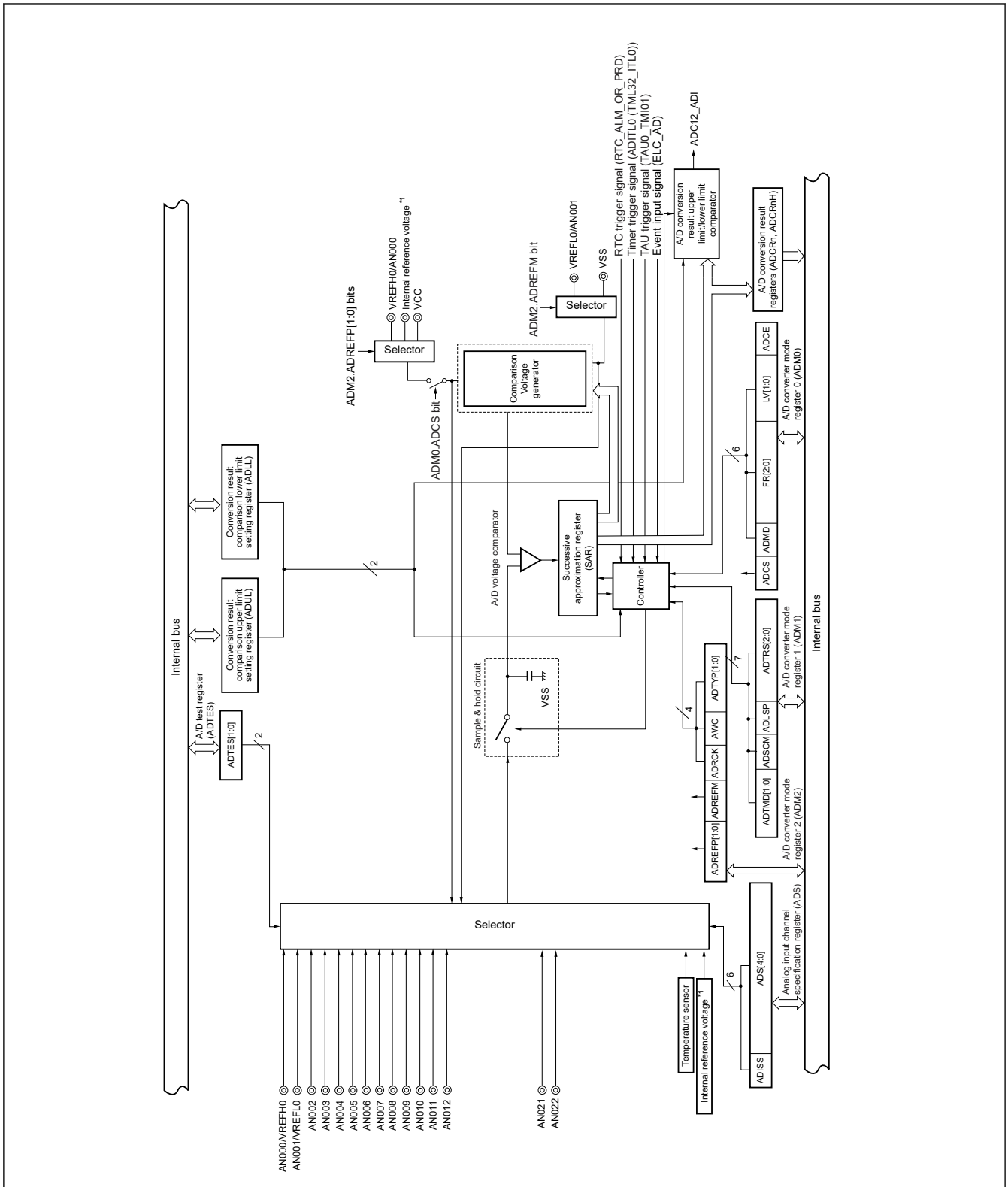
Function	Mode	Specification
Trigger mode	Software trigger no-wait mode	Conversion is started by setting the ADCE bit to 1 by software, and then setting ADCS to 1 after the A/D power supply stabilization wait time has passed.
	Software trigger wait mode	The power is turned on by setting the ADCS bit to 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 A/D converter is turned on by detecting a hardware trigger while the 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 AN000 to AN012 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 25.3 shows the sampling clock cycle for each operation voltage mode.

Table 25.3 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, VCC voltage, VREFH0 voltage, trigger mode, and PCLKB. See section 25.2.1. ADM0 : A/D Converter Mode Register 0 for details.



Note: n = 0 to 3
Note: Analog input pins in this figure are for a 32-pin product.
Note 1. For details about the internal reference voltage, see section 31, Electrical Characteristics.

Figure 25.1 Block diagram of A/D converter

The A/D converter includes the following hardware.

1. AN000 to AN012, AN021 and AN022 pins

These are the analog input pins of the 15 channels of the 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 A/D converter. This can be selected from VREFH0, the internal reference voltage*1, and VCC.

Note 1. For details about the internal reference voltage, see section 31, 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 (ADC12_ADI) 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 this register is used to specify 12-bit resolution, it holds the A/D conversion result in its lower 12 bits (the higher 4 bits are fixed to 0).
When this register is used to specify 10-bit resolution, 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 ADC12_ADI through the A/D conversion result upper limit/lower limit comparator.

9. VREFH0 pin
This pin inputs an external reference voltage (VREFH0).
If using VREFH0 as the '+' side reference voltage of the A/D converter, set the ADREFP[1:0] bits of A/D converter mode register 2 (ADM2) to 01b, respectively.
The analog signals input to AN000 to AN012, AN021 and AN022 are converted to digital signals based on the voltage applied between VREFH0 and the '-' side reference voltage (VREFL0/V_{SS}).

In addition to VREFH0, it is possible to select V_{CC} or the internal reference voltage*1 as the '+' side reference voltage of the A/D converter.

Note 1. For details about the internal reference voltage, see section 31, Electrical Characteristics.

10. VREFL0 pin
This pin inputs an external reference voltage (VREFL0). To use VREFL0 as the '-' side reference voltage of the A/D converter, set the ADREFM bit of the ADM2 register to 1.
In addition to VREFL0, it is possible to select V_{SS} as the '-' side reference voltage of the A/D converter.

11. Testing of the A/D converter
This test checks whether the A/D converter is operating normally by converting the A/D converter's positive and negative reference voltages, analog input channels (ANxxx), temperature sensor output voltage, and internal reference voltage.

Note: n = 0 to 3

25.2 Registers to Control the A/D Converter

The following registers are used to control the A/D converter.

- section 25.2.1. ADM0 : A/D Converter Mode Register 0
- section 25.2.2. ADM1 : A/D Converter Mode Register 1
- section 25.2.3. ADM2 : A/D Converter Mode Register 2
- section 25.2.4. ADCR/ADCRn: 12-bit or 10-bit A/D Conversion Result Register n (n = 0 to 3)
- section 25.2.5. ADCRH/ADCRnH : 8-bit A/D Conversion Result Register n (n = 0 to 3)
- section 25.2.6. ADS : Analog Input Channel Specification Register
- section 25.2.7. ADUL : Conversion Result Comparison Upper Limit Setting Register
- section 25.2.8. ADLL : Conversion Result Comparison Lower Limit Setting Register
- section 25.2.9. ADTES : A/D Test Register

25.2.1 ADM0 : A/D Converter Mode Register 0

Base address: ADC_D = 0x400A_1800

Offset address: 0x0000

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

Bit	Symbol	Function	R/W
6	ADMD	Specification of the A/D conversion channel selection mode 0: Select mode 1: Scan mode	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 1. For details of the FR[2:0], LV[1:0] bits, and A/D conversion, see Table 25.9 to Table 25.10.
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.

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 25.6. A/D Converter Setup Procedure.

Table 25.4 Relationship between ADCS and ADCE bits, including A/D operation status

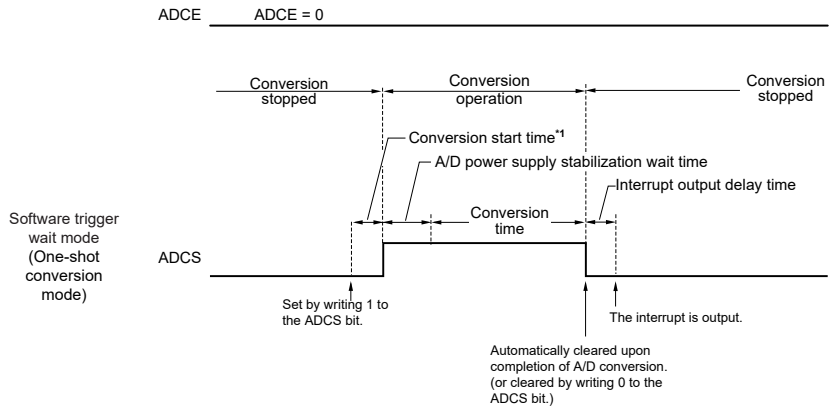
ADCS	ADCE	A/D conversion mode	A/D operation state
0	0	All modes	Conversion stopped state
0	1	Hardware trigger wait mode	Trigger standby state
		Other than hardware trigger wait mode	Conversion standby state
1	0	Software trigger wait mode	Conversion operation state
		Other than software trigger wait mode	Conversion stopped state
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 25.5 shows the conditions for setting and clearing the ADCS bit.

Table 25.5 Conditions for setting and clearing the ADCS bit

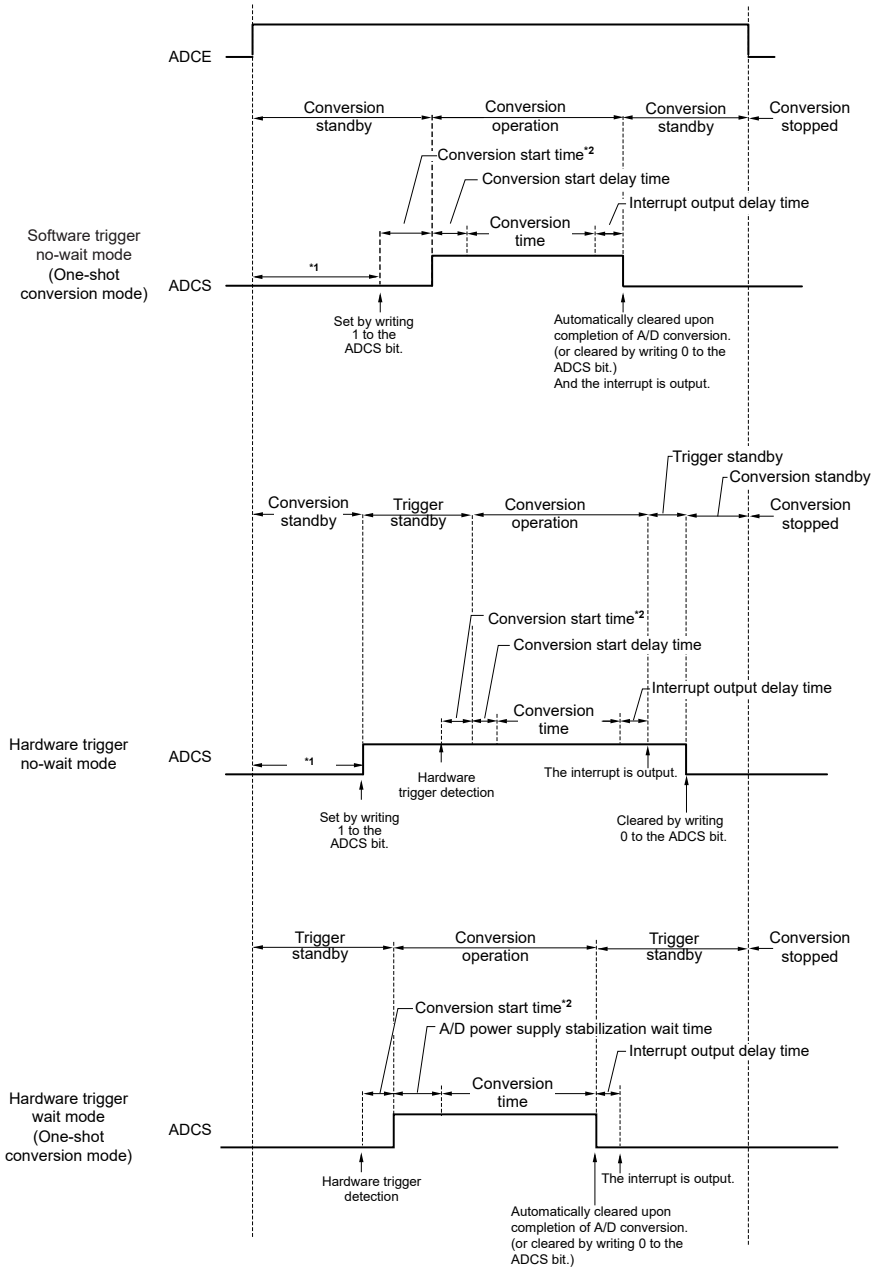
A/D conversion mode			Set conditions	Clear conditions
Software trigger no-wait mode	Select mode	Sequential conversion mode	When 1 is written to ADCS	When 0 is written to ADCS
		One-shot conversion mode		• When 0 is written to ADCS • The bit is automatically cleared to 0 when A/D conversion ends.
	Scan 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 25.2 and Figure 25.3.



Note 1. The maximum conversion start time takes the time shown in Table 25.6.

Figure 25.2 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 25.6.

Figure 25.3 Timing when 12-bit A/D Converter is used (other than software trigger wait mode)

Table 25.6 shows the conversion start time with FR[2:0] and ADLSP bits setting.

Table 25.6 Settings of conversion start time (1 of 2)

ADM1	ADM0		Conversion start time (number of PCLKB clock)	
ADSLP	FR[2:0]	Conversion clock (f_{AD})	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 25.6 Settings of conversion start time (2 of 2)

ADM1	ADM0	Conversion clock (f_{AD})	Conversion start time (number of PCLKB clock)	
ADSLP	FR[2:0]		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: 2 PCLKB clock cycles + conversion start time + conversion time
- Hardware trigger wait mode: 2 PCLKB clock cycles + conversion start time + A/D power supply stabilization wait time + conversion time + 5 µs

Table 25.7 shows the relationship between operation voltage mode and conversion time.

Table 25.7 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 25.8 shows the relationship between conversion start delay time, A/D power supply stabilization wait time, and interrupt output delay time.

Table 25.8 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 25.8 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 25.9 to Table 25.10 show the A/D conversion time with FR[2:0], LV[1:0], and ADLSP bits setting.

Table 25.9 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 = 32	2.4 V ≤ VREFH0 ≤ VCC ≤ 5.5 V	66 × 32/ PCLKB	72 × 32/ PCLKB	258 × 32/ PCLKB	264 × 32/ PCLKB
	0	001b	PCLKB/16	16 ≤ PCLKB ≤ 32		66 × 16/ PCLKB	72 × 16/ PCLKB	258 × 16/ PCLKB	264 × 16/ PCLKB
	0	010b	PCLKB/8	8 ≤ PCLKB ≤ 32		66 × 8/ PCLKB	74 × 8/ PCLKB	258 × 8/ PCLKB	266 × 8/ PCLKB
	0	011b	PCLKB/4	4 < PCLKB ≤ 32		66 × 4/ PCLKB	78 × 4/ PCLKB	258 × 4/ PCLKB	270 × 4/ PCLKB
	0	100b	PCLKB/2	4 < PCLKB ≤ 32		66 × 2/ PCLKB	86 × 2/ PCLKB	258 × 2/ PCLKB	278 × 2/ PCLKB
	0	101b	PCLKB	4 < PCLKB ≤ 32		66 × 1/ PCLKB	102 × 1/ PCLKB	258 × 1/ PCLKB	294 × 1/ PCLKB
	1	011b	PCLKB/4	PCLKB = 4		66 × 4/ PCLKB	72 × 4/ PCLKB	258 × 4/ PCLKB	264 × 4/ PCLKB
	1	100b	PCLKB/2	2 ≤ PCLKB ≤ 4		66 × 2/ PCLKB	72 × 2/ PCLKB	258 × 2/ PCLKB	264 × 2/ PCLKB
	1	101b	PCLKB	1 ≤ PCLKB ≤ 4		66 × 1/ PCLKB	74 × 1/ PCLKB	258 × 1/ PCLKB	266 × 1/ PCLKB
	Other than the above, Setting prohibited		—	—	—	—	—	—	—

Table 25.9 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 = 32	2.4 V ≤ VREFH0 ≤ VCC ≤ 5.5 V	183 × 32/ PCLKB	189 × 32/ PCLKB	726 × 32/ PCLKB	732 × 32/ PCLKB
	0	001b	PCLKB/16	16 ≤ PCLKB ≤ 32		183 × 16/ PCLKB	189 × 16/ PCLKB	726 × 16/ PCLKB	732 × 16/ PCLKB
	0	010b	PCLKB/8	8 ≤ PCLKB ≤ 32		183 × 8/ PCLKB	191 × 8/ PCLKB	726 × 8/ PCLKB	734 × 8/ PCLKB
	0	011b	PCLKB/4	4 < PCLKB ≤ 32		183 × 4/ PCLKB	195 × 4/ PCLKB	726 × 4/ PCLKB	738 × 4/ PCLKB
	0	100b	PCLKB/2	4 < PCLKB ≤ 32		183 × 2/ PCLKB	203 × 2/ PCLKB	726 × 2/ PCLKB	746 × 2/ PCLKB
	0	101b	PCLKB	4 < PCLKB ≤ 32		183 × 1/ PCLKB	219 × 1/ PCLKB	726 × 1/ PCLKB	762 × 1/ PCLKB
	1	011b	PCLKB/4	PCLKB = 4		183 × 4/ PCLKB	189 × 4/ PCLKB	726 × 4/ PCLKB	732 × 4/ PCLKB
	1	100b	PCLKB/2	2 ≤ PCLKB ≤ 4		183 × 2/ PCLKB	189 × 2/ PCLKB	726 × 2/ PCLKB	732 × 2/ PCLKB
	1	101b	PCLKB	1 ≤ PCLKB ≤ 4		183 × 1/ PCLKB	191 × 1/ PCLKB	726 × 1/ PCLKB	734 × 1/ PCLKB
	Other than the above, Setting prohibited		—	—	—	—	—	—	—

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 25.2, Figure 25.3, Table 25.7, and Table 25.8.
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 PCLKB frequency and VCC conditions, see section 9.5.2. Operating Range. Set the frequency and VCC to satisfy this condition and section 9.5.2. Operating Range.
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}).

Note: The A/D conversion time must also be within the relevant range of conversion times described in section 31.6.1. A/D Converter 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.

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.

Table 25.10 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 = 32	1.8 V ≤ VREFH0 ≤ VCC ≤ 5.5 V	82 × 32/ PCLKB	88 × 32/ PCLKB	322 × 32/ PCLKB	328 × 32/ PCLKB
	0	001b	PCLKB/16	16 ≤ PCLKB ≤ 32	1.8 V ≤ VREFH0 ≤ VCC ≤ 5.5 V	82 × 16/ PCLKB	88 × 16/ PCLKB	322 × 16/ PCLKB	328 × 16/ PCLKB
	0	010b	PCLKB/8	8 ≤ PCLKB ≤ 32	1.8 V ≤ VREFH0 ≤ VCC ≤ 5.5 V	82 × 8/ PCLKB	90 × 8/ PCLKB	322 × 8/ PCLKB	330 × 8/ PCLKB
	0	011b	PCLKB/4	4 < PCLKB ≤ 32	1.8 V ≤ VREFH0 ≤ VCC ≤ 5.5 V	82 × 4/ PCLKB	94 × 4/ PCLKB	322 × 4/ PCLKB	334 × 4/ PCLKB
	0	100b	PCLKB/2	4 < PCLKB ≤ 16	1.8 V ≤ VREFH0 ≤ VCC ≤ 5.5 V	82 × 2/ PCLKB	102 × 2/ PCLKB	322 × 2/ PCLKB	342 × 2/ PCLKB
				4 < PCLKB ≤ 32	2.4 V ≤ VREFH0 ≤ VCC ≤ 5.5 V				
	0	101b	PCLKB	4 < PCLKB ≤ 8	1.8 V ≤ VREFH0 ≤ VCC ≤ 5.5 V	82 × 1/ PCLKB	118 × 1/ PCLKB	322 × 1/ PCLKB	358 × 1/ PCLKB
				4 < PCLKB ≤ 16	2.4 V ≤ VREFH0 ≤ VCC ≤ 5.5 V				
				4 < PCLKB ≤ 24	2.7 V ≤ VREFH0 ≤ VCC ≤ 5.5 V				
	1	011b	PCLKB/4	PCLKB = 4	1.6 V ≤ VREFH0 ≤ VCC ≤ 5.5 V	82 × 4/ PCLKB	88 × 4/ PCLKB	322 × 4/ PCLKB	328 × 4/ PCLKB
	1	100b	PCLKB/2	2 ≤ PCLKB ≤ 4	1.6 V ≤ VREFH0 ≤ VCC ≤ 5.5 V	82 × 2/ PCLKB	88 × 2/ PCLKB	322 × 2/ PCLKB	328 × 2/ PCLKB
	1	101b	PCLKB	1 ≤ PCLKB ≤ 4	1.6 V ≤ VREFH0 ≤ VCC ≤ 5.5 V	82 × 1/ PCLKB	90 × 1/ PCLKB	322 × 1/ PCLKB	330 × 1/ PCLKB
	Other than the above, Setting prohibited		—	—	—	—	—	—	—

Table 25.10 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 = 32	1.8 V ≤ VREFH0 ≤ VCC ≤ 5.5 V	109 × 32/ PCLKB	115 × 32/ PCLKB	430 × 32/ PCLKB	436 × 32/ PCLKB
	0	001b	PCLKB/16	16 ≤ PCLKB ≤ 32	1.8 V ≤ VREFH0 ≤ VCC ≤ 5.5 V	109 × 16/ PCLKB	115 × 16/ PCLKB	430 × 16/ PCLKB	436 × 16/ PCLKB
	0	010b	PCLKB/8	8 ≤ PCLKB ≤ 32	1.8 V ≤ VREFH0 ≤ VCC ≤ 5.5 V	109 × 8/ PCLKB	117 × 8/ PCLKB	430 × 8/ PCLKB	438 × 8/ PCLKB
	0	011b	PCLKB/4	4 < PCLKB ≤ 32	1.8 V ≤ VREFH0 ≤ VCC ≤ 5.5 V	109 × 4/ PCLKB	121 × 4/ PCLKB	430 × 4/ PCLKB	442 × 4/ PCLKB
	0	100b	PCLKB/2	4 < PCLKB ≤ 16	1.8 V ≤ VREFH0 ≤ VCC ≤ 5.5 V	109 × 2/ PCLKB	129 × 2/ PCLKB	430 × 2/ PCLKB	450 × 2/ PCLKB
				4 < PCLKB ≤ 32	2.4 V ≤ VREFH0 ≤ VCC ≤ 5.5 V				
	0	101b	PCLKB	4 < PCLKB ≤ 8	1.8 V ≤ VREFH0 ≤ VCC ≤ 5.5 V	109 × 1/ PCLKB	145 × 1/ PCLKB	430 × 1/ PCLKB	466 × 1/ PCLKB
				4 < PCLKB ≤ 16	2.4 V ≤ VREFH0 ≤ VCC ≤ 5.5 V				
				4 < PCLKB ≤ 24	2.7 V ≤ VREFH0 ≤ VCC ≤ 5.5 V				
	1	011b	PCLKB/4	PCLKB = 4	1.6 V ≤ VREFH0 ≤ VCC ≤ 5.5 V	109 × 4/ PCLKB	115 × 4/ PCLKB	430 × 4/ PCLKB	436 × 4/ PCLKB
	1	100b	PCLKB/2	2 ≤ PCLKB ≤ 4	1.6 V ≤ VREFH0 ≤ VCC ≤ 5.5 V	109 × 2/ PCLKB	115 × 2/ PCLKB	430 × 2/ PCLKB	436 × 2/ PCLKB
	1	101b	PCLKB	1 ≤ PCLKB ≤ 4	1.6 V ≤ VREFH0 ≤ VCC ≤ 5.5 V	109 × 1/ PCLKB	117 × 1/ PCLKB	430 × 1/ PCLKB	438 × 1/ PCLKB
	Other than the above, Setting prohibited		—	—	—	—	—	—	—

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 25.2, Figure 25.3, Table 25.7, and Table 25.8.
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 PCLKB frequency and VCC conditions, see section 9.5.2. Operating Range. Set the frequency and VCC to satisfy this condition and section 9.5.2. Operating Range.
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}).

Note: The A/D conversion time must also be within the relevant range of conversion times described in section 31.6.1. A/D Converter 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.

Figure 25.4 shows the timing of sampling and A/D conversion.

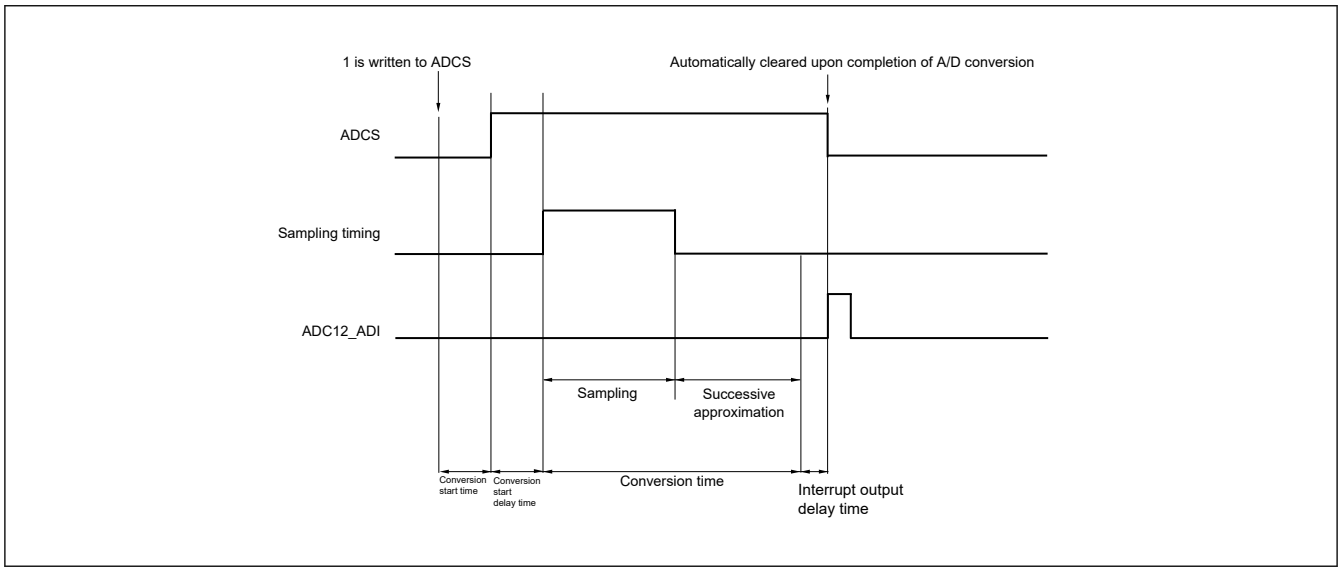


Figure 25.4 12-bit A/D converter sampling and A/D conversion timing (example for software trigger no-wait mode, select mode, and one-shot conversion mode)

25.2.2 ADM1 : A/D Converter Mode Register 1

Base address: ADC_D = 0x400A_1800

Offset address: 0x0002

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_TMI01) 0 1 0: Realtime clock interrupt signal (RTC_ALM_OR_PRD) 0 1 1: 32-bit interval timer event signal (ADITL0 (= TML32_ITL0)) 1 0 0: Event input signal (ELC_AD)*1 Others: Setting prohibited.	R/W
3	ADLSP	PCLKB Input Frequency Setting 0: 4 MHz < PCLKB ≤ 32 MHz 1: 1 MHz ≤ PCLKB ≤ 4 MHz	R/W
4	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	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 1. A/D converter can not be triggered by the ELC in the Snooze mode.

This register is used to specify the A/D conversion trigger, conversion mode, and hardware trigger signal.

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: 2 PCLKB clock cycles + conversion start time + A/D conversion time

Hardware trigger wait mode: 2 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_PRD or ADITL0 (= TML32_ITL0) is not recognized as a valid hardware trigger for up to 4 PCLKB cycles after the first RTC_ALM_OR_PRD or ADITL0 (= TML32_ITL0) is input.

25.2.3 ADM2 : A/D Converter Mode Register 2

Base address: ADC_D = 0x400A_1800

Offset address: 0x0110

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 resolution 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 (ADC12_ADI) is output when the ADLL register ≤ the ADCRn register ≤ the ADUL register (AREA 1). 1: The interrupt signal (ADC12_ADI) 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
5	ADREFM	Selection of the '-' Side Reference Voltage of the A/D Converter 0: Supplied from V_{SS} 1: Supplied from VREFL0/AN001	R/W
7:6	ADREFP[1:0]	Selection of the '+' Side Reference Voltage Source of the A/D Converter 0 0: Supplied from V_{CC} 0 1: Supplied from VREFH0/AN000 1 0: Supplied from the internal reference voltage*1 1 1: Discharge the internal circuitry	R/W

Note 1. For details about the internal reference voltage, see section 31, Electrical Characteristics.

This register is used to select the '+' side and '-' side reference voltages of the 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.

Note: Only rewrite the value of the ADM2 register while conversion operation is stopped (ADCS = 0, ADCE = 0).

Note: Do not set the ADREFP[1:0] bits to 10 when shifting to Software Standby 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 A/D converter reference voltage current (I_{ADREF}) indicated in section 31.3.2. Operating and Standby Current will be added.

Note: When using VREFH0 and VREFL0, specify AN000 and AN001 as the analog inputs and set the Pin Mode Control bit (PMC) to 1, the N-Channel Open-Drain Control bit (NCODR) to 0, and the Port Direction bit (PDR) to 0 in the Port mn Pin Function Select Register PmnPFS_A.

ADTYP[1:0] bits (Selection of the resolution)

These bits are used for selection of the resolution.

AWC bit (Specification of the Snooze Mode)

This bit is used for specification of the Snooze mode.

When there is a hardware trigger signal in the Software Standby mode, the Software Standby mode is exited, and A/D conversion is performed without operating the CPU (the Snooze mode).

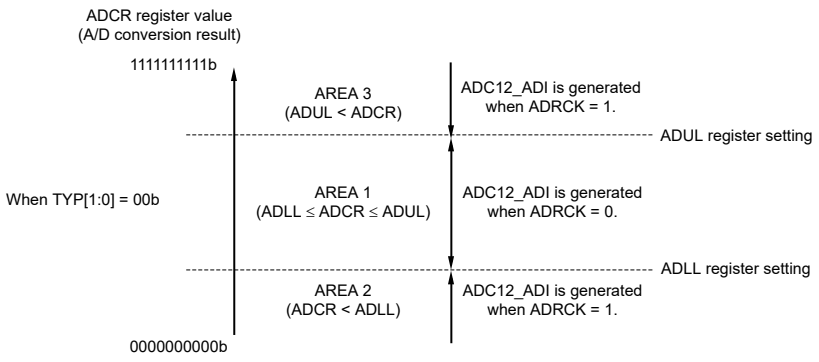
- When using the Snooze mode function, set AWC to 1 in hardware trigger wait mode.
- Using the Snooze mode function in the software trigger no-wait mode, software trigger wait mode, or hardware trigger no-wait mode is prohibited.
- Using the Snooze mode function in the sequential conversion mode and hardware trigger wait mode is prohibited.
- When using the Snooze mode function, specify a hardware trigger interval of at least "shift time to Snooze mode*1 + conversion start time + A/D power supply stabilization wait time + A/D conversion time + 2 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.

Note 1. Refer to Table 31.22 in section 31.4.2. Wakeup Time.

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 25.5 shows the generation range of the interrupt signal (ADC12_ADI) for AREA 1 to AREA 3.



Note: If ADC12_ADI does not occur, the A/D conversion result is not stored in the ADCRn or ADCRnH register.

Figure 25.5 ADRCK bit interrupt signal generation range (in 10-bit resolution mode)

ADREFM bit (Selection of the '-' Side Reference Voltage of the 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 A/D Converter)

These bits are used for selection of the '+' side reference voltage source of the 12-bit A/D converter.

Use Table 25.11 procedure to rewrite the ADREFP[1:0] bits.

Table 25.11 Register settings for ADREFP[1:0] rewrite

Step		Process	Remark
Register settings for ADREFP [1:0] Rewrite	<1>	Set ADM0.ADCE = 0	ADC is stopped.
	<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.CE = 1	ADC is start.
	<8>	Reference voltage stabilization wait time (B)	B = 1 µs + 2 cycles of conversion clock (f_{AD})
	<9>	Start the A/D conversion	—

25.2.4 ADCR/ADCRn: 12-bit or 10-bit A/D Conversion Result Register n (n = 0 to 3)

Base address: ADC_D = 0x400A_1800

Offset address: 0x0006 (ADCR)
0x0120 (ADCR0)
0x0122 (ADCR1)
0x0124 (ADCR2)
0x0126 (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	12-bit or 10-bit Resolution A/D Converter Result for Channel n	R

Note: When selecting 12-bit mode, the upper 4 bits are fixed at 0, and when selecting 10-bit mode, the lower 6 bits are fixed at 0.
Note: The contents of the ADCR register are stored in the ADCR0 register.

ADCRn is a 16-bit register that holds the A/D conversion result.

Each time A/D conversion ends, the conversion result is loaded from the successive approximation register (SAR).

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 25.5), the result is not stored.

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 ADCRn 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.

25.2.5 ADCRH/ADCRnH : 8-bit A/D Conversion Result Register n (n = 0 to 3)

Base address: ADC_D = 0x400A_1800

Offset address: 0x0007 (ADCRH)
0x0121 (ADCR0H)
0x0123 (ADCR1H)
0x0125 (ADCR2H)
0x0127 (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	8-bit Resolution A/D Converter Result for Channel n	R

Note: The contents of the ADCRH register are stored in theADCR0H register.

ADCRnH is an 8-bit register that holds the A/D conversion result. The higher 8 bits of 12-bit resolution are stored*1.

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 25.5), the result is not stored.

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.

25.2.6 ADS : Analog Input Channel Specification Register

Base address: ADC_D = 0x400A_1800

Offset address: 0x0001

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 25.12 to Table 25.13)	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 25.12 to Table 25.13) 0: External input 1: Internal circuit input	R/W

This register specifies the input channel of the analog voltage to be A/D converted.

Table 25.12 and Table 25.13 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 25.12 Input source selection by ADS[4:0] bits and ADISS bit in select mode

ADISS	ADS[4:0]	Analog input channel	Input source
0	00000b	AN000	P010
0	00001b	AN001	P011
0	00010b	AN002	P008
0	00011b	AN003	P009
0	00100b	AN004	P012
0	00101b	AN005	P013
0	00110b	AN006	P014
0	00111b	AN007	P015
0	01000b	AN008	P000
0	01001b	AN009	P001
0	01010b	AN010	P002
0	01011b	AN011	P003
0	01100b	AN012	P004
0	10101b	AN021	P101
0	10110b	AN022	P100
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 31, Electrical Characteristics.

<Scan mode (ADMD = 1)>

Table 25.13 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	AN000	AN001	AN002	AN003
0	00001b	AN001	AN002	AN003	AN004
0	00010b	AN002	AN003	AN004	AN005
0	00011b	AN003	AN004	AN005	AN006
0	00100b	AN004	AN005	AN006	AN007
0	00101b	AN005	AN006	AN007	AN008
0	00110b	AN006	AN007	AN008	AN009
0	00111b	AN007	AN008	AN009	AN010
0	01000b	AN008	AN009	AN010	AN011
0	01001b	AN009	AN010	AN011	AN012
Other than the above		Setting prohibited			

Note: Rewrite the value of the ADISS bit while conversion is stopped (ADCS = 0, ADCE = 0).

Note: If using VREFH0 as the '+' side reference voltage of the A/D converter, do not select AN000 as an A/D conversion channel.

Note: If using VREFL0 as the '-' side reference voltage of the A/D converter, do not select AN001 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

25.6.5. Example of Using the ADC12 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 31, 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 A/D converter reference voltage current (IADREF) indicated in section 31.3.2. Operating and Standby Current will be added.

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.

Note: When the setting of the ADISS bit is 1, the software trigger wait mode and one-shot conversion mode cannot be used at the same time.

25.2.7 ADUL : Conversion Result Comparison Upper Limit Setting Register

Base address: ADC_D = 0x400A_1800

Offset address: 0x0111

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 (ADC12_ADI) generation is controlled in the range specified by the ADRCK bit of A/D converter mode register 2 (ADM2) (shown in Figure 25.5).

25.2.8 ADLL : Conversion Result Comparison Lower Limit Setting Register

Base address: ADC_D = 0x400A_1800

Offset address: 0x0112

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 (ADC12_ADI) generation is controlled in the range specified by the ADRCK bit of A/D converter mode register 2 (ADM2) (shown in Figure 25.5).

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.

25.2.9 ADTES : A/D Test Register

Base address: ADC_D = 0x400A_1800

Offset address: 0x0113

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: ANxxx, 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 31, Electrical Characteristics.

This register is used to select the '+' side reference voltage or '-' side reference voltage for the converter, an analog input channel (ANxxx), the temperature sensor output voltage, or the internal reference voltage*1 as the target for A/D conversion.

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.

25.3 A/D Converter Operations

The 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 AV_{REF} 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 AV_{REF}, the MSB of the SAR register remains set to 1. If the analog input is smaller than 1/2 AV_{REF}, 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 (ADC12_ADI) can also be generated*1.

<8> Repeat steps <1> to <7>, until the ADCS bit is cleared to 0*2.

To stop the A/D converter, clear the ADCS bit to 0.

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 25.5), 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.

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 A/D converter. This can be selected from VREFH0, the internal reference voltage, and V_{CC}.

For details about the internal reference voltage, see section 31, Electrical Characteristics.

Note: n = 0 to 3

Figure 25.6 shows the conversion operation of the A/D converter during software trigger no-wait mode.

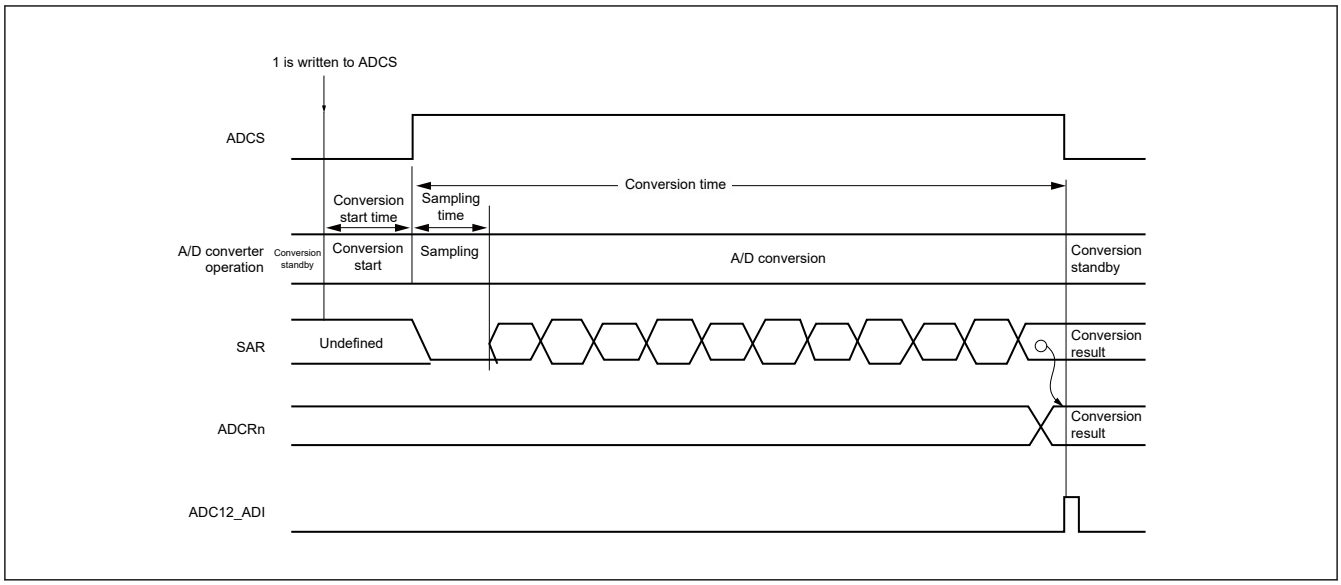


Figure 25.6 Conversion operation of A/D converter (software trigger no-wait mode)

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 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.

25.4 Input Voltage and Conversion Results

The relationship between the analog voltage input to the analog input pins (AN000 to AN012, AN021 and AN022) 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 25.7 shows relationship between analog input voltage and A/D conversion result.

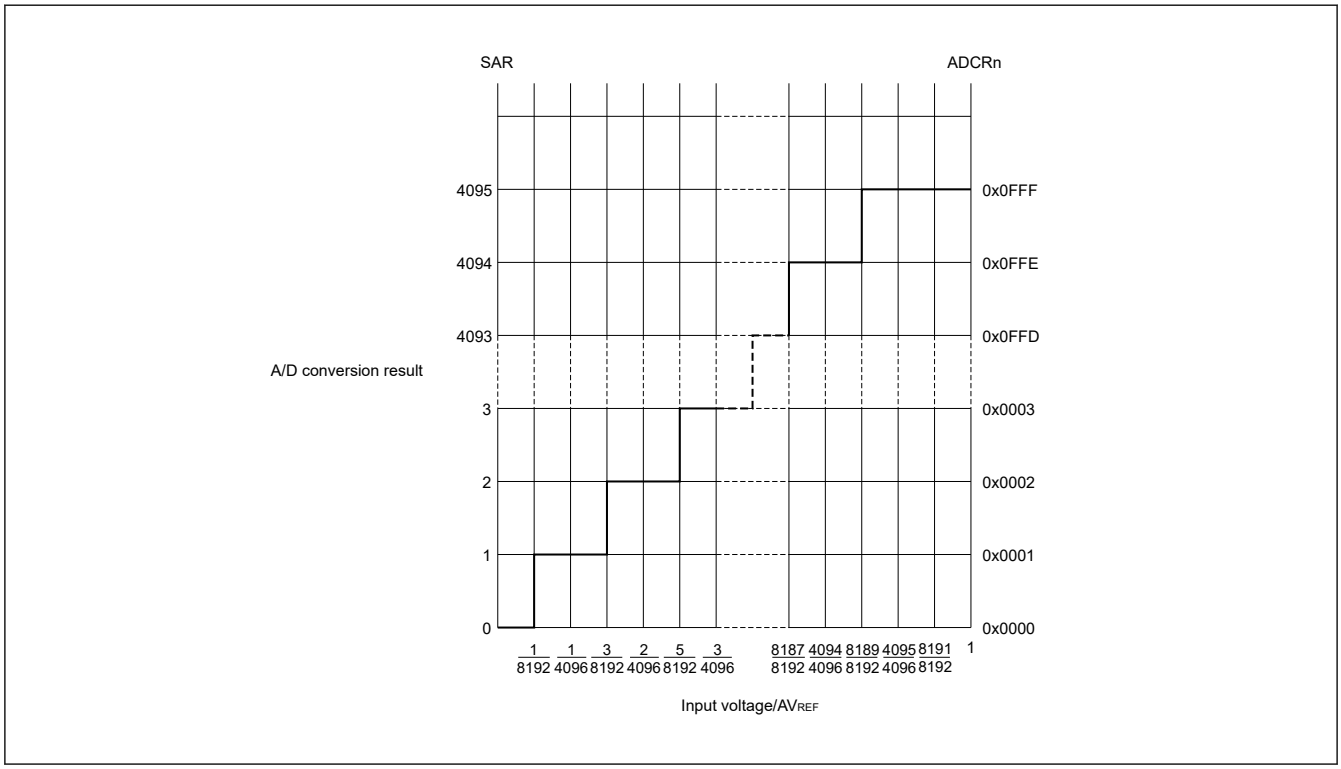


Figure 25.7 Relationship between analog input voltage and A/D conversion result

AV_{REF}: The '+' side reference voltage of the A/D converter. This can be selected from VREFH0 the internal reference voltage*1, and V_{CC}.

Note 1. For details about the internal reference voltage, see section 31, Electrical Characteristics.

25.5 A/D Converter Operation Modes

The operation of each A/D converter mode is described below. In addition, the procedure for specifying each mode is described in section 25.6. A/D Converter Setup Procedure.

25.5.1 Software Trigger No-wait Mode (Select Mode, Sequential Conversion Mode)

<1> In the stop state, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the A/D converter enters the 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 (ADC12_ADI) 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 A/D converter enters the standby state.

<8> When ADCE is cleared to 0 while in the A/D conversion standby state, the A/D converter enters the stop state.

Figure 25.8 shows the example of software trigger no-wait mode (select mode, sequential conversion mode) operation timing.

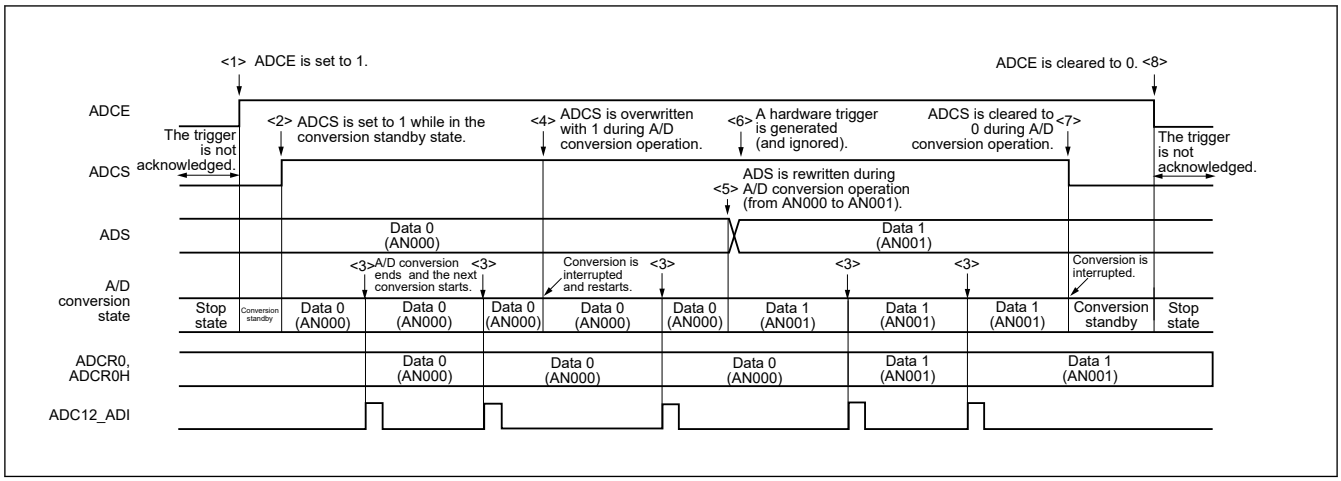


Figure 25.8 Example of software trigger no-wait mode (select mode, sequential conversion mode) operation timing

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 section 25.2.1. ADM0 : A/D Converter Mode Register 0 and section 25.2.1. ADM0 : A/D Converter Mode Register 0.)

25.5.2 Software Trigger No-wait Mode (Select Mode, One-shot Conversion Mode)

<1> In the stop state, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the A/D converter enters the 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 (ADC12_ADI) is generated.

<4> After A/D conversion ends, the ADCS bit is automatically cleared to 0, and the A/D converter enters the 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 A/D converter enters the standby state.

<8> When ADCE is cleared to 0 while in the A/D conversion standby state, the A/D converter enters the 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 25.9 shows the example of software select no-wait mode (select mode, one-shot conversion mode) operation timing.

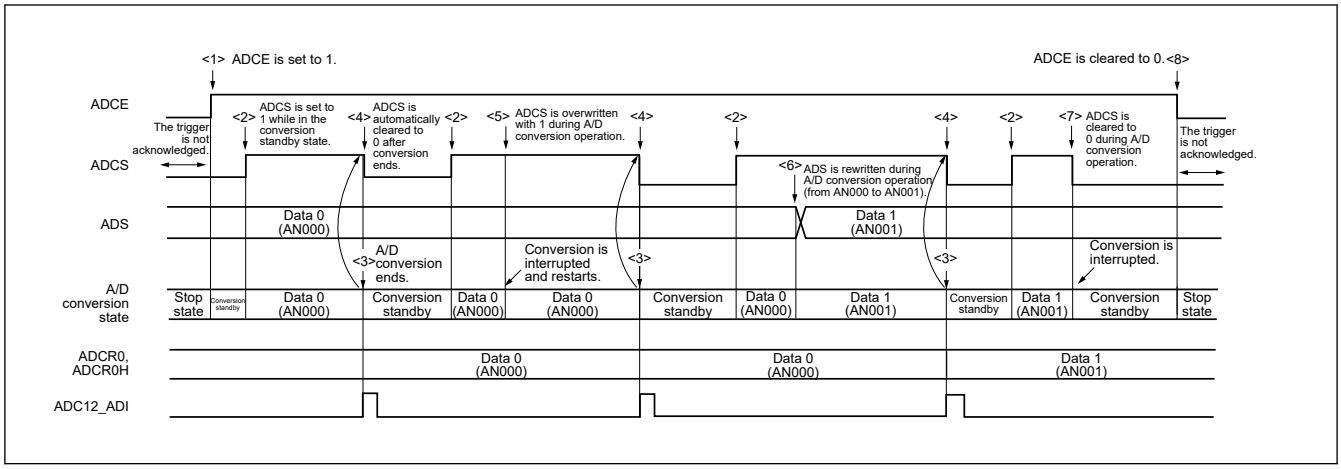


Figure 25.9 Example of software select no-wait mode (select mode, one-shot conversion mode) operation timing

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 section 25.2.1. ADM0 : A/D Converter Mode Register 0 and section 25.2.1. ADM0 : A/D Converter Mode Register 0.)

25.5.3 Software Trigger No-wait Mode (Scan Mode, Sequential Conversion Mode)

<1> In the stop state, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the A/D converter enters the 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 (ADC12_ADI) 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 A/D converter enters the standby state.

<8> When ADCE is cleared to 0 while in the A/D conversion standby state, the A/D converter enters the stop state.

Figure 25.10 shows the example of software trigger no-wait mode (scan mode, sequential conversion mode) operation timing.

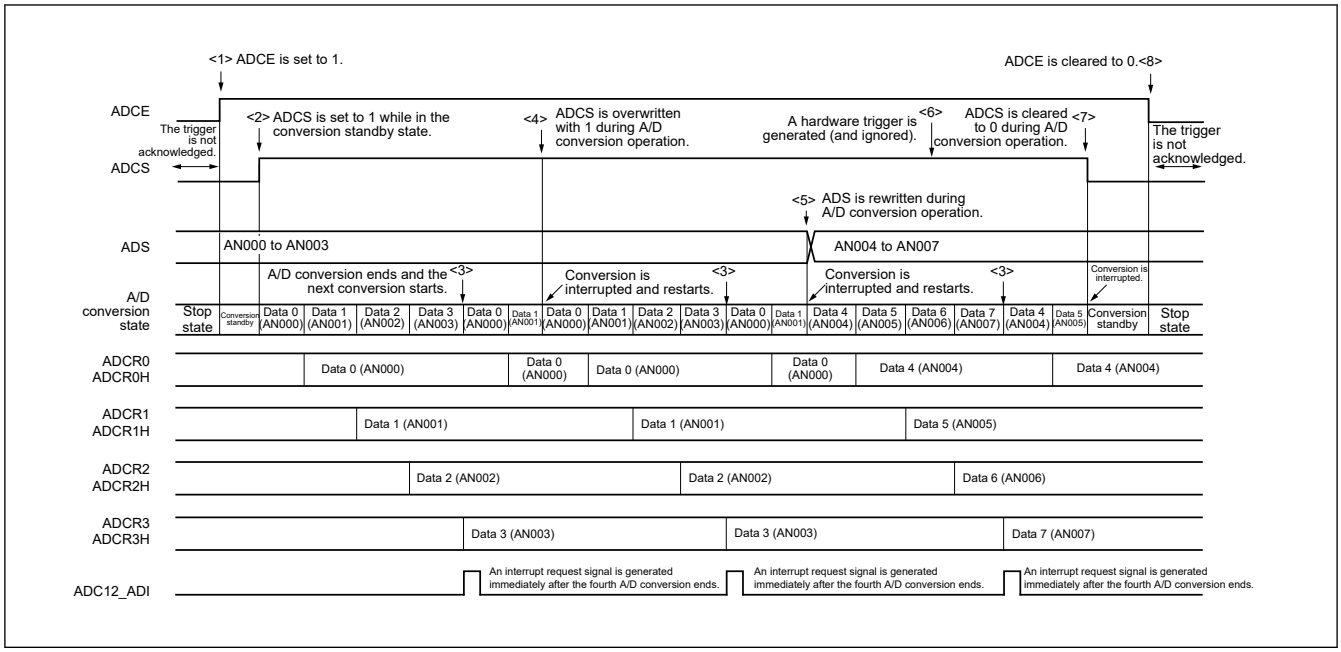


Figure 25.10 Example of software trigger no-wait mode (scan mode, sequential conversion mode) operation timing

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 section 25.2.1. ADM0 : A/D Converter Mode Register 0 and section 25.2.1. ADM0 : A/D Converter Mode Register 0.)

25.5.4 Software Trigger No-wait Mode (Scan Mode, One-shot Conversion Mode)

<1> In the stop state, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the A/D converter enters the 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 (ADC12_ADI) 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 A/D converter enters the 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 A/D converter enters the standby state.

<8> When ADCE is cleared to 0 while in the A/D conversion standby state, the A/D converter enters the 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 25.11 shows the example of software trigger no-wait mode (scan mode, one-shot conversion mode) operation timing.

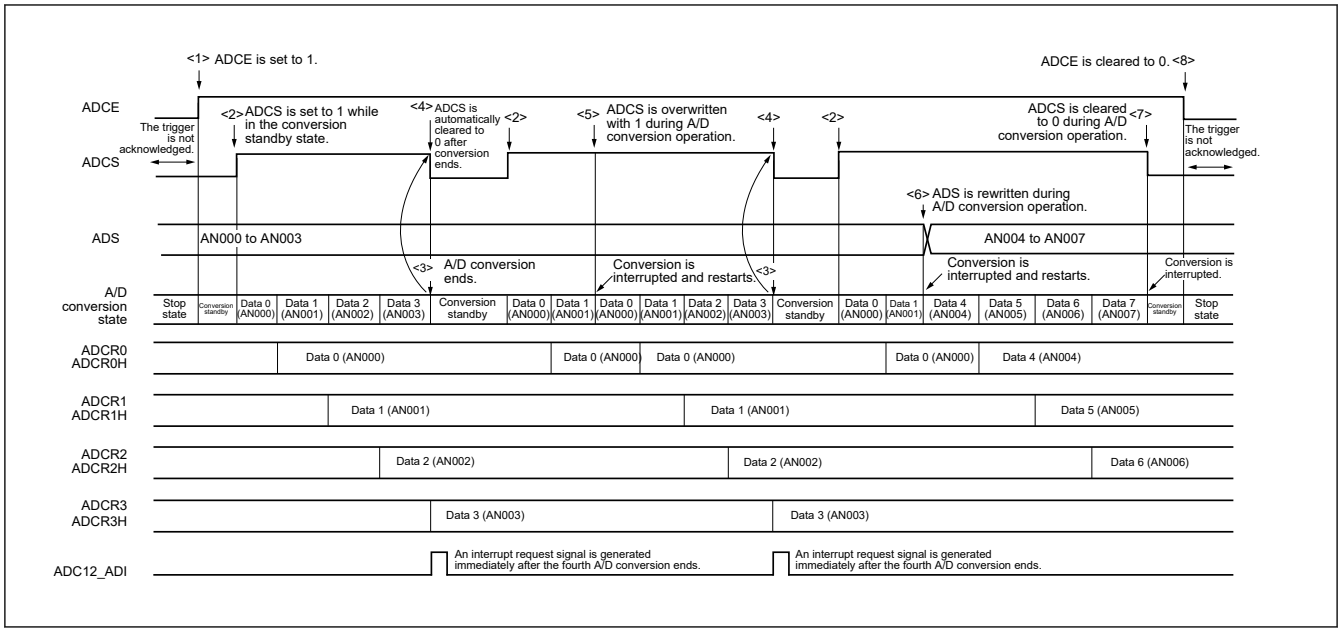


Figure 25.11 Example of software trigger no-wait mode (scan mode, one-shot conversion mode) operation timing

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 section 25.2.1. ADM0 : A/D Converter Mode Register 0 and section 25.2.1. ADM0 : A/D Converter Mode Register 0.)

25.5.5 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 (stop state).

<2> If ADCS is set to 1 in the 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 (ADC12_ADI) 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 A/D converter enters the stop state.

Figure 25.12 shows the example of software trigger wait mode (select mode, sequential conversion mode) operation timing.

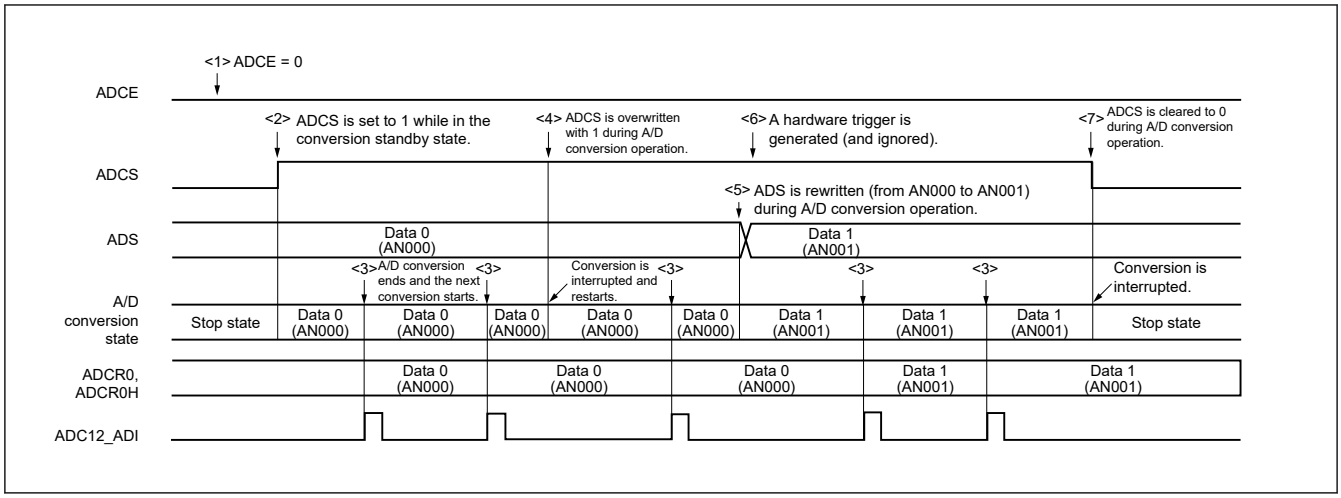


Figure 25.12 Example of software trigger wait mode (select mode, sequential conversion mode) operation timing

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 section 25.2.1. ADM0 : A/D Converter Mode Register 0 and section 25.2.1. ADM0 : A/D Converter Mode Register 0.)

25.5.6 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 (stop state).

<2> If ADCS is set to 1 in the 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 (ADC12_ADI) is generated.

<4> After A/D conversion ends, the ADCS bit is automatically cleared to 0, and the A/D converter enters the 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 A/D converter enters the stop state.

<8> When a hardware trigger is input during conversion operation, the trigger is not accepted.

Figure 25.13 shows the example of software trigger wait mode (select mode, one-shot conversion mode) operation timing.

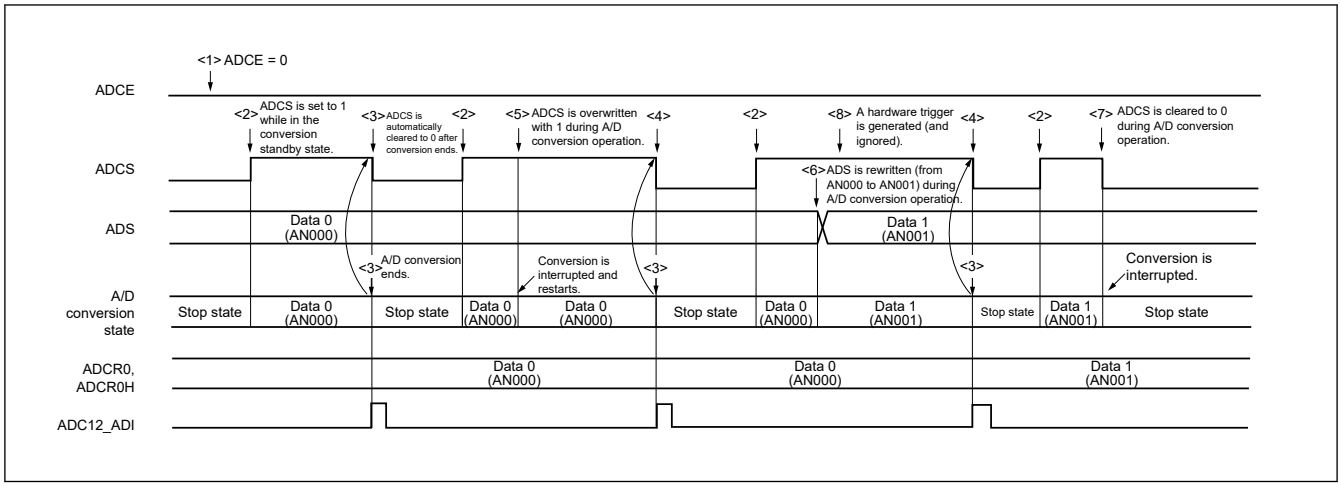


Figure 25.13 Example of software trigger wait mode (select mode, one-shot conversion mode) operation timing

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 section 25.2.1. ADM0 : A/D Converter Mode Register 0 and section 25.2.1. ADM0 : A/D Converter Mode Register 0.)

Note: In software trigger wait mode (select mode, one-shot conversion mode), the ADISS = 1 setting (input source = temperature sensor output voltage, internal reference voltage) cannot be used.

25.5.7 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 (stop state).

<2> If ADCS is set to 1 in the 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 (ADC12_ADI) 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 A/D converter enters the stop state.

Figure 25.14 shows the example of software trigger wait mode (scan mode, sequential conversion mode) operation timing.

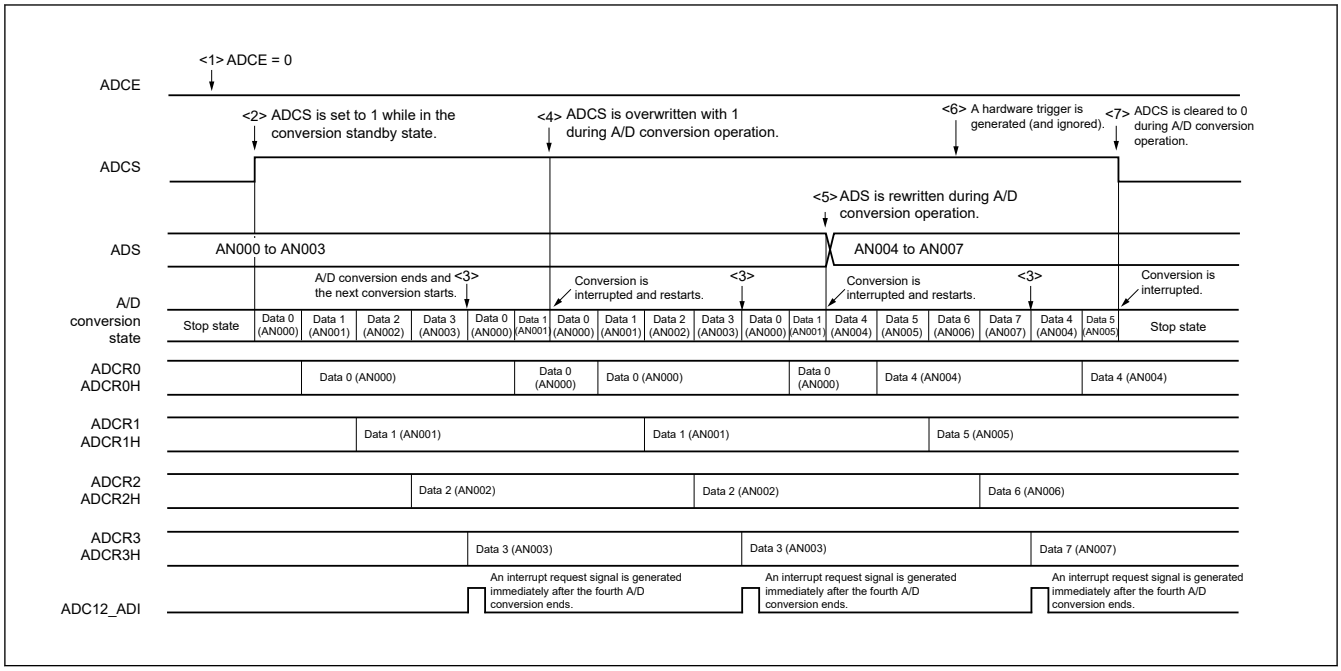


Figure 25.14 Example of software trigger wait mode (scan mode, sequential conversion mode) operation timing

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 section 25.2.1. ADM0 : A/D Converter Mode Register 0 and section 25.2.1. ADM0 : A/D Converter Mode Register 0.)

25.5.8 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 (stop state).

<2> If ADCS is set to 1 in the 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 (ADC12_ADI) 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 A/D converter enters the 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 A/D converter enters the stop state.

<8> When a hardware trigger is input during conversion operation, the trigger is not accepted.

Figure 25.15 shows the example of software trigger wait mode (scan mode, one-shot conversion mode) operation timing.

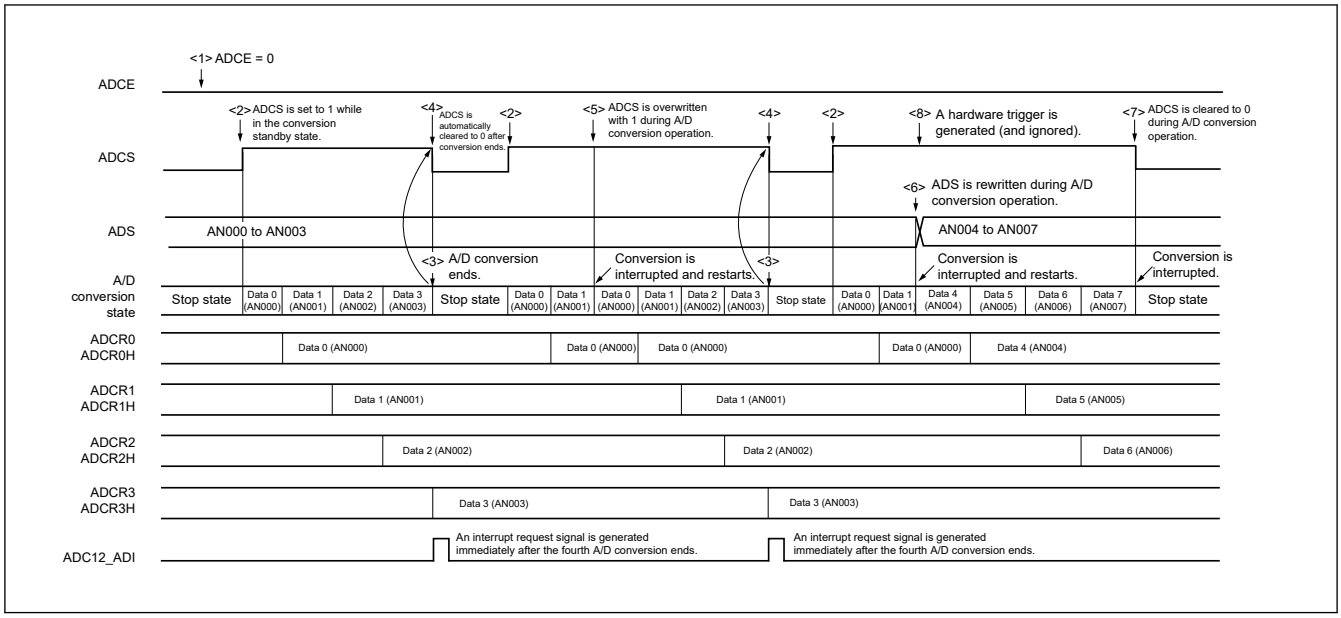


Figure 25.15 Example of software trigger wait mode (scan mode, one-shot conversion mode) operation timing

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 section 25.2.1. ADM0 : A/D Converter Mode Register 0 and section 25.2.1. ADM0 : A/D Converter Mode Register 0.)

25.5.9 Hardware Trigger No-wait Mode (Select Mode, Sequential Conversion Mode)

<1> In the stop state, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the A/D converter enters the 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 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 (ADC12_ADI) 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 A/D converter enters the standby state. However, the 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 A/D converter enters the stop state.

When ADCS = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 25.16 shows the example of hardware trigger no-wait mode (select mode, sequential conversion mode) operation timing.

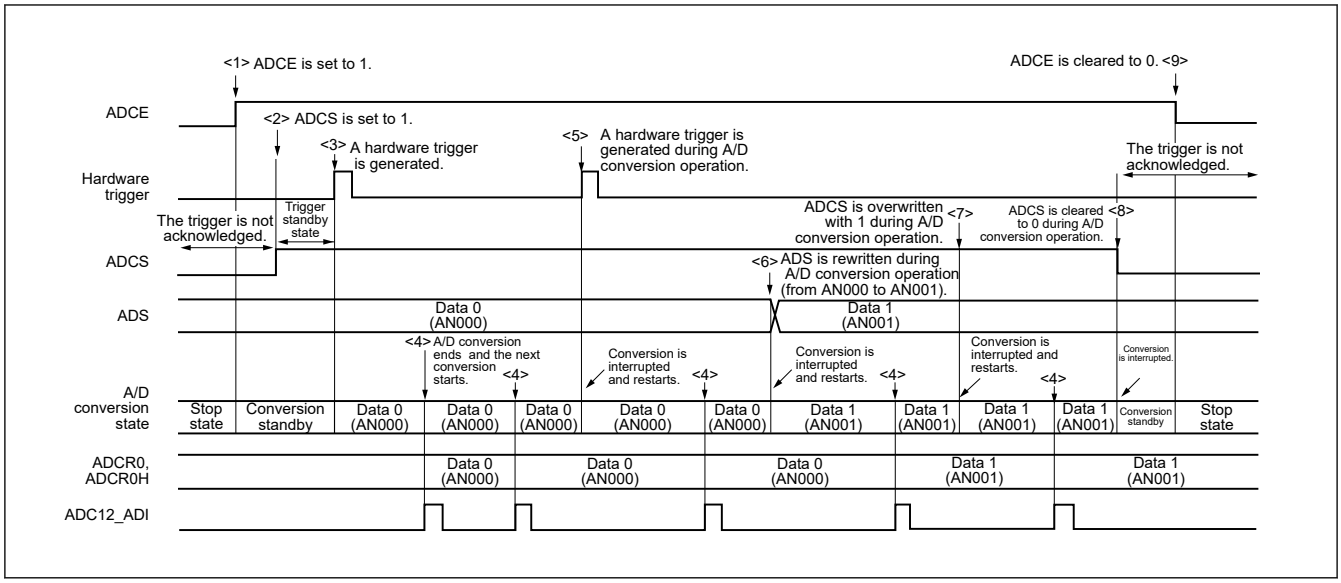


Figure 25.16 Example of hardware trigger no-wait mode (select mode, sequential conversion mode) operation timing

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 section 25.2.1. ADM0 : A/D Converter Mode Register 0 and section 25.2.1. ADM0 : A/D Converter Mode Register 0.)

25.5.10 Hardware Trigger No-wait Mode (Select Mode, One-shot Conversion Mode)

<1> In the stop state, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the A/D converter enters the 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 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 (ADC12_ADI) is generated.

<5> After A/D conversion ends, the ADCS bit remains set to 1, and the A/D converter enters the 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 A/D converter enters the standby state. However, the 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 A/D converter enters the stop state.

When ADCS = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 25.17 shows the example of hardware trigger no-wait mode (select mode, one-shot conversion mode) operation timing.

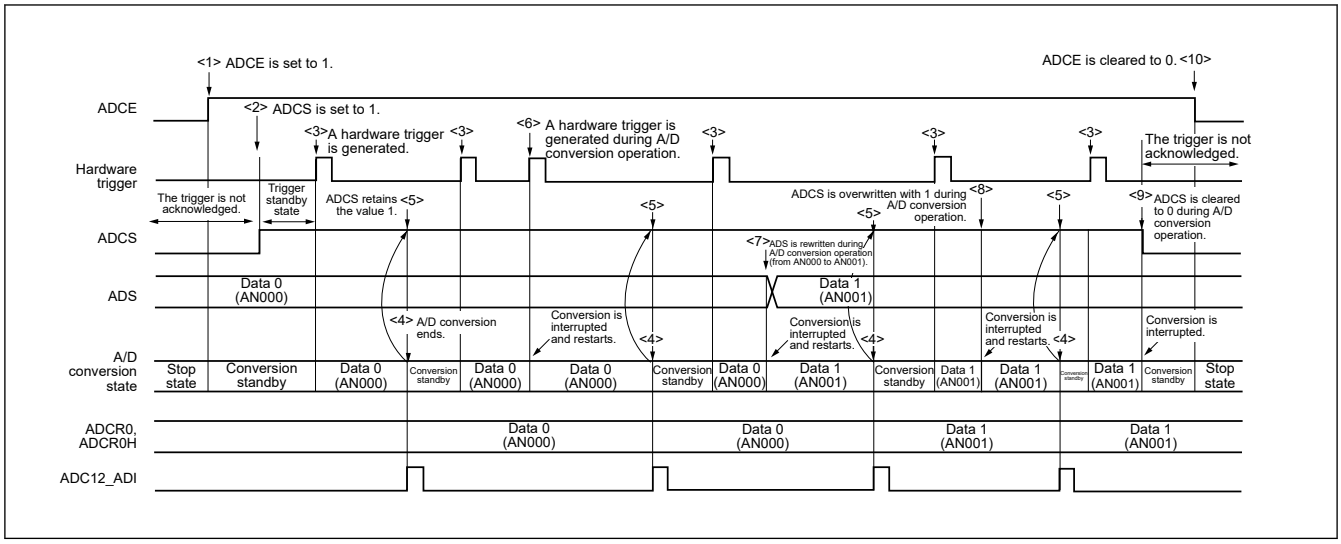


Figure 25.17 Example of hardware trigger no-wait mode (select mode, one-shot conversion mode) operation timing

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 section 25.2.1. ADM0 : A/D Converter Mode Register 0 and section 25.2.1. ADM0 : A/D Converter Mode Register 0.)

25.5.11 Hardware Trigger No-wait Mode (Scan Mode, Sequential Conversion Mode)

<1> In the stop state, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the A/D converter enters the 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 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 (ADC12_ADI) 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 A/D converter enters the standby state. However, the 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 A/D converter enters the stop state.

When ADCE = 0, any hardware trigger input is ignored and A/D conversion does not start.

Figure 25.18 shows the example of hardware trigger no-wait mode (scan mode, sequential conversion mode) operation timing.

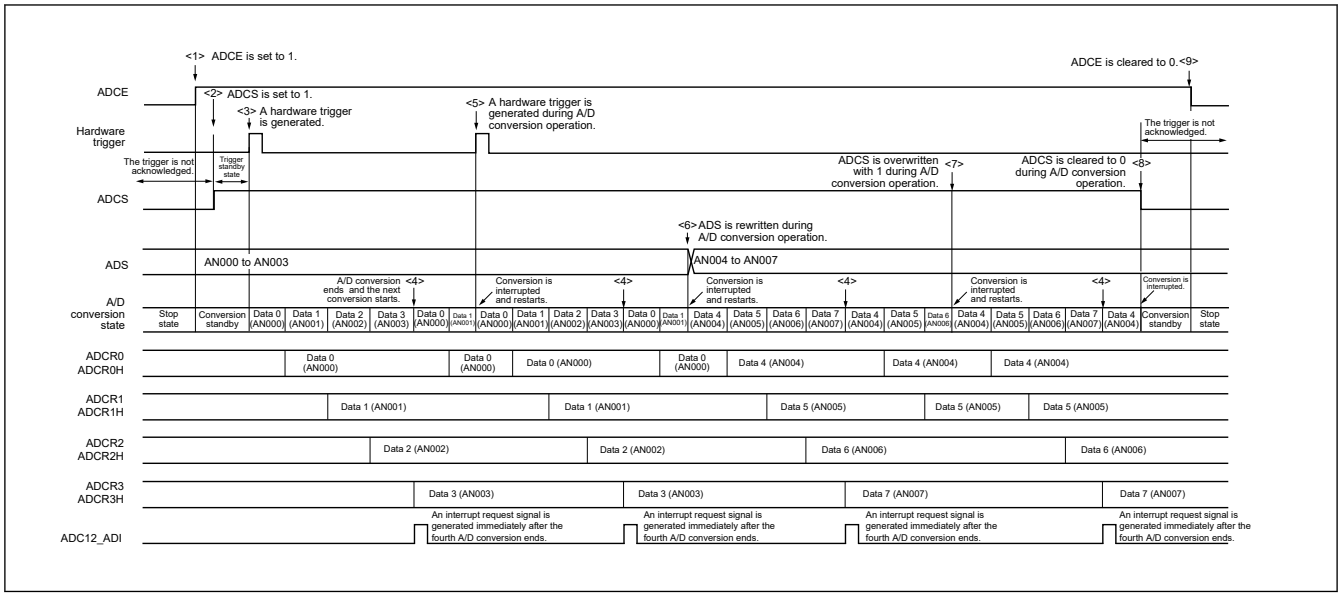


Figure 25.18 Example of hardware trigger no-wait mode (scan mode, sequential conversion mode) operation timing

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 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 section 25.2.1. ADM0 : A/D Converter Mode Register 0 and section 25.2.1. ADM0 : A/D Converter Mode Register 0.)

25.5.12 Hardware Trigger No-wait Mode (Scan Mode, One-shot Conversion Mode)

<1> In the stop state, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the A/D converter enters the 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 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 (ADC12_ADI) 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 A/D converter enters the 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 A/D converter enters the standby state. However, the 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 A/D converter enters the stop state.

When ADCS = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 25.19 shows the example of hardware trigger no-wait mode (scan mode, one-shot conversion mode) operation timing.

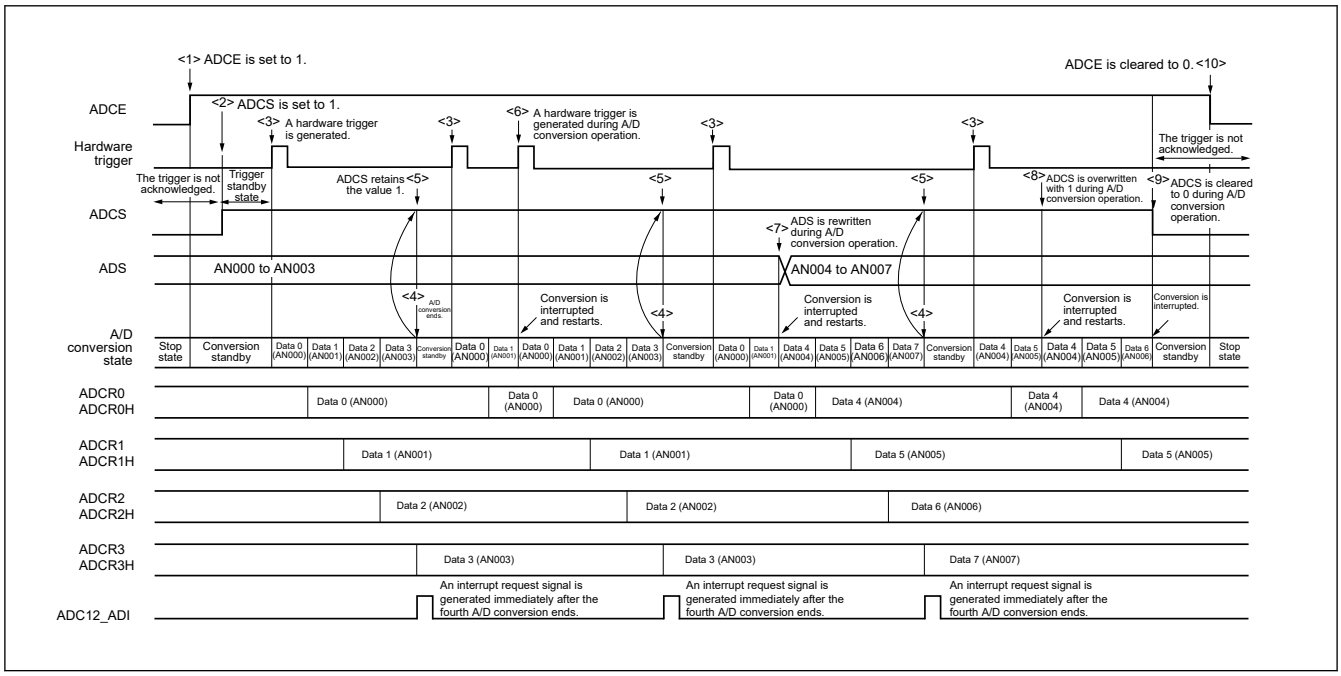


Figure 25.19 Example of hardware trigger no-wait mode (scan mode, one-shot conversion mode) operation timing

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 section 25.2.1. ADM0 : A/D Converter Mode Register 0 and section 25.2.1. ADM0 : A/D Converter Mode Register 0.)

25.5.13 Hardware Trigger Wait Mode (Select Mode, Sequential Conversion Mode)

<1> In the stop state, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the 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 (ADC12_ADI) 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 A/D converter enters the hardware trigger standby state, and the A/D converter enters the stop state.

<8> If ADCE = 0 is set during the hardware trigger wait state, the A/D converter is stopped. When ADCE = 0, the hardware trigger input is ignored and A/D conversion does not start.

Figure 25.20 shows the example of hardware trigger wait mode (select mode, sequential conversion mode) operation timing.

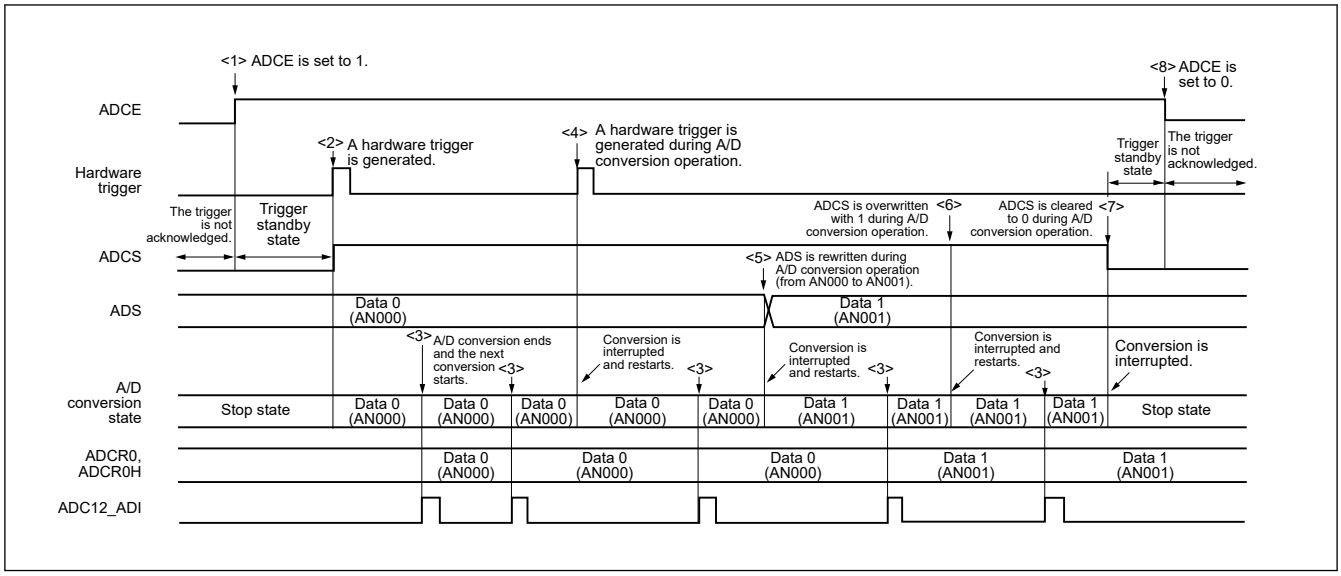


Figure 25.20 Example of hardware trigger no-wait mode (select mode, sequential conversion mode) operation timing

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 section 25.2.1. ADM0 : A/D Converter Mode Register 0 and section 25.2.1. ADM0 : A/D Converter Mode Register 0.)

25.5.14 Hardware Trigger Wait Mode (Select Mode, One-shot Conversion Mode)

<1> In the stop state, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the 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 (ADC12_ADI) is generated.

<4> After A/D conversion ends, the ADCS bit is automatically cleared to 0, and the A/D converter enters the 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 A/D converter enters the hardware trigger standby state, and the A/D converter enters the stop state.

<9> If ADCE = 0 is set during the hardware trigger wait state, the A/D converter is stopped. When ADCE = 0, the hardware trigger input is ignored and A/D conversion does not start.

Figure 25.21 shows the example of hardware trigger wait mode (select mode, one-shot conversion mode) operation timing.

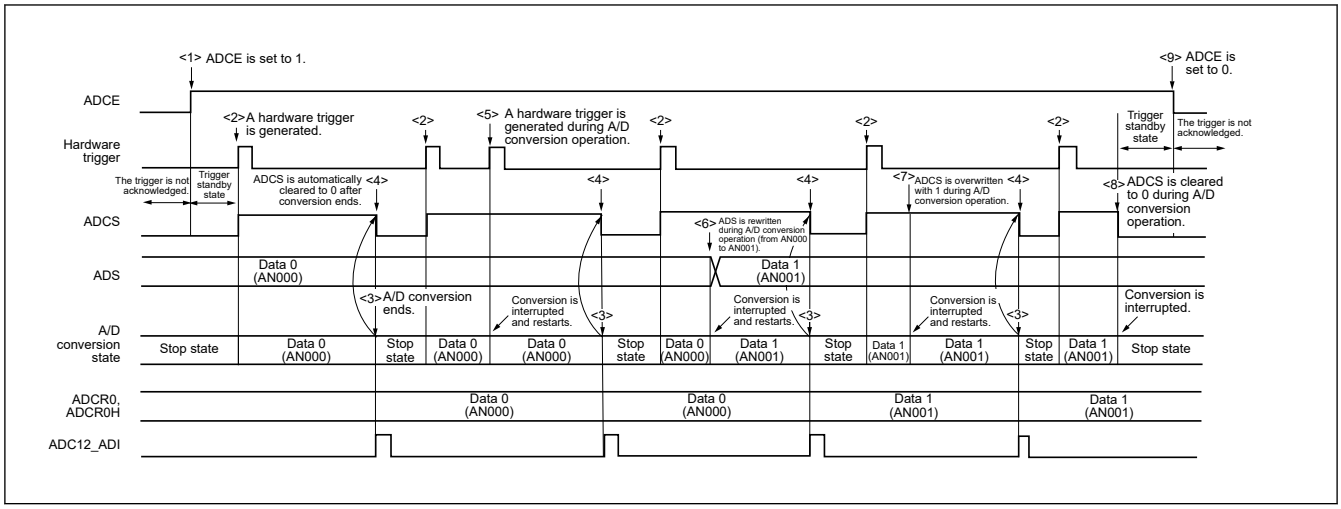


Figure 25.21 Example of hardware trigger wait mode (select mode, one-shot conversion mode) operation timing

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 section 25.2.1. ADM0 : A/D Converter Mode Register 0 and section 25.2.1. ADM0 : A/D Converter Mode Register 0.)

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).

25.5.15 Hardware Trigger Wait Mode (Scan Mode, Sequential Conversion Mode)

<1> In the stop state, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the 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 (ADC12_ADI) 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 A/D converter enters the hardware trigger standby state, and the A/D converter enters the stop state.

<8> If ADCE = 0 is set during the hardware trigger wait state, the A/D converter is stopped. When ADCE = 0, the hardware trigger input is ignored and A/D conversion does not start.

Figure 25.22 shows the example of hardware trigger wait mode (scan mode, sequential conversion mode) operation timing.

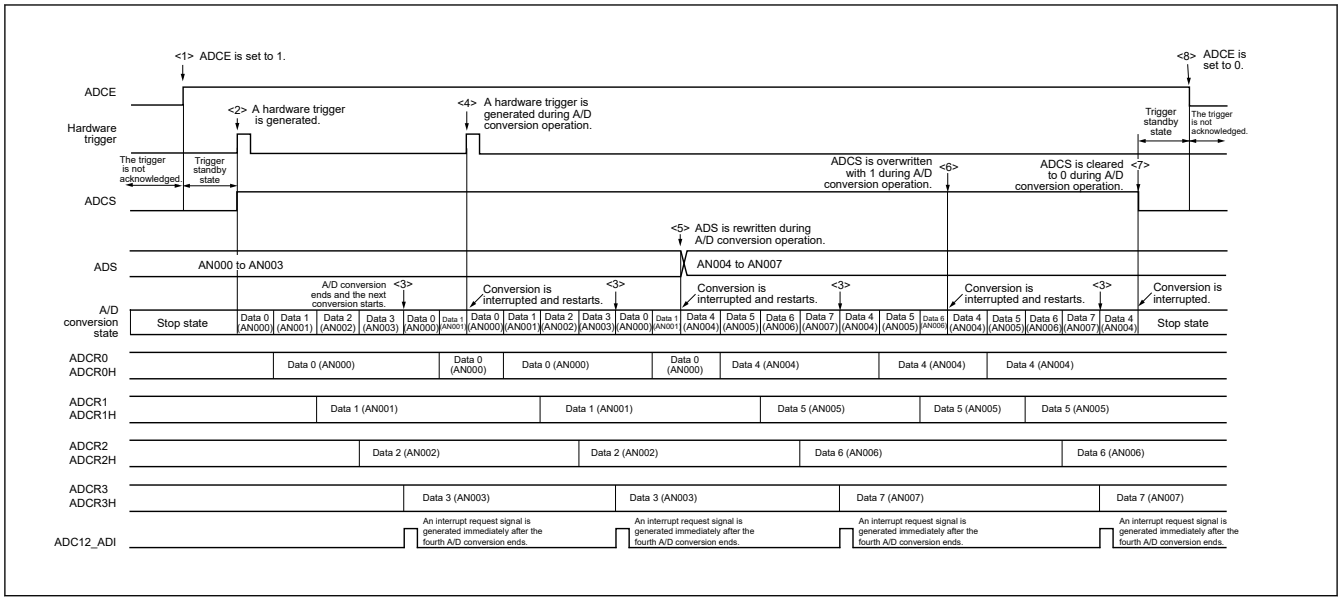


Figure 25.22 Example of hardware trigger wait mode (scan mode, sequential conversion mode) operation timing

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 section 25.2.1. ADM0 : A/D Converter Mode Register 0 and section 25.2.1. ADM0 : A/D Converter Mode Register 0.)

25.5.16 Hardware Trigger Wait Mode (Scan Mode, One-shot Conversion Mode)

<1> In the stop state, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the 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 (ADC12_ADI) 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 A/D converter enters the 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 A/D converter enters the hardware trigger standby state, and the A/D converter enters the stop state.

Figure 25.23 shows the example of hardware trigger wait mode (scan mode, one-shot conversion mode) operation timing.

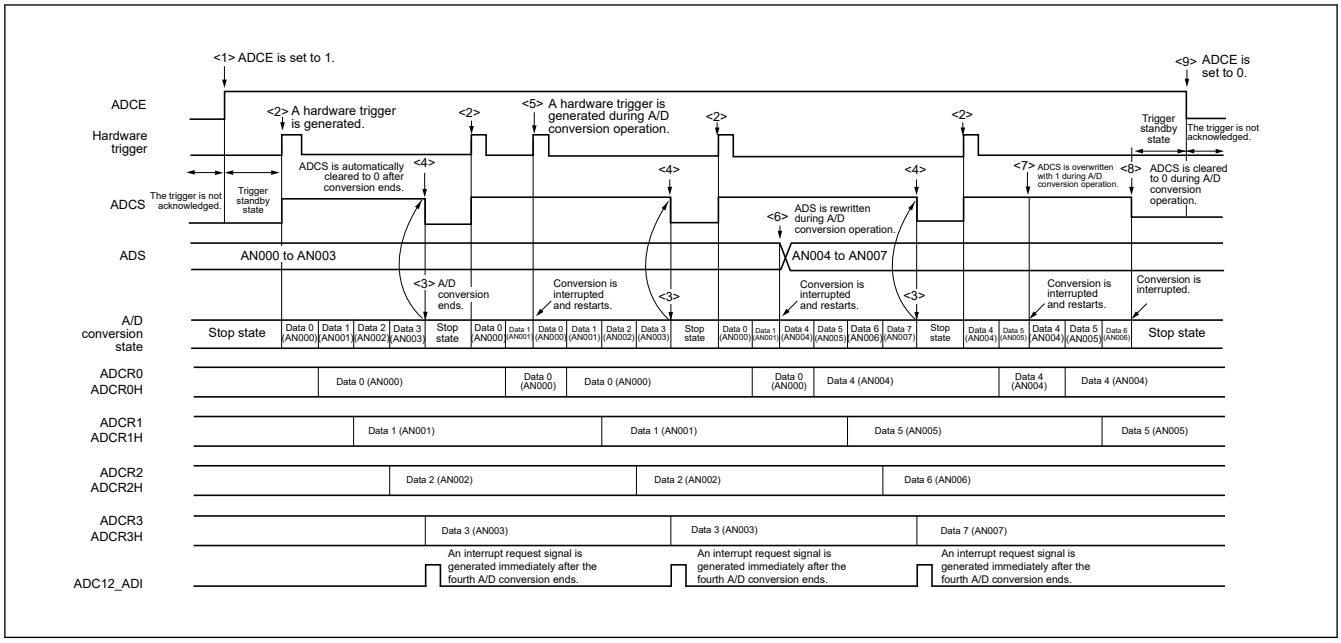


Figure 25.23 Example of hardware trigger wait mode (scan mode, one-shot conversion mode) operation timing

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 section 25.2.1. ADM0 : A/D Converter Mode Register 0 and section 25.2.1. ADM0 : A/D Converter Mode Register 0.)

25.6 A/D Converter Setup Procedure

The A/D converter setup procedure in each operation mode is described in the following section.

25.6.1 Setting up Software Trigger No-wait Mode

Table 25.14 shows the setup steps in software trigger no-wait mode.

Table 25.14 Setting up software trigger no-wait mode

Step		Process	Detail
Setting up Software Trigger No-Wait Mode	<1>	MSTPCRD register setting	The MSTPD16 bit of the MSTPCRD register is set to 0, and supplying the clock starts.
	<2>	PmnPFS_A register settings	The ports are set as the analog input. (See section 16.5.4. Notes on Using Analog Functions.)
	<3>	• 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 ADTMD1 and ADTMD0 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.
	<4>	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.
	<5>	Setting ADM2 register Changing the values of ADREFP [1:0]	• 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.
	<6>	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.
	<7>	ADCE bit setting	The ADCE bit of the ADM0 register is set to 1, and the A/D converter enters the standby state.
	<8>	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.
	<9>	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.
	<10>	Start of A/D conversion	—
	—	⋮	(The A/D conversion operations are performed)
	<11>	End of A/D conversion	The A/D conversion end interrupt (ADC12_ADI) is generated.*1
	<12>	Storage of conversion results in the ADCRn and ADCRnH register	The conversion results are stored in the ADCRn and 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.

25.6.2 Setting up Software Trigger Wait Mode

Table 25.15 shows the setup steps in software trigger wait mode.

Table 25.15 Setting up software trigger wait mode

Step		Process	Detail
Setting up Software Trigger Wait Mode	<1>	MSTPCRD register setting	The MSTPD16 bit of the MSTPCRD register is set to 0, and supplying the clock starts.
	<2>	PmnPFS_A register settings	The ports are set as the analog input. (See section 16.5.4. Notes on Using Analog Functions.)
	<3>	• 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 ADTMD1 and ADTMD0 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.
	<4>	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.
	<5>	Setting ADM2 register Changing the values of ADREFP [1:0]	• 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.
	<6>	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.
	<7>	ADCE bit setting	Do not set the ADCE bit of the ADM0 register (0). The A/D converter must remain in the stopped state.
	<8>	ADCS bit setting	The ADCS bit of the ADM0 register is set to 1, and the A/D converter enters the standby state.
	<9>	Stabilization wait time for A/D power supply	The A/D converter automatically counts up to the stabilization wait time for A/D power supply.
	<10>	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)
	<11>	End of A/D conversion	The A/D conversion end interrupt (ADC12_ADI) is generated.*1
	<12>	Storage of conversion results in the ADCRn and ADCRnH register	The conversion results are stored in the ADCRn and 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.

25.6.3 Setting up Hardware Trigger No-wait Mode

Table 25.16 shows the setup steps in hardware trigger no-wait mode.

Table 25.16 Setting up hardware trigger no-wait mode

Step		Process	Detail
Setting up Hardware Trigger No-Wait Mode	<1>	MSTPCRD register setting	The MSTPD16 bit of the MSTPCRD register is set to 0, and supplying the clock starts.
	<2>	PmnPFS_A register settings	The ports are set as the analog input. (See section 16.5.4. Notes on Using Analog Functions.)
	<3>	• 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 ADTMD1 and ADTMD0 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.
	<4>	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.
	<5>	Setting ADM2 register Changing the values of ADREFP [1:0]	• 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.
	<6>	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.
	<7>	ADCE bit setting	The ADCE bit of the ADM0 register is set to 1, and the A/D converter enters the standby state.
	<8>	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.
	<9>	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.
	<10>	Start of A/D conversion	—
	—	⋮	(The A/D conversion operations are performed)
	<11>	End of A/D conversion	The A/D conversion end interrupt (ADC12_ADI) is generated.*1
	<12>	Storage of conversion results in the ADCRn and ADCRnH register	The conversion results are stored in the ADCRn and 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.

25.6.4 Setting up Hardware Trigger Wait Mode

Table 25.17 shows the setup steps in hardware trigger wait mode.

Table 25.17 Setting up hardware trigger wait mode

Step		Process	Detail
Setting up Hardware Trigger Wait Mode	<1>	MSTPCRD register setting	The MSTPD16 bit of the MSTPCRD register is set to 0, and supplying the clock starts.
	<2>	PmnPFS_A register settings	The ports are set as the analog input. (See section 16.5.4. Notes on Using Analog Functions.)
	<3>	• 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 ADTMD1 and ADTMD0 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.
	<4>	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.
	<5>	Setting ADM2 register Changing the values of ADREFP [1:0]	• 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.
	<6>	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.
	<7>	ADCE bit setting	The ADCE bit of the ADM0 register is set to 1, and the A/D converter enters the standby state.
	<8>	Hardware trigger generation	Set the trigger signal output of other modules.
	<9>	Stabilization wait time for A/D power supply	The A/D converter automatically counts up to the stabilization wait time for A/D power supply.
	<10>	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)
	<11>	End of A/D conversion	The A/D conversion end interrupt (ADC12_ADI) is generated.*1
	<12>	Storage of conversion results in the ADCRn and ADCRnH register	The conversion results are stored in the ADCRn and 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.

25.6.5 Example of Using the ADC12 when Selecting the Temperature Sensor Output Voltage or Internal Reference Voltage, and Software Trigger No-wait Mode and One-shot Conversion Mode

Table 25.18 shows the setup steps When Temperature Sensor Output Voltage and Internal Reference Voltage Is Selected.

Table 25.18 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>	MSTPCRD register setting	The MSTPD16 bit of the MSTPCRD register is set to 0, and supplying the clock starts.
	<2>	• 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 ADTMD1 and ADTMD0 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.
	<3>	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.
	<4>	ADCE bit setting	The ADCE bit of the ADM0 register is set to 1, and the A/D converter enters the standby state.
	<5>	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.
	<6>	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.
	<7>	Start of A/D conversion	—
	<8>	End of A/D conversion	The A/D conversion end interrupt (ADC12_ADI) will be generated. After ADISS is set to 1, the initial conversion result cannot be used.
	<9>	ADCS bit setting	The ADCS bit of the ADM0 register is set to 1, and A/D conversion starts.
	<10>	Start of A/D conversion	—
	<11>	End of A/D conversion	The A/D conversion end interrupt (ADC12_ADI) is generated.*1
	<12>	Storage of conversion results in the ADCRn and ADCRnH register	The conversion results are stored in the ADCRn and 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.

25.6.6 Setting Up Test Mode

Table 25.19 shows the setup steps in test mode.

Table 25.19 Setting up test mode

Step		Process	Detail
Setting up Test Mode	<1>	MSTPCRD register setting	The MSTPD16 bit of the MSTPCRD register is set to 0, and supplying the clock starts.
	<2>	• 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 ADTMD1 and ADTMD0 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 AN000. • ADTES register ADTES[1:0] bits: VREFL0 or VREFH0.
	<3>	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.
	<4>	Setting ADM2 register Changing the values of ADREFP [1:0]	• 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.
	<5>	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.
	<6>	ADCE bit setting	The ADCE bit of the ADM0 register is set to 1, and the A/D converter enters the standby state.
	<7>	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.
	<8>	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.
	<9>	Start of A/D conversion	—
	<10>	End of A/D conversion	The A/D conversion end interrupt (ADC12_ADI) is generated.*1
	<11>	Storage of conversion results in the ADCRn and ADCRnH register	The conversion results are stored in the ADCRn and 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.

Note: For the procedure for testing the A/D converter, see section 25.8. Testing of the A/D Converter.

25.7 Snooze Mode Function

In Snooze mode, A/D conversion is triggered by inputting a hardware trigger in the Software Standby mode. Normally, A/D conversion is stopped while in the Software Standby mode, but, by using the Snooze mode function, A/D conversion can be performed without operating the CPU. This is effective for reducing the operating current.

25.7.1 A/D Conversion by Inputting a Hardware Trigger

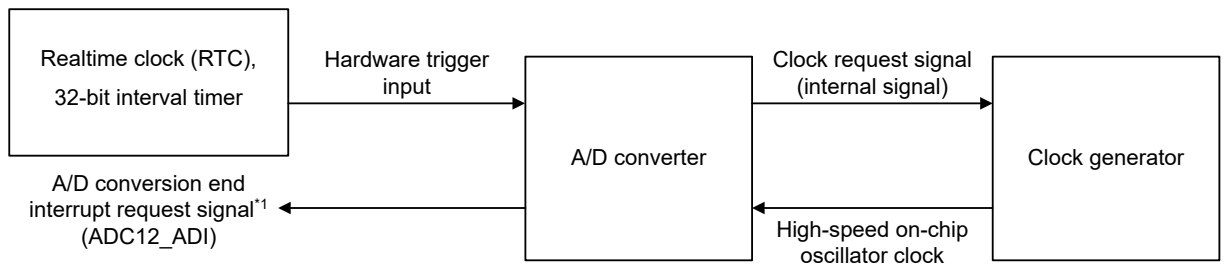
In Snooze mode, A/D conversion is triggered by inputting a hardware trigger.

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.

Note: The Snooze mode can only be specified when the high-speed on-chip oscillator clock or medium-speed on-chip oscillator clock is selected for PCLKB.



Note 1. Depending on the setting of the A/D conversion result comparison function (ADRCK bit, ADUL/ADLL register), there is a possibility of no interrupt signal being generated.

Figure 25.24 Block diagram when using Snooze mode in hardware trigger wait mode

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 25.20). 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 Software Standby mode, the high-speed on-chip oscillator clock is supplied to the A/D converter. After supplying this clock, the 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.*1

Note: Select the hardware trigger signal from among the realtime clock interrupt signal (RTC_ALM_OR_PRD), 32-bit interval timer event signal (ADITL0 (= TML32_ITL0)).

Note: A/D converter can not be triggered by the ELC in the Snooze mode.

(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 (ADC12_ADI) is generated.

- While in the select mode
  When A/D conversion ends and an A/D conversion end interrupt request signal (ADC12_ADI) is generated, if SBYEDCR0.ADC12ED = 1, the A/D converter returns to normal operation mode from Snooze mode. At this time, be

sure to clear bit 2 (AWC = 0: Snooze mode release) of the A/D converter mode register 2 (ADM2). If the AWC bit is left set to 1, A/D conversion will not start normally in the subsequent Snooze or normal operation mode.

- While in the 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 (ADC12_ADI) is generated, the A/D converter switches from the Snooze mode to the normal operation mode. At this time, be sure to clear bit 2 (AWC = 0: Snooze mode release) of the A/D converter mode register 2 (ADM2). If the AWC bit is left set to 1, A/D conversion will not start normally in the subsequent Snooze or normal operation mode.

Figure 25.25 shows an operation example when interrupt is generated after A/D conversion ends (while in scan mode).

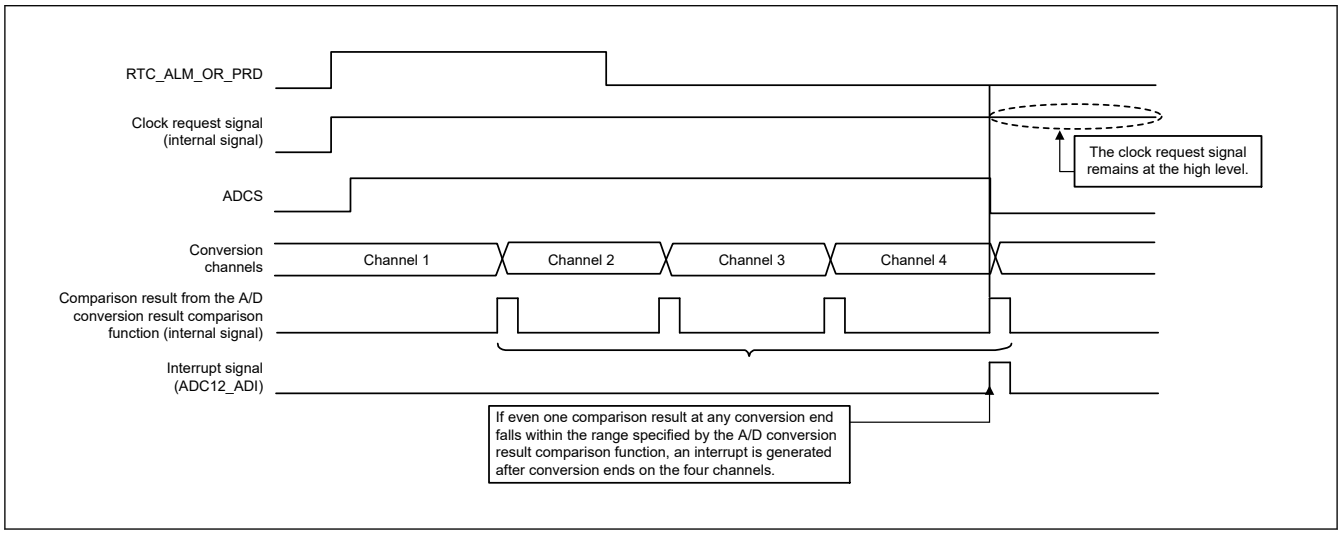


Figure 25.25 Operation example when interrupt is generated after A/D conversion ends (while in scan mode)

(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 (ADC12_ADI) is not generated.

- While in the select mode
  If the A/D conversion end interrupt request signal (ADC12_ADI) is not generated after A/D conversion ends, the clock request signal (an internal signal) is automatically set to the low level, and the clock is stopped. If a hardware trigger is input later, A/D conversion is performed again in the Snooze mode.
- While in the scan mode
  If the A/D conversion result values of the four channels do not fall within the range specified by the A/D conversion result comparison function even once, and the A/D conversion end interrupt request signal (ADC12_ADI) is not generated, the clock request signal (an internal signal) is automatically set to the low level after A/D conversion of the four channels ends, and the clock is stopped. If a hardware trigger is input later, A/D conversion is performed again in the Snooze mode.

Figure 25.26 shows an operation example when no interrupt is generated after A/D conversion ends (while in scan mode).

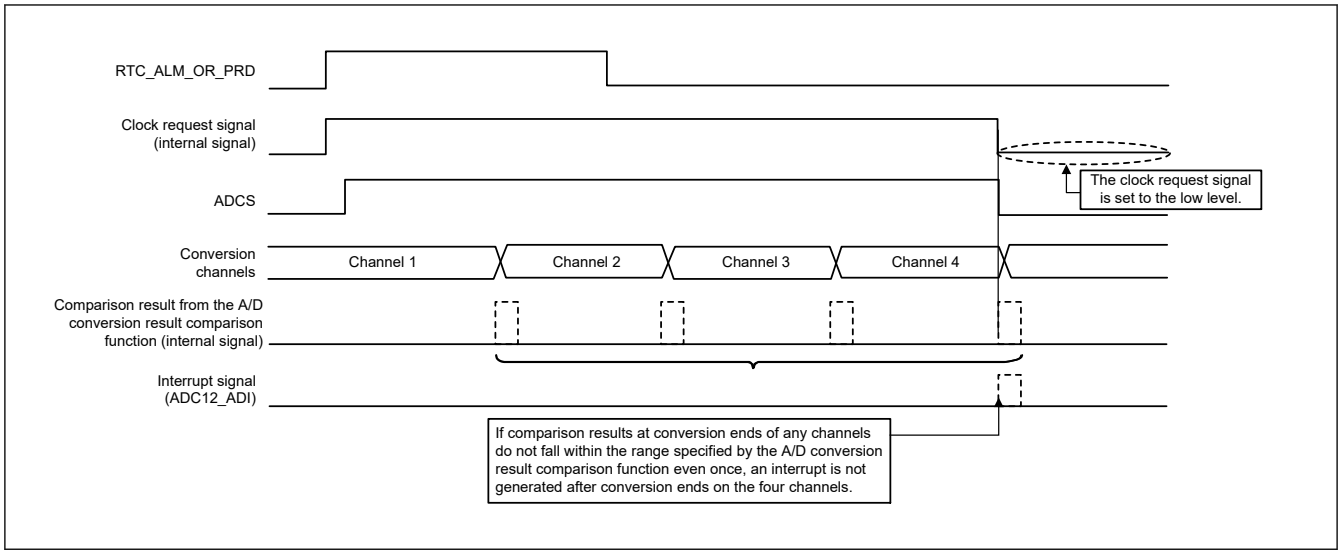


Figure 25.26 Operation example when no interrupt is generated after A/D conversion ends (while in scan mode)

Table 25.20 shows procedure for setting up Snooze mode (hardware trigger)

Table 25.20 Procedure for setting up Snooze mode (hardware trigger) (1 of 2)

Step		Process	Detail
Normal operation	<1>	MSTPCRD register setting	The MSTPD16 bit of the MSTPCRD register is set to 0, and supplying the clock starts.
	<2>	PmnPFS_A register settings	The ports are set as the analog input. (See section 16.5.4. Notes on Using Analog Functions.)
	<3>	• 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 ADTMD1 and ADTMD0 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.
	<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 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.
	<5>	AWC = 1	Immediately before entering the Software Standby mode, enable the Snooze mode by setting the AWC bit of the ADM2 register to 1.
	<6>	Normal operation	The ADCE bit of the ADM0 register is set to 1, and the A/D converter enters the standby state.

Table 25.20 Procedure for setting up Snooze mode (hardware trigger) (2 of 2)

Step		Process	Detail
Software Standby mode	<7>	Enter the Software Standby mode	—
Snooze mode	<8>	Hardware trigger generation	After hardware trigger is generated, the 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)
	<9>	End of A/D conversion	The A/D conversion end interrupt (ADC12_ADI) is generated.*1
	<10>	ADC12_ADI generation	• ADC12_ADI is generated : Go to step <11> • ADC12_ADI is not generated : The clock request signal (an internal signal) is automatically set to the low level in the Snooze mode. And go to step <7>.
Normal operation	<11>	Storage of conversion results in the ADCRn and ADCRnH register	The conversion results are stored in the ADCRn and ADCRnH register.
	<12>	AWC = 0	Release the Snooze mode by clearing the AWC bit of the ADM2 register to 0.*2
	<13>	Normal operation ⋮	—

Note 1. If the A/D conversion end interrupt request signal (ADC12_ADI) is not generated depending on the settings of the ADRCK bit, ADUL and ADLL registers, the result is not stored in the ADCRn and ADCRnH register. The A/D converter enters the Software Standby mode again. 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.

25.8 Testing of the A/D Converter

The IEC60730 standard mandates testing of the A/D converter. This test checks whether the A/D converter is operating normally by converting the A/D converter's positive and negative reference voltages, analog input channels (ANxxx), temperature sensor output voltage, and internal reference voltage.

Use the following procedure to check the analog multiplexer.

<1> Select the ANxxx pin for A/D conversion using the ADTES register (ADTES1 = 0, ADTES0 = 0).

<2> Perform A/D conversion for the ANxxx pin (conversion result 1-1).

<3> Select the A/D converter's negative reference voltage for A/D conversion using the ADTES register (ADTES1 = 1, ADTES0 = 0)

<4> Perform A/D conversion of the negative reference voltage of the A/D converter (conversion result 2-1).

<5> Select the ANxxx pin for A/D conversion using the ADTES register (ADTES1 = 0, ADTES0 = 0).

<6> Perform A/D conversion for the ANxxx pin (conversion result 1-2).

<7> Select the A/D converter's positive reference voltage for A/D conversion using the ADTES register (ADTES1 = 1, ADTES0 = 1)

<8> Perform A/D conversion of the positive reference voltage of the A/D converter (conversion result 2-2).

<9> Select the ANxxx pin for A/D conversion using the ADTES register (ADTES1 = 0, ADTES0 = 0).

<10>Perform A/D conversion for the ANxxx pin (conversion result 1-3).

<11>Check that the conversion results 1-1, 1-2, and 1-3 are equal.

<12>Check that the A/D conversion result 2-1 is all zero and the A/D conversion result 2-2 is all one.

Using the procedure above can confirm that the analog multiplexer is selected and all wiring is connected.

Note: If the analog input voltage is variable during A/D conversion in steps <1> to <10> above, use another method to check the analog multiplexer.

Note: The results of conversion might include an error. Consider an appropriate level of error in comparison of the results of conversion.

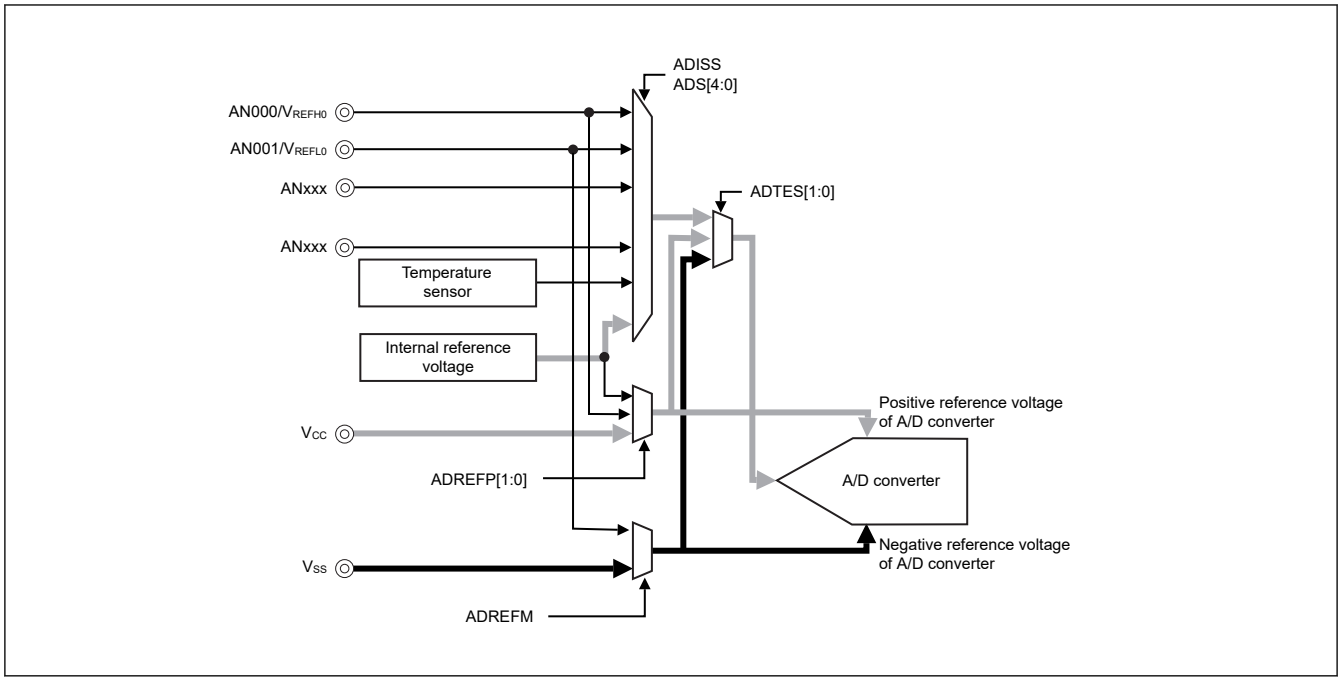


Figure 25.27 Configuration of testing of the A/D converter

25.9 How to Read A/D Converter Characteristics Table

Here, special terms unique to the 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 overall error.

(2) Overall error

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 overall error.

Note that the quantization error is not included in the overall error in the characteristics table.

(3) Quantization error

When analog values are converted to digital values, a ±1/2LSB error naturally occurs. In an A/D 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 overall error, zero-scale error, full-scale error, integral linearity error, and differential linearity error in the characteristics table.

Figure 25.28 shows the overall error, and Figure 25.29 shows the quantization error.

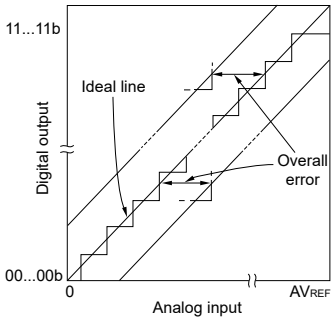


Figure 25.28 Overall error

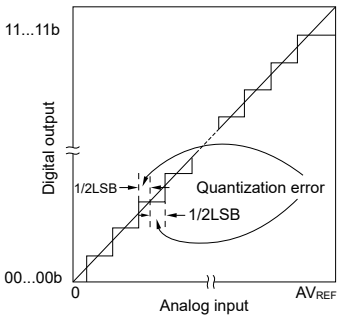


Figure 25.29 Quantization error

(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.

For differential inputs, the zero-scale error is shown in Figure 25.30, the full-scale error is shown in Figure 25.31, the integral linearity error is shown in Figure 25.32, and the differential linearity error is shown in Figure 25.33.

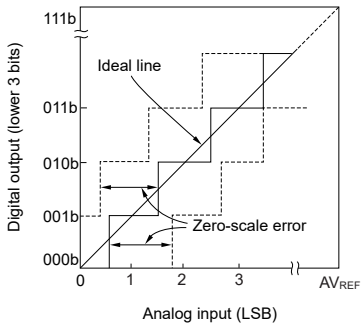


Figure 25.30 Zero-scale error

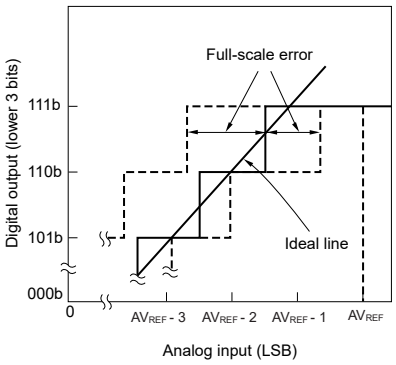


Figure 25.31 Full-scale error

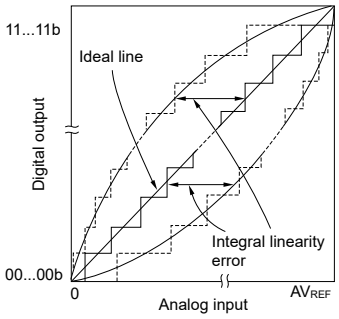


Figure 25.32 Integral linearity error

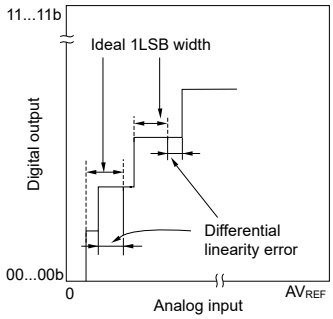


Figure 25.33 Differential linearity error

(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 25.34 shows the sampling time at the A/D conversion time.

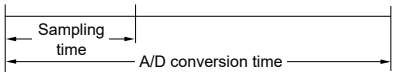


Figure 25.34 Sampling time at the A/D conversion time

25.10 Usage Notes

(1) Operating current in Software Standby mode

Shift to Software Standby mode after stopping the 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 standby state, clear the ADC12_ADI bit in the corresponding NVIC_ICPR0 register and start operation.

(2) Input range of AN000 to AN012, AN021 and AN022 pins

Observe the rated range of the AN000 to AN012, AN021 and AN022 pins input voltage. If a voltage exceeding VCC and VREFH0 or a voltage lower than VSS and VREFL0 (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 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 31, Electrical Characteristics.

(3) Conflicting operations

<1> Conflict between the conversion result being stored in the A/D conversion result registers (ADCRn and ADCRnH) at the end of conversion and the read access to the ADCRn and ADCRnH registers by instruction.

The ADCRn and ADCRnH registers read has priority. After the read operation, the new conversion result is written to the ADCRn and ADCRnH registers.

<2> Conflict between the conversion result being stored in the A/D conversion result registers (ADCRn and 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 has priority. The ADCRn and ADCRnH registers write is not performed, nor is the conversion end interrupt signal (ADC12_ADI) generated.

(4) Noise countermeasures

To maintain the 12-bit or 10-bit resolution, attention must be paid to noise input to the VREFH0, VCC, AN000 to AN012, AN021 and AN022 pins.

<1> Connect a capacitor with a low equivalent resistance and a good frequency response (capacitance of about 0.1 µF) through the shortest possible run of relatively thick wiring to the V_{CC} and VREFH0 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 25.35 is recommended.

<3> Do not switch these pins with other pins during conversion.

<4> The accuracy is improved if the Sleep mode is set immediately after the start of conversion.

Figure 25.35 shows connections of V_{CC}, VREFH0, and analog input pins.

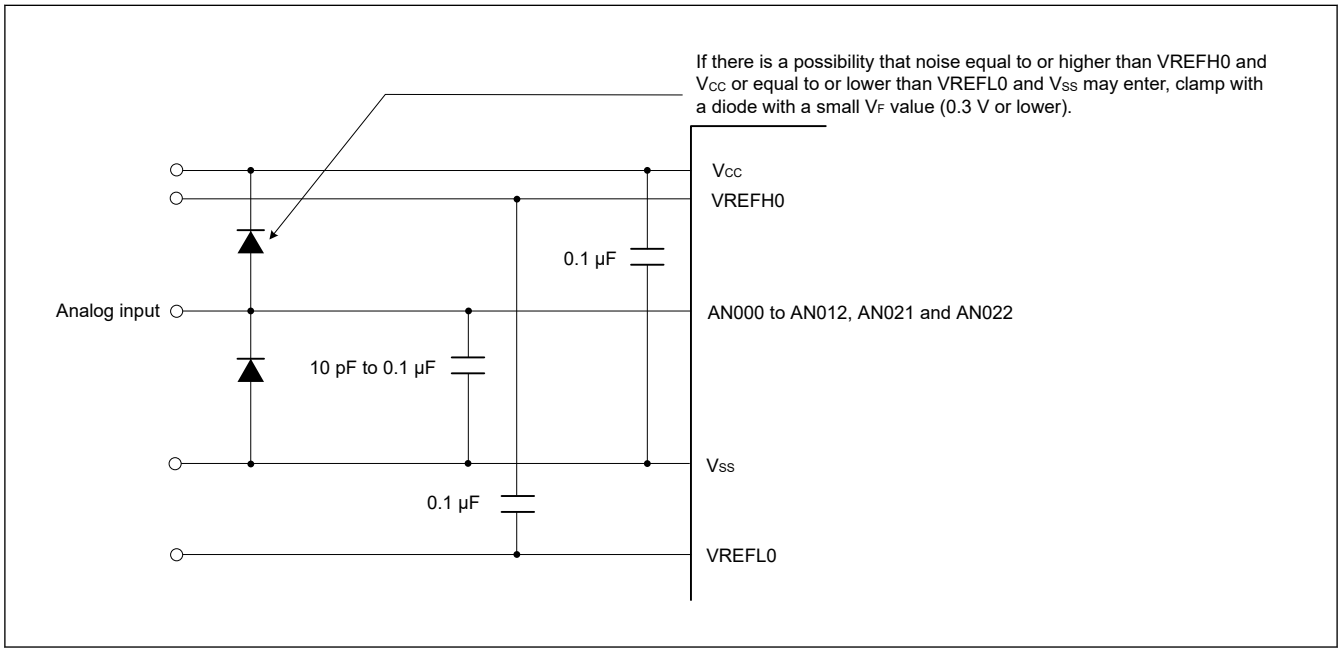


Figure 25.35 Connections of V_{CC}, VREFH0, and analog input pins

(5) Analog input (ANxxx) pins

<1> The analog input pins (AN000 to AN012, AN021 and AN022) are also used as input port pins (P000 to P004, P008 to P015, P100, and P101). When A/D conversion is performed with any of the AN000 to AN0012, AN021, and AN022 pins selected, do not change to output value P000 to P004, P008 to P015, P100, and P101 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 (ANxxx) 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 AN000 to AN012, AN021, and AN022 to which the source is connected (see Figure 25.35). 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 Clear-pending Register (NVIC_ICPR0)

The Interrupt Clear-pending Register (NVIC_ICPR0) 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 NVIC_ICPR0 for the pre-change analog input may have been set before the ADS register is rewritten. When reading the NVIC_ICPR0 immediately after rewriting to the ADS register, note that the NVIC_ICPR0 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 NVIC_ICPR0 before the A/D conversion operation is resumed.

Figure 25.36 shows timing of A/D conversion end interrupt request generation.

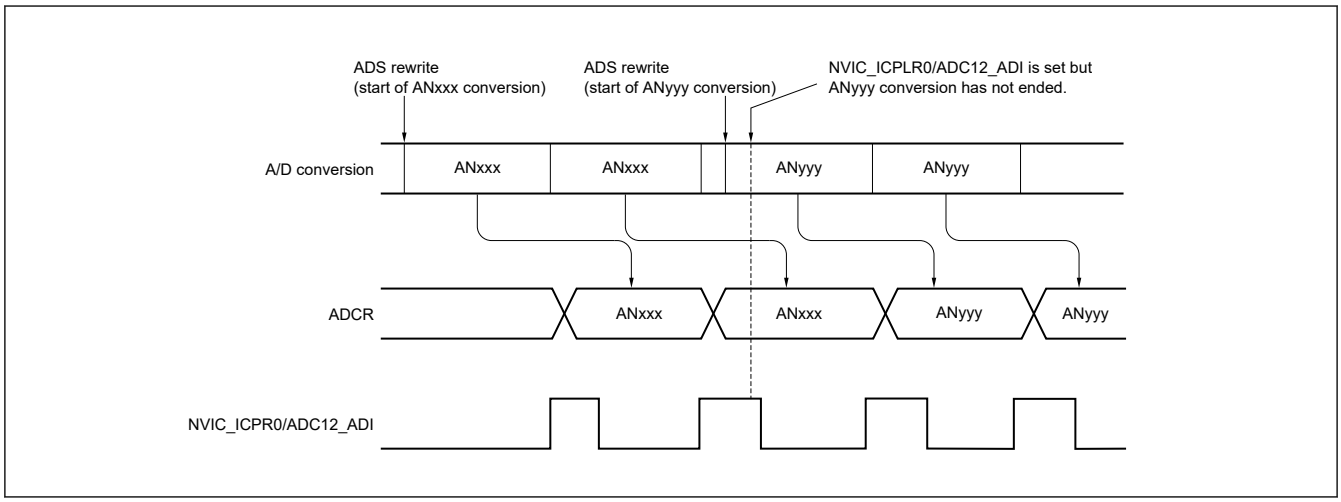


Figure 25.36 Timing of A/D conversion end interrupt request generation

(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 (ADC12_ADI) 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), I/O port registers (PODRm, PDRm, PORRm, POSRm, EORRm, EOSRm, and PmnPFS_A), 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, I/O port registers (PODRm, PDRm, PORRm, POSRm, EORRm, EOSRm, and PmnPFS_A), otherwise, an incorrect conversion result may be read.

(10) Starting the A/D converter

Start the A/D converter after the VREFH0 and V_{CC} voltages stabilize.

26. Temperature Sensor (TSN)

26.1 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 26.1 lists the TSN specifications, and Figure 26.1 shows a block diagram.

Table 26.1 TSN specifications

Item	Description
Temperature sensor voltage output	Temperature sensor outputs a voltage to the 12-bit A/D converter

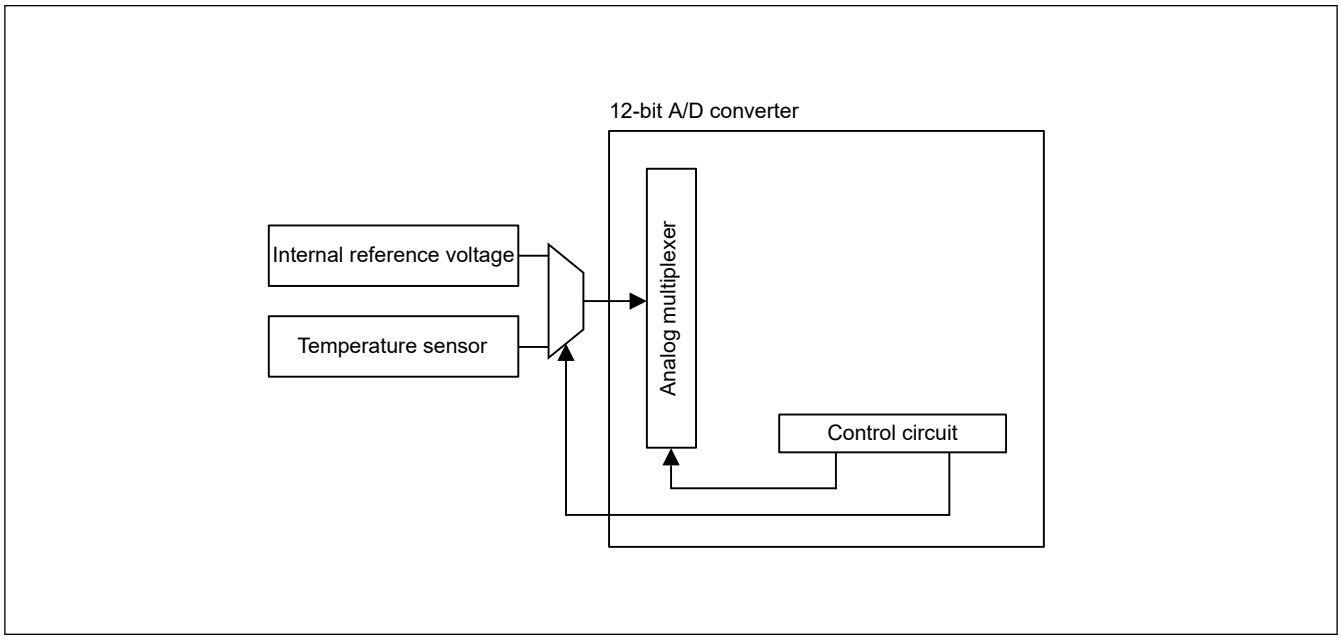


Figure 26.1 TSN block diagram

26.2 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.

26.2.1 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 31, 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.

26.2.2 Procedures for Using the Temperature Sensor

For details, see section 25, 12-bit A/D Converter (ADC12).

27. SRAM

27.1 Overview

The MCU provides an on-chip, high-density SRAM module with parity-bit checking. Parity check is performed on the all SRAM areas.

Table 27.1 lists the SRAM specifications.

Table 27.1 SRAM specifications

Parameter	Description
SRAM capacity	SRAM0: 16 KB
SRAM address	SRAM0: 0x2000_4000 to 0x2000_7FFF
Access*1	0 wait for both reading and writing
Parity	Even parity with 8-bit data and 1-bit parity
Error checking	Even parity error check

Note: SRAM0 and Trace RAM are shared. For the Trace RAM specifications, see *ARM® CoreSight™ MTB-M23 Technical Reference Manual (ARM DDI 0564C)*.

Note 1. For details, see section 27.3.3. Access Cycle.

27.2 Register Descriptions

27.2.1 PARIOAD : SRAM Parity Error Operation After Detection Register

Base address: SRAM = 0x4000_2000

Offset address: 0x0000

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.

27.2.2 SRAMPRCR : SRAM Protection Register

Base address: SRAM = 0x4000_2000

Offset address: 0x0004

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.

27.3 Operation

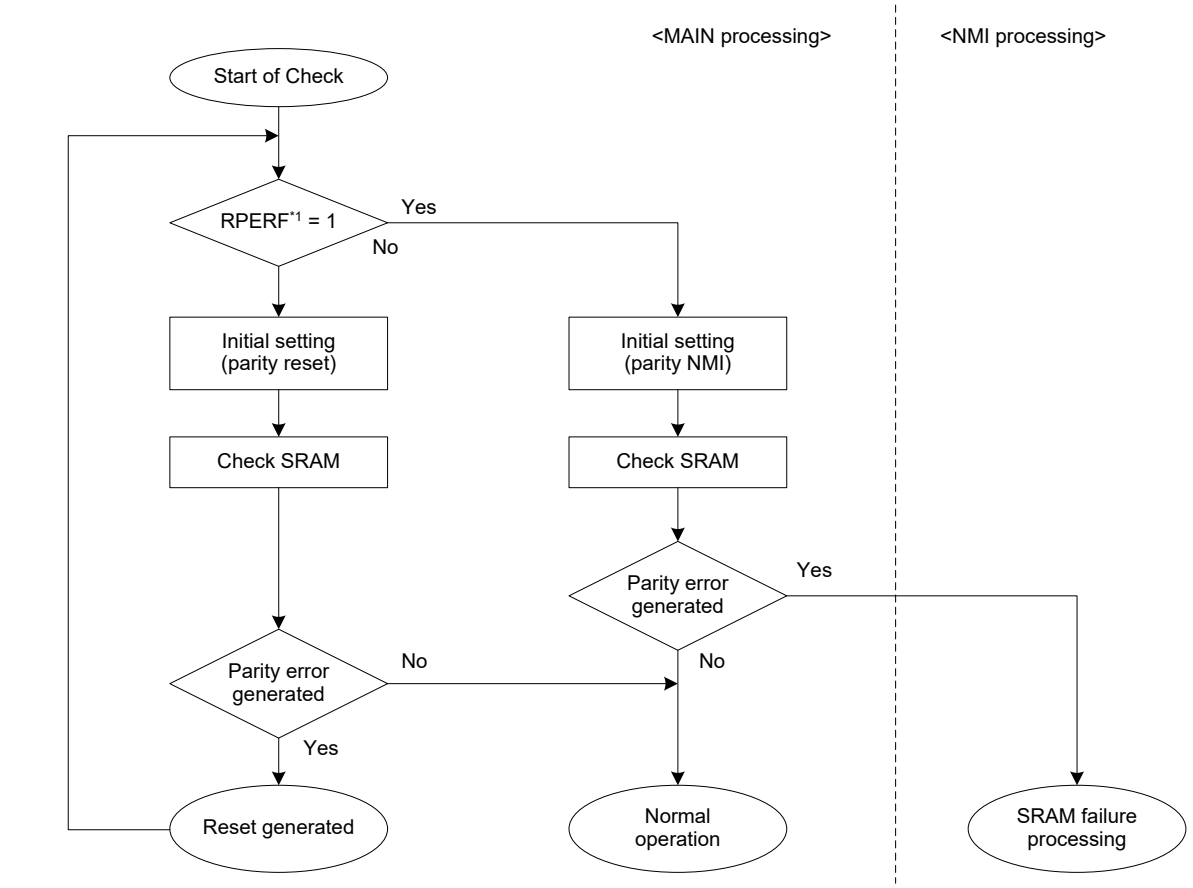
27.3.1 Parity Calculation Function

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.

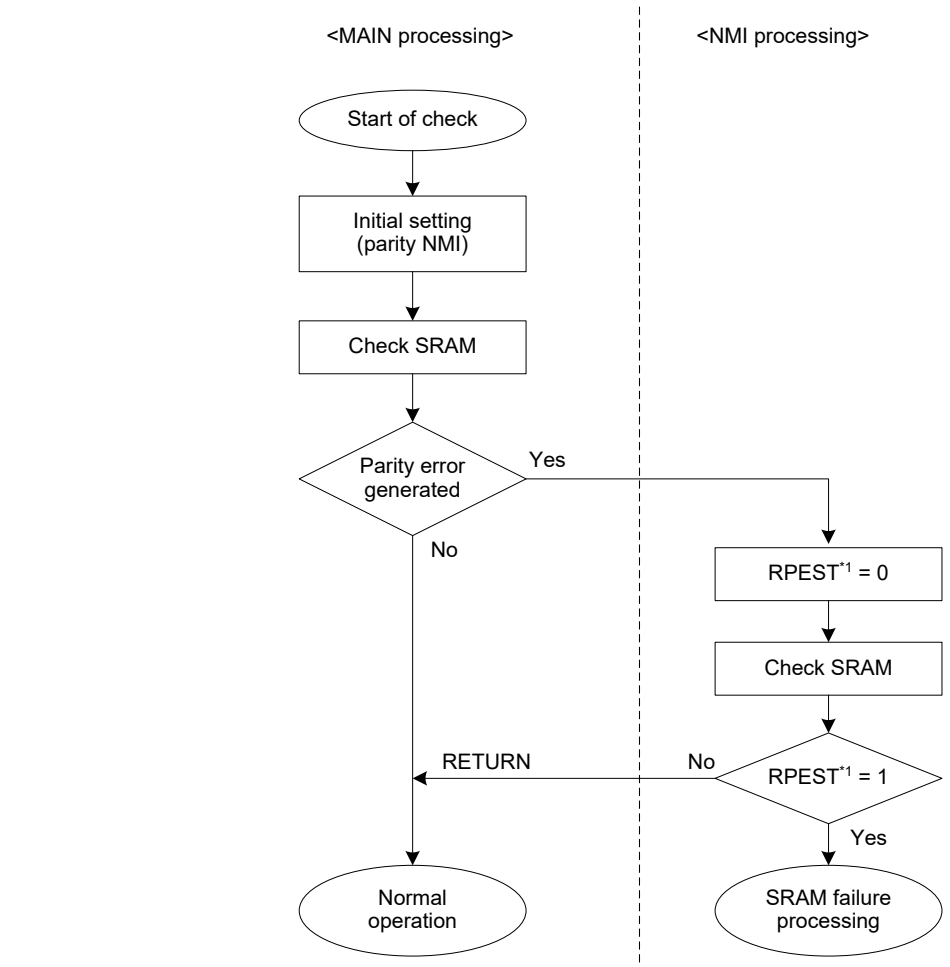
Parity errors can be occasionally caused by noise. To confirm whether the cause of the parity error is noise or corruption, follow the parity check flows shown in Figure 27.1 and Figure 27.2.

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: Internal Reset Request by RAM Parity Error Flag (RESF.RPERF bit)

Figure 27.1 Flow of SRAM parity check when SRAM parity reset is enabled



Note 1. RPEST: SRAM Parity Error Interrupt Status Flag (NMISR.RPEST bit)

Figure 27.2 Flow of SRAM parity check when SRAM parity interrupt is enabled

27.3.2 SRAM Error Sources

An SRAM error is a parity error. 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 parity errors.

Table 27.2 SRAM error sources

SRAM error source	DTC activation
Parity error (SRAM0 area)	Not possible

27.3.3 Access Cycle

Table 27.3 SRAM0 (parity area 0x2000_4000 to 0x2000_7FFF)

Read (cycles)		Write (cycles)	
Word access	Halfword/Byte access	Word access	Halfword/Byte access
2		2	

27.3.4 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 8 KB in the head area of SRAM0 (0x2000_4000 to 0x2000_5FFF) of SRAM0 (Parity area). For details on Software Standby mode, see section 9, Low Power Modes.

27.4 Usage Notes

27.4.1 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, 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.

27.4.2 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 NOP instruction, then read the SRAM (address = A)
- After writing to the SRAM (address = A), read data from area other than SRAM (address = A), then read the SRAM (address = A).

28. Flash Memory

28.1 Overview

The MCU provides up to 128-KB code flash memory and 2-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 28.1 lists the specifications of the code flash memory and data flash memory, and Figure 28.1 shows a block diagram of the related modules. Figure 28.2 shows the configuration of the code flash memory, and Figure 28.3 shows the configuration of the data flash memory.

Table 28.1 Code flash memory and data flash memory specifications

Parameter	Code flash memory	Data flash memory
Memory capacity	• 128-KB/64-KB of user area • Configuration setting area (See section 6, Option-Setting Memory)	2-KB of data area
Read cycle	• A read operation takes 2 cycles	• A read operation takes 6 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 of flash memory by user program (self-programming)*1.	
Security function	Protection against illicit tampering with or reading of data in flash memory	
Protection	Protection against erroneous overwriting of flash memory	
Background operation (BGO)	Code flash memory can be read during data flash memory programming	
Units of programming and erasure	• 32-bit units for programming in user area • 2-KB units for erasure in user area.	• 8-bit units for programming in data area • 256B units for erasure in data area.
Other functions	Interrupts accepted during self-programming	
	Option-setting memory can be set in the initial MCU settings	
On-board programming	Programming in on-chip debug mode: • SWD 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 28.9. Self-programming.

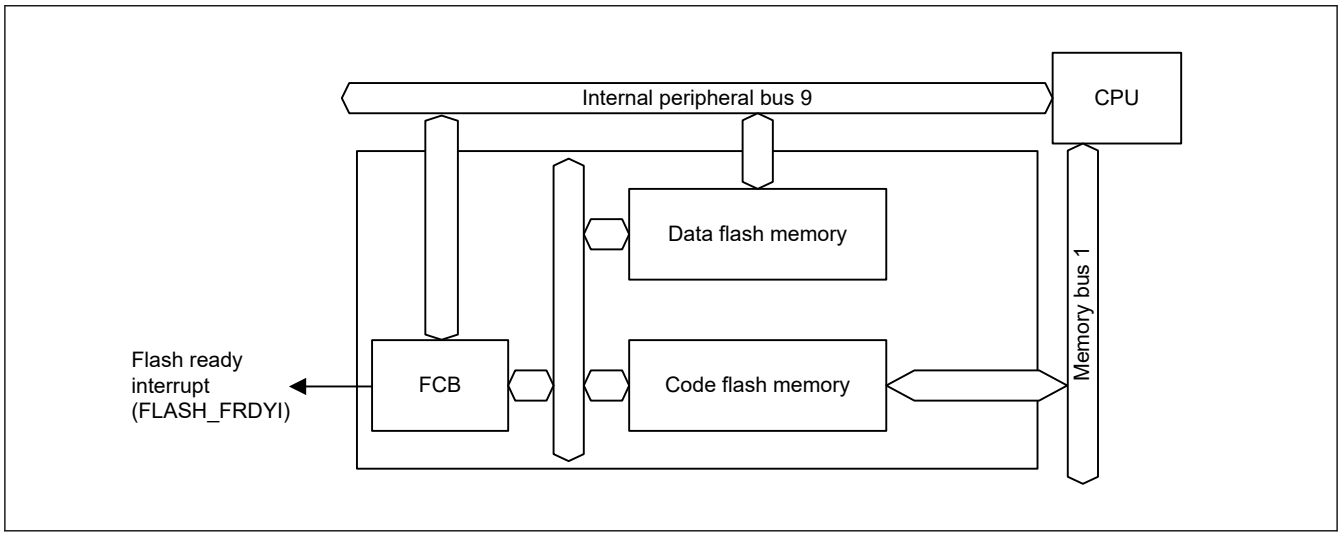


Figure 28.1 Flash memory-related modules block diagram

28.2 Memory Structure

Figure 28.2 shows the mapping of the code flash memory, and Table 28.2 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.

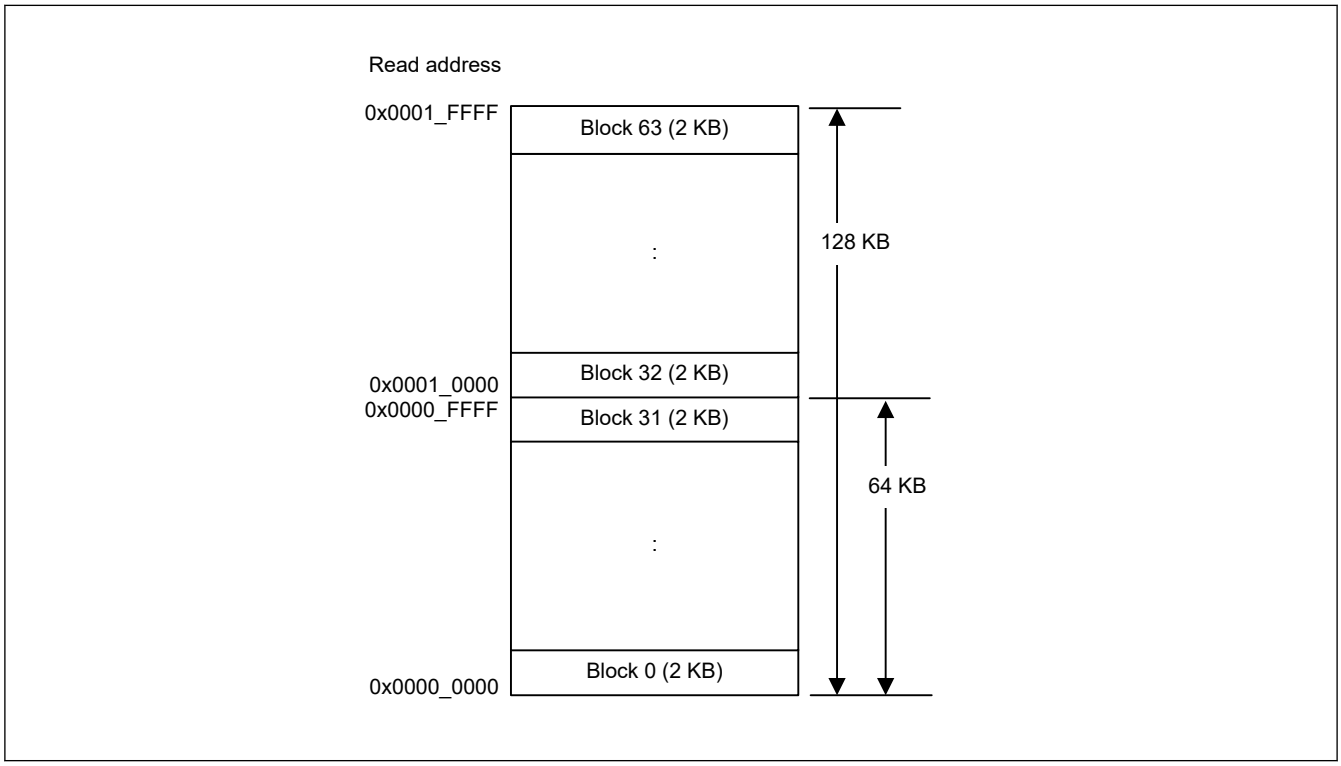


Figure 28.2 Mapping of the code flash memory

Table 28.2 Read and P/E addresses of the code flash memory

Size of code flash memory	Read address	P/E address	Block number
128 KB	0x0000_0000 to 0x0001_FFFF	0x0000_0000 to 0x0001_FFFF	0 to 63
64 KB	0x0000_0000 to 0x0000_FFFF	0x0000_0000 to 0x0000_FFFF	0 to 31

Figure 28.3 shows the mapping of the data flash memory, and Table 28.3 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 256-B blocks, with each being a unit for erasure.

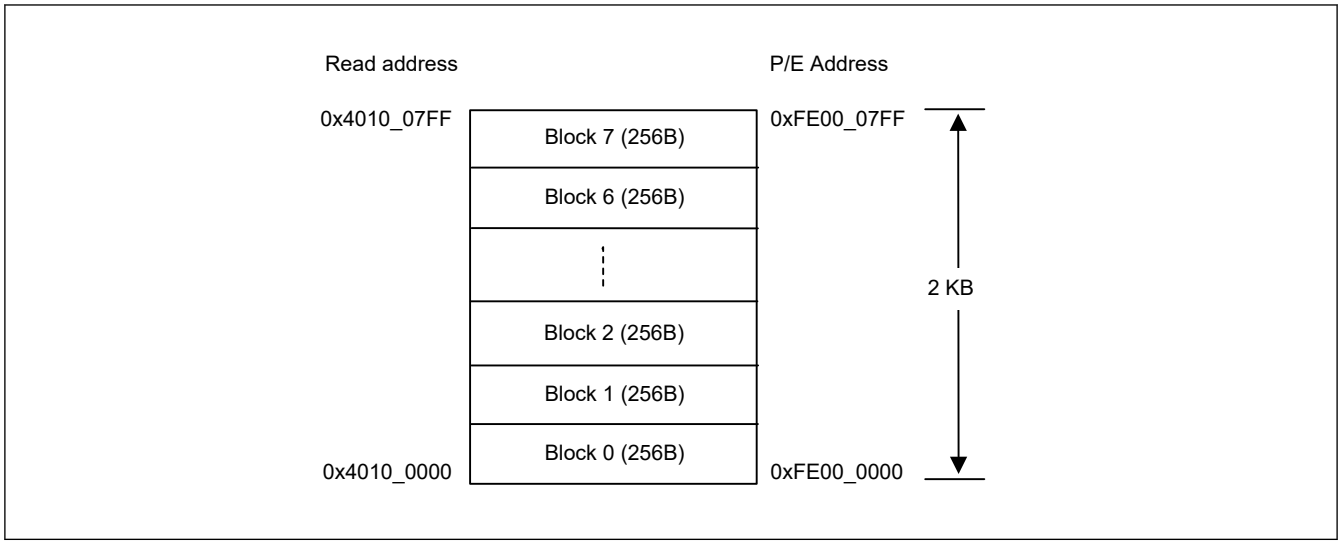


Figure 28.3 Mapping of the data flash memory

Table 28.3 Read and P/E addresses of the data flash memory

Size of data flash memory	Read address	P/E address	Block number
2-KB	0x4010_0000 to 0x4010_07FF	0xFE00_0000 to 0xFE00_07FF	0 to 7

28.3 Register Descriptions

28.3.1 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 security program, access window information program, and OCDID 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.

28.3.2 FENTRYR : Flash P/E Mode Entry Register

Base address: FLCN = 0x407E_C000

Offset address: 0x021A

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
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 28.10.1. 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 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.

FEKEY[7:0] bits (Key Code)

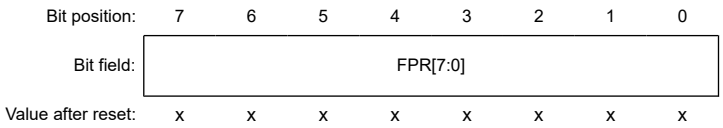
The FEKEY[7:0] bits protect from unauthorized setting of FENTRY0 bit or FENTRYD bit.

Setting 0xAA to FEKEY[7:0] allows setting the FENTRY0 bit or the FENTRYD bit. The FEKEY[7:0] bits are read as 0x00.

28.3.3 FPR : Protection Unlock Register

Base address: FLCN = 0x407E_C000

Offset address: 0x0180



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.	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 specified procedure is used to write data, the FPSR.PERR flag is set to 1.

28.3.4 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 28.3.3. FPR : Protection Unlock Register.

[Clearing conditions]

- The FPMCR register is accessed according to the procedure to unlock protection described in section 28.3.3. FPR : Protection Unlock Register.

28.3.5 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
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 28.13 and Figure 28.15 for this register write control method.

See section 28.3.3. 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 = 0, FMS0 = 1, and RPDIS = 0. Wait for the mode setup time t_{MS} (see section 31, Electrical Characteristics).

[How to enter the data flash from the read mode to the data flash P/E mode]

Set FMS1 = 1, FMS0 = 0, and RPDIS bit = 0.

[How to enter the code flash from the code flash P/E mode to the read mode]

Set FMS1 = 0, FMS0 = 0, and RPDIS = 1.

Wait for the read mode transition time (see section 31, Electrical Characteristics).

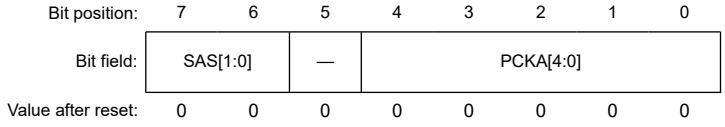
RPDIS bit (Code Flash P/E Disable)

RPDIS bit protects the code flash from unauthorized programming. Setting RPDIS bit to 0 allows the code flash to program.

28.3.6 FISR : Flash Initial Setting Register

Base address: FLCN = 0x407E_C000

Offset address: 0x01D8



Bit	Symbol	Function	R/W
4:0	PCKA[4:0]	Flash-IF Clock Notification	R/W
5	—	This bit is read as 0. The write value should be 0.	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 is 1. The FSPR bit is the protection flag of the access window and is stored in the extra area.

PCKA[4:0] bits (Flash-IF Clock Notification)

The hardware sequencer for the flash programming executes the commands according to the PCKA[4:0] bits. For this reason, it is necessary to set the PCKA[4:0] bits according to Flash-IF clock (ICLK) before execution of the programming and not during the programming.

Note: A wrong frequency setting may cause the flash memory to be damaged.

The following information describes how to set the PCKA[4:0] 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 = 11111b) 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 28.4 Frequency settings

Flash-IF clock frequency [MHz]	PCKA[4:0]	Flash-IF clock frequency [MHz]	PCKA[4:0]	Flash-IF clock frequency [MHz]	PCKA[4:0]
32	11111b	24	10111b	20	10011b
19	10010b	18	10001b	17	10000b
16	01111b	15	01110b	14	01101b
13	01100b	12	01011b	11	01010b
10	01001b	9	01000b	8	00111b
7	00110b	6	00101b	5	00100b
4	00011b	3	00010b	2	00001b
1	00000b	—	—	—	—

SAS[1:0] bits (Startup Area Select)

The SAS[1:0] bits select the startup area. To change the startup area, the following 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 accordingly. 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.

28.3.7 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, FWBL0, 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.

28.3.8 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.

28.3.9 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 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 and FWBL0 registers to the flash memory to the address pointed by the FSARH and FSARL registers.

[Blank check]

Verifies whether the flash memory 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 memory.

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 memory.

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, the block erase, and the read commands cannot be executed when the address of each command points to an area that is protected by the access window.

28.3.10 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 1 0: Access window information program Startup area selection and security setting 0 1 1: OCDID1 program 1 0 0: OCDID2 program 1 0 1: OCDID3 program 1 1 0: OCDID4 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 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:

- Startup area selection and security setting
- Access window information program
- OCDID program.

The following information describes the function of each software command.

[Startup area selection and security setting]

Setting data to the FWBL0/FWBH0 registers, this command is allowed to select the startup area from the default area (8 KB) to the alternative area (next 8 KB) and set the security. For details, see section 28.8.1. Startup Program Protection.

Bit [15] of the FWBH0 register is 0 and the alternative area (next 8 KB) is selected as the startup area.

Bit [15] of the FWBH0 register is 1 and the default area (8 KB) is selected as the startup area.

Bit [15] of the FWBL0 register is 0.

- The access window cannot be updated because the access window information program command cannot be executed.
- The startup area cannot be changed.
- Data of the SAS bits of the FISR register cannot be changed.

Note: The security setting command cannot be set to 1 for the corresponding bit of the extra area after 0 is set.

The following information describes mapping for the extra bit of the startup area selection and security setting.

Table 28.5 Mapping for the extra bit of the startup area selection and security setting (address (P/E) : 0x0000_0010)

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
BTFLG	—	—	—	—	FAWE[10:0]										
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
FSPR *1	—	—	—	—	FAWS[10:0]										

Note 1. Once 0 is set for these bits, it cannot be changed to 1.

[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. 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 28.6 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
BTFLG	—	—	—	—	FAWE[10:0]										
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
FSPR *1	—	—	—	—	FAWS[10:0]										

Note 1. Once 0 is set as data in these bits, it cannot be changed to 1.

[OCDID1-4 program]

These commands set the OCDID[127:0] bits.

Table 28.7 OCDID settings

Command	OCDID	FWBH0	FWBL0
OCDID1 program	OCDID [31:0]	OCDID [31:16]	OCDID [15:0]
OCDID2 program	OCDID [63:32]	OCDID [63:48]	OCDID [47:32]
OCDID3 program	OCDID [95:64]	OCDID [95:80]	OCDID [79:64]
OCDID4 program	OCDID [127:96]	OCDID [127:112]	OCDID [111:96]

The following information describes mapping for the extra bit of OCDID1-4 program.

Table 28.8 Mapping for the extra bit of OCDID1-4 program (address (P/E) : 0x0000_0018)

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
OCDID[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
OCDID[15:0]															

Table 28.9 Mapping for the extra bit of OCDID1-4 program (address (P/E) : 0x0000_0020)

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
OCDID[63:48]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
OCDID[47:32]															

Table 28.10 Mapping for the extra bit of OCDID1-4 program (address (P/E) : 0x0000_0028)

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
OCDID[95:80]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
OCDID[79:64]															

Table 28.11 Mapping for the extra bit of OCDID1-4 program (address (P/E) : 0x0000_0030)

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
OCDID[127:112]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
OCDID[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.

28.3.11 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 b4, and those bits are read as 0.

28.3.12 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.

Note: This product does not have the auto increment function of the program command. It is necessary to set the next address to the FSARH and FSARL registers every time programming flash.

See Figure 28.2 and Figure 28.3 for details on the addresses of the flash memory.

28.3.13 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 b4, and those bits are read as 0.

28.3.14 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 and the block erase command. 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 28.2 and Figure 28.3 for details on the addresses of the flash memory.

28.3.15 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.

28.3.16 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 security setting command, the access window information program command, and the OCDID 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 security setting command, the access window information program command, and the OCDID program command.

28.3.17 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
1:0	—	These bits are 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.

28.3.18 FSTATR2 : Flash Status Register 2

Base address: FLCN = 0x407E_C000

Offset address: 0x01F0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	EILGLERR	ILGLERR	BCERR	—	PRGERR	ERERR
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	ERERR	Erase Error Flag 0: Erasure terminates normally 1: An error occurs during erasure	R
1	PRGERR	Program Error Flag 0: Programming terminates normally 1: An error occurs during programming.	R
2	—	This bit is read as 0.	R
3	BCERR	Blank Check Error Flag 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

Bit	Symbol	Function	R/W
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
15:6	—	These bits are read as 0.	R

FSTATR2 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.

ERERR flag (Erase Error Flag)

The value of the ERERR bit is undefined when the FCR.STOP bit is set to 1 (processing is forcibly stopped) during erasure.

PRGERR flag (Program Error Flag)

The PRGERR bit is set when the program command of the FCR register or each command of the FEXCR register is abnormally terminated.

ILGLERR flag (Illegal Command Error Flag)

The ILGLERR flag indicates the execution of the software command of the FCR register with unexpected condition.

[Setting condition]

- Programming/erasure/read commands are executed to an area protected by the access window range
- The blank check and the block 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 and the blank check commands are executed when the FASR.EXS bit is 1
- 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 conditions]

- 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 condition]

- The software commands of the FEXCR register is executed when the EXS bit of the FASR register is 0
- The access window information program command is executed when the FSPR bit is 0

[Clearing conditions]

- The next software command is executed.

28.3.19 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

28.3.20 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

The error address is withdrawn from the FEAMH and FEAML registers after a software command execution. See Figure 28.2 and Figure 28.3 for details on the addresses of the flash memory.

28.3.21 FSCMR : Flash Start-up Setting Monitor Register

Base address: FLCN = 0x407E_C000

Offset address: 0x01C0

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

Note 1. The reset value depends on the state of the extra area.

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0.	R
8	SASMF	Startup Area Setting Monitor Flag 0: Setting to start up using the alternative area 1: Setting to start up using the default area	R
13:9	—	These bits are read as 0.	R
14	FSPR	Access Window Protection Flag 0: Access window setting disabled. 1: Access window setting enabled.	R
15	—	This bit is read as 0.	R

The FSCMR register monitors the extra area setting. Data of this register is updated at the reset sequence or execution of the software command of the FEXCR register.

28.3.22 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:	FSPR	—	—	—	—	FAWS[10:0]										
Value after reset:	x	0	0	0	0	The value set by the user*1										

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.

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
14:11	—	These bits are read as 0.	R
15	FSPR	Access Window Protection Flag This bit has the same value as the FSPR bit of the FSCMR register.	R

28.3.23 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:	SASMF	—	—	—	—	FAWE[10:0]										
Value after reset:	x	0	0	0	0	The value set by the user*1										

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.

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
14:11	—	These bits are read as 0.	R
15	SASMF	Startup Area Setting Monitor Flag This bit has the same value as the SASMF bit of the FSCMR register.	R

28.3.24 UIDRn : Unique ID Registers n (n = 0 to 3)

Address: 0x0101_1070 + n × 4

Bit position:	31	0
Bit field:	UID[31:0]	
Value after reset:	Unique value for each chip	

Bit	Symbol	Function	R/W
31:0	UID[31:0]	Unique ID	R

The UIDRn is a read-only register that stores a 16-byte ID code (unique ID) for identifying the individual MCU. The UIDRn register should be read in 32-bit units.

28.3.25 PNRn : Part Numbering Register n (n = 0 to 3)

Address: 0x0101_1080 + n × 4

Bit position:	31	0
Bit field:	PNR[31:0]	
Value after reset:	Unique value for each chip	

Bit	Symbol	Function	R/W
31:0	PNR[31:0]	Part Number	R

The PNRn is a read-only register that stores a 16-byte part numbering. The PNRn register should be read in 32-bit units. Each byte corresponds to the ASCII code representation of the product part number as detailed in product list.

In case of the part number is ’R7FA0E2073CNH’, 16-byte part numbering is stored as follows.

Address 0x0101_1080: ’H’, 0x48 in ASCII code

Address 0x0101_1081: ’N’, 0x4E in ASCII code

Address 0x0101_1082: ’C’, 0x43 in ASCII code

Address 0x0101_1083: ’3’, 0x33 in ASCII code

Address 0x0101_1084: ’7’, 0x37 in ASCII code

Address 0x0101_1085: ’0’, 0x30 in ASCII code

Address 0x0101_1086: ’2’, 0x32 in ASCII code

Address 0x0101_1087: ’E’, 0x45 in ASCII code

Address 0x0101_1088: ’0’, 0x30 in ASCII code

Address 0x0101_1089: ’A’, 0x41 in ASCII code

Address 0x0101_108A: ’F’, 0x46 in ASCII code

Address 0x0101_108B: ’7’, 0x37 in ASCII code

Address 0x0101_108C: ’R’, 0x52 in ASCII code

Address 0x0101_108D: ”(space) , 0x20 in ASCII code

Address 0x0101_108E: ”(space) , 0x20 in ASCII code

Address 0x0101_108F: ”(space) , 0x20 in ASCII code

28.3.26 MCUVER : MCU Version Register

Address: 0x0101_1090

Bit position:	7	6	5	4	3	2	1	0
Bit field:	MCUVE[7:0]							
Value after reset:	Value depend on the chip							

Bit	Symbol	Function	R/W
7:0	MCUVE[7:0]	MCU Version	R

The MCUVER is a read-only register that stores a MCU version. The MCUVER register should be read in 8-bit units. The higher the value, the newer MCU version.

28.4 Operating Modes Associated with the Flash Memory

Figure 28.4 shows a diagram of the mode transitions associated with the flash memory. For information on setting up the modes, see section 3, Operating Modes.

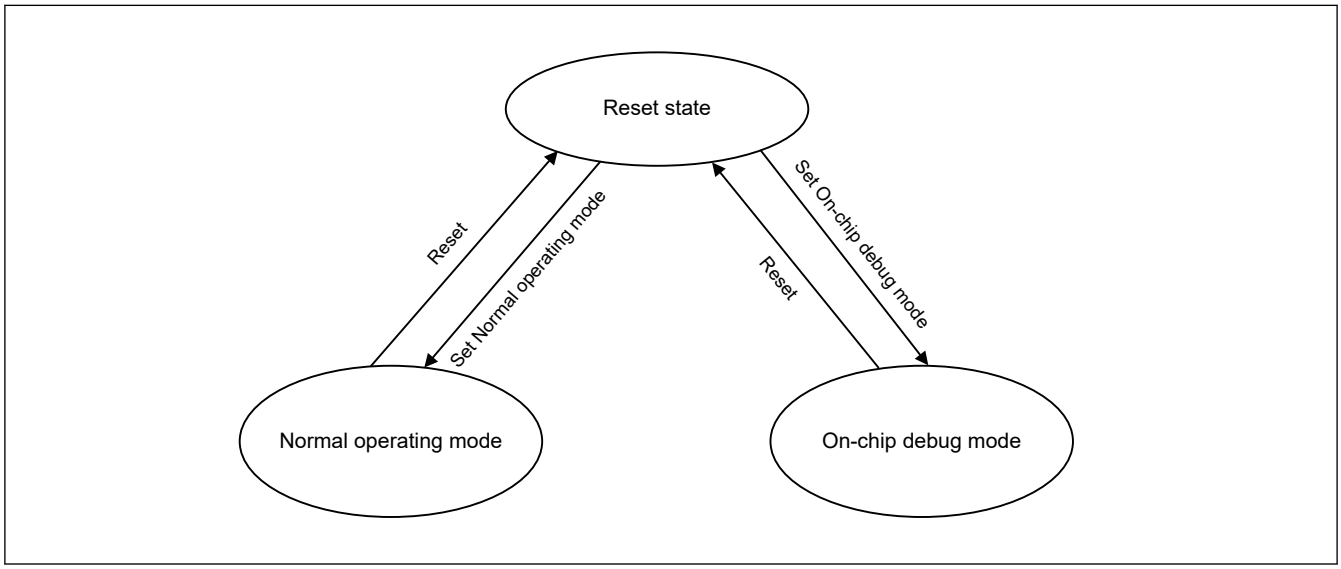


Figure 28.4 Mode transitions associated with flash memory

The flash memory areas where programming and erasure are permitted and where the boot program executes at a reset, differ with the mode. Table 28.12 shows the differences between the modes.

Table 28.12 Difference between modes

Parameter	Normal operating mode	On-chip debug mode
Programmable and erasable areas	• Code flash memory • Data flash memory.	• Code flash memory • Data flash memory.
Erasure in block units	Possible	Possible
Boot program at a reset	User area program	Depends on debug command

28.4.1 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 28.13 shows how the ID code determines the authentication method.

Table 28.13 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
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 on-chip debugger or serial programmer*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 on-chip debugger or serial programmer is permitted. Mismatching ID code indicates transition to the ID code protection wait state. When the ID code sent from the on-chip debugger or serial programmer is ALeRASE in ASCII code (0x414C_6552_4153_45FF_FFFF_FFFF_FFFF_FFFF), the content of the user flash area is erased and all bits in the OSIS register are 1. However, when the AWS.FSPR bit is 0, the content of the user flash area is not erased.
	Bit [127] = 1 and bit [126] = 0	Protection enabled	Matching ID code indicates that authentication is complete and connection to the on-chip debugger or serial programmer 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 on-chip debugger or serial programmer is prohibited.

Note 1. Never send the ID code from on-chip debugger. Or send ID code 0xFF (all bytes 0xFF) from on-chip debugger.

28.5 Overview of Functions

By using a dedicated flash-memory programmer to program the on-chip flash memory through SWD interface (on-chip debug mode), the device can be programmed before or after it is mounted on the target system. Additionally, security 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 28.14 lists the programming methods and the associated operating modes.

Table 28.14 Programming methods

Programming method	Functional overview	Operating mode
Self-programming	A user program written to memory 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
SWD programming	A dedicated flash-memory programmer or an on-chip debugger connected through SWD 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 SWD and a dedicated programming adapter board allow off-board programming of the flash memory, before it is mounted on the target system.	

Table 28.15 lists the functions of the on-chip flash memory. For self-programming, use the programming commands to read the on-chip flash memory or run the user program.

Table 28.15 Basic functions

Function	Functional overview	Availability	
		Self-programming	SWD programming
Blank check	Checks a specified block to ensure that writing to it has not already proceeded.	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	Not supported (read by user program is possible)	Not supported
ID code check	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.	Not supported (ID authentication is not performed)	Supported
Security configuration	Configures the protection of security function (Access window and Start-up area selection)	Supported with conditions See section 28.8. Protection	Supported with conditions See section 28.8. Protection
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 security function for use with SWD programming. Table 28.16 lists the available operations and security settings.

Table 28.16 Available operations and security settings

Function	All security settings and erasure, programming, and read operations		Constraints on the security setting configuration
	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 ● Security configuration commands: not supported ● Protection configuration commands: not supported. When the ID codes match: ● Block erasure commands: supported ● Programming commands: supported ● Read commands: supported ● Security configuration commands: supported ● Protection configuration commands: supported.	● Blank check: supported ● Block erasure: supported ● Programming: supported ● Security configuration: supported ● Protection configuration: supported.	ID authentication is not performed in the Self-programming mode.

28.5.1 Configuration Area Bit Map

The bits used for ID authentication, startup area select, access window protection, and security configuration functions are mapped in Figure 28.5. The boot program must use these bits as hexadecimal data.

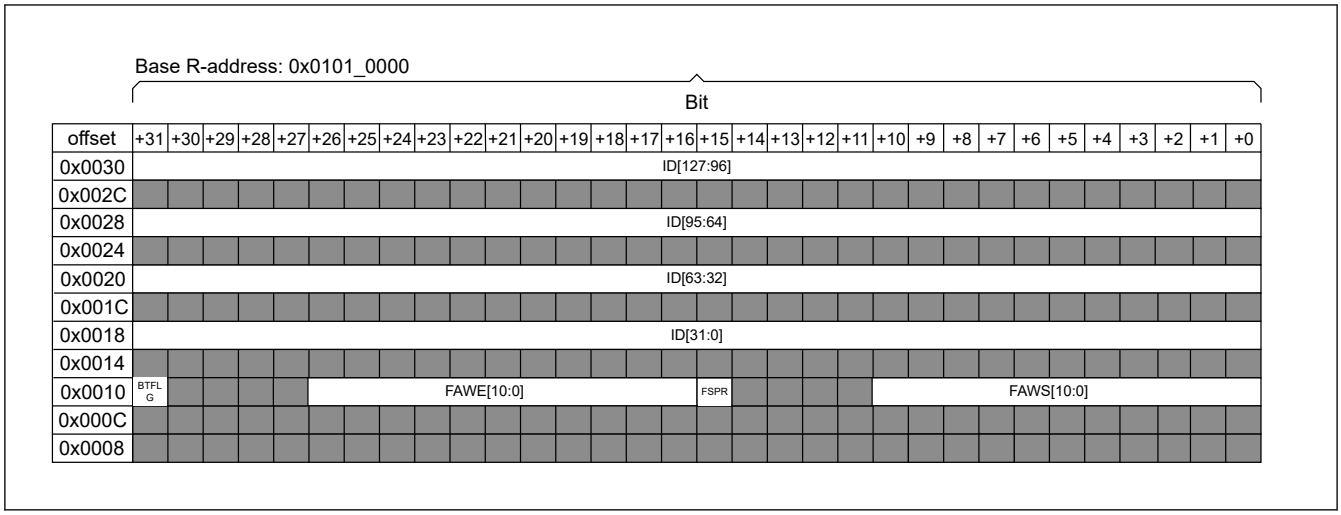


Figure 28.5 Configuration area bit map

28.5.2 Startup Area Select

The startup area select function allows the boot program to be safely updated. The startup area is 8 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 AWS register. The startup area can be locked by the FSPR bit.

Figure 28.6 shows an overview of the startup program protection.

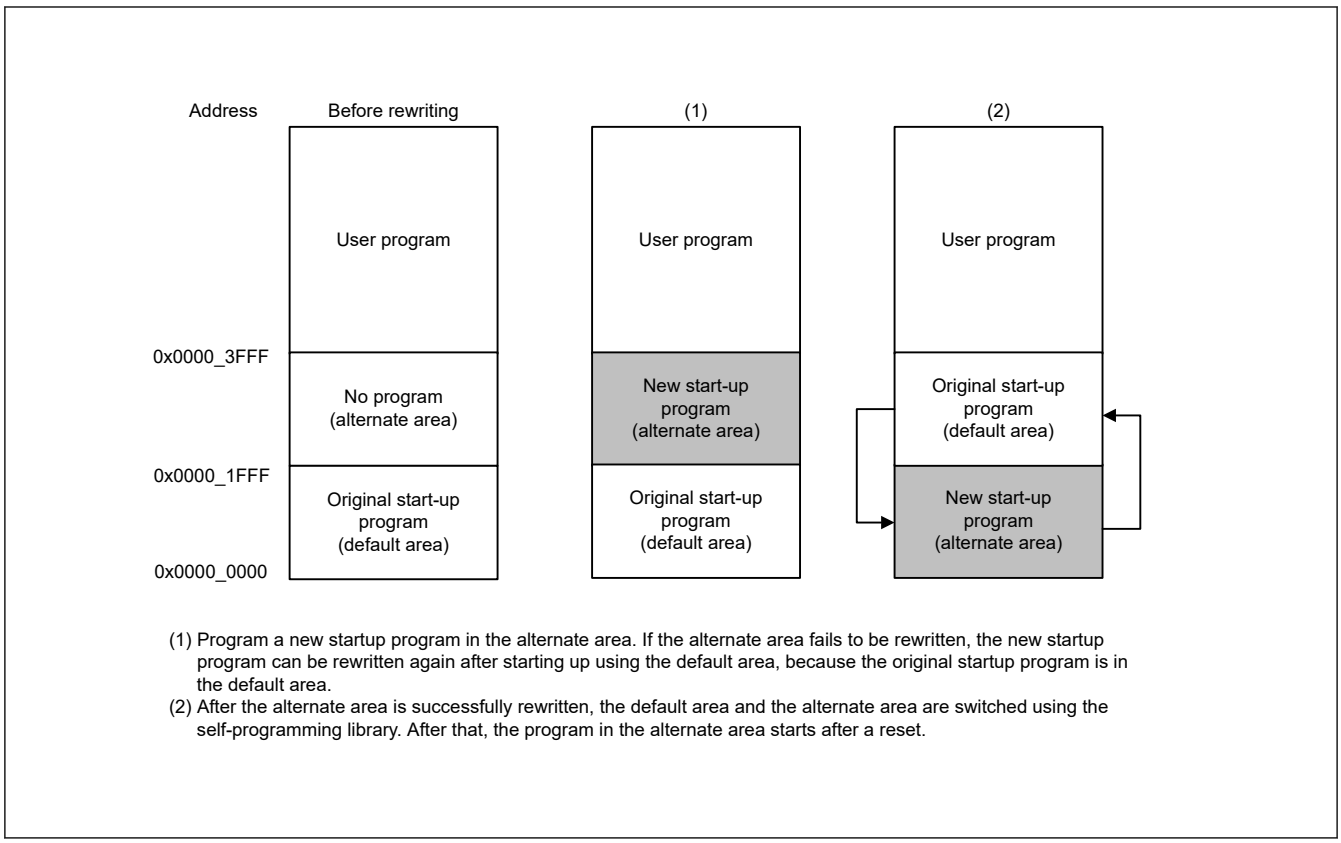


Figure 28.6 Overview of startup program protection

28.5.3 Protection by Access Window

Issuing the program or block erase command to a flash memory area outside 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, and on-chip debug modes. Figure 28.7 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 6.2.3. 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 command can execute anywhere in the user area of the code flash memory
- FAWE [10:0] > FAWS [10:0]: The P/E 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 command cannot execute anywhere in the user area of the code flash memory.

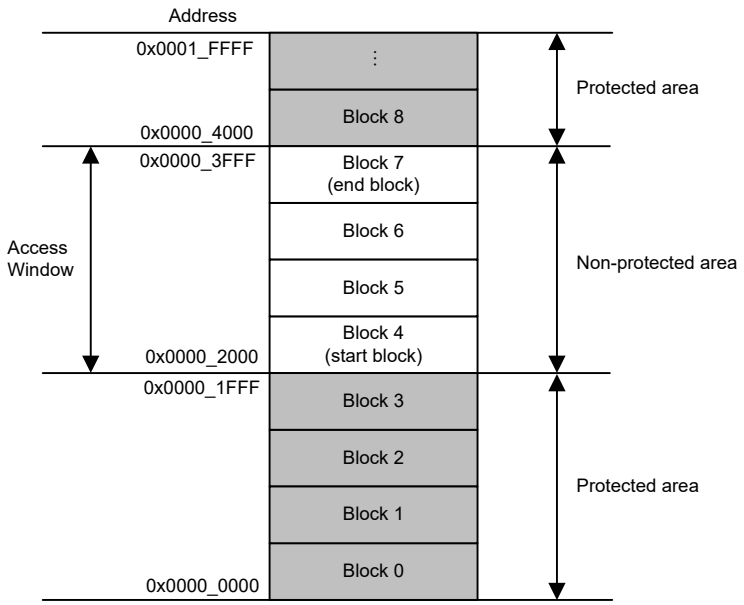


Figure 28.7 Flash area protection overview

28.6 Programming Commands

The FCB controls the programming commands.

28.7 Suspend Operation

The forced stop command forces the blank check command or the block 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.

28.8 Protection

The types of protection provided include:

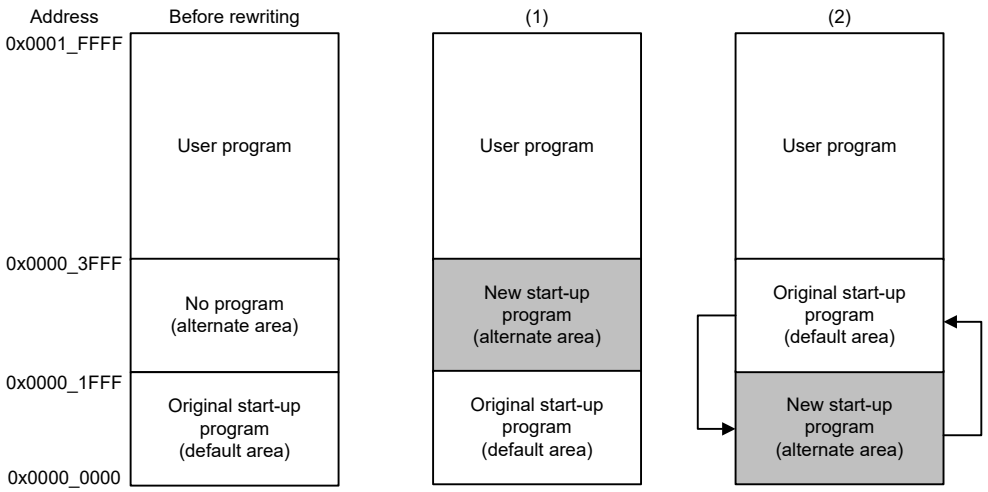
- Startup Program Protection
- Area Protection

28.8.1 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 28.8 shows an overview of the Startup Program Protection. In this figure, the default area indicates the 8-KB region from the start address and the alternate area indicates the next 8-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 28.8 Overview of the startup program protection

28.8.2 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 (self-programming mode and OCD mode).

Figure 28.9 shows an overview of area protection.

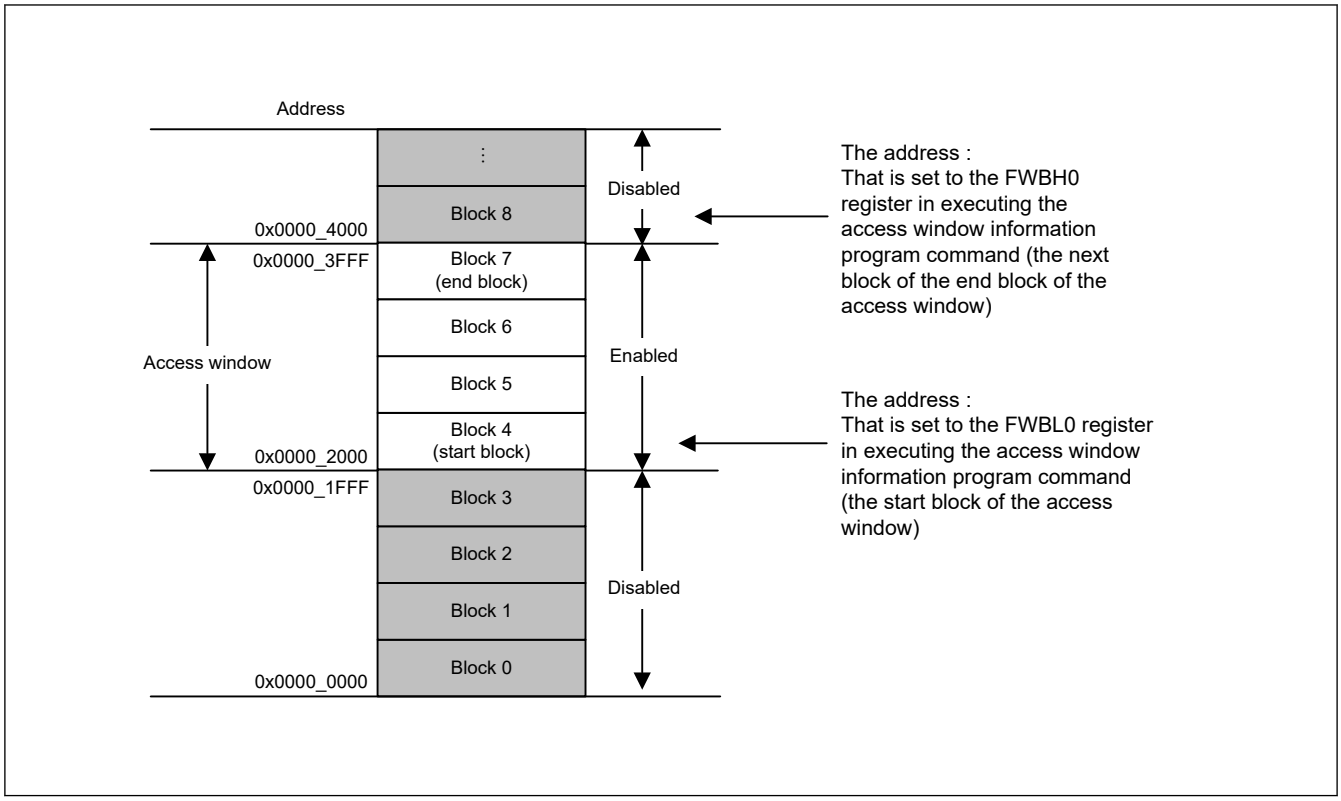


Figure 28.9 Area protection overview

28.9 Self-programming

28.9.1 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.

In self-programming, it is necessary to supply a stable HOCO clock to the flash memory in order to generate the program voltage and erase voltage. Therefore, in case that the HOCO is stopped where another clock source is selected as the system clock, it is necessary to start the HOCO operation and ensure that the oscillation is in a stable state before executing the self-programming. For details of HOCO clock oscillation stabilization check, see section 8.2.14. OSCSF : Oscillation Stabilization Flag Register.

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 28.10. 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.

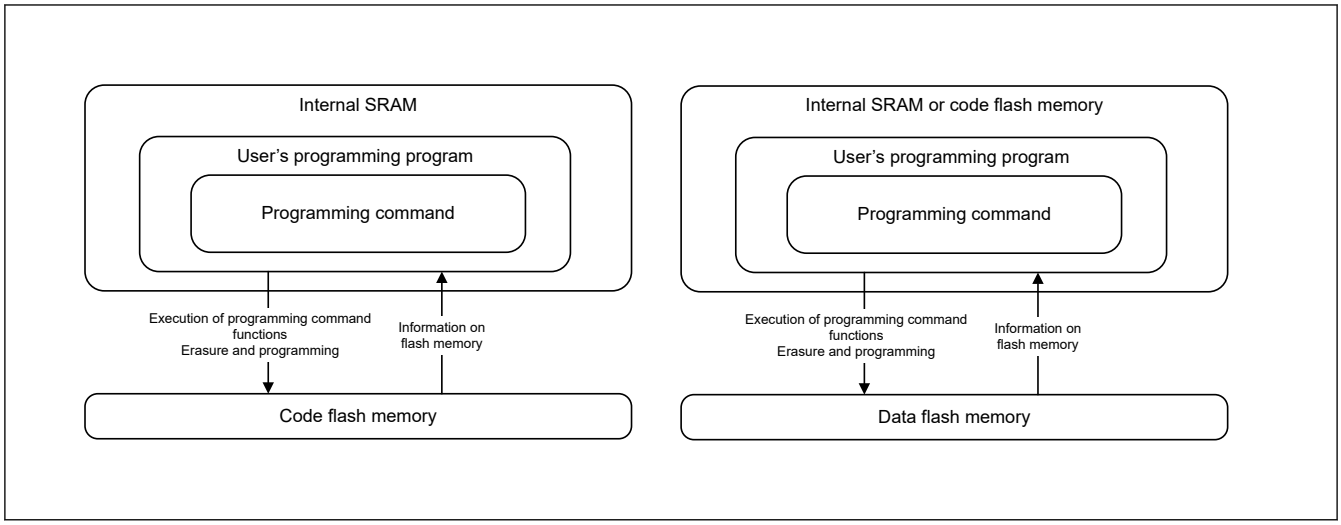


Figure 28.10 Schematic view of self-programming

28.9.2 Background Operation

Background operation can be used when a combination of the flash memory for writing and reading is as listed in Table 28.17.

Table 28.17 Conditions under which background operation is available

Product	Writable range	Readable range
All products	Data flash memory	Code flash memory

28.10 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 single-chip mode.

28.10.1 Sequencer Modes

The sequencer has four modes and transitions between modes occur by writing to the DFLCTL register, or by issuing commands to set the FPMCR register. Figure 28.11 shows mode transitions of the flash memory.

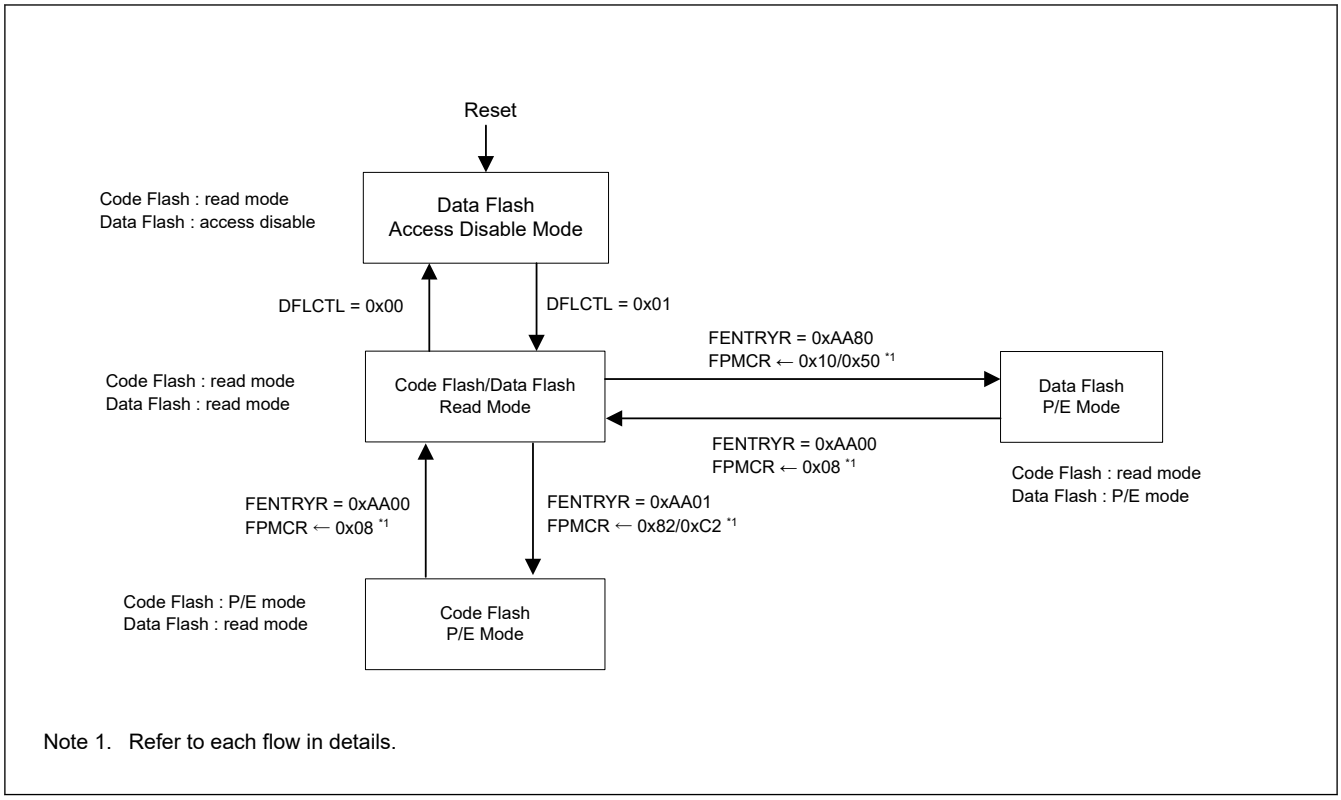


Figure 28.11 Mode transitions of the flash memory

28.10.1.1 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.

28.10.1.2 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.

28.10.1.3 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.

28.10.2 Software Commands

Software commands consist of commands for programming and erasure, and commands for programming startup program area information and access window information. Table 28.18 lists the software commands for use with the flash memory.

Table 28.18 Software commands

Command	Function
Program	Code flash programming (4 bytes) Data flash programming (1 byte)
Block 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 security program	Set the FSPR or the SASMF to the extra area
Access window information program	Set the access window used for area protection to the extra area
OCDID program	Set the OCDID to the extra area

28.10.3 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 28.12 shows the procedure for entering the code flash/data flash read mode from the data flash access disable mode.

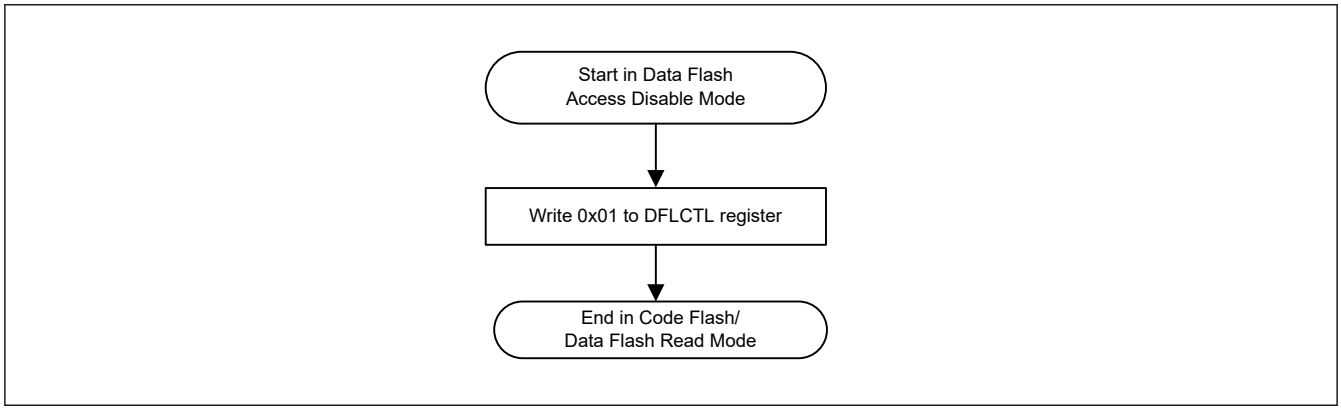
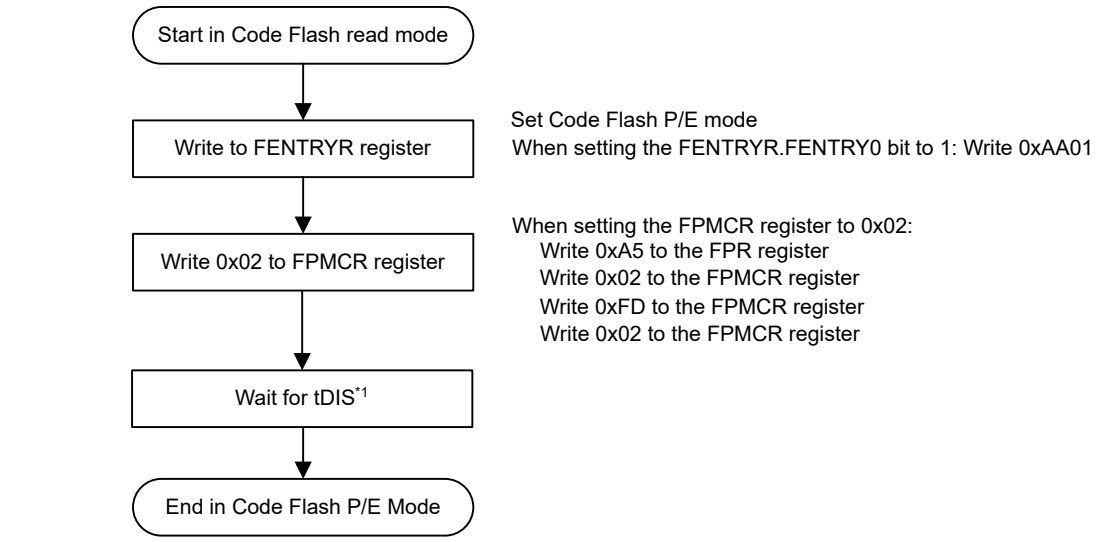


Figure 28.12 Mode transitions to read mode from data flash access disable mode

(2) Switching to Code Flash P/E Mode

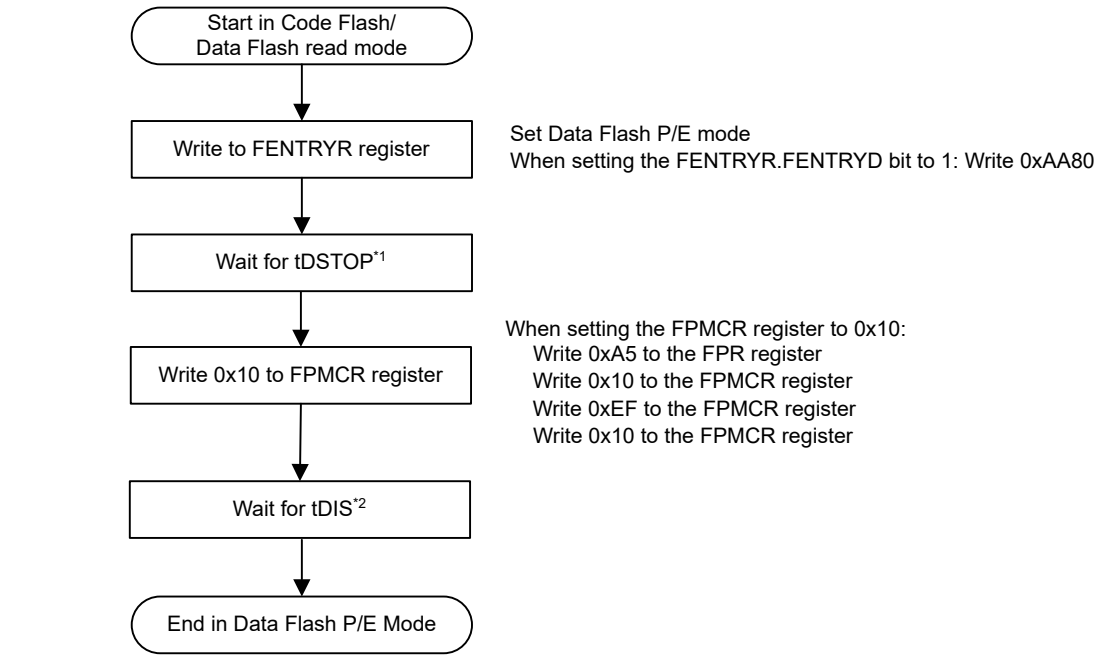
It is necessary to enter the code flash P/E mode by setting the FENTRY0 bit of the FENTRYR register before executing the software command for the code flash. Figure 28.13 shows the procedure for entering code flash P/E Mode.



Note 1. t_{DIS}: Flash memory mode transition wait time 1 (See section 31, Electrical Characteristics.)

Figure 28.13 Procedure for changing from read mode to code flash P/E mode

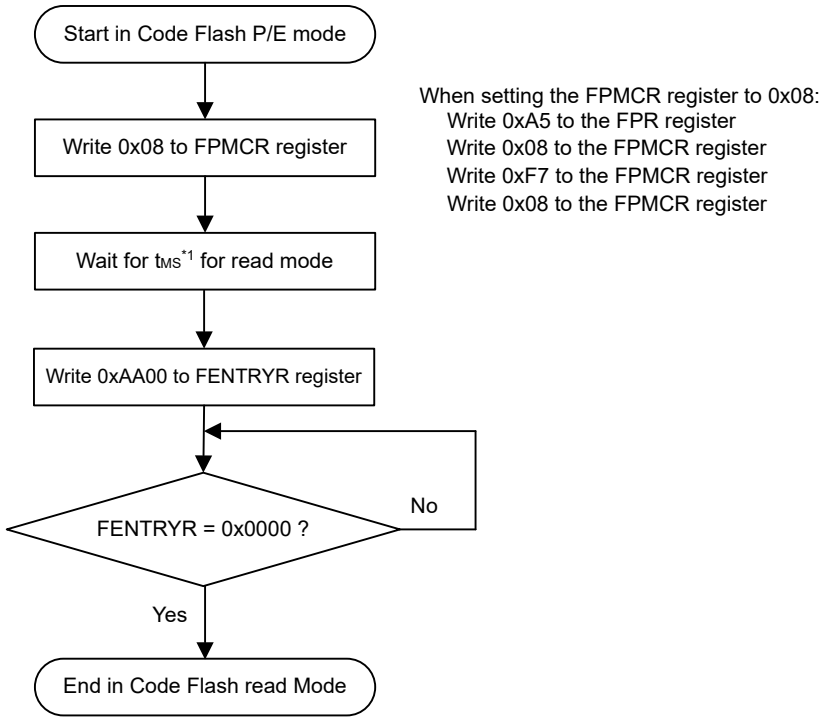
It is necessary to enter the data flash P/E mode by setting the FENTRYD bit of the FENTRYR register before executing the software command for the data flash. Figure 28.14 shows the procedure for entering to the data flash P/E Mode.



Note 1. t_{DSTOP}: STOP recovery time (See section 31, Electrical Characteristics.)
Note 2. t_{DIS}: Flash memory mode transition wait time 1 (See section 31, Electrical Characteristics.)

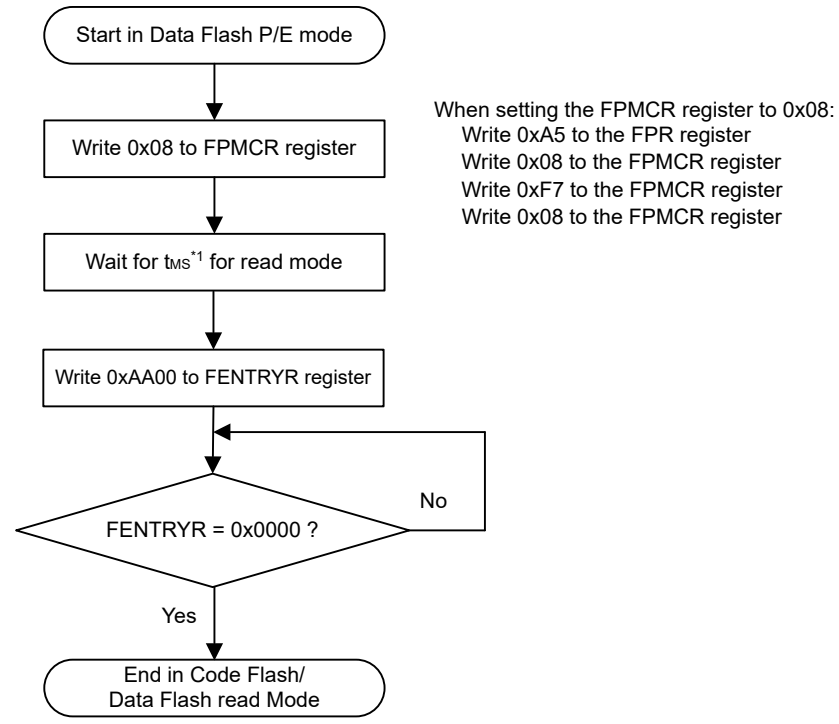
Figure 28.14 Procedure for changing from read mode to data flash P/E mode

(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 31, Electrical Characteristics.)

Figure 28.15 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 31, Electrical Characteristics.)

Figure 28.16 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.

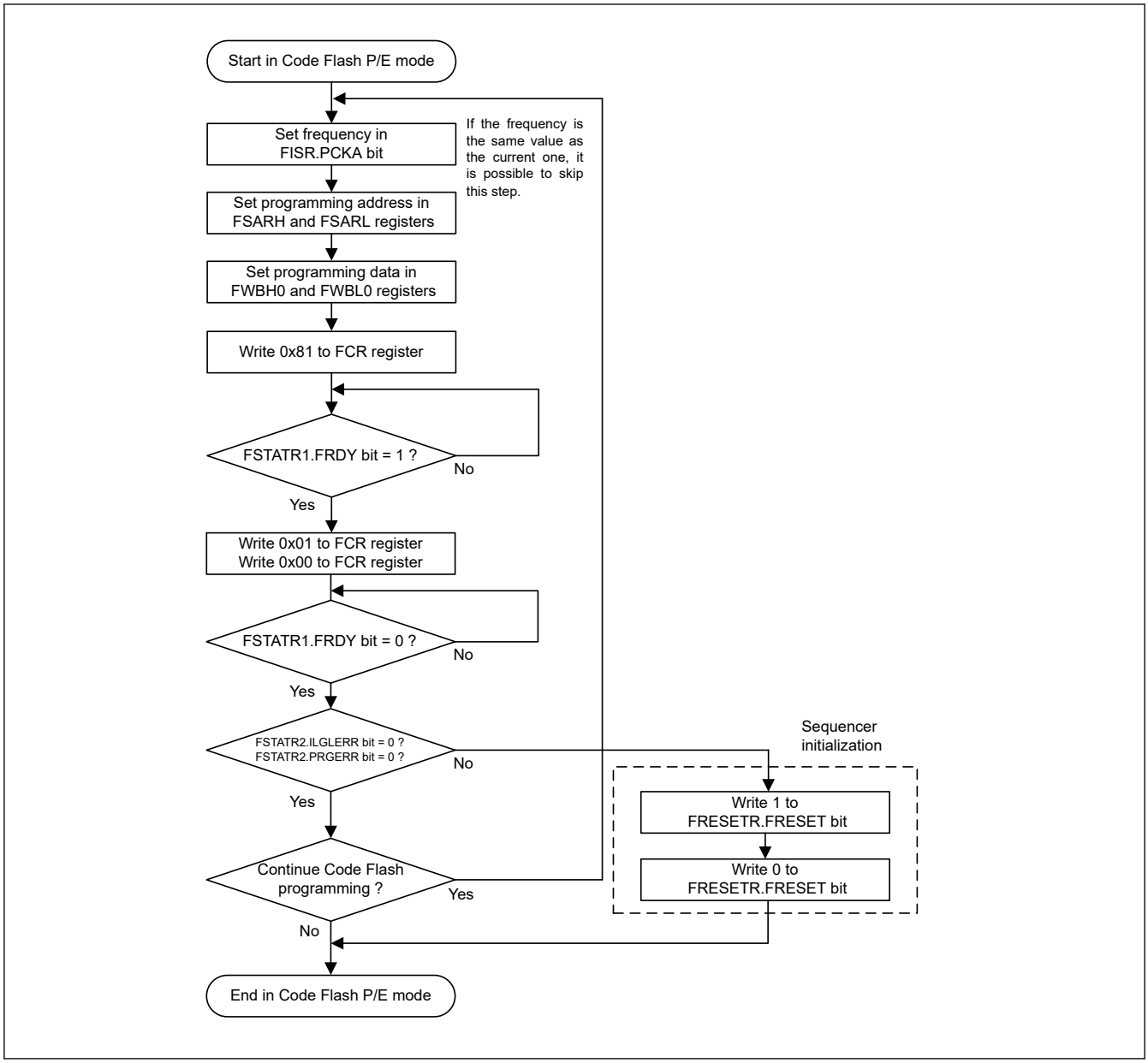


Figure 28.17 Flowchart for programming of the code flash

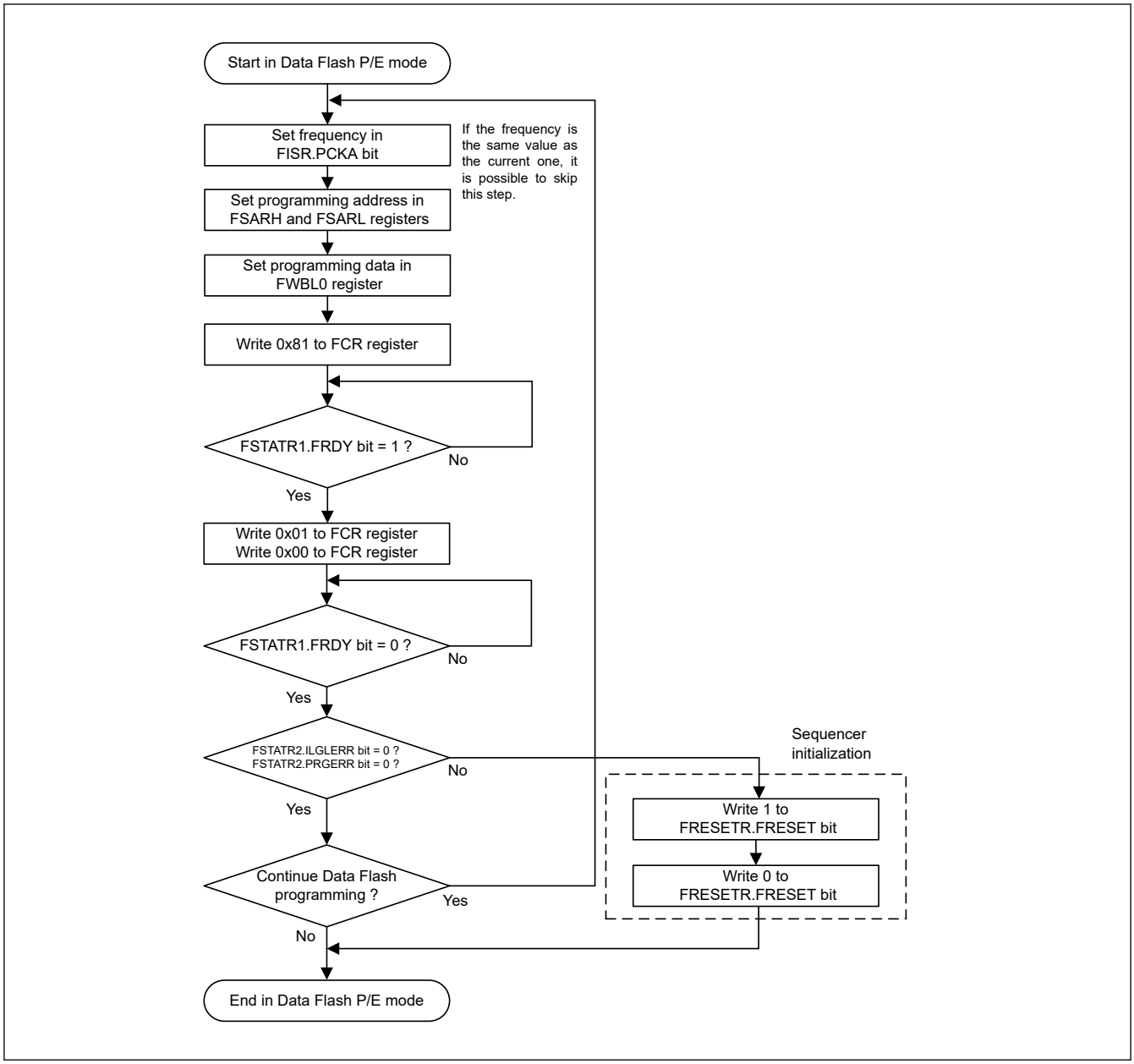


Figure 28.18 Flowchart for programming of the data flash

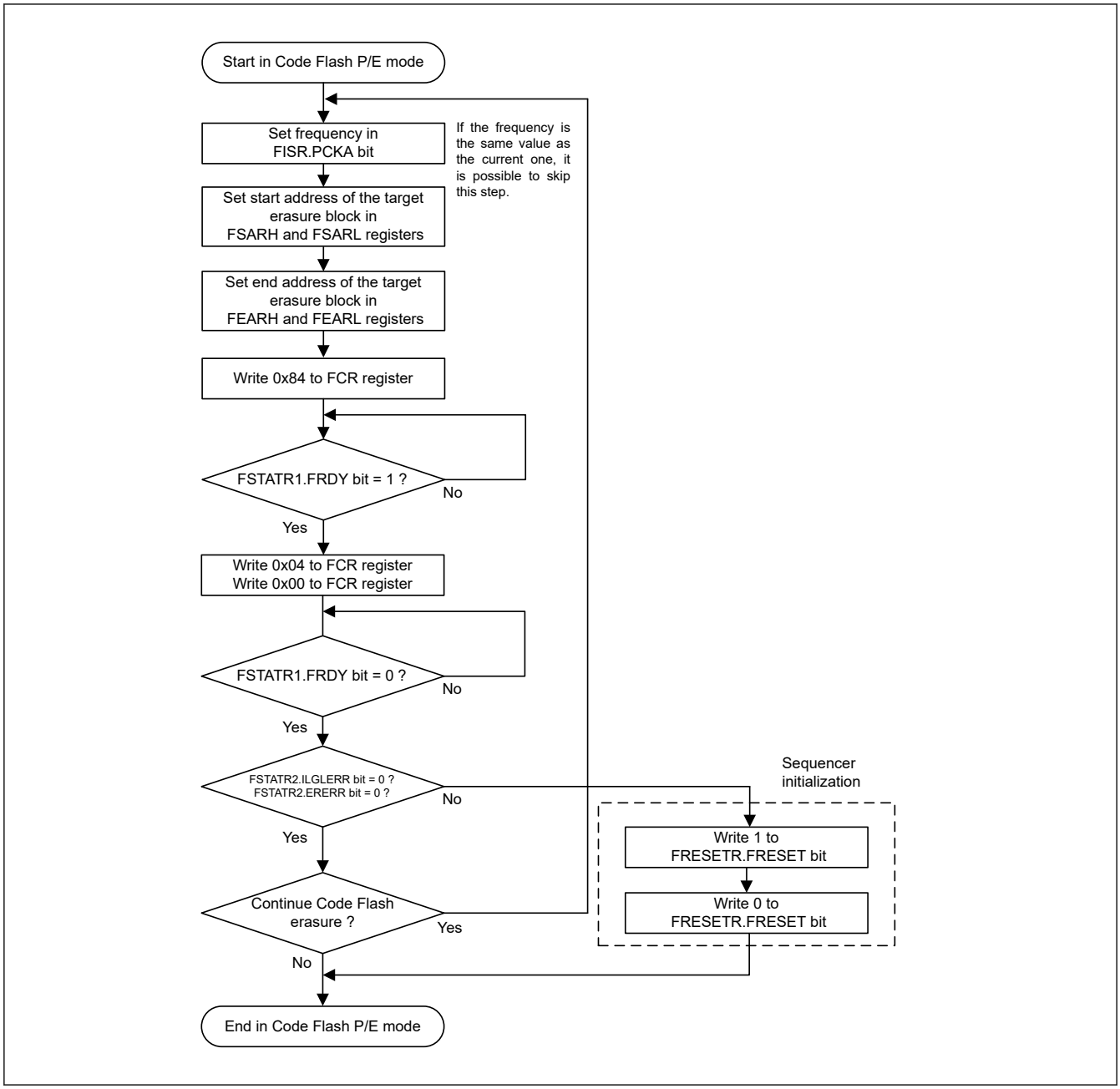


Figure 28.19 Flowchart for the code flash block erase procedure

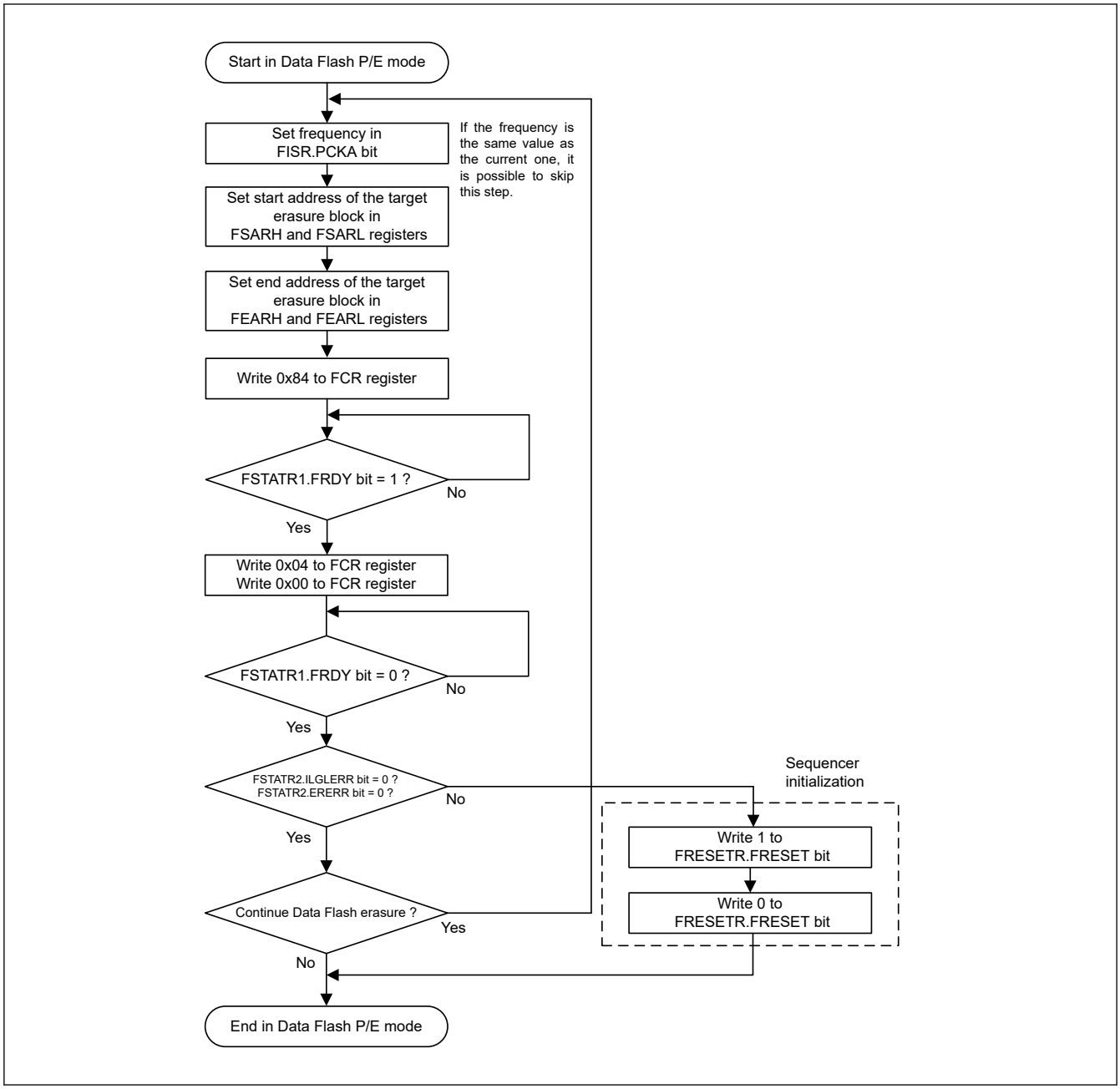


Figure 28.20 Flowchart for the data flash block erase procedure

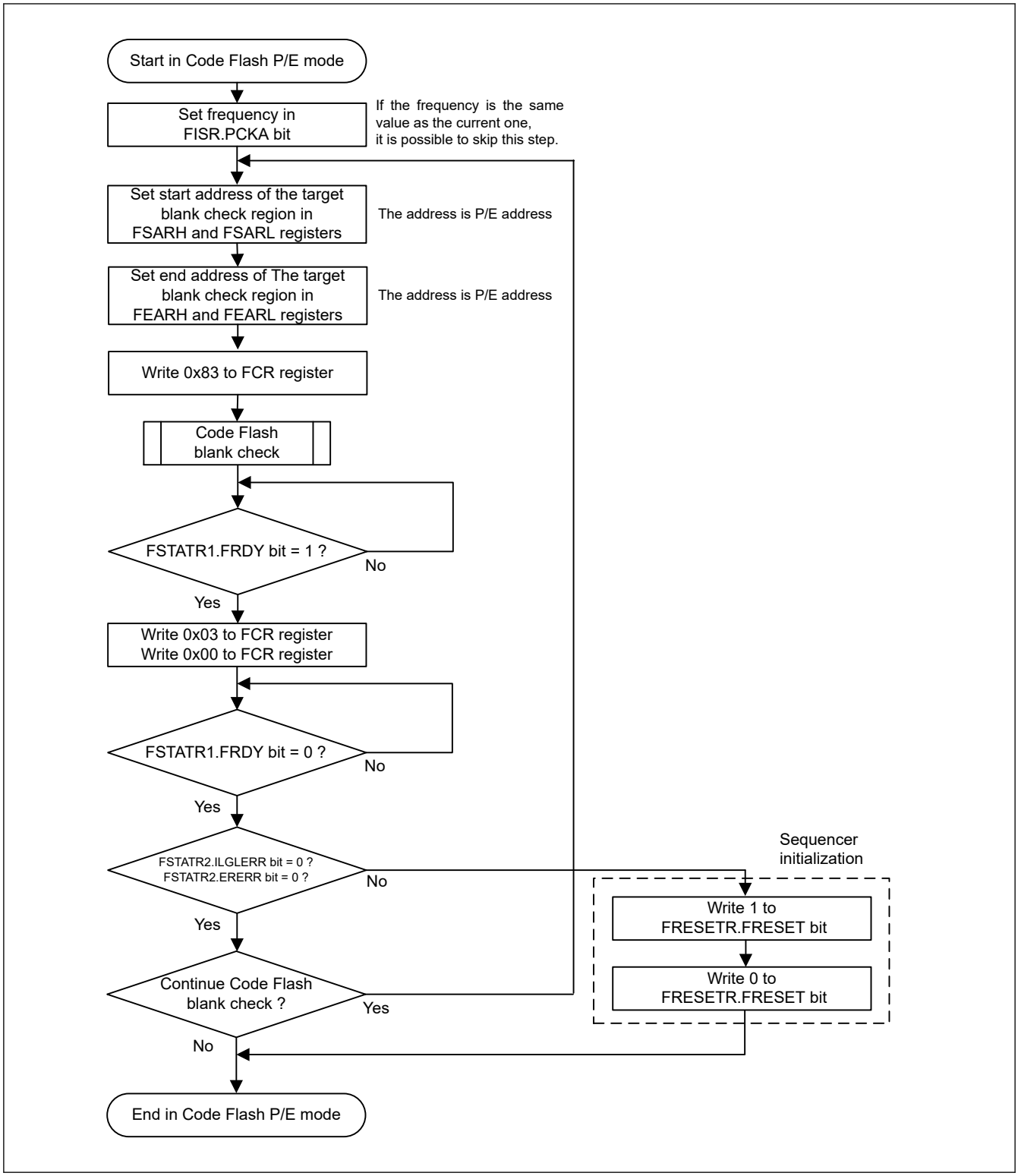


Figure 28.21 Flowchart for the code flash blank check procedure

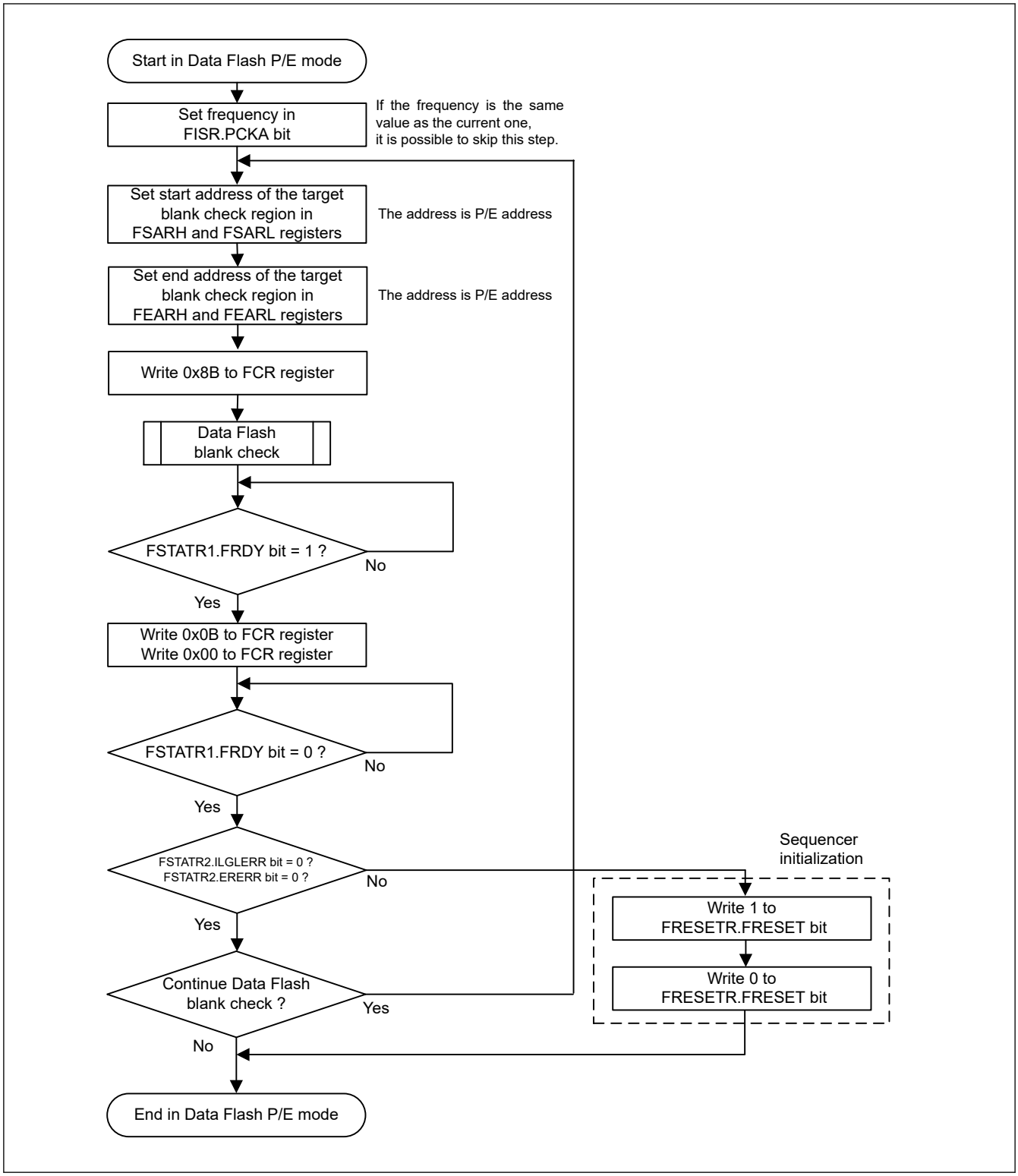
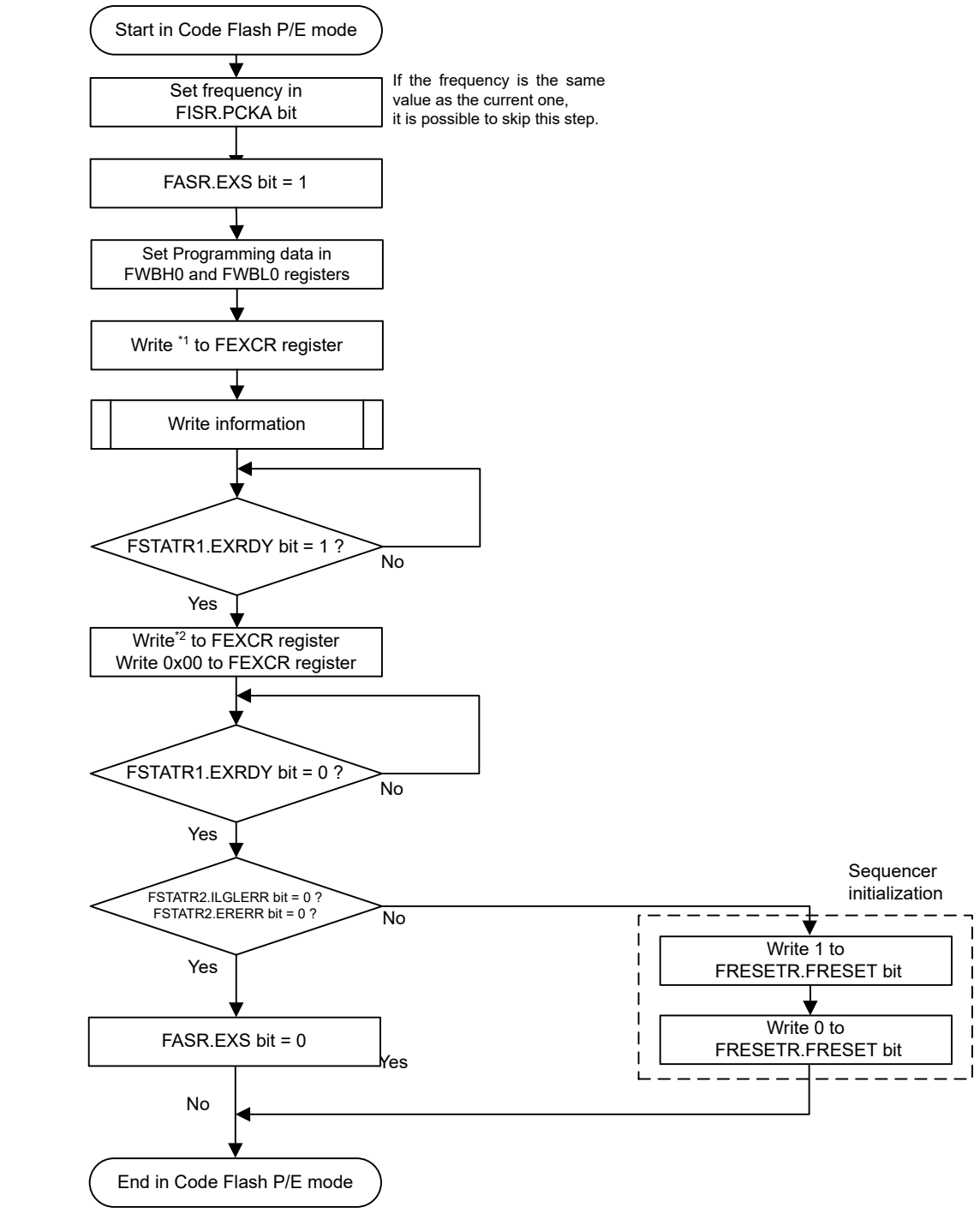


Figure 28.22 Flowchart for the data flash blank check procedure

(5) Startup Area Information and FSPR Program/Access Window Information Program/OCDID information Program

Figure 28.23 is a simple flowchart of the procedure for the startup area information and FSPR program/access window information program/OCDID information program.



Note: Order of the FSPR Bit Setting by Startup Area Information and FSPR Program
Note 1. Write data :
0x81 : Reserved
0x82 : start-up area information and FSPR bit access window information
0x83 : OCDID1 information
0x84 : OCDID2 information
0x85 : OCDID3 information
0x86 : OCDID4 information
Note 2. 0x0y (y = 1 to 6)

Figure 28.23 Simple flowchart for the procedure for Startup Area Information and FSPR Program/Access Window Information Program/OCDID information Program

Set the FSPR bit after programing of the startup area information and the access window information. If the FSPR bit is set before programing of the startup area information and the access window information, the programming cannot be performed because of the security function in the FSPR. When programming using the hex file, programming in the ascending order of the address. In this case, the FSPR bit is written before the access window information. Therefore, divide the hex file for FSPR into another file, and use it after setting the access window information.

(6) Forced Stop by Software Command

Figure 28.24 shows a simple flowchart for the forced stop procedure to stop the blank check command or the block erase 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.

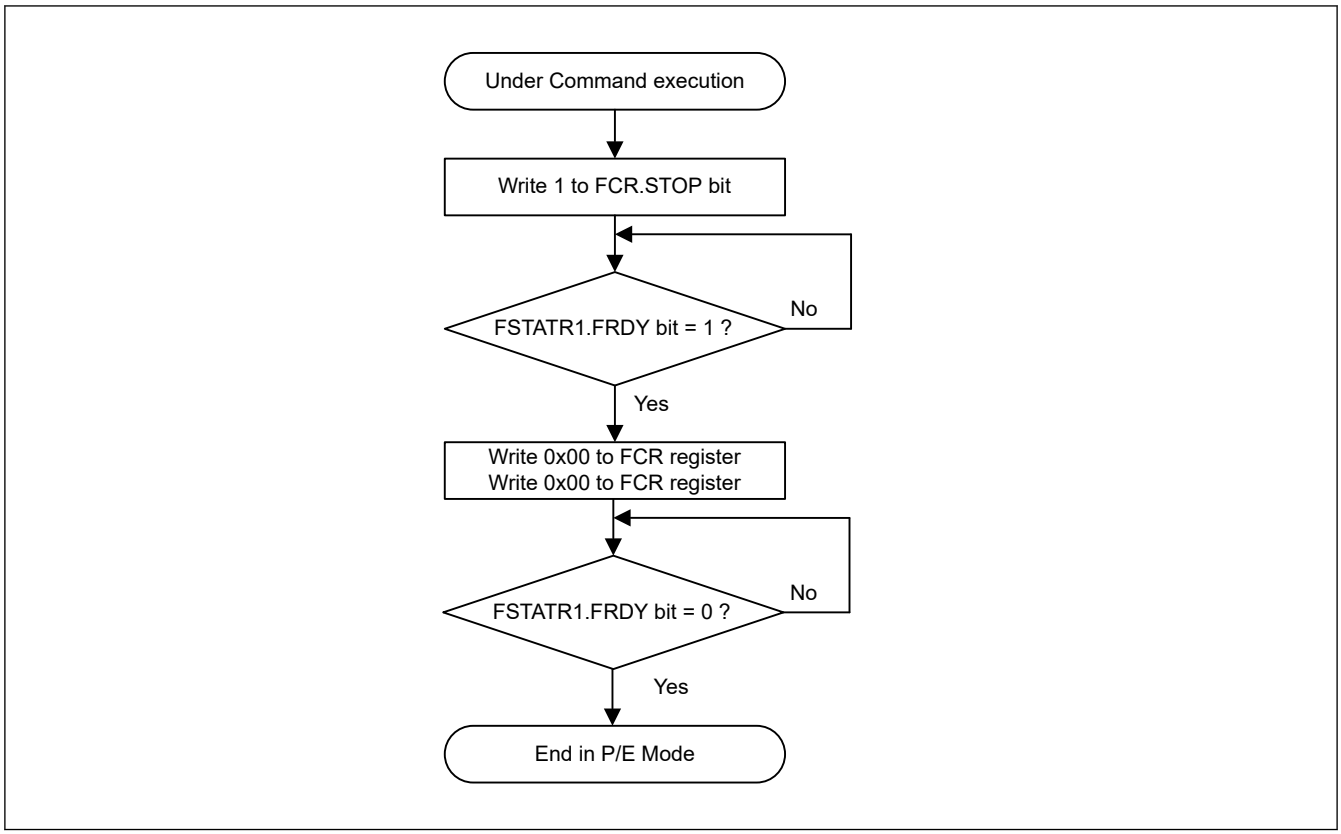


Figure 28.24 Simple flowchart for the forced stop procedure

28.11 Reading the Flash Memory

28.11.1 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.

28.11.2 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.

28.12 Usage Notes

28.12.1 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.

28.12.2 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.

28.12.3 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 31.4.1. 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.

28.12.4 Non-Maskable Interrupt Disabled during Programming and Erasure

When a non-maskable interrupt[*1] occurs during a programming or erasure operation, the vectors are fetched from the code flash memory, and undefined data is read. Therefore, do not generate a non-maskable interrupt during programming and erasure operations in the code flash memory. This constraint applies only to the code flash memory.

Note 1. A non-maskable interrupt is an NMI pin interrupt, IWDT underflow or refresh error, voltage monitor interrupt, SRAM parity error.

28.12.5 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.

28.12.6 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 ICLKSCR register for low-power consumption functions.

28.12.7 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 28.12.8. Actions Prohibited during Programming and Erasure, erase the area again.

28.12.8 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 FLMODE.MODE[1:0] bits value
- Do not update the ICLKSCR.CKSEL 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 data flash access control setting during a program or erase operation to the data flash memory.

28.12.9 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).

29. True Random Number Generator (TRNG)

29.1 Overview

The true random number generator generates 32-bit random number seeds (which are true random numbers).

The data generated by testing a seed itself and a random number which is generated from a seed (using the continuous random number generator test prescribed in NIST FIPS140-2) are the same by a fixed probability according to the bit length of the two generated random numbers.

The probability that a random number of a comparative target is identical in the nth bit (the theoretical value) is $1/2^n$.

Table 29.1 TRNG specifications

Item	Description
Seed specification	32-bit random number seeds
Operating clock	Peripheral module clock (PCLKB)
Interrupts	Generates the read request signal, TRNG_RDREQ
Module-stop function	Module-stop state can be set to reduce power consumption*1

Note 1. For details, see section 9, Low Power Modes.

29.2 Register Descriptions

29.2.1 TRNGSDR : TRNG Seed Data Register

Base address: TRNG = 0x400D_1000

Offset address: 0x0000

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	Seed Data The seed is generated as 32-bit data. The TRNGSDR register should be read four times when TRNGSCR0.RDRDY = 1. The TRNGSCR0.RDRDY bit is automatically set to 0 by hardware. When TRNGSCR0.RDRDY = 0, the read value is 0x00.	R

Note: To use seed data as a random number, encrypt the read.

29.2.2 TRNGSCR0 : TRNG Seed Command Register 0

Base address: TRNG = 0x400D_1000

Offset address: 0x0002

Bit position:	7	6	5	4	3	2	1	0
Bit field:	RDRDY	—	—	—	SGCEN	SGSTART	—	—
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	SGSTART	Seed Generation Start 0: No effect 1: Start to generate the seed data	W

Bit	Symbol	Function	R/W
3	SGCEN	Seed Generation Circuit Enable 0: Seed generation circuit is disabled 1: Seed generation circuit is enabled	R/W
6:4	—	These bits are read as 0. The write value should be 0.	R/W
7	RDRDY	Read Ready When generating the seed data is completed, the RDRDY bit becomes 1. If SGCEN = 0, this bit value is 0.	R

29.2.3 TRNGSCR1 : TRNG Seed Command Register 1

Base address: TRNG = 0x400D_1000

Offset address: 0x0003

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

Bit	Symbol	Function	R/W
0	INTEN	TRNG Interrupt Enable 0: TRNG interrupt is disabled. 1: TRNG interrupt is enabled.	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
4	—	The read value is undefined. The write value should be 0.	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

29.3 Operation

29.3.1 Overall Processing Flow

Table 29.2 shows the overall processing flow of TRNG activation.

Table 29.2 Procedure for using the True Random Number Generator to generate a random number seed

No	Step Name	Description
1	Module stop setting	Set the MSTPCRC.MSTPC28 = 0 to cancel the module-stop state.
2	Wait	Wait for the peripheral module clock (PCLKB) × 6.
3	TRNG enable setting	Set the TRNGSCR0.SGCEN = 1 to enable the true random number generator.
4	TRNG interrupt setting	Set the TRNGSCR1.INTEN bit to enable/disable the TRNG interrupt output.
5	TRNG operation start setting	Set the TRNGSCR0.SGSTART = 1 to start the generation of a random number seed.
6	Read the seed data	There are 2 operation for TRNG seed generation, Polling and Interrupt. 1. Polling operation ; Read TRNGSDR for 4 times after the TRNGSCR0.RDRDY = 1 2. Interrupt operation ; Read TRNGSDR register for 4 times after TRNG interrupt is generated.
7	TRNG operation stop setting	Set the TRNGSCR0.SGCEN = 0 to disable the true random number generator. Set the TRNGSCR0.SGSTART = 0 to stop the generation of a random number seed.
8	Module stop setting	Set the MSTPCRC.MSTPC28 = 1 to enter the module-stop state.

29.4 Usage Notes

TRNG operation is prohibited for a period of 20 µs before and after MCU operation mode transition.

30. Internal Voltage Regulator

30.1 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.

30.2 Operation

Table 30.1 lists the LDO pin settings, and Figure 30.1 shows the LDO settings.

Table 30.1 LDO pin

Pins	Setting descriptions
VCC	• Connect VCC to the system power supply. • Connect VCC 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 0.47 µF to 1 µF multilayer ceramic capacitor. Place the capacitor close to the pin.

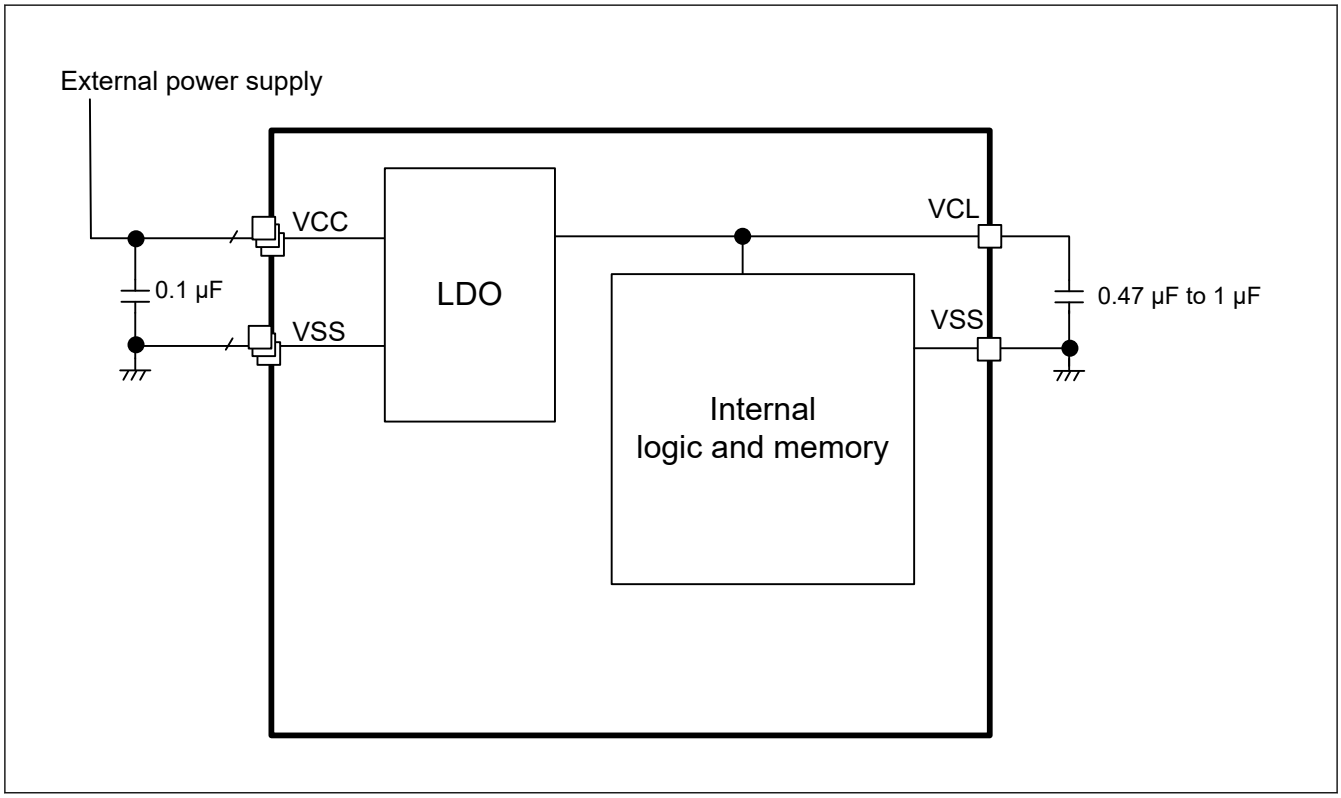


Figure 30.1 LDO settings

31. Electrical Characteristics

Unless otherwise specified, the electrical characteristics of the MCU are defined under the following conditions:

VCC[*1] = VREFH0 = 1.6 to 5.5 V

VSS = VREFL0 = 0 V, Ta = T_{opr}

Note 1. The typical condition is set to VCC = 3.3 V.

Figure 31.1 shows the timing conditions.

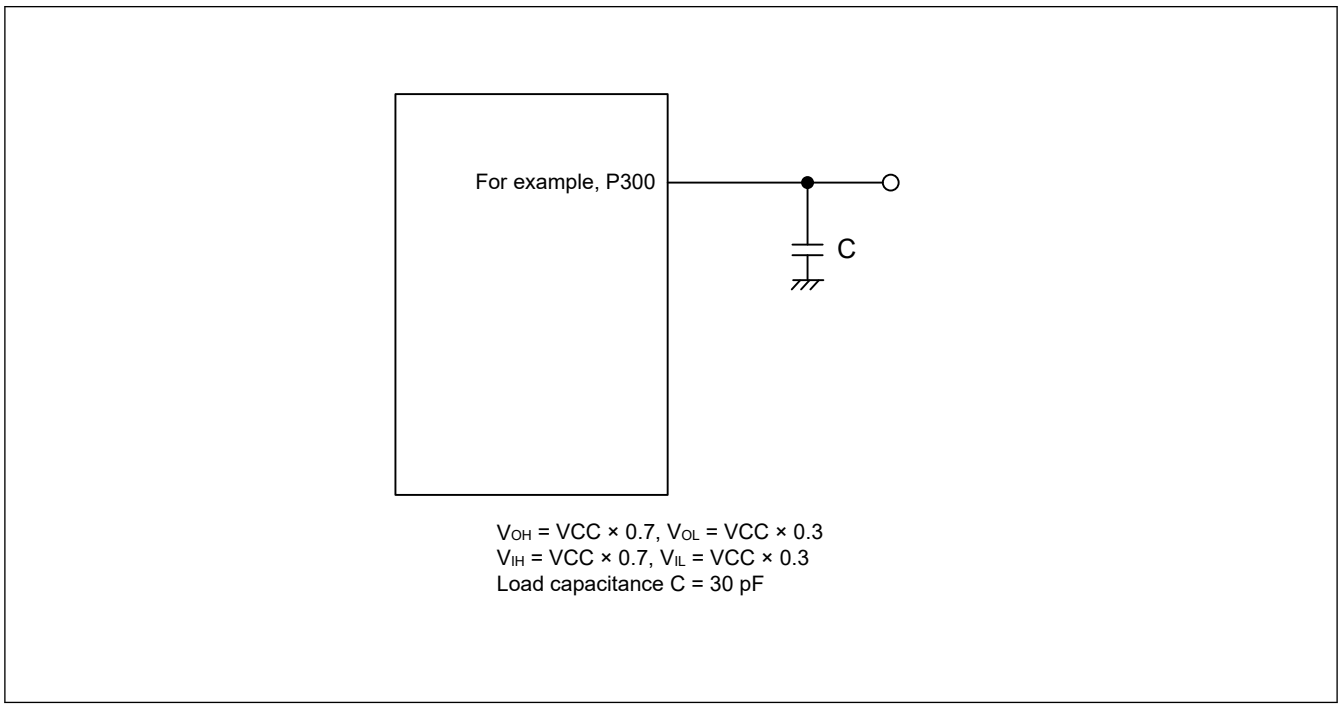


Figure 31.1 Input or output timing measurement conditions

31.1 Absolute Maximum Ratings

Table 31.1 Absolute maximum ratings (1 of 2)

Parameter		Symbol	Value	Unit
Power supply voltage		VCC	-0.5 to +6.5	V
VCL pin input voltage		V_{IVCL}	-0.3 to +2.1 and -0.3 to VCC + 0.3[*1]	V
Input voltage	P100 to P115, P200, P201, P204 to P208, P300 to P304, P402, P403, P407 to P411, P500 to P502, P915, RES	V_{I1}	-0.3 to VCC + 0.3[*2]	V
	P400, P401, P913, P914 (5 V tolerant)	V_{I2}	-0.3 to +6.5	V
	P000 to P004, P008 to P015, P212 to P215	V_{I3}	-0.3 to VCC + 0.3[*2]	V
Output voltage	P100 to P115, P201, P204 to P208, P300 to P304, P402, P403, P407 to P411, P500 to P502, P915	V_{O1}	-0.3 to VCC + 0.3[*2]	V
	P400, P401, P913, P914 (N-ch open-drain)	V_{O2}	-0.3 to +6.5	V
	P000 to P004, P008 to P015, P212, P213	V_{O3}	-0.3 to VCC + 0.3[*2]	V
Analog input voltage	AN000 to AN012	V_{AI1}	-0.3 to VCC + 0.3 and -0.3 to VREFH0 + 0.3[*2 *3]	V
	AN021 to AN022	V_{AI2}	-0.3 to VCC + 0.3 and -0.3 to VREFH0 + 0.3[*2 *3]	V

Table 31.1 Absolute maximum ratings (2 of 2)

Parameter			Symbol	Value	Unit
High-level output current	P100 to P115, P201, P204 to P208, P300 to P304, P402, P403, P407 to P411, P500 to P502, P915	Per pin	I_{OH1}	-40	mA
	P402, P403	Total of all pins		-70	mA
	P100 to P115, P201, P204 to P208, P300 to P304, P407 to P411, P500 to P502, P915			-100	mA
	P000 to P004, P008 to P015, P212, P213	Per pin	I_{OH2}	-5	mA
		Total of all pins		-20	mA
Low-level output current	P100 to P115, P201, P204 to P208, P300 to P304, P400 to P403, P407 to P411, P500 to P502, P913 to P915	Per pin	I_{OL1}	40	mA
	P400 to P403	Total of all pins		70	mA
	P100 to P115, P201, P204 to P208, P300 to P304, P407 to P411, P500 to P502, P913 to P915			100	mA
	P000 to P004, P008 to P015, P212, P213	Per pin	I_{OL2}	10	mA
		Total of all pins		20	mA
Ambient operating temperature	In normal operation mode		Ta	-40 to +105 -40 to +125	°C
	In flash memory programming mode			-40 to +105 -40 to +125	°C
Storage temperature			Tstg	-65 to +150	°C

Note 1. Connect the VCL pin to VSS through a capacitor (0.47 to 1 µF). The listed value is the absolute maximum rating of the VCL pins. Only use the capacitor connection. Do not apply a specific voltage to this pin.
Note 2. This voltage must be no higher than 6.5 V.
Note 3. The voltage on a pin in use for A/D conversion must not exceed VREFH0 + 0.3.

Note: The characteristics of functions multiplexed on a given pin are the same as those for the port pin unless otherwise specified.

Note: VREFH0 refers to the positive reference voltage of the A/D converter.

Note: The reference voltage is VSS.

Caution: Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

Table 31.2 Recommended operating conditions

Parameter	Symbol		Min	Typ	Max	Unit
Power supply voltages	VCC		1.6	—	5.5	V
	VSS		—	0	—	V
Analog power supply voltages	VREFH0	When used as ADC12 Reference	1.6	—	VCC	V
	VREFL0		—	0	—	V

31.1.1 Tj/Ta Definition

Table 31.3 Tj/Ta definition

Conditions: Products with operating temperature Ta = -40 to +125°C

Parameter	Symbol	Typ	Max*1	Unit	Test conditions
Permissible junction temperature	Tj	—	140 125	°C	High-speed mode Middle-speed mode Low-speed mode Subosc-speed mode

Note 1. The upper limit of operating temperature is 105°C or 125°C depending on the product. For details, see section 1.3. Part Numbering. If the part number shows the operation temperature at 105°C, then the maximum value of Tj is 125°C, otherwise it is 140°C.

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.

31.2 Oscillators Characteristics

31.2.1 Main clock Oscillator Characteristics

Table 31.4 Main clock oscillator characteristics

Conditions: VCC = 1.6 to 5.5 V, VSS = 0 V, Ta = -40 to +125°C

Parameter		Min	Typ	Max	Unit	Test conditions
Main clock oscillation allowable input cycle time*1	Ceramic resonator Crystal resonator	0.05	—	1	μs	—

Note 1. The listed time and frequency indicate permissible ranges of the oscillator. For actual applications, request evaluation by the manufacturer of the oscillator circuit mounted on a board so you can use appropriate values. Refer to AC Characteristics for instruction execution time.

Note: Since the CPU is started by the high-speed on-chip oscillator clock after release from the reset state, the user should use the oscillation stabilization time counter status register (OSTC) to check the X1 clock oscillation stabilization time. Specify the values for the oscillation stabilization time in the OSTC register and the oscillation stabilization time select register (OSTS) after having sufficiently evaluated the oscillation stabilization time with the resonator to be used.

31.2.2 Sub-clock Oscillator Characteristics

Table 31.5 Sub-clock oscillator characteristics

Conditions: VCC = 1.6 to 5.5 V, VSS = 0 V, Ta = -40 to +125°C

Parameter		Min	Typ	Max	Unit	Test conditions
Sub-clock oscillation frequency (f_{SOSC})*1	Crystal resonator	—	32.768	—	kHz	—

Note 1. The listed time and frequency indicate permissible ranges of the oscillator. For actual applications, request evaluation by the manufacturer of the oscillator circuit mounted on a board so you can use appropriate values. Refer to AC Characteristics for instruction execution time.

31.2.3 On-chip Oscillators Characteristics

Table 31.6 On-chip oscillators characteristics (1 of 2)

Conditions: VCC = 1.6 to 5.5 V, VSS = 0 V, Ta = -40 to +125°C

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
High-speed on-chip oscillator clock frequency		f_{HOCO}	1	—	32	MHz	—
High-speed on-chip oscillator clock frequency accuracy	OSCSF.HOCOSF = 1	—	-1.0	—	+1.0	%	Ta = -40 to +125°C, 1.6 V ≤ VCC ≤ 5.5 V
	OSCSF.HOCOSF = 0*3	—	-15	—	0	%	—
High-speed on-chip oscillator clock frequency trimming resolution		—	—	0.05	—	%	—

Table 31.6 On-chip oscillators characteristics (2 of 2)

Conditions: VCC = 1.6 to 5.5 V, VSS = 0 V, Ta = -40 to +125°C

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
High-speed on-chip oscillator clock oscillation stabilization time*4	t_{HOCO}	—	—	4.4	µs	—
Middle-speed on-chip oscillator clock frequency*1	f_{MOCO}	1	—	4	MHz	—
Middle-speed on-chip oscillator clock frequency accuracy	—	-12	—	12	%	—
Middle-speed on-chip oscillator clock frequency trimming resolution	—	—	0.15	—	%	—
Middle-speed on-chip oscillator clock oscillation stabilization time	t_{MOCO}	—	—	1	µs	—
Middle-speed on-chip oscillator frequency temperature coefficient	—	—	—	±0.17*2	%/°C	—
Low-speed on-chip oscillator clock frequency*1	f_{LOCO}	—	32.768	—	kHz	—
Low-speed on-chip oscillator clock frequency accuracy	—	-15	—	15	%	—
Low-speed on-chip oscillator clock frequency trimming resolution	—	—	0.3	—	%	—
Low-speed on-chip oscillator clock oscillation stabilization time	t_{LOCO}	—	—	100	µs	—
Low-speed on-chip oscillator frequency temperature coefficient	—	—	—	±0.21*2	%/°C	—

Note 1. The listed values only indicate the characteristics of the oscillators. Refer to AC Characteristics for instruction execution time.
Note 2. These values are the results of characteristic evaluation and are not checked for shipment.
Note 3. The listed condition applies when OFS1.HOCOFRQ1[2:0] = 010b.
Note 4. Check OSCSF.HOCOSF to confirm whether stabilization time has elapsed.

31.3 DC Characteristics

31.3.1 Pin Characteristics

Table 31.7 I/O I_{OH}

Conditions: VCC = 1.6 to 5.5 V, VSS = 0 V, Ta = -40 to +125°C

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Allowable high-level output current*1	Per pin for P100 to P115, P201, P204 to P208, P300 to P304, P402, P403, P407 to P411, P500 to P502, P915	I_{OH1}	—	—	-10*2	mA	1.6 V ≤ VCC ≤ 5.5 V
	Total of P402, P403 (when duty ≤ 70%*3)		—	—	-55*4	mA	4.0 V ≤ VCC ≤ 5.5 V
			—	—	-10	mA	2.7 V ≤ VCC < 4.0 V
			—	—	-5	mA	1.8 V ≤ VCC < 2.7 V
			—	—	-2.5	mA	1.6 V ≤ VCC < 1.8 V
	Total of P100 to P115, P201, P204 to P208, P300 to P304, P407 to P411, P500 to P502, P915 (when duty ≤ 70%*3)		—	—	-80*5	mA	4.0 V ≤ VCC ≤ 5.5 V
			—	—	-19	mA	2.7 V ≤ VCC < 4.0 V
			—	—	-10	mA	1.8 V ≤ VCC < 2.7 V
			—	—	-5	mA	1.6 V ≤ VCC < 1.8 V
	Total of all pins (when duty ≤ 70%*3)		—	—	-135*6	mA	1.6 V ≤ VCC ≤ 5.5 V
	Per pin for P000 to P004, P008 to P015, P212, P213	I_{OH2}	—	—	-3*2	mA	4.0 V ≤ VCC ≤ 5.5 V
			—	—	-1*2	mA	2.7 V ≤ VCC < 4.0 V
			—	—	-1*2	mA	1.8 V ≤ VCC < 2.7 V
			—	—	-0.5*2	mA	1.6 V ≤ VCC < 1.8 V
	Total of all pins (when duty ≤ 70%*3)		—	—	-20	mA	4.0 V ≤ VCC ≤ 5.5 V
			—	—	-10	mA	2.7 V ≤ VCC < 4.0 V
			—	—	-5	mA	1.8 V ≤ VCC < 2.7 V
			—	—	-5	mA	1.6 V ≤ VCC < 1.8 V

Note 1. Device operation is guaranteed at the listed currents even if current is flowing from the VCC pin to an output pin.
Note 2. The combination of these and other pins must also not exceed the value for maximum total current.
Note 3. The listed currents apply when the duty cycle is no greater than 70%. Use the following formula to calculate the output current when the duty cycle is greater than 70%, where n is the duty cycle.

- Total output current from the listed pins = (I_{OH} × 0.7)/(n × 0.01)
  Example when n = 80% and I_{OH} = -10.0 mA
  Total output current from the listed pins = (-10.0 × 0.7)/(80 × 0.01) = -8.75 mA
  Note that the duty cycle has no effect on the current that is allowed to flow into a single pin. A current higher than the absolute maximum rating must not flow into a single pin.

Note 4. The maximum value is -30 mA with an ambient operating temperature range of 85°C to 125°C.
Note 5. The maximum value is -50 mA with an ambient operating temperature range of 85°C to 125°C.
Note 6. The maximum value is -60 mA with an ambient operating temperature range of 85°C to 125°C.

Note: The following pins are not capable of the output of high-level signals in the N-ch open-drain mode.

P100 to P107, P109 to P115, P201, P204 to P208, P212, P213, P301 to P304, P402, P403, P407 to P411, P500 to P502 and P915

Note: The characteristics of functions multiplexed on a given pin are the same as those for the port pin unless otherwise specified.

Table 31.8 I/O I_{OL}

Conditions: VCC = 1.6 to 5.5 V, VSS = 0 V, Ta = -40 to +125°C

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Allowable low-level output current[*1]	Per pin for P100 to P115, P201, P204 to P208, P300 to P304, P402, P403, P407 to P411, P500 to P502, P915	I_{OL1}	—	—	20[*2]	mA	—
	Per pin for P400, P401, P913, P914		—	—	15[*2]	mA	—
	Total of P400 to P403 (when duty ≤ 70%[*3])		—	—	70[*4]	mA	4.0 V ≤ VCC ≤ 5.5 V
			—	—	15	mA	2.7 V ≤ VCC < 4.0 V
			—	—	9	mA	1.8 V ≤ VCC < 2.7 V
			—	—	4.5	mA	1.6 V ≤ VCC < 1.8 V
	Total of P100 to P115, P201, P204 to P208, P300 to P304, P407 to P411, P500 to P502, P913 to P915 (when duty ≤ 70%[*3])		—	—	80[*4]	mA	4.0 V ≤ VCC ≤ 5.5 V
			—	—	35	mA	2.7 V ≤ VCC < 4.0 V
			—	—	20	mA	1.8 V ≤ VCC < 2.7 V
			—	—	10	mA	1.6 V ≤ VCC < 1.8 V
	Total of all pins (when duty ≤ 70%[*3])		—	—	150[*5]	mA	1.6 V ≤ VCC ≤ 5.5 V
	Per pin for P000 to P004, P008 to P015, P212, P213	I_{OL2}	—	—	8.5[*2]	mA	4.0 V ≤ VCC ≤ 5.5 V
			—	—	1.5[*2]	mA	2.7 V ≤ VCC < 4.0 V
			—	—	0.6[*2]	mA	1.8 V ≤ VCC < 2.7 V
			—	—	0.4[*2]	mA	1.6 V ≤ VCC < 1.8 V
	Total of all pins (when duty ≤ 70%[*3])		—	—	20	mA	4.0 V ≤ VCC ≤ 5.5 V
			—	—	20	mA	2.7 V ≤ VCC < 4.0 V
			—	—	15	mA	1.8 V ≤ VCC < 2.7 V
			—	—	10	mA	1.6 V ≤ VCC < 1.8 V

Note 1. Device operation is guaranteed at the listed currents even if current is flowing from an output pin to VSS pin.
Note 2. The combination of these and other pins must also not exceed the value for maximum total current.
Note 3. The listed currents apply when the duty cycle is no greater than 70%. Use the following formula to calculate the output current when the duty cycle is greater than 70%, where n is the duty cycle.

- Total output current from the listed pins = (I_{OL} × 0.7)/(n × 0.01)
  Example when n = 80% and I_{OL} = 10.0 mA
  Total output current from the listed pins = (10.0 × 0.7)/(80 × 0.01) = 8.75 mA
  Note that the duty cycle has no effect on the current that is allowed to flow into a single pin.
  A current higher than the absolute maximum rating must not flow into a single pin.

Note 4. The maximum value is 40 mA with an ambient operating temperature range of 85°C to 125°C.
Note 5. The maximum value is 80 mA with an ambient operating temperature range of 85°C to 125°C.

Note: The characteristics of functions multiplexed on a given pin are the same as those for the port pin unless otherwise specified.

Table 31.9 I/O V_{IH}, V_{IL}

Conditions: VCC = 1.6 to 5.5 V, VSS = 0 V, Ta = -40 to +125°C

Parameter			Symbol	Min	Typ	Max	Unit	Test conditions
Input voltage, high	P100 to P115, P200, P201, P204 to P208, P300 to P304, P402, P403, P407 to P411, P500 to P502, P915, RES	Normal input buffer	V_{IH1}	VCC × 0.8	—	VCC	V	—
	P100 to P115, P201, P204 to P208, P300 to P304, P402, P403, P407 to P411, P500 to P502, P915	TTL input buffer	V_{IH2}	2.2	—	VCC	V	4.0 V ≤ VCC ≤ 5.5 V
				2.0	—	VCC	V	3.3 V ≤ VCC < 4.0 V
				1.5	—	VCC	V	1.6 V ≤ VCC < 3.3 V
	P000 to P004, P008 to P015		V_{IH3}	VCC × 0.7	—	VCC	V	—
	P400, P401, P913, P914		V_{IH4}	VCC × 0.7	—	6.0	V	—
	P212 to P215		V_{IH5}	VCC × 0.8	—	VCC	V	—
Input voltage, low	P100 to P115, P200, P201, P204 to P208, P300 to P304, P402, P403, P407 to P411, P500 to P502, P915, RES	Normal input buffer	V_{IL1}	0	—	VCC × 0.2	V	—
	P100 to P115, P201, P204 to P208, P300 to P304, P402, P403, P407 to P411, P500 to P502, P915	TTL input buffer	V_{IL2}	0	—	0.8	V	4.0 V ≤ VCC ≤ 5.5 V
				0	—	0.5	V	3.3 V ≤ VCC < 4.0 V
				0	—	0.32	V	1.6 V ≤ VCC < 3.3 V
	P000 to P004, P008 to P015		V_{IL3}	0	—	VCC × 0.3	V	—
	P400, P401, P913, P914		V_{IL4}	0	—	VCC × 0.3	V	—
	P212 to P215		V_{IL5}	0	—	VCC × 0.2	V	—

Note: The maximum value of V_{IH} of pins P100 to P107, P109 to P115, P201, P204 to P208, P212, P213, P301 to P304, P402, P403, P407 to P411, P500 to P502 and P915 is VCC, even in the N-ch open-drain mode.

Note: The characteristics of functions multiplexed on a given pin are the same as those for the port pin unless otherwise specified.

Table 31.10 I/O V_{OH}, V_{OL}

Conditions: VCC = 1.6 to 5.5 V, VSS = 0 V, Ta = -40 to +125°C

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Output voltage, high	P100 to P115, P201, P204 to P208, P300 to P304, P402, P403, P407 to P411, P500 to P502, P915	V_{OH1}	VCC - 1.5	—	—	V	4.0 V ≤ VCC ≤ 5.5 V $I_{OH}1$ = -10 mA
			VCC - 0.7	—	—	V	4.0 V ≤ VCC ≤ 5.5 V $I_{OH}1$ = -3 mA
			VCC - 0.6	—	—	V	2.7 V ≤ VCC ≤ 5.5 V $I_{OH}1$ = -2 mA
			VCC - 0.5	—	—	V	1.8 V ≤ VCC ≤ 5.5 V $I_{OH}1$ = -1.5 mA
			VCC - 0.5	—	—	V	1.6 V ≤ VCC ≤ 5.5 V $I_{OH}1$ = -1 mA
	P000 to P004, P008 to P015, P212, P213	V_{OH2}	VCC - 0.7	—	—	V	4.0 V ≤ VCC ≤ 5.5 V $I_{OH}2$ = -3 mA
			VCC - 0.5	—	—	V	2.7 V ≤ VCC < 4.0 V $I_{OH}2$ = -1 mA
			VCC - 0.5	—	—	V	1.8 V ≤ VCC < 2.7 V $I_{OH}2$ = -1 mA
			VCC - 0.5	—	—	V	1.6 V ≤ VCC < 1.8 V $I_{OH}2$ = -0.5 mA
Output voltage, low	P100 to P115, P201, P204 to P208, P300 to P304, P402, P403, P407 to P411, P500 to P502, P915	V_{OL1}	—	—	1.3	V	4.0 V ≤ VCC ≤ 5.5 V $I_{OL}1$ = 20 mA
			—	—	0.7	V	4.0 V ≤ VCC ≤ 5.5 V $I_{OL}1$ = 8.5 mA
			—	—	0.6	V	2.7 V ≤ VCC ≤ 5.5 V $I_{OL}1$ = 3 mA
			—	—	0.4	V	2.7 V ≤ VCC ≤ 5.5 V $I_{OL}1$ = 1.5 mA
			—	—	0.4	V	1.8 V ≤ VCC ≤ 5.5 V $I_{OL}1$ = 0.6 mA
			—	—	0.4	V	1.6 V ≤ VCC ≤ 5.5 V $I_{OL}1$ = 0.3 mA
	P000 to P004, P008 to P015, P212, P213	V_{OL2}	—	—	0.7	V	4.0 V ≤ VCC ≤ 5.5 V $I_{OL}2$ = 8.5 mA
			—	—	0.5	V	2.7 V ≤ VCC < 4.0 V $I_{OL}2$ = 1.5 mA
			—	—	0.4	V	1.8 V ≤ VCC < 2.7 V $I_{OL}2$ = 0.6 mA
			—	—	0.4	V	1.6 V ≤ VCC < 1.8 V $I_{OL}2$ = 0.4 mA
	P400, P401, P913, P914	V_{OL3}	—	—	2.0	V	4.0 V ≤ VCC ≤ 5.5 V $I_{OL}3$ = 15 mA
			—	—	0.4	V	4.0 V ≤ VCC ≤ 5.5 V $I_{OL}3$ = 5 mA
			—	—	0.4	V	2.7 V ≤ VCC ≤ 5.5 V $I_{OL}3$ = 3 mA
			—	—	0.4	V	1.8 V ≤ VCC ≤ 5.5 V $I_{OL}3$ = 2 mA
			—	—	0.4	V	1.6 V ≤ VCC ≤ 5.5 V $I_{OL}3$ = 1 mA

Note: P100 to P107, P109 to P115, P201, P204 to P208, P212, P213, P301 to P304, P402, P403, P407 to P411, P500 to P502 and P915 do not output high-level signals in the N-ch open-drain mode.

Note: The characteristics of functions multiplexed on a given pin are the same as those for the port pin unless otherwise specified.

Table 31.11 I/O other characteristics

Conditions: VCC = 1.6 to 5.5 V, VSS = 0 V, Ta = -40 to +125°C

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Input leakage current, high	P100 to P115, P200, P201, P204 to P208, P300 to P304, P400 to P403, P407 to P411, P500 to P502, P913 to P915, RES	I_{LIH1}	—	—	1	μA	V_I = VCC
	P000 to P004, P008 to P015	I_{LIH2}	—	—	1	μA	V_I = VCC
	P212 to P215	I_{LIH3}	—	—	1	μA	V_I = VCC
Input leakage current, low	P100 to P115, P200, P201, P204 to P208, P300 to P304, P400 to P403, P407 to P411, P500 to P502, P913 to P915, RES	I_{LIL1}	—	—	-1	μA	V_I = VSS
	P000 to P004, P008 to P015	I_{LIL2}	—	—	-1	μA	V_I = VSS
	P212 to P215	I_{LIL3}	—	—	-1	μA	V_I = VSS
On-chip pull-up resistance	P100 to P115, P201, P204 to P208, P212, P213, P300 to P304, P402, P403, P407 to P411, P500 to P502, P915, RES	R_U	10	20	100	kΩ	V_I = VSS In input port
Input capacitance	P200	Cin	—	—	30	pF	Vin = 0 V, f = 1 MHz, Ta = 25°C
	Other input pins		—	—	15		

Note: The characteristics of functions multiplexed on a given pin are the same as those for the port pin unless otherwise specified.

31.3.2 Operating and Standby Current

Table 31.12 Operating and standby current (1) (1 of 2)

Conditions: VCC = 1.6 to 5.5 V

Parameter					Symbol	Typ*5	Max	Unit	Test Conditions
Supply current*1	High-speed mode*2	Normal mode	All peripheral clocks disabled, CoreMark code executing from flash	ICLK = 32 MHz	Icc	2.8	—	mA	—
			All peripheral clocks enabled, CoreMark code executing from flash*6	ICLK = 32 MHz		—	5.1		—
		Sleep mode	All peripheral clocks disabled	ICLK = 32 MHz		0.89	—		—
			All peripheral clocks enabled*6	ICLK = 32 MHz		—	2.8		—
	Middle-speed mode*2	Normal mode	All peripheral clocks disabled, CoreMark code executing from flash	ICLK = 24 MHz		2.1	—		—
				ICLK = 16 MHz		1.6	—		—
				ICLK = 8 MHz		1.0	—		—
				ICLK = 4 MHz		0.70	—		—
			All peripheral clocks enabled, CoreMark code executing from flash*6	ICLK = 24 MHz		—	3.8		—
				ICLK = 16 MHz		—	2.8		—
				ICLK = 8 MHz		—	1.6		—
				ICLK = 4 MHz		—	1.1		—
		Sleep mode	All peripheral clocks disabled	ICLK = 24 MHz		0.73	—		—
				ICLK = 16 MHz		0.64	—		—
				ICLK = 8 MHz		0.52	—		—
				ICLK = 4 MHz		0.46	—		—
			All peripheral clocks enabled*6	ICLK = 24 MHz		—	2.2		—
				ICLK = 16 MHz		—	1.7		—
				ICLK = 8 MHz		—	1.1		—
				ICLK = 4 MHz		—	0.8		—
	Low-speed mode*3	Normal mode	All peripheral clocks disabled, CoreMark code executing from flash	ICLK = 2 MHz		189	—	µA	—
			All peripheral clocks enabled, CoreMark code executing from flash*6	ICLK = 2 MHz		—	332		—
		Sleep mode	All peripheral clocks disabled	ICLK = 2 MHz		52	—		—
			All peripheral clocks enabled*6	ICLK = 2 MHz		—	167		—

Table 31.12 Operating and standby current (1) (2 of 2)

Conditions: VCC = 1.6 to 5.5 V

Parameter						Symbol	Typ*5	Max	Unit	Test Conditions
Supply current*1	Subosc-speed mode*4	Normal mode	Peripheral clocks disabled	ICLK = 32.768 kHz	Ta = -40°C	Icc	3.2	—	µA	—
					Ta = 25°C		3.5	—		
					Ta = 50°C		3.8	—		
					Ta = 70°C		4.2	—		
					Ta = 85°C		4.7	—		
					Ta = 105°C		6.3	—		
					Ta = 125°C		9.7	—		
			Peripheral clocks enabled*7	ICLK = 32.768 kHz	Ta = -40°C		—	7.1		
					Ta = 25°C		—	7.5		
					Ta = 50°C		—	9.6		
					Ta = 70°C		—	14		
					Ta = 85°C		—	22		
					Ta = 105°C		—	40		
					Ta = 125°C		—	89		
		Sleep mode	Peripheral clocks disabled	ICLK = 32.768 kHz	Ta = -40°C		0.9	—		—
					Ta = 25°C		1.1	—		
					Ta = 50°C		1.3	—		
					Ta = 70°C		1.5	—		
					Ta = 85°C		1.9	—		
					Ta = 105°C		3.1	—		
					Ta = 125°C		5.4	—		
			Peripheral clocks enabled*7	ICLK = 32.768 kHz	Ta = -40°C		—	4.6		
					Ta = 25°C		—	4.9		
					Ta = 50°C		—	7.0		
					Ta = 70°C		—	11		
					Ta = 85°C		—	18		
					Ta = 105°C		—	36		
					Ta = 125°C		—	84		

Note 1. Supply current is the total current flowing into VCC. 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 clock source is high-speed on-chip oscillator (HOCO).
Note 3. The clock source is middle-speed on-chip oscillator (MOCO).
Note 4. The clock source is the Sub-clock oscillator (SOSC) and CMC.SODRV[1:0] are 10b (Low power mode 2).
Note 5. VCC = 3.3 V.
Note 6. Includes operating current for PCLBUZ, TAU, SAU, and IICA functions only. For other peripheral operating currents, please add the current in Peripheral Functions Supply current in Table 31.14.
Note 7. Includes operating current for PCLBUZ, TAU and SAU functions only. For other peripheral operating currents, please add the current in Peripheral Functions Supply current in Table 31.14.

Table 31.13 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	PSMCR.RAMSD[1:0] are 00b	All SRAMs (0x2000_4000 to 0x2000_7FFF) are on	Ta = -40°C	Icc	0.20	1.2	µA	—
					Ta = 25°C		0.25	1.2		
					Ta = 50°C		0.35	3.0		
					Ta = 70°C		0.60	7.0		
					Ta = 85°C		0.95	14		
					Ta = 105°C		2.2	32		
					Ta = 125°C		4.6	80		
			PSMCR.RAMSD[1:0] are 11b	Only 8 KB SRAM (0x2000_4000 to 0x2000_5FFF) is on	Ta = -40°C		0.20	1.2		—
					Ta = 25°C		0.25	1.2		
					Ta = 50°C		0.35	3.0		
					Ta = 70°C		0.55	6.5		
					Ta = 85°C		0.90	13		
					Ta = 105°C		2.0	28		
					Ta = 125°C		4.3	75		

Note 1. Supply current is the total current flowing into VCC. 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.

Table 31.14 Peripheral Functions Supply current

Conditions: VCC = 1.6 to 5.5 V

Parameter				Symbol	Typ*12	Max	Unit	Test conditions
Peripheral Functions Supply current*1	High-speed on chip oscillator operating current*1		OFS1.HOCOFRQ1[2:0] are 010b	I_{HOCO}	320	—	μA	—
	Middle-speed on chip oscillator operating current*1			I_{MOCO}	20	—	μA	—
	Low-speed on chip oscillator operating current*1			I_{LOCO}	0.24	—	μA	—
	Main-clock oscillator	CMC.MODRV = 0	f_{MOCO} = 10 MHz	I_{MOSC}	160	—	μA	—
		CMC.MODRV = 1	f_{MOCO} = 20 MHz		330	—	μA	—
	Sub-clock oscillator	SBYCR.RTCLPC is 1	CMC.SODRV[1:0] are 11b (Low power mode 3)	I_{SOSC}	0.13	—	μA	—
			CMC.SODRV[1:0] are 10b (Low power mode 2)		0.34	—	μA	—
			CMC.SODRV[1:0] are 00b (Low power mode 1)		0.49	—	μA	—
			CMC.SODRV[1:0] are 01b (Normal mode)		0.62	—	μA	—
		SBYCR.RTCLPC is 0	CMC.SODRV[1:0] are 11b (Low power mode 3)		0.30	—	μA	
			CMC.SODRV[1:0] are 10b (Low power mode 2)		0.51	—	μA	
			CMC.SODRV[1:0] are 00b (Low power mode 1)		0.65	—	μA	
			CMC.SODRV[1:0] are 01b (Normal mode)		0.80	—	μA	
	RTC*1*2*3	RTCC0.RTC128EN is 0		I_{RTC}	0.006	—	μA	—
		RTCC0.RTC128EN is 1			0.001	—	μA	—
	32-bit interval timer operating current*1*2*4			I_{IT}	0.06	—	μA	—
	Independent watchdog timer operating current*1*2*5		f_{LOCO} = 32.768 kHz (typ.)	I_{IWDT}	0.03	—	μA	—
	A/D converter operating current*1*6	When conversion at maximum speed	Normal mode, VREFH0 = VCC = 5.0 V	I_{ADC}	0.85	1.6	mA	—
			Low voltage mode, VREFH0 = VCC = 3.0 V		0.46	0.75	mA	—
	VREFH0 current*7		VREFH0 = 5.0 V	I_{ADREF}	68	—	μA	—
	A/D converter internal reference voltage current*1			I_{ADREF}	86	—	μA	—
	Temperature sensor operating current*1			I_{TMPS}	100	—	μA	—
	LVD operating current*1		LVD0 is enabled*8	I_{LVD0}	0.03	—	μA	—
			LVD1 is enabled*9	I_{LVD1}	0.03	—	μA	—
	Self-programming operating current*1*10			I_{FSP}	—	12.2	mA	—
	Data flash rewrite operating current*1*11			I_{BGO}	—	12.2	mA	—
	Operating current of the true random number generator*1			I_{TRNG}	1.1	—	mA	—
	DTC		Data transfer to RAM	I_{DTC}	1.82	—	mA	—

Note 1. This current flows into V_{CC}.

Note 2. The listed currents apply when the high-speed on-chip oscillator (HOCO), middle-speed on-chip oscillator (MOCO), and Main clock oscillator (MOSC) are stopped.

Note 3. This current flows into the realtime clock (RTC). It does not include the operating current of the low-speed on-chip oscillator (LOCO) or the Sub-clock oscillator (SOSC).
The supply current of the RA0 microcontrollers is the sum of either Icc, and I_{RTC}.

When the low-speed on-chip oscillator (LOCO) is selected, I_{LOCO} should be included in the supply current.
When the Sub-clock oscillator (SOSC) is selected, I_{SOSC} should be included in the supply current.

Note 4. This current only flows to the 32-bit interval timer. It does not include the operating current of the low-speed on-chip oscillator (LOCO) or Sub-clock oscillator (SOSC).
The supply current of the RA0 microcontrollers is the sum of either Icc and I_{IT}.
When the low-speed on-chip oscillator (LOCO) is selected, I_{LOCO} should be included in the supply current.
When the Sub-clock oscillator (SOSC) is selected, I_{SOSC} should be included in the supply current.

Note 5. This current only flows to the independent watchdog timer. It does not include the operating current of the low-speed on-chip oscillator (LOCO) .
The supply current of the RA0 microcontrollers is the sum of either Icc, I_{IWDT} and I_{LOCO}.

Note 6. This current only flows to the A/D converter. The supply current of the RA0 microcontrollers is the sum of Icc and I_{ADC} when the A/D converter is operating or in the SLEEP mode.

Note 7. This current flows into VREFH0.

Note 8. This current only flows to the LVD0 circuit. The supply current of the RA0 microcontrollers is the sum of Icc and I_{LVD0} when the LVD0 circuit is in operation.

Note 9. This current only flows to the LVD1 circuit. The supply current of the RA0 microcontrollers is the sum of Icc and I_{LVD1} when the LVD1 circuit is in operation.

Note 10. This current only flows during self programming.

Note 11. This current only flows while the data flash memory is being rewritten.

Note 12. VCC = 3.3 V.

31.3.3 Thermal Characteristics

The maximum value of junction temperature (Tj) must not exceed the value specified in the section 31.1.1. Tj/Ta Definition.

Tj is calculated by either of the following equations.

- Tj = Ta + θja × Total power consumption
- Tj = Tt + Ψjt × Total power consumption
  Tj : Junction Temperature (°C)
  Ta : Ambient Temperature (°C)
  Tt : Top Center Case Temperature (°C)
  θja : Thermal Resistance of “Junction”-to-“Ambient” (°C/W)
  Ψjt : Thermal Resistance of “Junction”-to-“Top Center Case” (°C/W)
- Total power consumption = Voltage × (Leakage current + Dynamic current)
- Leakage current of IO = Σ (I_{OL} × V_{OL}) / Voltage + Σ (|I_{OH}| × |VCC – V_{OH}|) / Voltage
- Dynamic current of IO = Σ IO (Cin + Cload) × IO switching frequency × Voltage
  Cin: Input capacitance
  Cload: Output capacitance

Regarding θja and Ψjt, see Table 31.15.

Table 31.15 Thermal resistance

Parameter	Package	Symbol	Value*1	Unit	Test condition
Thermal resistance	64-pin LFQFP	θja	57.0	°C/W	JESD 51-2 and 51-7 compliant
	48-pin LFQFP		65.9		
	48-pin HWQFN		20.2		
	32-pin LQFP		65.6		
	32-pin HWQFN		23.8		
	64-pin LFQFP	Ψjt	4.02	°C/W	JESD 51-2 and 51-7 compliant
	48-pin LFQFP		6.26		
	48-pin HWQFN		0.28		
	32-pin LQFP		6.58		
	32-pin HWQFN		0.32		

Note 1. The values are reference values when the 4-layer board is used. Thermal resistance depends on the number of layers or size of the board. For details, refer to the JEDEC standards.

31.4 AC Characteristics

Table 31.16 AC characteristics

Conditions: VCC = 1.6 to 5.5 V, VSS = 0 V, Ta = -40 to +125°C

Parameter			Symbol	Min	Typ	Max	Unit	Test conditions
Instruction cycle (minimum instruction execution time)	Main system clock (FMAIN) operation	High-speed mode	T_{CY}	0.03125	—	1	µs	1.8 V ≤ VCC ≤ 5.5 V
				0.25	—	1	µs	1.6 V ≤ VCC < 1.8 V
		Middle-speed mode		0.04167	—	1	µs	1.8 V ≤ VCC ≤ 5.5 V
				0.25	—	1	µs	1.6 V ≤ VCC < 1.8 V
		Low-speed mode		0.5	—	1	µs	1.6 V ≤ VCC ≤ 5.5 V
	Subsystem clock (FSUB) operation			26.041	30.5	31.3	µs	1.6 V ≤ VCC ≤ 5.5 V
	In the self-programming mode	High-speed mode		0.03125	—	1	µs	1.8 V ≤ VCC ≤ 5.5 V
		Middle-speed mode		0.04167	—	1	µs	1.8 V ≤ VCC ≤ 5.5 V
External system clock frequency			f_{EX}	1.0	—	20.0	MHz	1.8 V ≤ VCC ≤ 5.5 V
				1.0	—	4.0	MHz	1.6 V ≤ VCC < 1.8 V
External system clock input high-level width, low-level width			t_{EXH} t_{EXL}	24	—	—	ns	1.8 V ≤ VCC ≤ 5.5 V
				120	—	—	ns	1.6 V ≤ VCC < 1.8 V
TI00 to TI07 input high-level width, low-level width			t_{TIH} t_{TIL}	$1/f_{MCK}$ +10[*1]	—	—	ns	
TO00 to TO07 output frequency		High-speed mode Middle-speed mode	f_{TO}	—	—	16[*2]	MHz	4.0 V ≤ VCC ≤ 5.5 V
				—	—	8	MHz	2.7 V ≤ VCC < 4.0 V
				—	—	4	MHz	1.8 V ≤ VCC < 2.7 V
				—	—	2	MHz	1.6 V ≤ VCC < 1.8 V
		Low-speed mode		—	—	2	MHz	1.6 V ≤ VCC ≤ 5.5 V
PCLBUZ0, PCLBUZ1 output frequency		High-speed mode Middle-speed mode	f_{PCL}	—	—	16[*2]	MHz	4.0 V ≤ VCC ≤ 5.5 V
				—	—	8	MHz	2.7 V ≤ VCC < 4.0 V
				—	—	4	MHz	1.8 V ≤ VCC < 2.7 V
				—	—	2	MHz	1.6 V ≤ VCC < 1.8 V
		Low-speed mode		—	—	2	MHz	1.6 V ≤ VCC ≤ 5.5 V
Interrupt input high-level width, low-level width		NMI/IRQ0, IRQ1 to IRQ7	f_{IRQH} f_{IRQL}	1	—	—	µs	1.6 V ≤ VCC ≤ 5.5 V

Note 1. f_{MCK}: Timer array unit operating clock frequency
To set this operating clock, use the CKS[1:0] bits of the timer mode register 0n (TMR0n).
m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Note 2. The maximum value is 12MHz with an ambient operating temperature range of 105°C to 125°C.

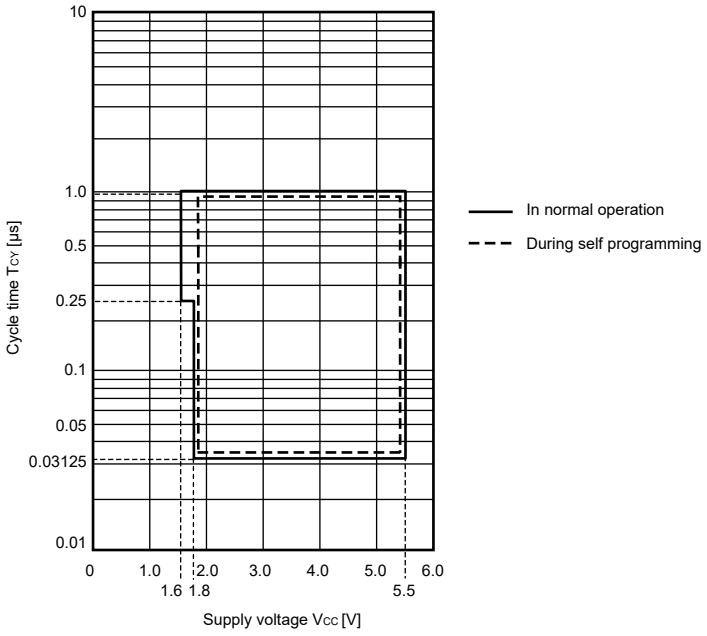


Figure 31.2 T_{CY} vs V_{CC} in High-speed mode

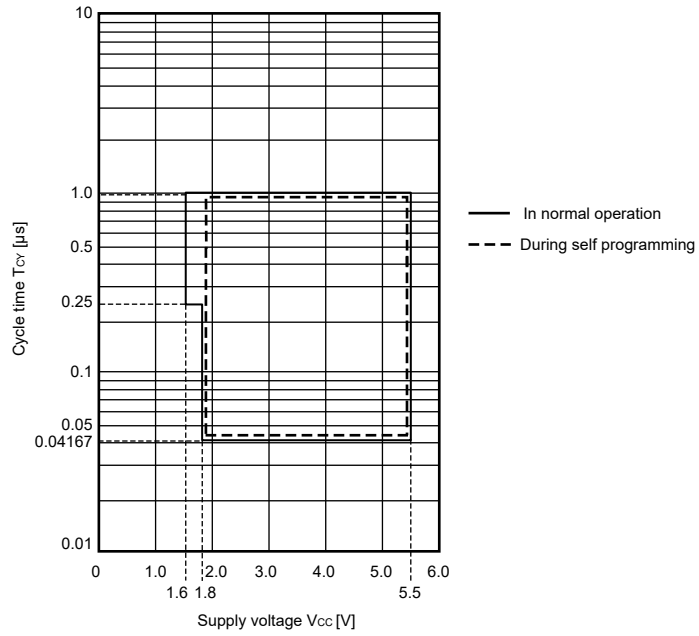


Figure 31.3 T_{CY} vs V_{CC} in Middle-speed mode

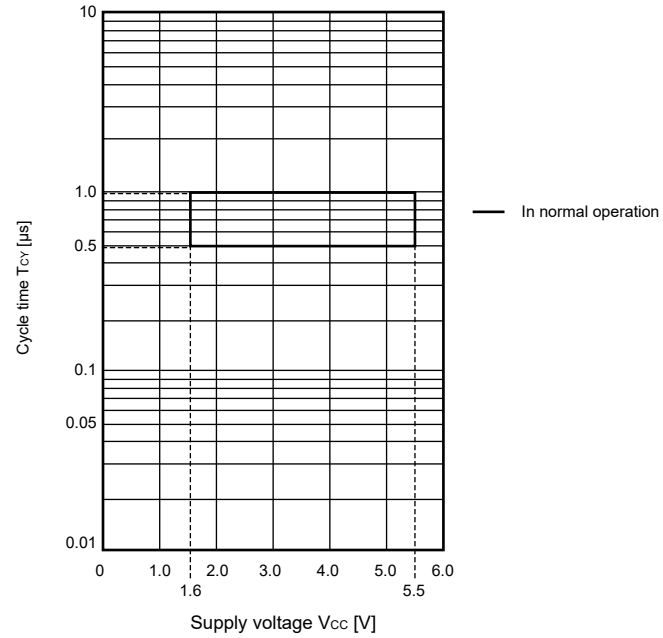


Figure 31.4 T_{CY} vs V_{CC} in Low-speed mode

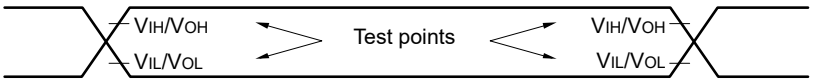


Figure 31.5 AC timing test points

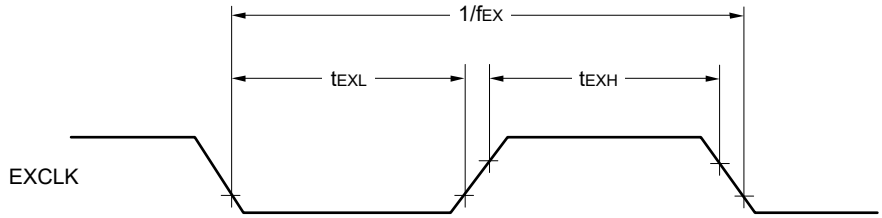


Figure 31.6 External system clock timing

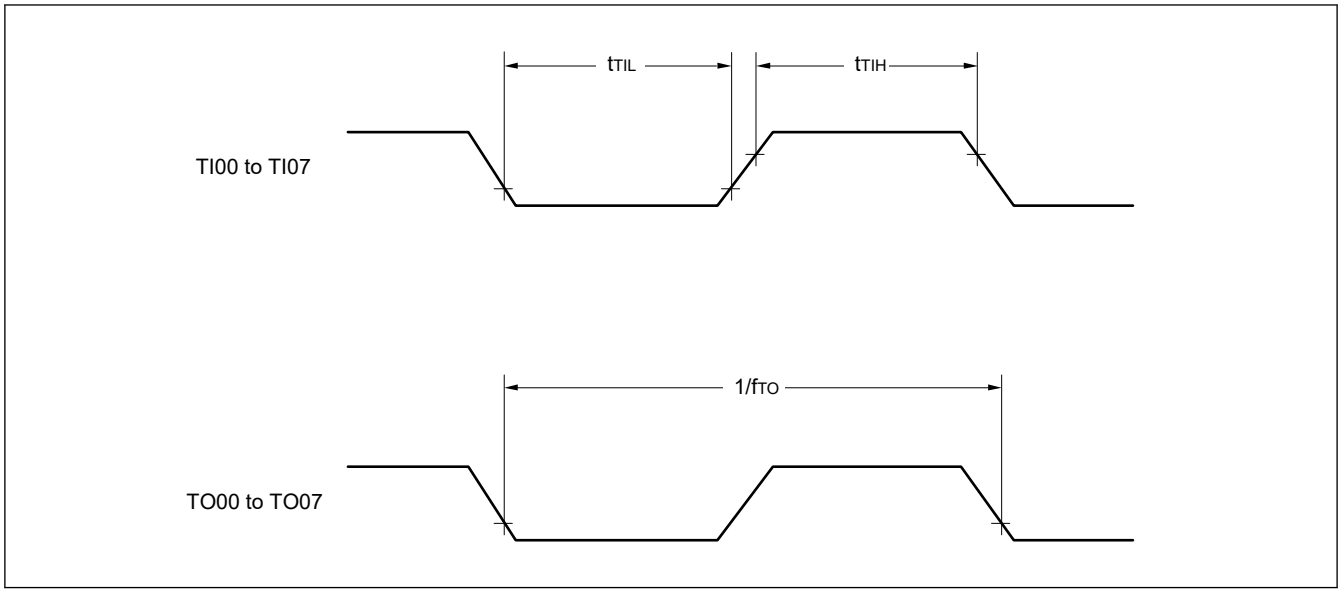


Figure 31.7 TI/TO timing

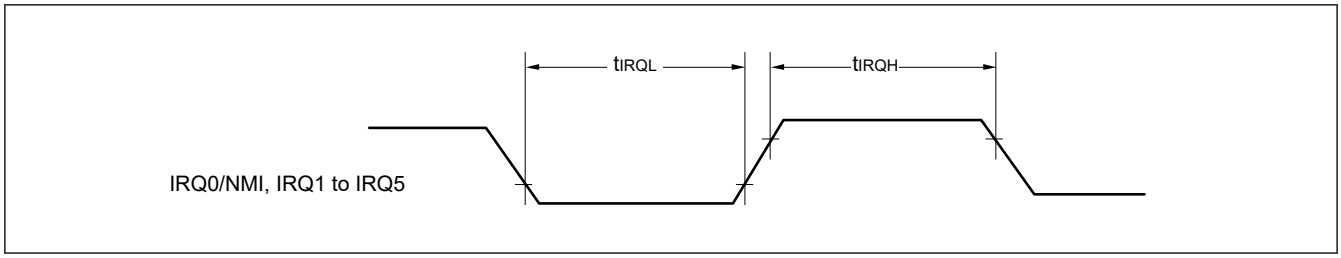


Figure 31.8 IRQ interrupt input timing

31.4.1 Reset Timing

Table 31.17 Reset timing

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
RES pulse width	At power-on*3	t_{RESWP}	9.9	—	—	ms	—
	Not at power-on	t_{RESW}	10	—	—	µs	—
Wait time after RES cancellation (at power-on)	LVD0 enabled*1	t_{RESWT}	—	0.506	0.694	ms	—
	LVD0 disabled*2		—	0.201	0.335	ms	—
Wait time after RES cancellation (during powered-on state)	LVD0 enabled*1	t_{RESWT2}	—	0.476	0.616	ms	—
	LVD0 disabled*2		—	0.170	0.257	ms	—
Internal reset by Independent watch dog timer reset, SRAM parity error reset, software reset		t_{RESW2}	—	0.04	0.041	ms	—

Note 1. When OFS1.LVDAS = 0.
Note 2. When OFS1.LVDAS = 1.
Note 3. When RES pin is not used as the external reset input, this specification can be ignore.

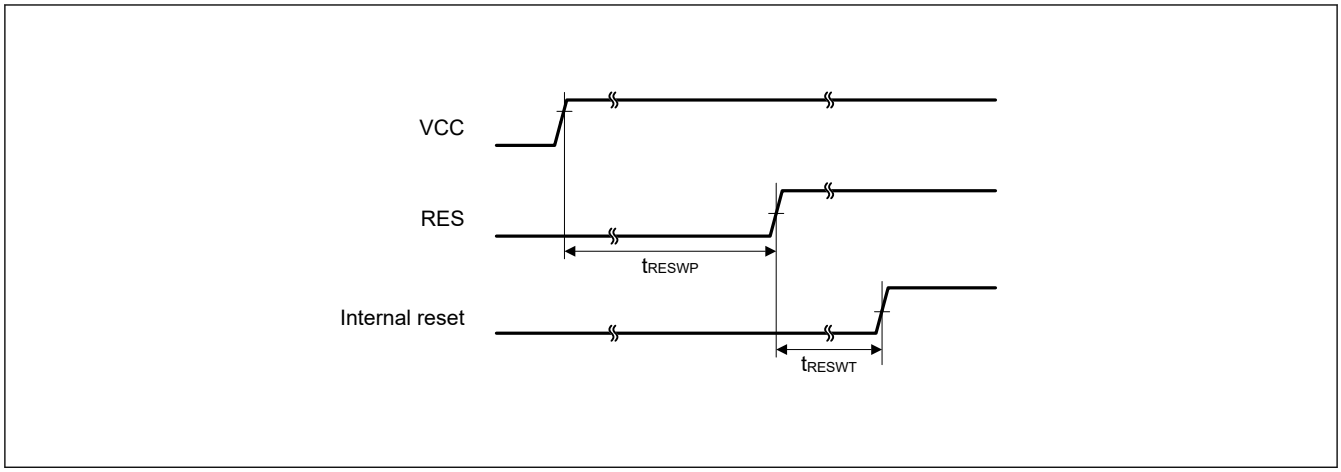


Figure 31.9 Reset input timing at power-on

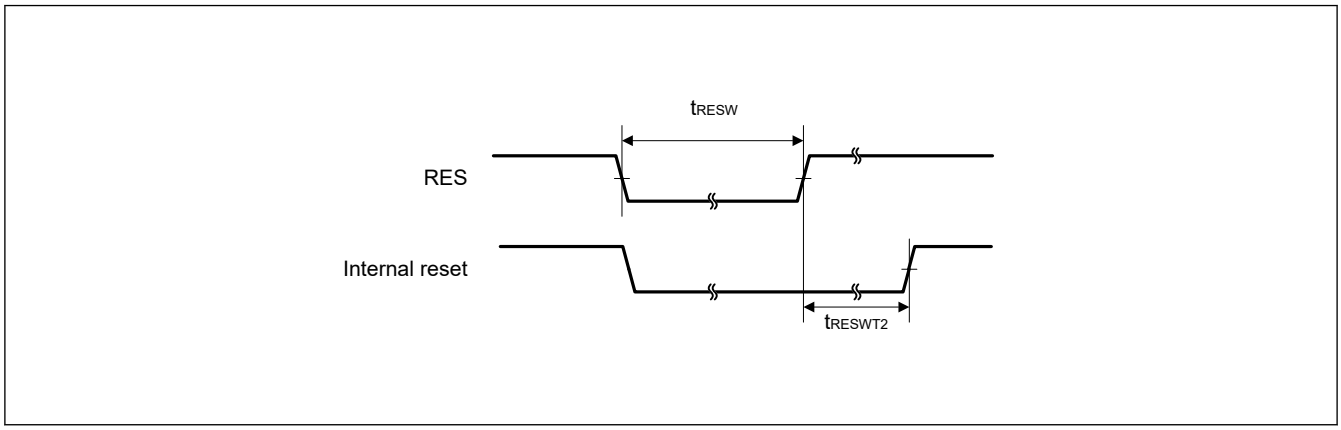


Figure 31.10 Reset input timing (1)

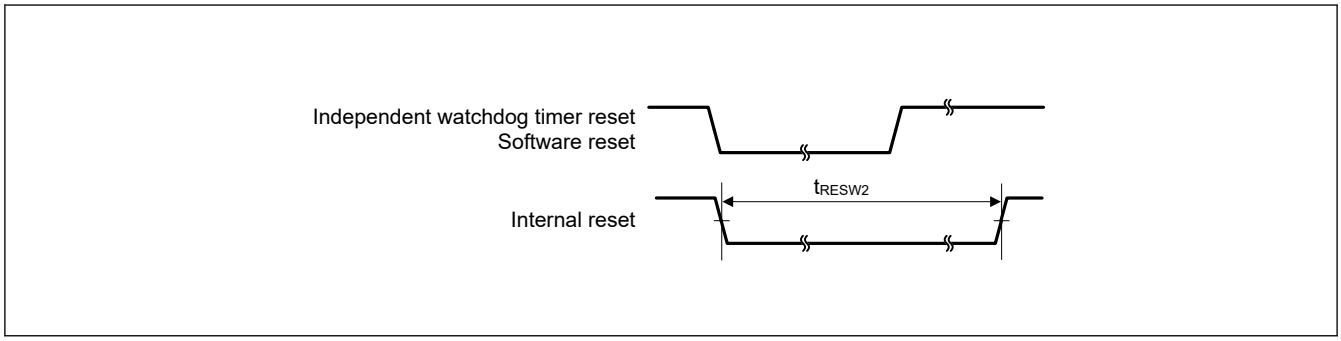


Figure 31.11 Reset input timing (2)

31.4.2 Wakeup Time

Table 31.18 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	System clock source is HOCO	System clock source is HOCO (32 MHz) VCC = 1.8 V to 5.5 V	t_{SBYHO}	—	4.9	5.5	µs	Figure 31.12
			System clock source is HOCO (4 MHz) VCC = 1.6 V to 1.8 V		—	6.6	7.3	µs	

Note 1. The division ratio of ICLK is the minimum division ratio within the allowable frequency range. The recovery time is determined by the system clock source.

Table 31.19 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	System clock source is HOCO	System clock source is HOCO (24 MHz) VCC = 1.8 V to 5.5 V	t_{SBYHO}	—	4.9	5.5	µs	Figure 31.12
			System clock source is HOCO (3 MHz) VCC = 1.6 V to 1.8 V		—	7.4	8.1	µs	

Note 1. The division ratio of ICLK is the minimum division ratio within the allowable frequency range. The recovery time is determined by the system clock source.

Table 31.20 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	System clock source is HOCO (2 MHz)	t_{SBYHO}	—	9.1	9.9	µs	Figure 31.12

Note 1. The division ratio of ICLK is the minimum division ratio within the allowable frequency range. The recovery time is determined by the system clock source.

Table 31.21 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 LOCO (32.768 kHz)	t_{SBYLO}	—	0.29	0.36	ms	Figure 31.12

Note 1. The LOCO itself continues oscillating in Software Standby mode during Subosc-speed mode.

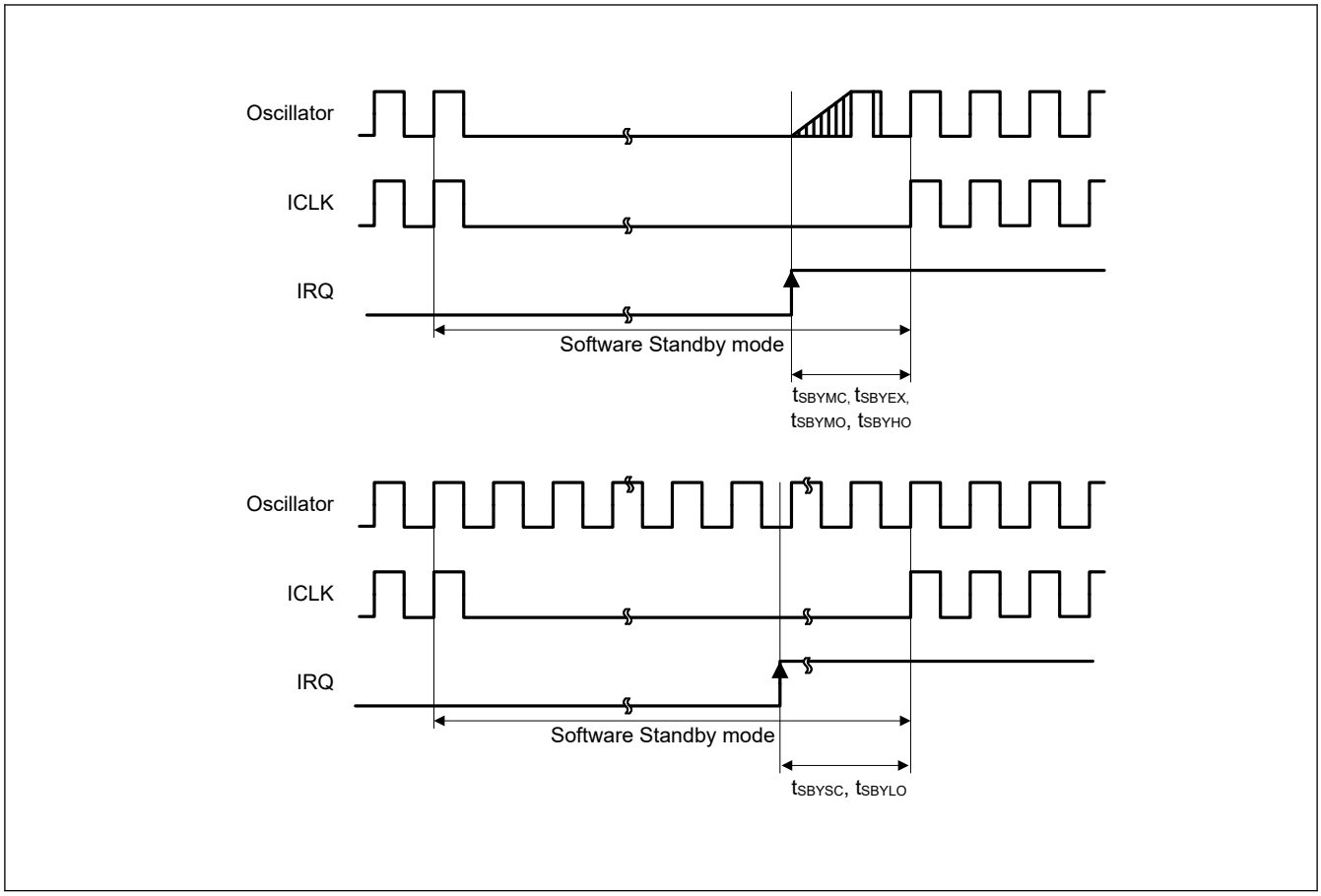
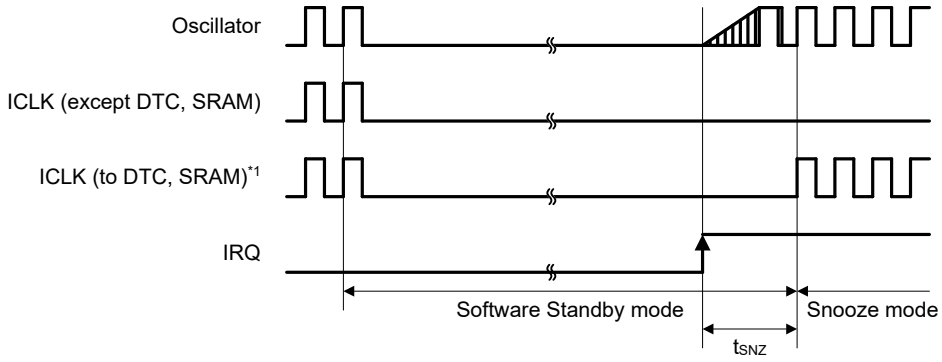


Figure 31.12 Software Standby mode cancellation timing

Table 31.22 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	SBYCR.FWKUP = 0	t_{SNZ}	—	4.1	4.4	µs	Figure 31.13
		SBYCR.FWKUP = 1		—	0.9	1.0	µs	
	Middle-speed mode System clock source is HOCO (24 MHz) VCC = 1.8 V to 5.5 V		t_{SNZ}	—	4.2	4.4	µs	
	Middle-speed mode System clock source is HOCO (3 MHz) VCC = 1.6 V to 1.8 V		t_{SNZ}	—	4.8	5.3	µs	
	Low-speed mode System clock source is MOCO (2 MHz)		t_{SNZ}	—	4.0	5.4	µs	



Note 1. When SNZCR.SNZDTCEN bit is set to 1, ICLK is supplied to DTC and SRAM.

Figure 31.13 Recovery timing from Software Standby mode to Snooze mode

31.5 Peripheral Function Characteristics

31.5.1 Serial Array Unit (SAU)

Table 31.23 In UART communications with devices operating at the same voltage levels

Conditions: VCC = 1.6 to 5.5 V, VSS = 0 V, Ta = -40 to +125°C

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 31.15
		Theoretical value of the maximum transfer rate f_{MCK} = PCLKB*2		—	5.3	—	4	—	0.33	Mbps	

Note 1. The transfer rate in SNOOZE mode is within the range from 4800 to 9600 bps when SBYCR.FWKUP = 0, and within the range from 4800 to 115200 bps when SBYCR.FWKUP = 1.

Note 2. The maximum operating frequencies of the peripheral module clock (PCLKB) are as follows.
High-speed mode: 32 MHz (1.8 V ≤ VCC ≤ 5.5 V), 4 MHz (1.6 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: 2 MHz (1.6 V ≤ VCC ≤ 5.5 V)

Note: Select the normal input buffer for the RXDq pin and the normal output mode for the TXDq pin by using the Port gh Pin Function Select Register (PghPFS_A.PIM and PghPFS_A.NCODR).

gh: Port number (gh = 100, 101, 109, 110, 212, 213, 402, 403, 501, 502)

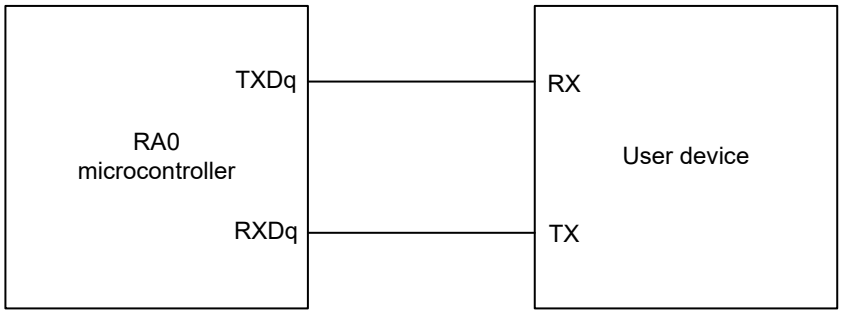


Figure 31.14 Connection in the UART communications with devices operating at the same voltage levels

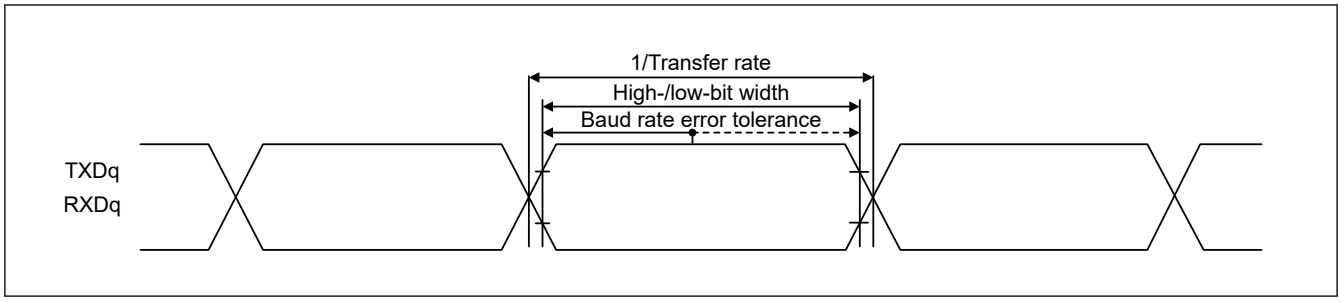


Figure 31.15 Bit width in the UART communications when interfacing devices operate at the same voltage level (reference)

Note:
- q: UART number (q = 0 to 2), gh: Port number (gh = 100, 101, 109, 110, 212, 213, 402, 403, 501, 502)
- f_{MCK}: Serial array unit operation clock frequency
  To set this operating clock, set the CKS bit in the serial mode register mn (SMRmn).
  m: Unit number, n: Channel number (mn = 00 to 03, 10, 11)

Table 31.24 In simplified SPI communications in the master mode with devices operating at the same voltage levels with the internal SCKp clock (the ratings below are only applicable to SPI00)

Conditions: VCC = 2.7 to 5.5 V, VSS = 0 V, Ta = -40 to +85°C

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 31.17 Figure 31.18
		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			$t_{KCY1}/2 - 10$	—	$t_{KCY1}/2 - 15$	—	$t_{KCY1}/2 - 50$	—	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 1. The setting applies when SCRmn.DCP0[1:0] = 00b or 11b. The setting for the SIp setup time becomes to SCKp↓ and that for the SIp hold time becomes from SCKp↓ when SCRmn.DCP0[1:0] = 01b or 10b.
Note 2. This setting applies when SCRmn.DCP0[1:0] = 00b or 11b. The setting for the delay time to SOp output becomes from SCKp↑ when SCRmn.DCP0[1:0] = 01b or 10b.
Note 3. C is the load capacitance of the SCKp and SOp output lines.

Note: Select the normal input buffer for the SIp pin and the normal output mode for the SOp pin and SCKp pin by using the Port gh Pin Function Select Register (PghPFS_A.PIM and PghPFS_A.NCODR).

Note:
- The listed times are only valid when the peripheral I/O redirect function of SPI00 is not in use.
- p: Simplified SPI number (p = 00), m: Unit number (m = 0), n: Channel number (n = 0), gh: Port number (gh = 100 to 103, 112, 201, 500 to 502)
- f_{MCK}: Serial array unit operation clock frequency
  To set this operating clock, use the CKS bit in the serial mode register mn (SMRmn).
  m: Unit number, n: Channel number (mn = 00)

Table 31.25 In simplified SPI communications in the master mode with devices operating at the same voltage levels with the internal SCKp clock

Conditions: VCC = 1.6 to 5.5 V, VSS = 0 V, Ta = -40 to +125°C

Parameter			Symbol	High-speed mode		Middle-speed mode		Low-speed mode		Unit	Test Conditions
				Min.	Max.	Min.	Max.	Min.	Max.		
SCKp cycle time	$t_{KCY1} \geq 4/PCLKB$	2.7 V ≤ VCC ≤ 5.5 V	t_{KCY1}	125	—	166	—	2000	—	ns	Figure 31.17 Figure 31.18
		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	—	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			$t_{KCY1}/2 - 100$	—	$t_{KCY1}/2 - 100$	—	$t_{KCY1}/2 - 100$	—	ns	
SIp setup time (to SCKp↑)*1	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	—	220	—	ns	
SIp hold time (from SCKp↑)*1	1.6 V ≤ VCC ≤ 5.5 V		t_{KSI1}	19	—	19	—	19	—	ns	
Delay time from SCKp↓ to SOp output*2	1.6 V ≤ VCC ≤ 5.5 V C = 30 pF*3		t_{KSO1}	—	25	—	25	—	25	ns	

Note 1. This setting applies when SCRmn.DCP[1:0] = 00b or 11b. The setting for the SIp setup time becomes to SCKp↓ and that for the SIp hold time becomes 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 becomes from SCKp↑ when SCRmn.DCP[1:0] = 01b or 10b.

Note 3. C is the load capacitance of the SCKp and SOp output lines.

Note: Select the normal input buffer for the SIp pin and the normal output mode for the SOp pin and SCKp pin by using the Port gh Pin Function Select Register (PghPFS_A.PIM and PghPFS_A.NCODR).

Note:
- p: Simplified SPI number (p = 00, 01, 10, 11, 20, 21), m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), gh: Port number (gh = 100 to 106, 109, 110, 112 to 115, 201, 204 to 208, 212, 213, 301 to 303, 402, 403, 407, 409 to 411, 500 to 502, 915)
- f_{MCK}: Serial array unit operation clock frequency
  To set this operating clock, use the CKS bit in the serial mode register mn (SMRmn).
  m: Unit number, n: Channel number (mn = 00 to 03, 10, 11)

Table 31.26 In simplified SPI communications in the slave mode with devices operating at the same voltage levels with the SCKp external clock

Conditions: VCC = 1.6 to 5.5 V, VSS = 0 V, Ta = -40 to +125°C

Item	Conditions		Symbol	High-speed mode		Middle-speed mode		Low-speed mode		Unit	Test Conditions
				Min.	Max.	Min.	Max.	Min.	Max.		
SCKp cycle time*4	4.0 V ≤ VCC ≤ 5.5 V	20 MHz < f_{MCK}	t_{KCY2}	8/f_{MCK}	—	8/f_{MCK}	—	—	—	ns	Figure 31.17 Figure 31.18
		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			Greater of: 6/f_{MCK} or 500	—	Greater of: 6/f_{MCK} or 500	—	Greater of: 6/f_{MCK} or 500	—	ns	
	1.8 V ≤ VCC ≤ 5.5 V			Greater of: 6/f_{MCK} or 750	—	Greater of: 6/f_{MCK} or 750	—	Greater of: 6/f_{MCK} or 750	—	ns	
	1.6 V ≤ VCC ≤ 5.5 V			Greater of: 6/f_{MCK} or 1500	—	Greater of: 6/f_{MCK} or 1500	—	Greater of: 6/f_{MCK} or 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	—	t_{KCY2}/2 - 66	—	ns	
SIp setup time (to SCKp↑)*1	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	—	1/f_{MCK} + 40	—	ns	
SIp hold time (from SCKp↑)*1	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	—	1/f_{MCK} + 250	—	ns	
Delay time from SCKp↓ to SOp output*2	C = 30 pF*3	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	—	2/f_{MCK} + 220	ns	

Note 1. This setting applies when SCRmn.DCP[1:0] = 00b or 11b. The setting for the SIp setup time becomes to SCKp↓ and that for the SIp hold time becomes 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 becomes from SCKp↑ when SCRmn.DCP[1:0] = 01b or 10b.
Note 3. C is the load capacitance of the SOp output line.
Note 4. Transfer rate in the Snooze mode is 1 Mbps at the maximum.

Note: Select the normal input buffer for the SIp pin and SCKp pin and the normal output mode for the SOp pin by using the Port gh Pin Function Select Register (PghPFS_A.PIM and PghPFS_A.NCODR).

Note:
- p: Simplified SPI number (p = 00, 01, 10, 11, 20, 21), m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), gh: Port number (gh = 100 to 106,109,110,112 to 115, 201, 204 to 208, 212, 213, 301 to 303, 402, 403, 407, 409 to 411, 500 to 502, 915)
- f_{MCK}: Serial array unit operation clock frequency
  To set this operating clock, use the CKS bit in the serial mode register mn (SMRmn).
  m: Unit number, n: Channel number (mn = 00 to 03, 10, 11)

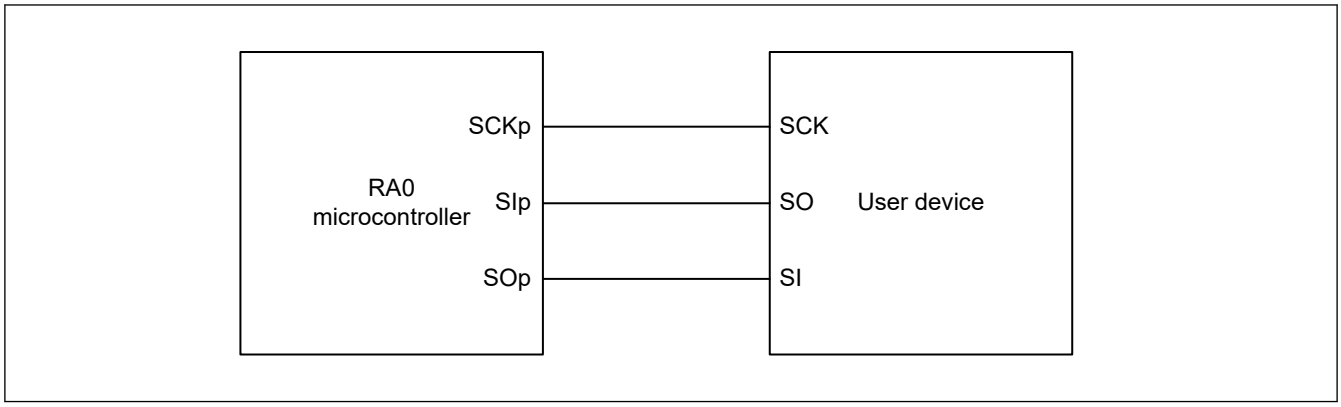


Figure 31.16 Connection in the simplified SPI communications with devices operating at the same voltage levels

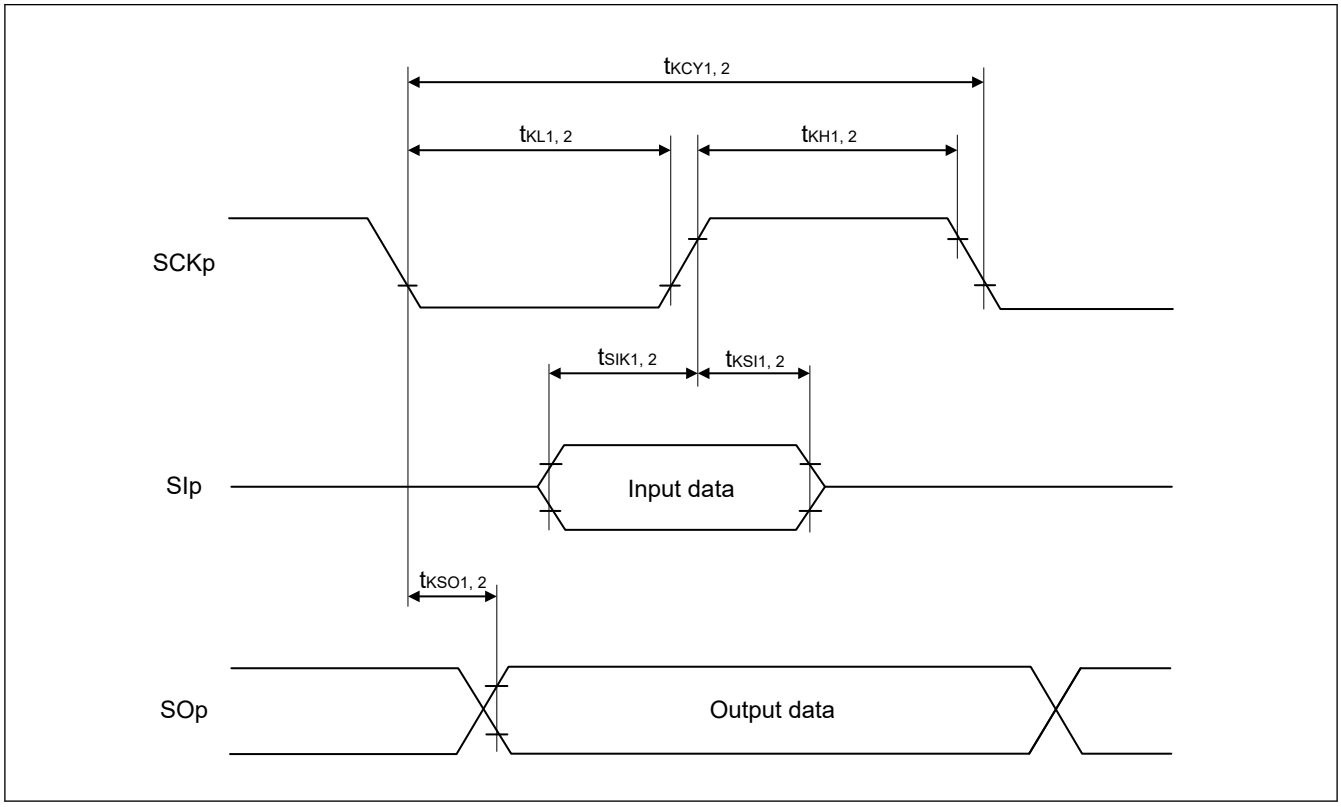


Figure 31.17 Timing of serial transfer in the simplified SPI communications with devices operating at the same voltage levels when SCRmn.DCP[1:0] = 00b or 11b

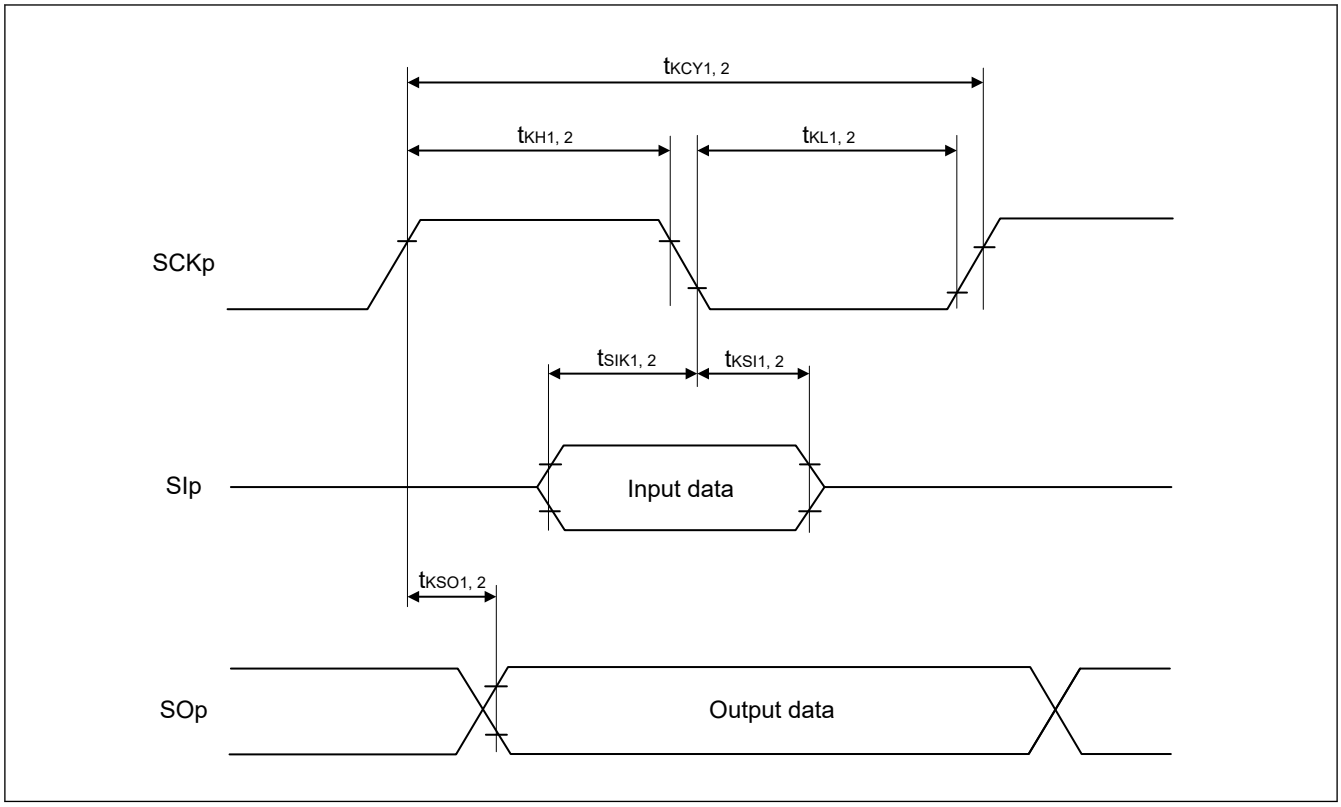


Figure 31.18 Timing of serial transfer in the simplified SPI communications with devices operating at the same voltage levels when SCRmn.DCP[1:0] = 01b or 10b

Note:
- p: Simplified SPI number (p = 00, 01, 10, 11, 20, 21)
- m: Unit number, n: Channel number (mn = 00 to 03, 10, 11)

Table 31.27 In simplified IIC communications with devices operating at the same voltage levels (1 of 2)

Conditions: VCC = 1.6 to 5.5 V, VSS = 0 V, Ta = -40 to +125°C

Parameter		Symbol	High-speed mode		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, C_b = 50 pF, R_b = 2.7 kΩ	f_{SCL}	—	1000*1	—	1000*1	—	400*1	kHz	Figure 31.20
	1.8 V ≤ VCC ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ		—	400*1	—	400*1	—	400*1	kHz	
	1.8 V ≤ VCC < 2.7 V, C_b = 100 pF, R_b = 5 kΩ		—	300*1	—	300*1	—	300*1	kHz	
	1.6 V ≤ VCC < 1.8 V, C_b = 100 pF, R_b = 5 kΩ		—	250*1	—	250*1	—	250*1	kHz	
Hold time when SCLr is low	2.7 V ≤ VCC ≤ 5.5 V, C_b = 50 pF, R_b = 2.7 kΩ	t_{LOW}	475	—	475	—	1150	—	ns	
	1.8 V ≤ VCC ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ		1150	—	1150	—	1150	—	ns	
	1.8 V ≤ VCC < 2.7 V, C_b = 100 pF, R_b = 5 kΩ		1550	—	1550	—	1550	—	ns	
	1.6 V ≤ VCC < 1.8 V, C_b = 100 pF, R_b = 5 kΩ		1850	—	1850	—	1850	—	ns	
Hold time when SCLr is high	2.7 V ≤ VCC ≤ 5.5 V, C_b = 50 pF, R_b = 2.7 kΩ	t_{HIGH}	475	—	475	—	1150	—	ns	
	1.8 V ≤ VCC ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ		1150	—	1150	—	1150	—	ns	
	1.8 V ≤ VCC < 2.7 V, C_b = 100 pF, R_b = 5 kΩ		1550	—	1550	—	1550	—	ns	
	1.6 V ≤ VCC < 1.8 V, C_b = 100 pF, R_b = 5 kΩ		1850	—	1850	—	1850	—	ns	

Table 31.27 In simplified IIC communications with devices operating at the same voltage levels (2 of 2)

Conditions: VCC = 1.6 to 5.5 V, VSS = 0 V, Ta = -40 to +125°C

Parameter		Symbol	High-speed mode		Middle-speed mode		Low-speed mode		Unit	Test Conditions
			Min.	Max.	Min.	Max.	Min.	Max.		
Data setup time (reception)	2.7 V ≤ VCC ≤ 5.5 V, C_b = 50 pF, R_b = 2.7 kΩ	$t_{SU:DAT}$	$1/f_{MCK}$ + 85*2	—	$1/f_{MCK}$ +85*2	—	$1/f_{MCK}$ +145*2	—	ns	Figure 31.20
	1.8 V ≤ VCC ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ		$1/f_{MCK}$ + 145*2	—	$1/f_{MCK}$ + 145*2	—	$1/f_{MCK}$ +145*2	—	ns	
	1.8 V ≤ VCC < 2.7 V, C_b = 100 pF, R_b = 5 kΩ		$1/f_{MCK}$ + 230*2	—	$1/f_{MCK}$ + 230*2	—	$1/f_{MCK}$ + 230*2	—	ns	
	1.6 V ≤ VCC < 1.8 V, C_b = 100 pF, R_b = 5 kΩ		$1/f_{MCK}$ + 290*2	—	$1/f_{MCK}$ + 290*2	—	$1/f_{MCK}$ + 290*2	—	ns	
Data hold time (transmission)	2.7 V ≤ VCC ≤ 5.5 V, C_b = 50 pF, R_b = 2.7 kΩ	$t_{HD:DAT}$	0	305	0	305	0	305	ns	
	1.8 V ≤ VCC ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ		0	355	0	355	0	355	ns	
	1.8 V ≤ VCC < 2.7 V, C_b = 100 pF, R_b = 5 kΩ		0	405	0	405	0	405	ns	
	1.6 V ≤ VCC < 1.8 V, C_b= 100 pF, R_b = 5 kΩ		0	405	0	405	0	405	ns	

Note 1. The listed times must be no greater than $f_{MCK}/4$.

Note 2. Set f_{MCK} so that it does not exceed the hold time when SCLr is low or high.

Note: Select the normal input buffer and the N-ch open drain output [withstand voltage of VCC] mode for the SDAr pin and the normal output mode for the SCLr pin by using the Port gh Pin Function Select Register (PghPFS_A.PIM and PghPFS_A.NCODR).

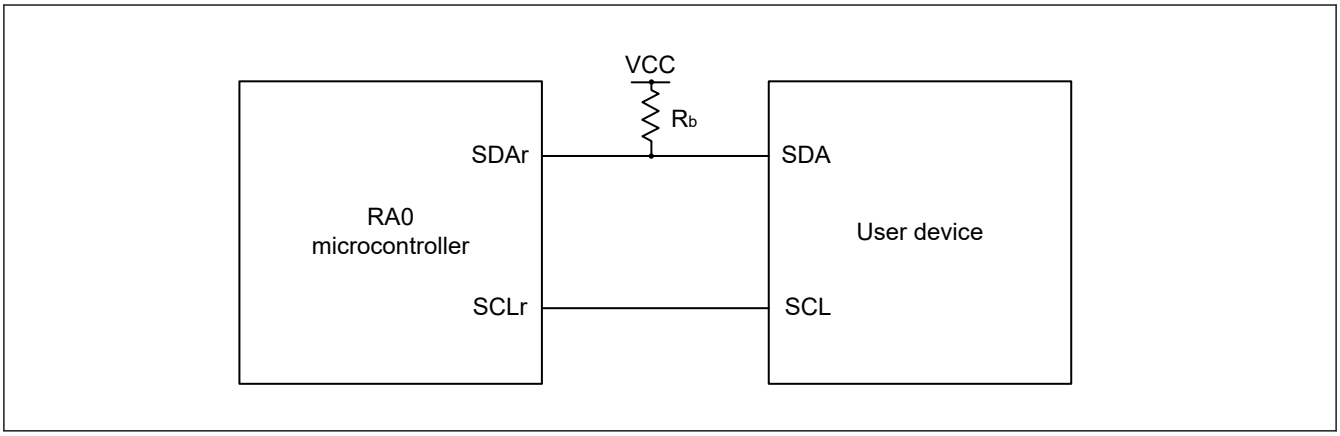


Figure 31.19 Connection in the simplified IIC communications with devices operating at the same voltage levels

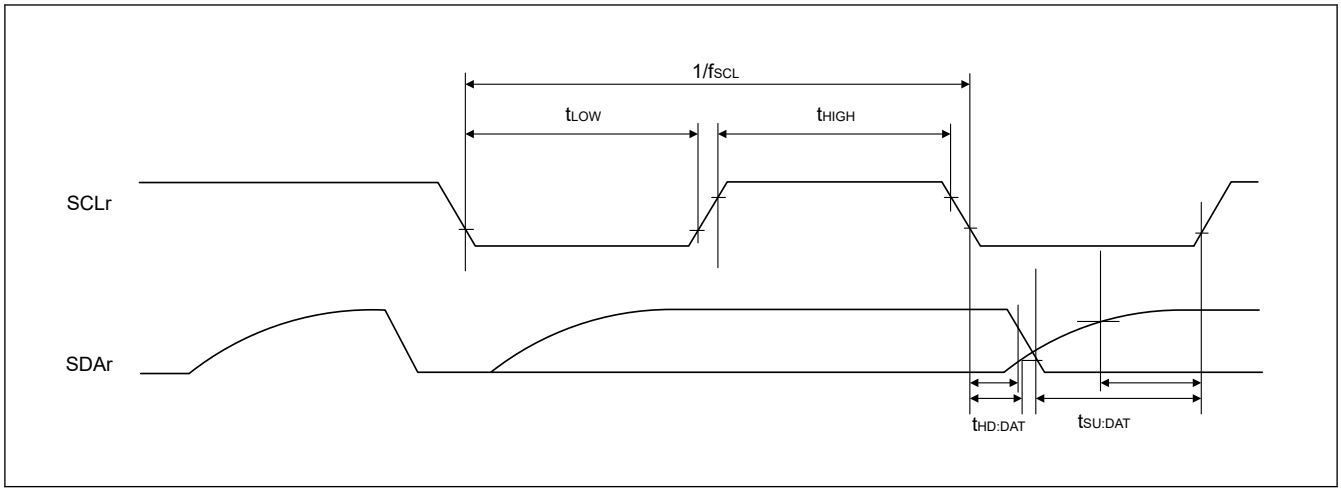


Figure 31.20 Timing of serial transfer in the simplified IIC communications with devices operating at the same voltage levels

Note: ● R_b[Ω]: Communication line (SDAr) pull-up resistance, C_b[F]: Communication line (SDAr, SCLr) load capacitance

● r: IIC number (r = 00, 01, 10, 11, 20, 21), gh: Port number (gh = 100, 102, 104, 105, 110, 112, 114, 115, 201, 204, 205, 207, 208, 212, 301, 302, 403, 407, 409 to 411, 500, 502)

● f_{MCK}: Serial array unit operation clock frequency
To set this operating clock, use the CKSmn bit in the serial mode register mn (SMRmn).
m: Unit number, n: Channel number (mn = 00 to 03, 10, 11)

Table 31.28 In UART communications with devices operating at different voltage levels (1.8 V, 2.5 V, 3 V) (1)

Conditions: VCC = 1.8 to 5.5 V, VSS = 0 V, Ta = -40 to +125°C

Parameter			Symbol	High-speed mode		Middle-speed mode		Low-speed mode		Unit	Test Conditions
				Min.	Max.	Min.	Max.	Min.	Max.		
Transfer rate	Reception	4.0 V ≤ VCC ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V	—	—	$f_{MCK}/6$*1	—	$f_{MCK}/6$*1	—	$f_{MCK}/6$*1	bps	Figure 31.22
		Theoretical value of the maximum transfer rate f_{MCK} = PCLKB*3		—	5.3	—	4	—	0.33	Mbps	
		2.7 V ≤ VCC < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V		—	$f_{MCK}/6$*1	—	$f_{MCK}/6$*1	—	$f_{MCK}/6$*1	bps	
		Theoretical value of the maximum transfer rate f_{MCK}*3 = PCLKB*3		—	5.3	—	4	—	0.33	Mbps	
		1.8 V ≤ VCC < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V		—	$f_{MCK}/6$*1 *2	—	$f_{MCK}/6$*1 *2	—	$f_{MCK}/6$*1 *2	bps	
		Theoretical value of the maximum transfer rate f_{MCK} = PCLKB*3		—	5.3	—	4	—	0.33	Mbps	

Note 1. Transfer rate in the Snooze mode is within the range from 4800 to 9600 bps.
Note 2. Use this rate with VCC ≥ V_b.
Note 3. The maximum operating frequencies of the system clock (PCLKB) are:
High-speed mode: 32 MHz (1.8 V ≤ VCC ≤ 5.5 V), 4 MHz (1.6 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: 2 MHz (1.6 V ≤ VCC ≤ 5.5 V)

Note: Select the TTL input buffer for the RXDq pin and the N-ch open drain output [withstand voltage of VCC] mode for the TXDq pin by using the Port gh Pin Function Select Register (PghPFS_A.PIM and PghPFS_A.NCODR). For V_{IH} and V_{IL}, see the DC characteristics with the TTL input buffer selected.

Note: ● V_b[V]: Communication line voltage

● q: UART number (q = 0 to 2), gh: Port number (gh = 100, 101, 109, 110, 212, 213, 402, 403, 501, 502)

- f_{MCK}: Serial array unit operation clock frequency
  To set this operating clock, use the CKS bit in the serial mode register mn (SMRmn).
  m: Unit number, n: Channel number (mn = 00 to 03, 10, 11)
- Communications by using P212 and P213 with devices operating at different voltage levels are not possible since P212PFS_A and P213PFS_A registers do not have PIM bit.

Table 31.29 In UART communications with devices operating at different voltage levels (1.8 V, 2.5 V, 3 V) (2)

Conditions: VCC = 1.8 to 5.5 V, VSS = 0 V, Ta = -40 to +125°C

Parameter			Symbol	High-speed mode		Middle-speed mode		Low-speed mode		Unit	Test Conditions
				Min.	Max.	Min.	Max.	Min.	Max.		
Transfer rate	Transmission	4.0 V ≤ VCC ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V	—	—	*1	—	*1	—	*1	bps	Figure 31.22
		Theoretical value of the maximum transfer rate C_b = 50 pF, R_b = 1.4 kΩ, V_b = 2.7 V		—	2.8*2	—	2.8*2	—	2.8*2	Mbps	
		2.7 V ≤ VCC < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V		—	*3	—	*3	—	*3	bps	
		Theoretical value of the maximum transfer rate C_b = 50 pF, R_b = 2.7 kΩ, V_b = 2.3 V		—	1.2*4	—	1.2*4	—	1.2*4	Mbps	
		1.8 V ≤ VCC < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V		—	*5 *6	—	*5 *6	—	*5 *6	bps	
		Theoretical value of the maximum transfer rate C_b = 50 pF, R_b = 5.5 kΩ, V_b = 1.6 V		—	0.43*7	—	0.43*7	—	0.43*7	Mbps	

Note 1. The smaller maximum transfer rate derived by using f_{MCK}/6 or the following expression is the valid maximum transfer rate.
Expression for calculating the transfer rate when 4.0 V ≤ VCC ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V

$$\text{Maximum transfer rate} = \frac{1}{\left\{-C_b \times R_b \times \ln\left(1-\frac{2.2}{V_b}\right)\right\} \times 3}[\text{bps}]$$

$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \left\{-C_b \times R_b \times \ln\left(1-\frac{2.2}{V_b}\right)\right\}}{\left(\frac{1}{\text{Transfer rate}}\right) \times \text{Number of transferred bits}} \times 100[\%]$$

This value is the theoretical value of the relative difference between the transmission and reception sides.

Note 2. This rate is calculated as an example when the conditions described in the Conditions column are met. See *1 above to calculate the maximum transfer rate under the conditions of the customer.

Note 3. The smaller maximum transfer rate derived by using f_{MCK}/6 or the following expression is the valid maximum transfer rate.
Expression for calculating the transfer rate when 2.7 V ≤ VCC < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V

$$\text{Maximum transfer rate} = \frac{1}{\left\{-C_b \times R_b \times \ln\left(1-\frac{2.0}{V_b}\right)\right\} \times 3}[\text{bps}]$$

$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \left\{-C_b \times R_b \times \ln\left(1-\frac{2.0}{V_b}\right)\right\}}{\left(\frac{1}{\text{Transfer rate}}\right) \times \text{Number of transferred bits}} \times 100[\%]$$

This value is the theoretical value of the relative difference between the transmission and reception sides.

Note 4. This rate is calculated as an example when the conditions described in the Conditions column are met. See *3 above to calculate the maximum transfer rate under the conditions of the customer.

Note 5. Use this rate with VCC ≥ V_b.

Note 6. The smaller maximum transfer rate derived by using f_{MCK}/6 or the following expression is the valid maximum transfer rate.
Expression for calculating the transfer rate when 1.8 V ≤ VCC < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V

$$\text{Maximum transfer rate} = \frac{1}{\left\{-C_b \times R_b \times \ln\left(1-\frac{1.5}{V_b}\right)\right\} \times 3}[\text{bps}]$$

$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \left\{-C_b \times R_b \times \ln\left(1 - \frac{1.5}{V_b}\right)\right\}}{\left(\frac{1}{\text{Transfer rate}}\right) \times \text{Number of transferred bits}} \times 100[\%]$$

This value is the theoretical value of the relative difference between the transmission and reception sides.

Note 7. This rate is calculated as an example when the conditions described in the Conditions column are met. See *6 above to calculate the maximum transfer rate under the conditions of the customer.

Note: Select the TTL input buffer for the RXDq pin and the N-ch open drain output [withstand voltage of VCC] mode for the TXDq pin by using the Port gh Pin Function Select Register (PghPFS_A.PIM and PghPFS_A.NCODR). For V_{IH} and V_{IL}, see the DC characteristics with the TTL input buffer selected.

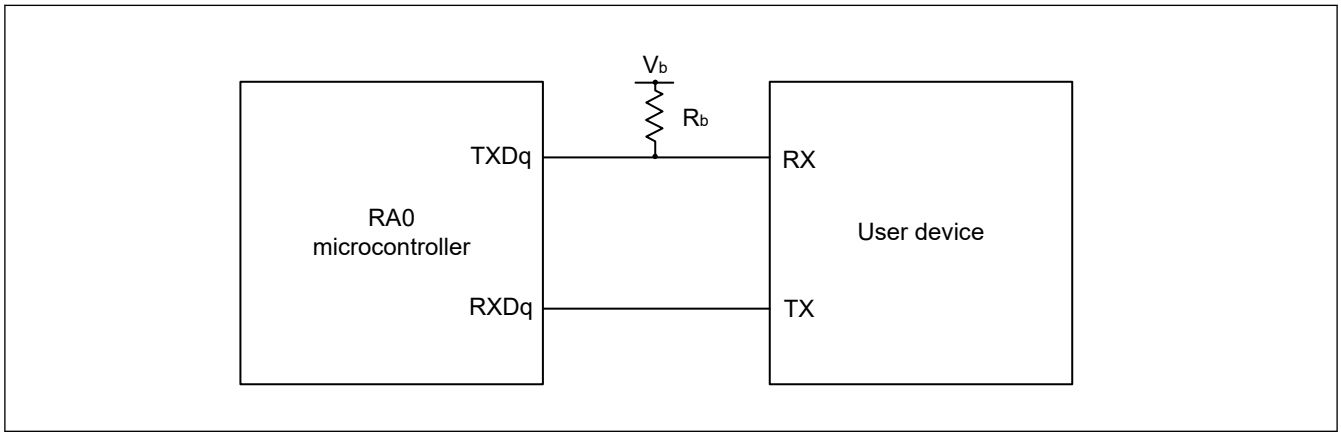


Figure 31.21 In UART communications with devices operating at different voltage levels

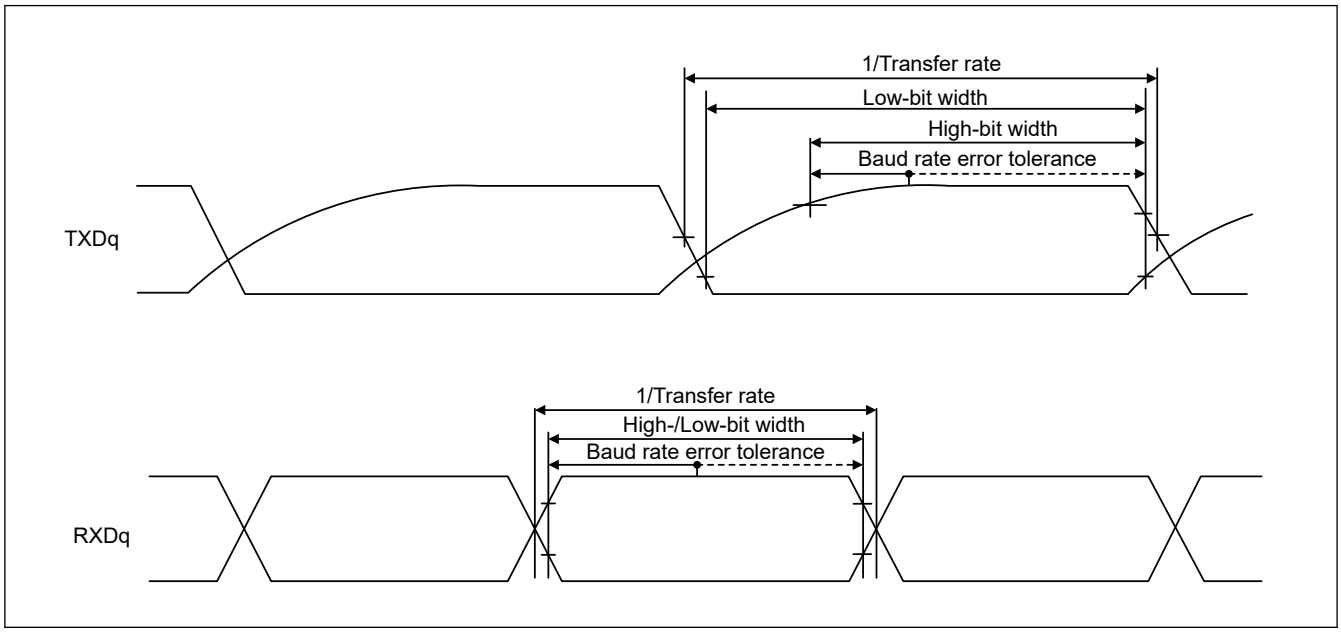


Figure 31.22 Bit width in the UART communications with devices operating at different voltage levels (reference)

Note:
- R_b[Ω]: Communication line (TXDq) pull-up resistance, C_b[F]: Communication line (TXDq) load capacitance, V_b[V]: Communication line voltage
- q: UART number (q = 0 to 2), gh: Port number (gh = 100, 101, 109, 110, 212, 213, 402, 403, 501, 502)
- f_{MCK}: Serial array unit operation clock frequency
  To set this operating clock, use the CKS bit in the serial mode register mn (SMRmn).
  m: Unit number, n: Channel number (mn = 00 to 03, 10, 11)
- Communications by using P212 and P213 with devices operating at different voltage levels are not possible since P212PFS_A and P213PFS_A registers do not have PIM bit.

Table 31.30 In simplified SPI communications in the master mode with devices operating at different voltage levels (2.5 V or 3 V) with the internal SCKp clock (the ratings below are only applicable to SPI00)

Conditions: VCC = 2.7 to 5.5 V, VSS = 0 V, Ta = -40 to +105°C

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} ≥ 2/PCLKB	4.0 V ≤ VCC ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ	t_{KCY1}	200	—	200	—	2300	—	ns	Figure 31.24 Figure 31.25
		2.7 V ≤ VCC < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ		300	—	300	—	2300	—	ns	
SCKp high-level width	4.0 V ≤ VCC ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ		t_{KH1}	t_{KCY1}/2 - 50	—	t_{KCY1}/2 - 50	—	t_{KCY1}/2 - 50	—	ns	
	2.7 V ≤ VCC < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ			t_{KCY1}/2 - 120	—	t_{KCY1}/2 - 120	—	t_{KCY1}/2 - 120	—	ns	
SCKp low-level width	4.0 V ≤ VCC ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ		t_{KL1}	t_{KCY1}/2 -7	—	t_{KCY1}/2 -7	—	t_{KCY1}/2 - 50	—	ns	
	2.7 V ≤ VCC < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ			t_{KCY1}/2 - 10	—	t_{KCY1}/2 - 10	—	t_{KCY1}/2 - 50	—	ns	
SIp setup time (to SCKp↑)*1	4.0 V ≤ VCC ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ		t_{SIK1}	58	—	58	—	479	—	ns	
	2.7 V ≤ VCC < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ			121	—	121	—	479	—	ns	
SIp hold time (from SCKp↑)*1	4.0 V ≤ VCC ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ		t_{KSI1}	10	—	10	—	10	—	ns	
	2.7 V ≤ VCC < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ			10	—	10	—	10	—	ns	
Delay time from SCKp↓ to SOp output*1	4.0 V ≤ VCC ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ		t_{KSO1}	—	60	—	60	—	60	ns	
	2.7 V ≤ VCC < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ			—	130	—	130	—	130	ns	
SIp setup time (to SCKp↓)*2	4.0 V ≤ VCC ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ		t_{SIK1}	23	—	23	—	110	—	ns	
	2.7 V ≤ VCC < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ			33	—	33	—	110	—	ns	
SIp hold time (from SCKp↓)*2	4.0 V ≤ VCC ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ		t_{KSI1}	10	—	10	—	10	—	ns	
	2.7 V ≤ VCC < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ			10	—	10	—	10	—	ns	
Delay time from SCKp↑ to SOp output*2	4.0 V ≤ VCC ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ		t_{KSO1}	—	10	—	10	—	10	ns	
	2.7 V ≤ VCC < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ			—	10	—	10	—	10	ns	

Note 1. This setting applies when SCRmn.DCP[1:0] = 00b or 11b.
Note 2. This setting applies when SCRmn.DCP[1:0] = 01b or 10b.

Note: Select the TTL input buffer for the SIp pin and the N-ch open drain output [withstand voltage of VCC] mode for the SOp pin and SCKp pin by using the Port gh Pin Function Select Register (PghPFS_A.PIM and PghPFS_A.NCODR). For V_{IH} and V_{IL}, see the DC characteristics with the TTL input buffer selected.

Note:
- R_b[Ω]: Communication line (SCKp, SOp) pull-up resistance, C_b[F]: Communication line (SCKp, SOp) load capacitance, V_b[V]: Communication line voltage
- p: Simplified SPI number (p = 00), m: Unit number (m = 0), n: Channel number (n = 0), gh: Port number (gh = 100 to 103, 112, 201, 500 to 502)
- f_{MCK}: Serial array unit operation clock frequency

To set this operating clock, use the CKSmn bit in the serial mode register mn (SMRmn).
m: Unit number, n: Channel number (mn = 00)

Table 31.31 In simplified SPI communications in the master mode with devices operating at different voltage levels (1.8 V, 2.5 V, or 3 V) with the internal SCKp clock (1)

Conditions: VCC = 1.8 to 5.5 V, VSS = 0 V, Ta = -40 to +125°C

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} ≥ 4/ PCLKB	4.0 V ≤ VCC ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	t_{KCY1}	300	—	300	—	2300	—	ns	Figure 31.24 Figure 31.25
		2.7 V ≤ VCC < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		500	—	500	—	2300	—	ns	
		1.8 V ≤ VCC < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V*1, C_b = 30 pF, R_b = 5.5 kΩ		1150	—	1150	—	2300	—	ns	
SCKp high-level width	4.0 V ≤ VCC ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		t_{KH1}	t_{KCY1}/2 - 75	—	t_{KCY1}/2 - 75	—	t_{KCY1}/2 - 75	—	ns	
	2.7 V ≤ VCC < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ			t_{KCY1}/2 - 170	—	t_{KCY1}/2 - 170	—	t_{KCY1}/2 - 170	—	ns	
	1.8 V ≤ VCC < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V*1, C_b = 30 pF, R_b = 5.5 kΩ			t_{KCY1}/2 - 458	—	t_{KCY1}/2 - 458	—	t_{KCY1}/2 - 458	—	ns	
SCKp low-level width	4.0 V ≤ VCC ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		t_{KL1}	t_{KCY1}/2 -12	—	t_{KCY1}/2 -12	—	t_{KCY1}/2 - 50	—	ns	
	2.7 V ≤ VCC < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ			t_{KCY1}/2 - 18	—	t_{KCY1}/2 - 18	—	t_{KCY1}/2 - 50	—	ns	
	1.8 V ≤ VCC < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V*1, C_b = 30 pF, R_b = 5.5 kΩ			t_{KCY1}/2 - 50	—	t_{KCY1}/2 - 50	—	t_{KCY1}/2 - 50	—	ns	

Note 1. Use this setting with VCC ≥ V_b.

Note: Select the TTL input buffer for the SIp pin and the N-ch open drain output [withstand voltage of VCC] mode for the SOp pin and SCKp pin by using the Port gh Pin Function Select Register (PghPFS_A.PIM and PghPFS_A.NCODR). For V_{IH} and V_{IL}, see the DC characteristics with the TTL input buffer selected.

Table 31.32 In simplified SPI communications in the master mode with devices operating at different voltage levels (1.8 V, 2.5 V, or 3 V) with the internal SCKp clock (2)

Conditions: VCC = 1.8 to 5.5 V, VSS = 0 V, Ta = -40 to +125°C

Parameter		Symbol	High-speed mode		Middle-speed mode		Low-speed mode		Unit	Test Conditions
			Min.	Max.	Min.	Max.	Min.	Max.		
SIp setup time (to SCKp↑)*1	4.0 V ≤ VCC ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	t_{SIK1}	81	—	81	—	479	—	ns	Figure 31.24 Figure 31.25
	2.7 V ≤ VCC < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		177	—	177	—	479	—	ns	
	1.8 V ≤ VCC < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V*2, C_b = 30 pF, R_b = 5.5 kΩ		479	—	479	—	479	—	ns	
SIp hold time (from SCKp↑)*1	4.0 V ≤ VCC ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	t_{KSI1}	19	—	19	—	19	—	ns	
	2.7 V ≤ VCC < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		19	—	19	—	19	—	ns	
	1.8 V ≤ VCC < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V*2, C_b = 30 pF, R_b = 5.5 kΩ		19	—	19	—	19	—	ns	
Delay time from SCKp↓ to SOp output*1	4.0 V ≤ VCC ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	t_{KSO1}	—	100	—	100	—	100	ns	
	2.7 V ≤ VCC < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		—	195	—	195	—	195	ns	
	1.8 V ≤ VCC < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V*2, C_b = 30 pF, R_b = 5.5 kΩ		—	483	—	483	—	483	ns	

Note 1. This setting applies when SCRmn.DCP[1:0] = 00b or 11b.
Note 2. Use this setting with VCC ≥ V_b.

Note: Select the TTL input buffer for the SIp pin and the N-ch open drain output [withstand voltage of VCC] mode for the SOp pin and SCKp pin by using the Port gh Pin Function Select Register (PghPFS_A.PIM and PghPFS_A.NCODR). For V_{IH} and V_{IL}, see the DC characteristics with the TTL input buffer selected.

Table 31.33 In simplified SPI communications in the master mode with devices operating at different voltage levels (1.8 V, 2.5 V, or 3 V) with the internal SCKp clock (3)

Conditions: VCC = 1.8 to 5.5 V, VSS = 0 V, Ta = -40 to +125°C

Parameter		Symbol	High-speed mode		Middle-speed mode		Low-speed mode		Unit	Test Conditions
			Min.	Max.	Min.	Max.	Min.	Max.		
SIp setup time (to SCKp↓)*1	4.0 V ≤ VCC ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	t_{SIK1}	44	—	44	—	110	—	ns	Figure 31.24 Figure 31.25
	2.7 V ≤ VCC < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		44	—	44	—	110	—	ns	
	1.8 V ≤ VCC < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V*2, C_b = 30 pF, R_b = 5.5 kΩ		110	—	110	—	110	—	ns	
SIp hold time (from SCKp↓)*1	4.0 V ≤ VCC ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	t_{KSI1}	19	—	19	—	19	—	ns	
	2.7 V ≤ VCC < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		19	—	19	—	19	—	ns	
	1.8 V ≤ VCC < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V*2, C_b = 30 pF, R_b = 5.5 kΩ		19	—	19	—	19	—	ns	
Delay time from SCKp↑ to SOp output*1	4.0 V ≤ VCC ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	t_{KSO1}	—	25	—	25	—	25	ns	
	2.7 V ≤ VCC < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		—	25	—	25	—	25	ns	
	1.8 V ≤ VCC < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V*2, C_b = 30 pF, R_b = 5.5 kΩ		—	25	—	25	—	25	ns	

Note 1. This setting applies when SCRmn.DCP[1:0] = 01b or 10b.
Note 2. Use this setting with VCC ≥ V_b.

Note: Select the TTL input buffer for the SIp pin and the N-ch open drain output [withstand voltage of VCC] mode for the SOp pin and SCKp pin by using the Port gh Pin Function Select Register (PghPFS_A.PIM and PghPFS_A.NCODR). For V_{IH} and V_{IL}, see the DC characteristics with the TTL input buffer selected.

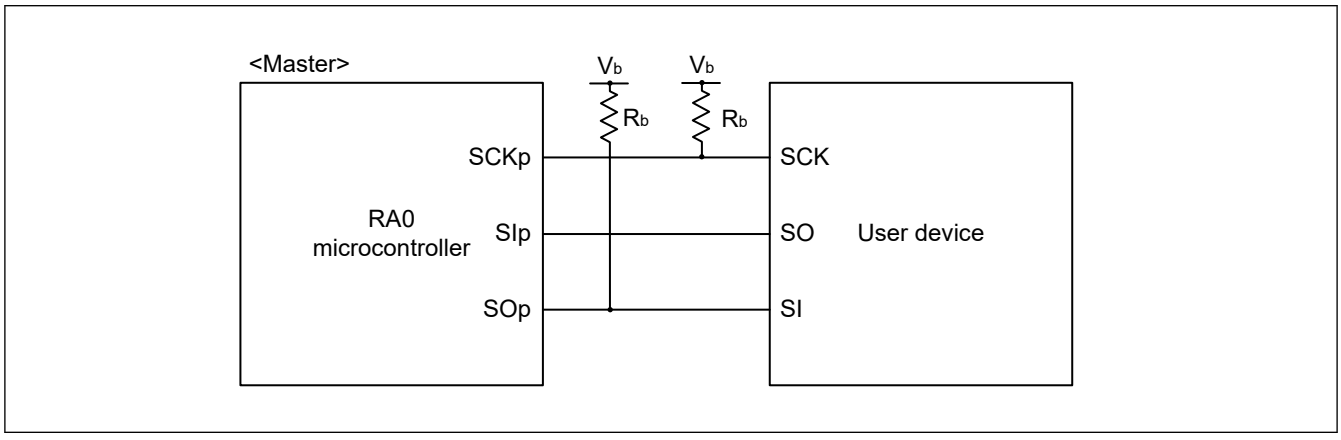


Figure 31.23 Connection in the simplified SPI communications with devices operating at different voltage levels

Note: • R_b[Ω]: Communication line (SCKp, SOp) pull-up resistance, C_b[F]: Communication line (SCKp, SOp) load capacitance, V_b[V]: Communication line voltage

- p: Simplified SPI number (p = 00, 01, 10, 11, 20, 21), m: Unit number, n: Channel number (mn = 00 to 03, 10, 11), gh: Port number (gh = 100 to 106, 109, 110, 112 to 115, 201, 204 to 208, 212, 213, 301 to 303, 402, 403, 407, 409 to 411, 500 to 502, 915)
- f_{MCK}: Serial array unit operation clock frequency
  To set this operating clock, use the CKS bit in the serial mode register mn (SMRmn).
  m: Unit number, n: Channel number (mn = 00 to 03, 10, 11)
- Communications by using P212 and P213 with devices operating at different voltage levels are not possible since P212PFS_A and P213PFS_A registers do not have PIM bit.

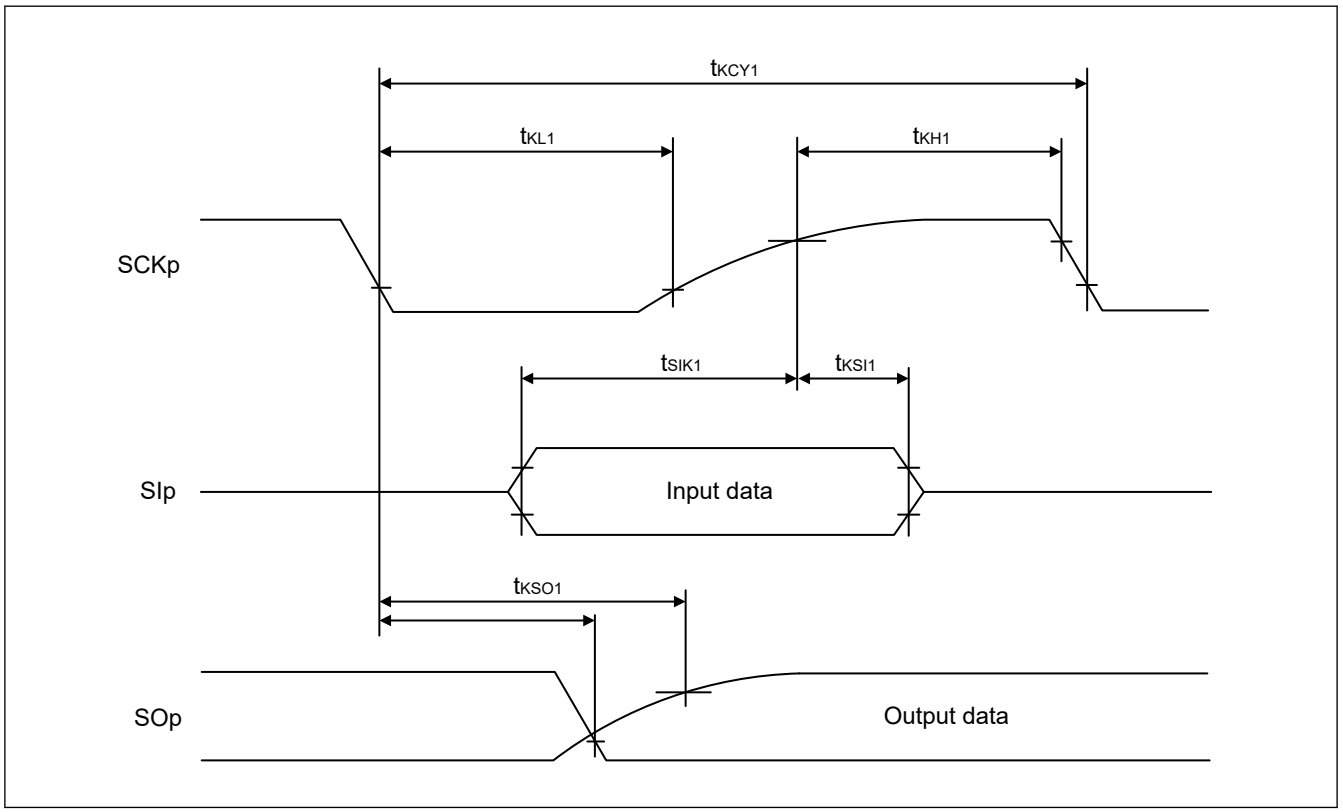


Figure 31.24 Timing of serial transfer in the simplified SPI communications in the master mode with devices operating at different voltage levels when SCRmn.DCP[1:0] = 00b or 11b

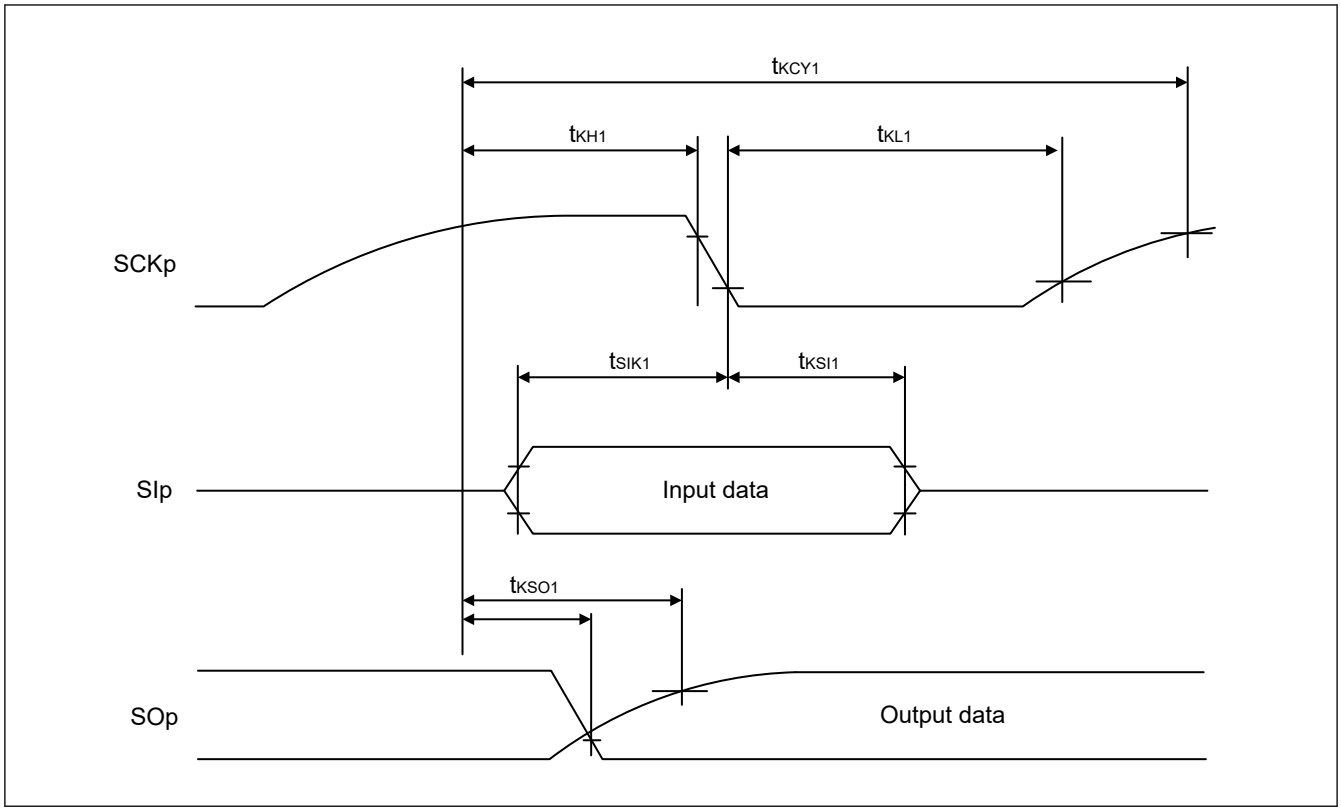


Figure 31.25 Timing of serial transfer in the simplified SPI communications in the master mode with devices operating at different voltage levels when SCRmn.DCP[1:0] = 01b or 10b

Note:
- p: Simplified SPI number (p = 00, 01, 10, 11, 20, 21), m: Unit number, n: Channel number (mn = 00 to 03, 10, 11), gh: Port number (gh = 100 to 106, 109, 110, 112 to 115, 201, 204 to 208, 212, 213, 301 to 303, 402, 403, 407, 409 to 411, 500 to 502, 915)
- Communications by using P212 and P213 with devices operating at different voltage levels are not possible since P212PFS_A and P213PFS_A registers do not have PIM bit.

Table 31.34 In simplified SPI communications in the slave mode with devices operating at different voltage levels (1.8 V, 2.5 V, or 3 V) with the external SCKp clock

Conditions: VCC = 1.8 to 5.5 V, VSS = 0 V, Ta = -40 to +125°C

Parameter			Symbol	High-speed mode		Middle-speed mode		Low-speed mode		Unit	Test Conditions
				Min.	Max.	Min.	Max.	Min.	Max.		
SCKp cycle time*1	4.0 V ≤ VCC ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V	24 MHz < f_{MCK}	t_{KCY2}	14/f_{MCK}	—	—	—	—	—	ns	Figure 31.27 Figure 31.28
		20 MHz < f_{MCK} ≤ 24 MHz		12/f_{MCK}	—	12/f_{MCK}	—	—	—	ns	
		8 MHz < f_{MCK} ≤ 20 MHz		10/f_{MCK}	—	10/f_{MCK}	—	—	—	ns	
		4 MHz < f_{MCK} ≤ 8 MHz		8/f_{MCK}	—	8/f_{MCK}	—	—	—	ns	
		f_{MCK} ≤ 4 MHz		6/f_{MCK}	—	6/f_{MCK}	—	10/f_{MCK}	—	ns	
	2.7 V ≤ VCC < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V	24 MHz < f_{MCK}		20/f_{MCK}	—	—	—	—	—	ns	
		20 MHz < f_{MCK} ≤ 24 MHz		16/f_{MCK}	—	16/f_{MCK}	—	—	—	ns	
		16 MHz < f_{MCK} ≤ 20 MHz		14/f_{MCK}	—	14/f_{MCK}	—	—	—	ns	
		8 MHz < f_{MCK} ≤ 16 MHz		12/f_{MCK}	—	12/f_{MCK}	—	—	—	ns	
		4 MHz < f_{MCK} ≤ 8 MHz		8/f_{MCK}	—	8/f_{MCK}	—	—	—	ns	
		f_{MCK} ≤ 4 MHz		6/f_{MCK}	—	6/f_{MCK}	—	10/f_{MCK}	—	ns	
	1.8 V ≤ VCC < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V*2	24 MHz < f_{MCK}		48/f_{MCK}	—	—	—	—	—	ns	
		20 MHz < f_{MCK} ≤ 24 MHz		36/f_{MCK}	—	36/f_{MCK}	—	—	—	ns	
		16 MHz < f_{MCK} ≤ 20 MHz		32/f_{MCK}	—	32/f_{MCK}	—	—	—	ns	
		8 MHz < f_{MCK} ≤ 16 MHz		26/f_{MCK}	—	26/f_{MCK}	—	—	—	ns	
		4 MHz < f_{MCK} ≤ 8 MHz		16/f_{MCK}	—	16/f_{MCK}	—	—	—	ns	
		f_{MCK} ≤ 4 MHz		10/f_{MCK}	—	10/f_{MCK}	—	10/f_{MCK}	—	ns	
SCKp high-/low-level width	4.0 V ≤ VCC ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V		t_{KH2}, t_{KL2}	t_{KCY2}/2 - 12	—	t_{KCY2}/2 - 12	—	t_{KCY2}/2 - 50	—	ns	
	2.7 V ≤ VCC < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V			t_{KCY2}/2 - 18	—	t_{KCY2}/2 - 18	—	t_{KCY2}/2 - 50	—	ns	
	1.8 V ≤ VCC < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V*2			t_{KCY2}/2 - 50	—	t_{KCY2}/2 - 50	—	t_{KCY2}/2 - 50	—	ns	
SIp setup time (to SCKp↑)*3	4.0 V ≤ VCC ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V		t_{SIK2}	1/f_{MCK} + 20	—	1/f_{MCK} + 20	—	1/f_{MCK} + 30	—	ns	
	2.7 V ≤ VCC < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V			1/f_{MCK} + 20	—	1/f_{MCK} + 20	—	1/f_{MCK} + 30	—	ns	
	1.8 V ≤ VCC < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V*2			1/f_{MCK} + 30	—	1/f_{MCK} + 30	—	1/f_{MCK} + 30	—	ns	
SIp hold time (from SCKp↑)*3			t_{KSI2}	1/f_{MCK} + 31	—	1/f_{MCK} + 31	—	1/f_{MCK} + 31	—	ns	
Delay time from SCKp↓ to SOp output*4	4.0 V ≤ VCC ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		t_{KSO2}	—	2/f_{MCK} + 120	—	2/f_{MCK} + 120	—	2/f_{MCK} + 573	ns	
	2.7 V ≤ VCC < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ			—	2/f_{MCK} + 214	—	2/f_{MCK} + 214	—	2/f_{MCK} + 573	ns	
	1.8 V ≤ VCC < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V*2, C_b = 30 pF, R_b = 5.5 kΩ			—	2/f_{MCK} + 573	—	2/f_{MCK} + 573	—	2/f_{MCK} + 573	ns	

Note 1. Transfer rate in the Snooze mode: 1 Mbps (max.)
Note 2. Use this setting with VCC ≥ V_b.
Note 3. This setting applies when SCRmn.DCP[1:0] = 00b or 11b. The SIp setup time becomes to SCKp↓ and SIp hold time becomes from SCKp↓ when SCRmn.DCP[1:0] = 01b or 10b.
Note 4. This setting applies when SCRmn.DCP[1:0] = 00b or 11b. The delay time to SOp output becomes from SCKp↑ when SCRmn.DCP[1:0] = 01b or 10b.

Note: Select the TTL input buffer for the SIp pin and the N-ch open drain output [withstand voltage of VCC] mode for the SOp pin and SCKp pin by using the Port gh Pin Function Select Register (PghPFS_A.PIM and PghPFS_A.NCODR). For V_{IH} and V_{IL}, see the DC characteristics with the TTL input buffer selected.

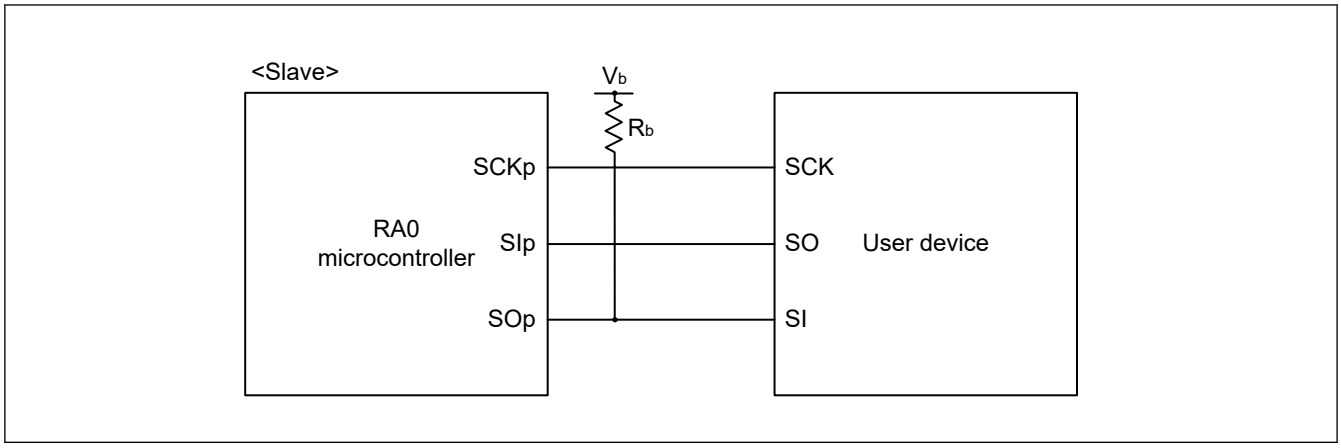


Figure 31.26 Connection in the simplified SPI communications with devices operating at different voltage levels

Note:
- R_b[Ω]: Communication line (SOp) pull-up resistance, C_b[F]: Communication line (SOp) load capacitance, V_b[V]: Communication line voltage
- p: Simplified SPI number (p = 00, 01, 10, 11, 20, 21), m: Unit number, n: Channel number (mn = 00 to 03, 10, 11), gh: Port number (gh = 100 to 106, 109, 110, 112 to 115, 201, 204 to 208, 212, 213, 301 to 303, 402, 403, 407, 409 to 411, 500 to 502, 915)
- f_{MCK}: Serial array unit operation clock frequency
  To set this operating clock, use the CKS bit in the serial mode register mn (SMRmn).
  m: Unit number, n: Channel number (mn = 00 to 03, 10, 11)
- Communications by using P212 and P213 with devices operating at different voltage levels are not possible since P212PFS_A and P213PFS_A registers do not have PIM bit.

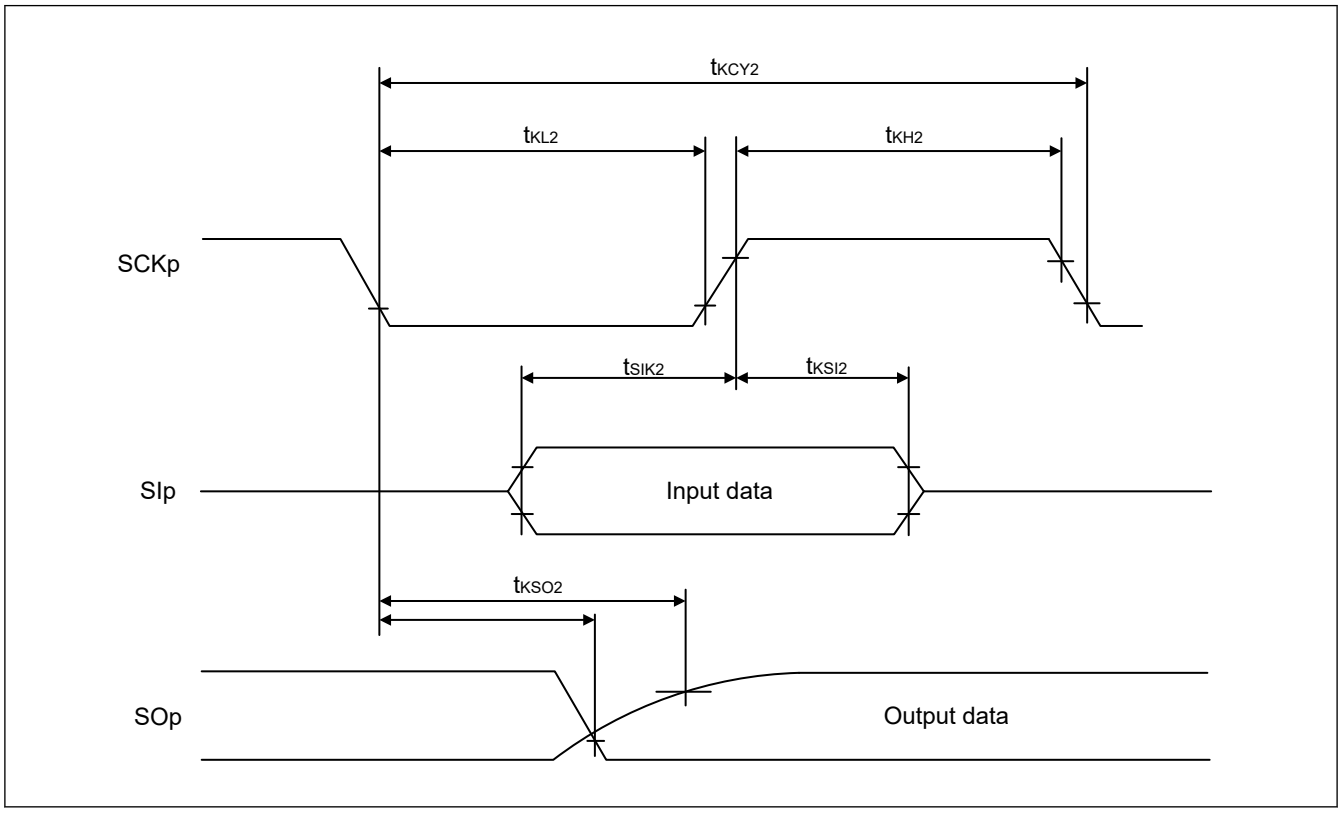


Figure 31.27 Timing of serial transfer in the simplified SPI communications in the slave mode with devices operating at different voltage levels when SCRmn.DCP[1:0] = 00b or 11b

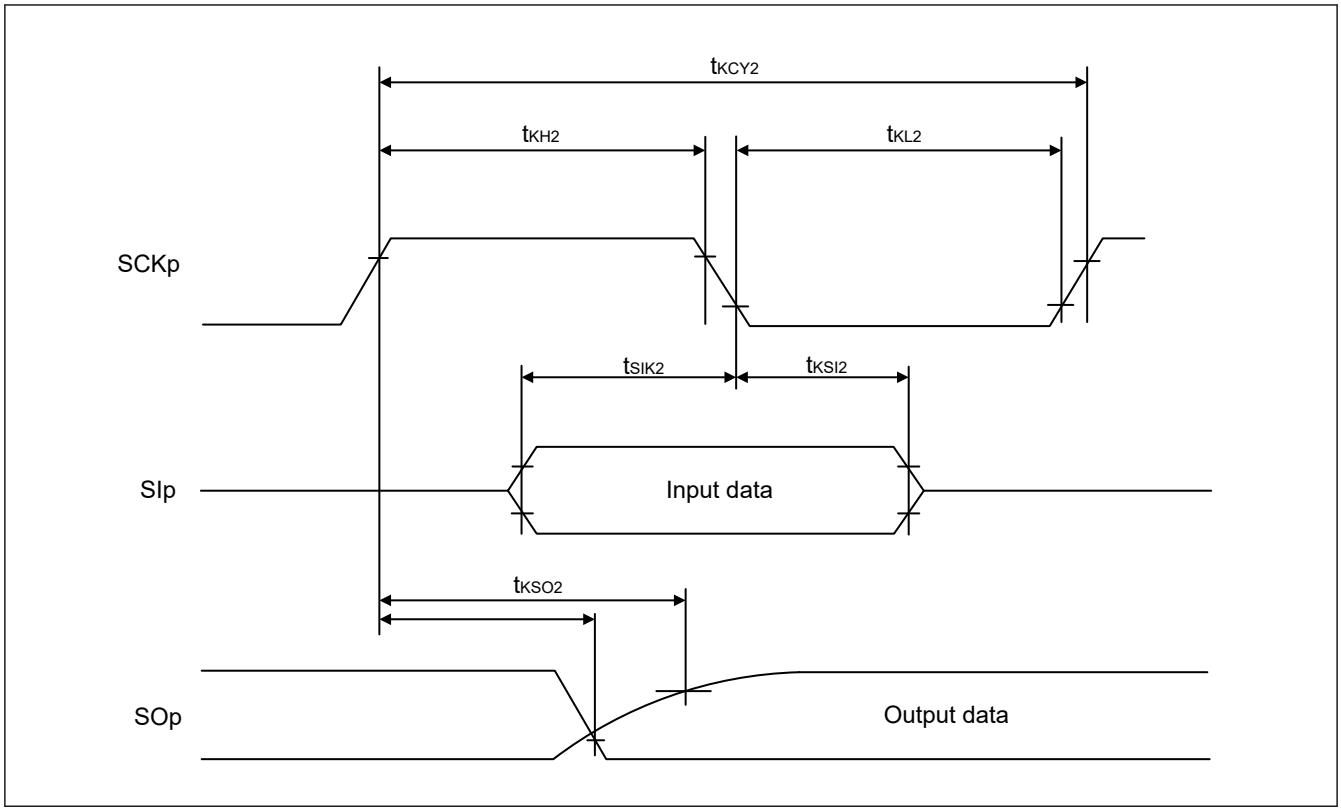


Figure 31.28 Timing of serial transfer in the simplified SPI communications in the slave mode with devices operating at different voltage levels when SCRmn.DCP[1:0] = 01b or 10b

Note:
- p: Simplified SPI number (p = 00, 01, 10, 11, 20, 21), m: Unit number, n: Channel number (mn = 00 to 03, 10, 11), gh: Port number (gh = 100 to 106,109,110,112 to 115, 201, 204 to 208, 212, 213, 301 to 303, 402, 403, 407, 409 to 411, 500 to 502, 915)
- Communications by using P212 and P213 with devices operating at different voltage levels are not possible since P212PFS_A and P213PFS_A registers do not have PIM bit.

Table 31.35 Simplified IIC communications with devices operating at different voltage levels (1.8 V, 2.5 V, or 3 V) (1 of 2)

Conditions: VCC = 1.8 to 5.5 V, VSS = 0 V, Ta = -40 to +125°C

Parameter		Symbol	High-speed mode		Middle-speed mode		Low-speed mode		Unit	Test Conditions
			Min.	Max.	Min.	Max.	Min.	Max.		
SCLr clock frequency	4.0 V ≤ VCC ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 2.7 kΩ	f_{SCL}	—	1000*1	—	1000*1	—	300*1	kHz	Figure 31.30
	2.7 V ≤ VCC < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ		—	1000*1	—	1000*1	—	300*1	kHz	
	4.0 V ≤ VCC ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ		—	400*1	—	400*1	—	300*1	kHz	
	2.7 V ≤ VCC < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ		—	400*1	—	400*1	—	300*1	kHz	
	1.8 V ≤ VCC < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V*2, C_b = 100 pF, R_b = 5.5 kΩ		—	300*1	—	300*1	—	300*1	kHz	
Hold time when SCLr is low	4.0 V ≤ VCC ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 2.7 kΩ	t_{LOW}	475	—	475	—	1550	—	ns	
	2.7 V ≤ VCC < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ		475	—	475	—	1550	—	ns	
	4.0 V ≤ VCC ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ		1150	—	1550	—	1550	—	ns	
	2.7 V ≤ VCC < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ		1150	—	1550	—	1550	—	ns	
	1.8 V ≤ VCC < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V*2, C_b = 100 pF, R_b = 5.5 kΩ		1550	—	1550	—	1550	—	ns	
Hold time when SCLr is high	4.0 V ≤ VCC ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 2.7 kΩ	t_{HIGH}	245	—	245	—	610	—	ns	
	2.7 V ≤ VCC < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ		200	—	200	—	610	—	ns	
	4.0 V ≤ VCC ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ		675	—	675	—	610	—	ns	
	2.7 V ≤ VCC < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ		600	—	600	—	610	—	ns	
	1.8 V ≤ VCC < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V*2, C_b = 100 pF, R_b = 5.5 kΩ		610	—	610	—	610	—	ns	

Table 31.35 Simplified IIC communications with devices operating at different voltage levels (1.8 V, 2.5 V, or 3 V) (2 of 2)

Conditions: VCC = 1.8 to 5.5 V, VSS = 0 V, Ta = -40 to +125°C

Parameter		Symbol	High-speed mode		Middle-speed mode		Low-speed mode		Unit	Test Conditions
			Min.	Max.	Min.	Max.	Min.	Max.		
Data setup time (reception)	4.0 V ≤ VCC ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 2.7 kΩ	$t_{SU:DAT}$	$1/f_{MCK}$ +135*3	—	$1/f_{MCK}$ +135*3	—	$1/f_{MCK}$ +190*3	—	ns	Figure 31.30
	2.7 V ≤ VCC < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ		$1/f_{MCK}$ +135*3	—	$1/f_{MCK}$ +135*3	—	$1/f_{MCK}$ +190*3	—	ns	
	4.0 V ≤ VCC ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ		$1/f_{MCK}$ +190*3	—	$1/f_{MCK}$ +190*3	—	$1/f_{MCK}$ +190*3	—	ns	
	2.7 V ≤ VCC < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ		$1/f_{MCK}$ +190*3	—	$1/f_{MCK}$ +190*3	—	$1/f_{MCK}$ +190*3	—	ns	
	1.8 V ≤ VCC < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V*2, C_b = 100 pF, R_b = 5.5 kΩ		$1/f_{MCK}$ +190*3	—	$1/f_{MCK}$ +190*3	—	$1/f_{MCK}$ +190*3	—	ns	
Data hold time (transmission)	4.0 V ≤ VCC ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 2.7 kΩ	$t_{HD:DAT}$	0	305	0	305	0	305	ns	
	2.7 V ≤ VCC < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ		0	305	0	305	0	305	ns	
	4.0 V ≤ VCC ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ		0	355	0	355	0	355	ns	
	2.7 V ≤ VCC < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ		0	355	0	355	0	355	ns	
	1.8 V ≤ VCC < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V*2, C_b = 100 pF, R_b = 5.5 kΩ		0	405	0	405	0	405	ns	

Note 1. The listed times must be no greater than $f_{MCK}/4$.

Note 2. Use this setting with VCC ≥ V_b.

Note 3. Set f_{MCK} so that it does not exceed the hold time when SCLr is low or high.

Note: Select the TTL input buffer and the N-ch open drain output [withstand voltage of VCC] mode for the SDAr pin and the N-ch open drain output [withstand voltage of VCC] mode for the SCLr pin by using the Port gh Pin Function Select Register (PghPFS_A.PIM and PghPFS_A.NCODR). For V_{IH} and V_{IL}, see the DC characteristics with the TTL input buffer selected.

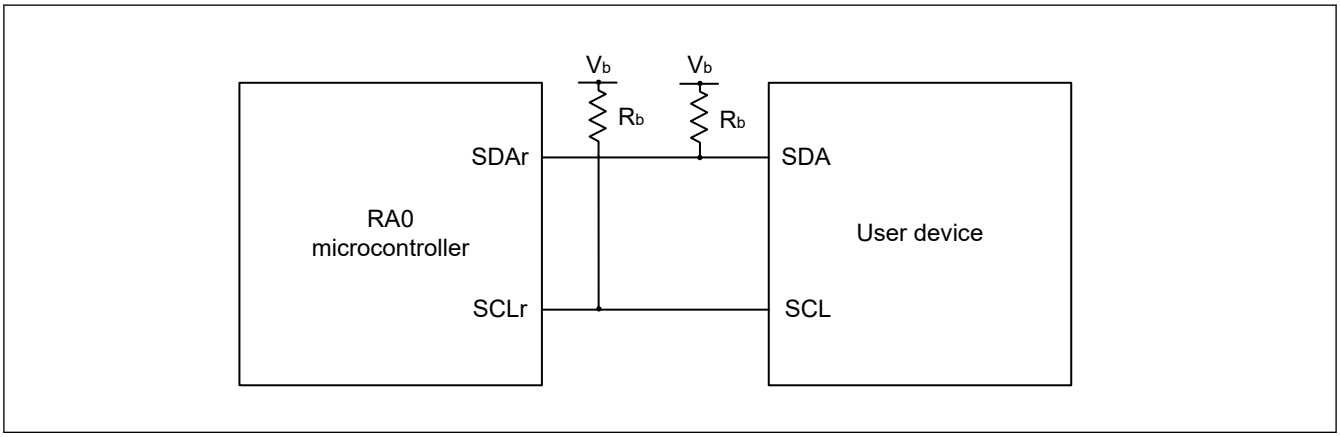


Figure 31.29 Connection in the IIC communications with devices operating at different voltage levels

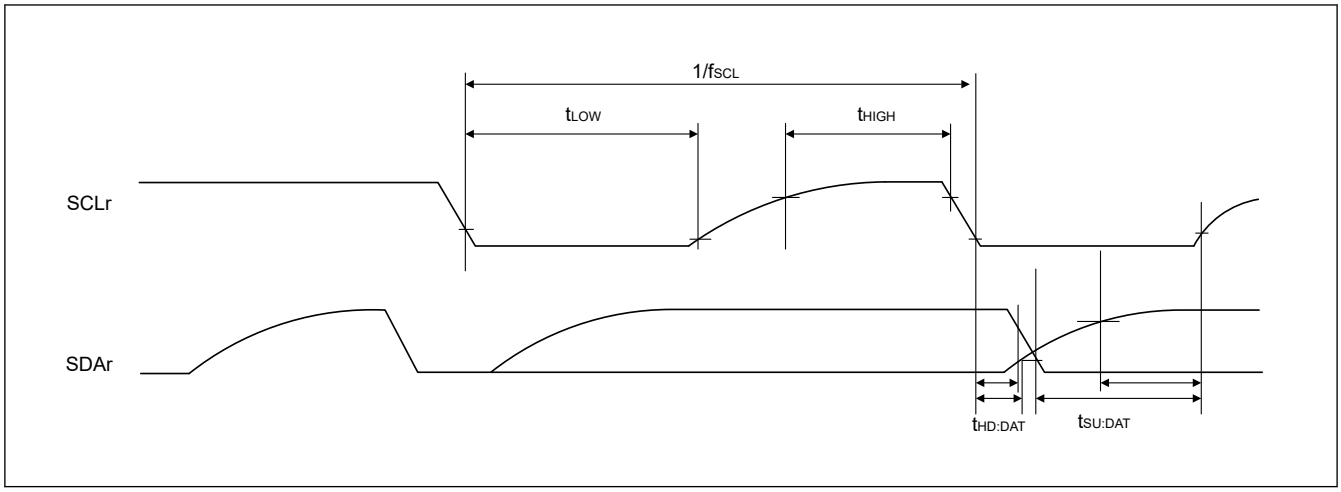


Figure 31.30 Timing of serial transfer in simplified IIC communications with devices operating at different voltage levels

Note:
- R_b[Ω]: Communication line (SDAr, SCLr) pull-up resistance, C_b[F]: Communication line (SDAr, SCLr) load capacitance, V_b[V]: Communication line voltage
- r: Simplified IIC number (r = 00, 01, 10, 11, 20, 21), gh: Port number (gh = 100, 102, 104, 105, 110, 112, 114, 115, 201, 204, 205, 207, 208, 212, 301, 302, 403, 407, 409 to 411, 500, 502)
- f_{MCK}: Serial array unit operation clock frequency
  To set this operating clock, use the CKS bit in the serial mode register mn (SMRmn).
  m: Unit number, n: Channel number (mn = 00 to 03, 10, 11)

31.5.2 UART Interface (UARTA)

Table 31.36 UARTA communications

Conditions: VCC = 1.6 to 5.5 V, VSS = 0 V, Ta = -40 to +125°C

Parameter	Symbol	Min.	Typ.	Max.	Unit	Test conditions
Transfer rate	—	200	0	153600	bps	—

Note: Select the normal input buffer for the RXDAn pin and the normal output mode for the TXDAn pin by using the Port gh Pin Function Select Register (PghPFS_A.PIM and PghPFS_A.NCODR).

Note: n: Unit number (n = 0, 1), gh: Port number (gh = 100 to 103, 105, 106, 109, 110, 205, 206 to 208, 212, 213, 301, 302, 402, 403, 410, 411, 501, 502)

Note: Communications by using P212 and P213 with devices operating at different voltage levels are not possible since P212PFS_A and P213PFS_A registers do not have PIM bit.

31.5.3 I²C Bus Interface (IICA)

Table 31.37 I²C standard mode

Conditions: VCC = 1.6 to 5.5 V, VSS = 0 V, Ta = -40 to +125°C

Parameter		Symbol	Min.	Typ.	Max.	Unit	Test conditions
SCLAn clock frequency	Standard mode: PCLKB ≥ 1 MHz	f_{SCL}	0	—	100	kHz	Figure 31.31
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 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 will be inserted on reception of an acknowledgment (ACK) signal.

Note: n: Unit number (0,1)

Note: Communications by using P212 and P213 with devices operating at different voltage levels are not possible since P212PFS_A and P213PFS_A registers do not have PIM bit.

Note: The maximum value of communication line capacitance (C_b) and communication line pull-up resistor (R_b) are as follows.

C_b = 400 pF, R_b = 2.7 kΩ

Table 31.38 I²C fast mode

Conditions: VCC = 1.8 to 5.5 V, VSS = 0 V, Ta = -40 to +125°C

Parameter		Symbol	Min.	Typ.	Max.	Unit	Test conditions
SCLAn clock frequency	Fast mode: PCLKB ≥ 3.5 MHz 1.8 V ≤ VCC ≤ 5.5 V	f_{SCL}	0	—	400	kHz	Figure 31.31
Setup time of restart condition	1.8 V ≤ VCC ≤ 5.5 V	$t_{SU:STA}$	0.6	—	—	µs	
Hold time*1	1.8 V ≤ VCC ≤ 5.5 V	$t_{HD:STA}$	0.6	—	—	µs	
Hold time when SCLAn is low	1.8 V ≤ VCC ≤ 5.5 V	t_{LOW}	1.3	—	—	µs	
Hold time when SCLAn is high	1.8 V ≤ VCC ≤ 5.5 V	t_{HIGH}	0.6	—	—	µs	
Data setup time (reception)	1.8 V ≤ VCC ≤ 5.5 V	$t_{SU:DAT}$	100	—	—	ns	
Data hold time (transmission)*2	1.8 V ≤ VCC ≤ 5.5 V	$t_{HD:DAT}$	0	—	0.9	µs	
Setup time of stop condition	1.8 V ≤ VCC ≤ 5.5 V	$t_{SU:STO}$	0.6	—	—	µs	
Bus-free time	1.8 V ≤ VCC ≤ 5.5 V	t_{BUF}	1.3	—	—	µs	

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 will be inserted on reception of an acknowledgment (ACK) signal.

Note: Communications by using P212 and P213 with devices operating at different voltage levels are not possible since P212PFS_A and P213PFS_A registers do not have PIM bit.

Note: The maximum value of communication line capacitance (C_b) and communication line pull-up resistor (R_b) are as follows.

C_b = 320 pF, R_b = 1.1 kΩ

Table 31.39 I²C fast mode plus

Conditions: VCC = 2.7 to 5.5 V, VSS = 0 V, Ta = -40 to +125°C

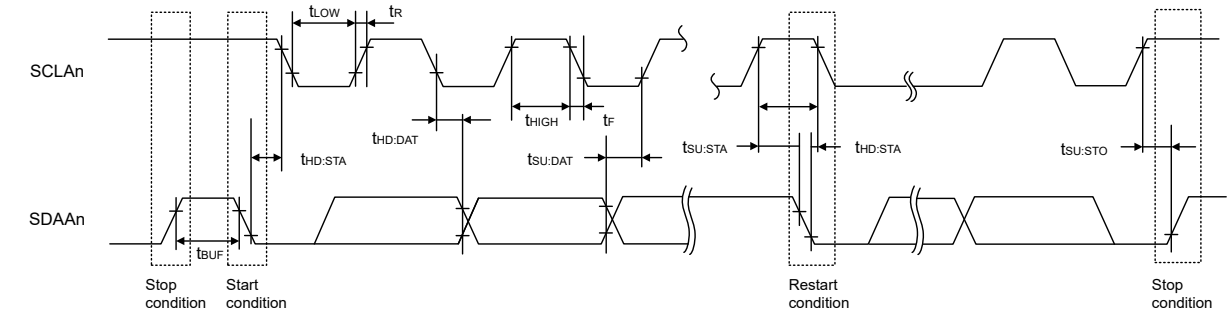
Parameter		Symbol	Min.	Typ.	Max.	Unit	Test conditions
SCLAn clock frequency	Fast mode plus: PCLKB ≥ 10 MHz 2.7 V ≤ VCC ≤ 5.5 V	f_{SCL}	0	—	1000	kHz	Figure 31.31
Setup time of restart condition	2.7 V ≤ VCC ≤ 5.5 V	$t_{SU:STA}$	0.26	—	—	µs	
Hold time*1	2.7 V ≤ VCC ≤ 5.5 V	$t_{HD:STA}$	0.26	—	—	µs	
Hold time when SCLAn is low	2.7 V ≤ VCC ≤ 5.5 V	t_{LOW}	0.5	—	—	µs	
Hold time when SCLAn is high	2.7 V ≤ VCC ≤ 5.5 V	t_{HIGH}	0.26	—	—	µs	
Data setup time (reception)	2.7 V ≤ VCC ≤ 5.5 V	$t_{SU:DAT}$	50	—	—	ns	
Data hold time (transmission)*2	2.7 V ≤ VCC ≤ 5.5 V	$t_{HD:DAT}$	0	—	0.45	µs	
Setup time of stop condition	2.7 V ≤ VCC ≤ 5.5 V	$t_{SU:STO}$	0.26	—	—	µs	
Bus-free time	2.7 V ≤ VCC ≤ 5.5 V	t_{BUF}	0.5	—	—	µs	

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 will be inserted on reception of an acknowledgment (ACK) signal.

Note: Communications by using P212 and P213 with devices operating at different voltage levels are not possible since P212PFS_A and P213PFS_A registers do not have PIM bit.

Note: The maximum value of communication line capacitance (C_b) and communication line pull-up resistor (R_b) are as follows.

C_b = 120 pF, R_b = 1.1 kΩ



Note: n = 0

Figure 31.31 IICA serial transfer timing

31.6 Analog Characteristics

31.6.1 A/D Converter Characteristics

Table 31.40 A/D conversion characteristics in Normal modes 1 and 2 (1 of 2)

Conditions: 2.4 V ≤ VREFH0 ≤ VCC ≤ 5.5 V, VSS = 0 V, Ta = -40 to +125 °C
Reference voltage range applied to the VREFH0 (ADREFP[1:0] = 01b) and VREFL0 (ADREFM = 1b).
Target pins: AN000 to AN012, AN021 to AN022, internal reference voltage, and temperature sensor output voltage

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
Resolution	RES	8	—	12	bit	—
Conversion clock	f_{AD}	1	—	32	MHz	—

Table 31.40 A/D conversion characteristics in Normal modes 1 and 2 (2 of 2)

Conditions: 2.4 V ≤ VREFH0 ≤ VCC ≤ 5.5 V, VSS = 0 V, Ta = -40 to +125 °C
Reference voltage range applied to the VREFH0 (ADREFP[1:0] = 01b) and VREFL0 (ADREFM = 1b).
Target pins: AN000 to AN012, AN021 to AN022, internal reference voltage, and temperature sensor output voltage

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Overall error*1 *3 *4 *5	12-bit resolution	AINL	—	—	±7.5	LSB	4.5 V ≤ VREFH0 = VCC ≤ 5.5 V
			—	—	±9.0	LSB	2.7 V ≤ VREFH0 = VCC ≤ 5.5 V
			—	—	±9.0	LSB	2.4 V ≤ VREFH0 = VCC ≤ 5.5 V
Conversion time*6	12-bit resolution	t_{CONV}	2.0	—	—	µs	4.5 V ≤ VREFH0 = VCC ≤ 5.5 V
			2.0	—	—	µs	2.7 V ≤ VREFH0 = VCC ≤ 5.5 V
			2.0	—	—	µs	2.4 V ≤ VREFH0 = VCC ≤ 5.5 V
Zero-scale error*1 *2 *3 *4 *5	12-bit resolution	E_{ZS}	—	—	±0.17	%FSR	4.5 V ≤ VREFH0 = VCC ≤ 5.5 V
			—	—	±0.21	%FSR	2.7 V ≤ VREFH0 = VCC ≤ 5.5 V
			—	—	±0.21	%FSR	2.4 V ≤ VREFH0 =VCC ≤ 5.5 V
Full-scale error*1 *2 *3 *4 *5	12-bit resolution	E_{FS}	—	—	±0.17	%FSR	4.5 V ≤ VREFH0 = VCC ≤ 5.5 V
			—	—	±0.21	%FSR	2.7 V ≤ VREFH0 = VCC ≤ 5.5 V
			—	—	±0.21	%FSR	2.4 V ≤ VREFH0 = VCC ≤ 5.5 V
Integral linearity error*1 *4 *5	12-bit resolution	ILE	—	—	±3.0	LSB	4.5 V ≤ VREFH0 = VCC ≤ 5.5 V
			—	—	±3.0	LSB	2.7 V ≤ VREFH0 = VCC ≤ 5.5 V
			—	—	±3.0	LSB	2.4 V ≤ VREFH0 = VCC ≤ 5.5 V
Differential linearity error*1	12-bit resolution	DLE	—	±1.0	—	LSB	4.5 V ≤ VREFH0 = VCC ≤ 5.5 V
			—	±1.0	—	LSB	2.7 V ≤ VREFH0 = VCC ≤ 5.5 V
			—	±1.0	—	LSB	2.4 V ≤ VREFH0 = VCC ≤ 5.5 V
Analog input voltage		V_{AIN}	0	—	VREFH0	V	—

Note 1. This value does not include the quantization error (±1/2 LSB).
Note 2. This value is indicated as a ratio (%FSR) to the full-scale value.
Note 3. When pins AN021 to AN022 are selected as the target pins for conversion, the maximum values are as follows.
Overall error: Add ±3 LSB to the maximum value.
Zero-scale/full-scale error: Add ±0.04%FSR to the maximum value.
Note 4. When reference voltage (+) = VCC (ADREFP[1:0] = 00b) and reference voltage (-) = VSS (ADREFM = 0b), the maximum values are as follows.
Overall error: Add ±10 LSB to the maximum value.
Zero-scale/full-scale error: Add ±0.25%FSR to the maximum value.
Integral linearity error: Add ±4 LSB to the maximum value.
Note 5. When VREFH0 < VCC, the maximum values are as follows.
Overall error / Zero-scale error / Full-scale error: Add ±0.75 LSB × (VCC − VREFH0) to the maximum value.
Integral linearity error: Add ±0.2 LSB × (VCC − VREFH0) to the maximum value.
Note 6. 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 standard mode 2 with the longer sampling time.

Table 31.41 A/D conversion characteristics in Low-voltage modes 1 and 2 (1) (1 of 2)

Conditions: 1.6 V ≤ VREFH0 ≤ VCC ≤ 5.5 V, VSS = 0 V, Ta = -40 to +125°C
Reference voltage range applied to the VREFH0 (ADREFP[1:0] = 01b) and VREFL0 (ADREFM = 1b).
Target pins: AN000 to AN012, AN021 to AN022, internal reference voltage*7, and temperature sensor output voltage*7.

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Resolution		RES	8	—	12	bit	—
Conversion clock		f_{AD}	1	—	24	MHz	—
Overall error*1 *3 *4 *5	12-bit resolution	AINL	—	—	±9	LSB	2.7 V ≤ VREFH0 = VCC ≤ 5.5 V
			—	—	±9	LSB	2.4 V ≤ VREFH0 = VCC ≤ 5.5 V
			—	—	±11.5	LSB	1.8 V ≤ VREFH0 = VCC ≤ 5.5 V
			—	—	±12.0	LSB	1.6 V ≤ VREFH0 = VCC ≤ 5.5 V

Table 31.41 A/D conversion characteristics in Low-voltage modes 1 and 2 (1) (2 of 2)

Conditions: 1.6 V ≤ VREFH0 ≤ VCC ≤ 5.5 V, VSS = 0 V, Ta = -40 to +125°C
Reference voltage range applied to the VREFH0 (ADREFP[1:0] = 01b) and VREFL0 (ADREFM = 1b).
Target pins: AN000 to AN012, AN021 to AN022, internal reference voltage*7, and temperature sensor output voltage*7.

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Conversion time*6	12-bit resolution	t_{CONV}	3.3	—	—	µs	2.7 V ≤ VREFH0 = VCC ≤ 5.5 V
			5.0	—	—	µs	2.4 V ≤ VREFH0 = VCC ≤ 5.5 V
			10.0	—	—	µs	1.8 V ≤ VREFH0 = VCC ≤ 5.5 V
			20.0	—	—	µs	1.6 V ≤ VREFH0 = VCC ≤ 5.5 V
Zero-scale error*1 *2 *3 *4 *5	12-bit resolution	E_{ZS}	—	—	±0.21	%FSR	2.7 V ≤ VREFH0 = VCC ≤ 5.5 V
			—	—	±0.21	%FSR	2.4 V ≤ VREFH0 = VCC ≤ 5.5 V
			—	—	±0.27	%FSR	1.8 V ≤ VREFH0 = VCC ≤ 5.5 V
			—	—	±0.28	%FSR	1.6 V ≤ VREFH0 = VCC ≤ 5.5 V
Full-scale error*1 *2 *3 *4 *5	12-bit resolution	E_{FS}	—	—	±0.21	%FSR	2.7 V ≤ VREFH0 = VCC ≤ 5.5 V
			—	—	±0.21	%FSR	2.4 V ≤ VREFH0 = VCC ≤ 5.5 V
			—	—	±0.27	%FSR	1.8 V ≤ VREFH0 = VCC ≤ 5.5 V
			—	—	±0.28	%FSR	1.6 V ≤ VREFH0 = VCC ≤ 5.5 V
Integral linearity error*1 *4 *5	12-bit resolution	ILE	—	—	±4.0	LSB	2.7 V ≤ VREFH0 = VCC ≤ 5.5 V
			—	—	±4.0	LSB	2.4 V ≤ VREFH0 = VCC ≤ 5.5 V
			—	—	±4.5	LSB	1.8 V ≤ VREFH0 = VCC ≤ 5.5 V
			—	—	±4.5	LSB	1.6 V ≤ VREFH0 = VCC ≤ 5.5 V
Differential linearity error*1	12-bit resolution	DLE	—	±1.5	—	LSB	2.7 V ≤ VREFH0 = VCC ≤ 5.5 V
			—	±1.5	—	LSB	2.4 V ≤ VREFH0 = VCC ≤ 5.5 V
			—	±2.0	—	LSB	1.8 V ≤ VREFH0 = VCC ≤ 5.5 V
			—	±2.0	—	LSB	1.6 V ≤ VREFH0 = VCC ≤ 5.5 V
Analog input voltage		V_{AIN}	0	—	VREFH0	V	—

Note 1. This value does not include the quantization error (±1/2 LSB).
Note 2. This value is indicated as a ratio (%FSR) to the full-scale value.
Note 3. When pins AN021 to AN022 are selected as the target pins for conversion, the maximum values are as follows.
Overall error: Add ±3 LSB to the maximum value.
Zero-scale/full-scale error: Add ±0.04%FSR to the maximum value.
Note 4. When reference voltage (+) = VCC (ADREFP[1:0] = 00b) and reference voltage (-) = VSS (ADREFM = 0b), the maximum values are as follows.
Overall error: Add ±10 LSB to the maximum value.
Zero-scale/full-scale error: Add ±0.25%FSR to the maximum value.
Integral linearity error: Add ±4 LSB to the maximum value.
Note 5. When VREFH0 < VCC, the maximum values are as follows.
Overall error / Zero-scale error / Full-scale error: Add ±0.75 LSB × (VCC − VREFH0) to the maximum value.
Integral linearity error: Add ±0.2 LSB × (VCC − VREFH0) to the maximum value.
Note 6. 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 standard mode 2 with the longer sampling time, and use the conversion clock (f_{AD}) of no more than 16 MHz.
Note 7. If the internal reference voltage or temperature sensor output voltage is to be A/D converted, VCC must be at least 1.8 V.

Table 31.42 A/D conversion characteristics in Low-voltage modes 1 and 2 (2) (1 of 2)

Conditions: 1.8 V ≤ VCC ≤ 5.5 V, VSS = 0 V, Ta = -40 to +125°C
Reference voltage range applied to the internal reference voltage (ADREFP[1:0] = 10b) and VREFL0 (ADREFM = 1b).

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
Resolution	RES	8			bit	—
Conversion clock	f_{AD}	1	—	2	MHz	1.8 V ≤ VCC ≤ 5.5 V
Zero-scale error*1 *2 *4	E_{ZS}	—	—	±0.6	%FSR	1.8 V ≤ VCC ≤ 5.5 V
Integral linearity error*1 *4	ILE	—	—	±2.0	LSB	1.8 V ≤ VCC ≤ 5.5 V

Table 31.42 A/D conversion characteristics in Low-voltage modes 1 and 2 (2) (2 of 2)

Conditions: 1.8 V ≤ VCC ≤ 5.5 V, VSS = 0 V, Ta = -40 to +125°C
Reference voltage range applied to the internal reference voltage (ADREFP[1:0] = 10b) and VREFL0 (ADREFM = 1b).

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
Differential linearity error*1	DLE	—	±1.0	—	LSB	1.8 V ≤ VCC ≤ 5.5 V
Analog input voltage	V_{AIN}	0	—	VBGR*3	V	—

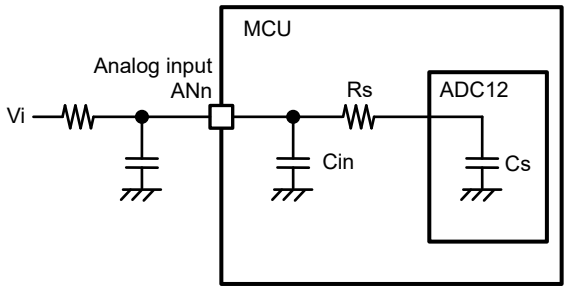
Note 1. This value does not include the quantization error (±1/2 LSB).
Note 2. This value is indicated as a ratio (%FSR) to the full-scale value.
Note 3. Refer to Table 31.44.
Note 4. When the reference voltage (-) is selected as VSS, the maximum values are as follows.
Zero-scale error: Add ±0.35%FSR to the maximum value.
Integral linearity error: Add ±0.5 LSB to the maximum value.

Table 31.43 Resistance and capacitance values of equivalent circuit (Reference data)

Parameter			Min	Typ	Max	Unit	Test conditions
Analog input capacitance	Cin	Refer to I/O input capacitance (Cin), see Table 31.11.					
	Cs*2	High-precision channel*1	—	—	9	pF	—
		Normal-precision channel*1	—	—	10		—
Analog input resistance	Rs*2	High-precision channel*1	—	—	11	kΩ	VCC = 2.4 to 5.5 V
			—	—	55		VCC = 1.8 to 2.4 V
			—	—	110		VCC = 1.6 to 1.8 V
		Normal-precision channel*1	—	—	12		VCC = 2.4 to 5.5 V
			—	—	60		VCC = 1.8 to 2.4 V
			—	—	120		VCC = 1.6 to 1.8 V

Note 1. AN000 to AN012 are high-precision channels. AN021 and AN022 are normal-precision channels.
Note 2. These values are based on simulation. They are not production tested.

Figure 31.32 shows the equivalent circuit for analog input.



Note: Terminal leakage current is not shown in this figure.

Figure 31.32 Equivalent circuit for analog input

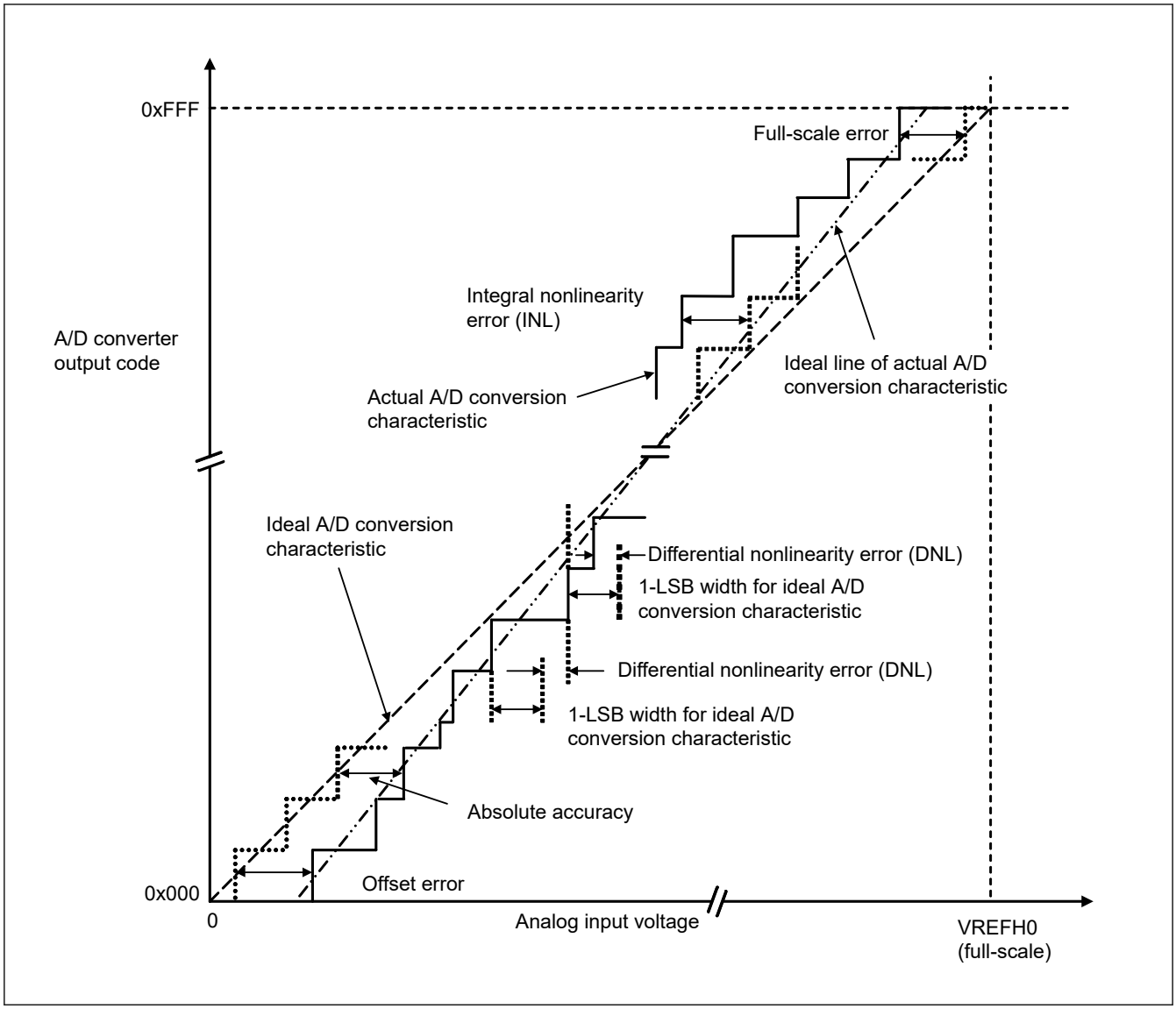


Figure 31.33 Illustration of 12-bit A/D converter characteristic terms

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 VREFH0 = 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 the 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.

31.6.2 Temperature Sensor/Internal Reference Voltage Characteristics

Table 31.44 Temperature sensor/internal reference voltage characteristics

Conditions: 1.8 V ≤ VCC ≤ 5.5 V, VSS = 0 V, Ta = -40 to +125 °C

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
Temperature sensor output voltage	V_{TMPS25}	—	1.05	—	V	25 °C
Internal reference voltage	V_{BGR}	1.40	1.48	1.56	V	—
Temperature coefficient	F_{VTMPS}	—	-3.3	—	mV/°C	—
Operation stabilization wait time	t_{AMP}	5	—	—	µs	—

31.6.3 POR Characteristics

Table 31.45 POR characteristics

Conditions: VSS = 0 V, Ta = -40 to +125°C

Parameter	Symbol	Min	Typ	Max	Unit	Test Conditions
Detection voltage	VPOR VPDR	1.43	1.50	1.57	V	—
Minimum pulse width*1	TPW	300	—	—	µs	—

Note 1. This width is the minimum time required for a POR reset when VCC falls below VPDR. This width is also the minimum time required for a POR reset from when VCC falls below 0.7 V to when VCC exceeds VPOR in the Software standby mode or while the main system clock is stopped through setting HOCOCR.HCSTOP bit and MOSCCR.MOSTP bit.

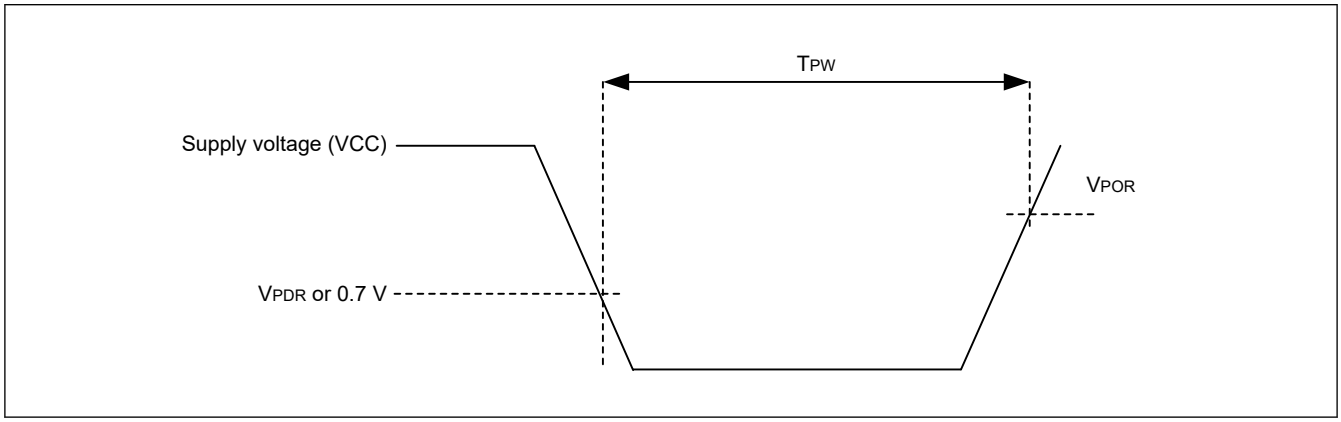


Figure 31.34 Minimum VCC pulse width

31.6.4 LVD Characteristics

Table 31.46 LVD0 characteristics

Conditions: VPDR ≤ VCC ≤ 5.5 V, VSS = 0 V, Ta = -40 to +125°C

Parameter		Symbol	Min	Typ	Max	Unit	Test Conditions
Detection voltage	Supply voltage level	V_{det0_0}	3.84	3.96	4.08	V	The power supply voltage is rising.
			3.76	3.88	4.00	V	The power supply voltage is falling.
		V_{det0_1}	2.88	2.97	3.06	V	The power supply voltage is rising.
			2.82	2.91	3.00	V	The power supply voltage is falling.
		V_{det0_2}	2.59	2.67	2.75	V	The power supply voltage is rising.
			2.54	2.62	2.70	V	The power supply voltage is falling.
		V_{det0_3}	2.31	2.38	2.45	V	The power supply voltage is rising.
			2.26	2.33	2.40	V	The power supply voltage is falling.
		V_{det0_4}	1.84	1.90	1.95	V	The power supply voltage is rising.
			1.80	1.86	1.91	V	The power supply voltage is falling.
		V_{det0_5}	1.64	1.69	1.74	V	The power supply voltage is rising.
			1.60	1.65	1.70	V	The power supply voltage is falling.
Minimum pulse width		t_{LW0}	500	—	—	µs	—
Detection delay time		t_{det0}	—	—	500	µs	—

Table 31.47 LVD1 characteristics (1 of 2)

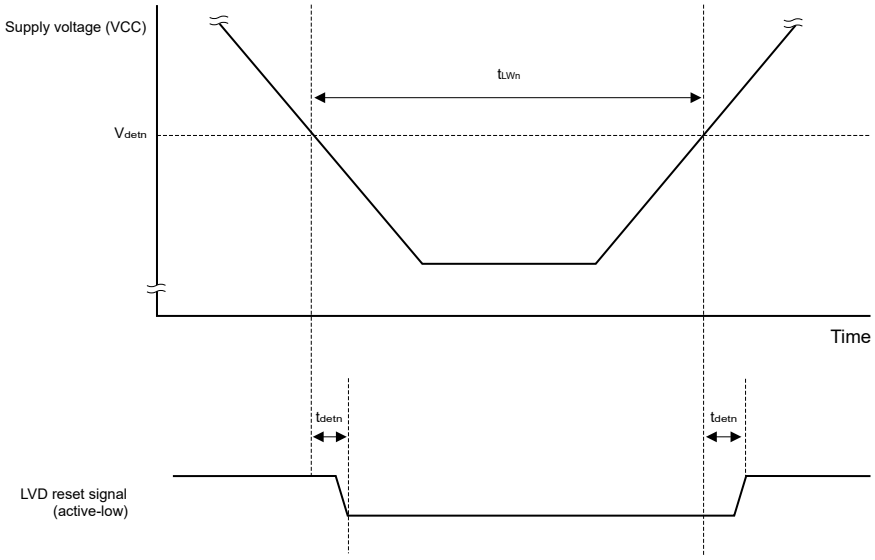
Conditions: VPDR ≤ VCC ≤ 5.5 V, VSS = 0 V, Ta = -40 to +125°C

Parameter		Symbol	Min	Typ	Max	Unit	Test Conditions
Detection voltage	Supply voltage level	V_{det1_0}	4.08	4.16	4.24	V	The power supply voltage is rising.
			4.00	4.08	4.16	V	The power supply voltage is falling.
		V_{det1_1}	3.88	3.96	4.04	V	The power supply voltage is rising.
			3.80	3.88	3.96	V	The power supply voltage is falling.
		V_{det1_2}	3.68	3.75	3.82	V	The power supply voltage is rising.
			3.60	3.67	3.74	V	The power supply voltage is falling.
		V_{det1_3}	3.48	3.55	3.62	V	The power supply voltage is rising.
			3.40	3.47	3.54	V	The power supply voltage is falling.
		V_{det1_4}	3.28	3.35	3.42	V	The power supply voltage is rising.
			3.20	3.27	3.34	V	The power supply voltage is falling.
		V_{det1_5}	3.07	3.13	3.19	V	The power supply voltage is rising.
			3.00	3.06	3.12	V	The power supply voltage is falling.
		V_{det1_6}	2.91	2.97	3.03	V	The power supply voltage is rising.
			2.85	2.91	2.97	V	The power supply voltage is falling.
		V_{det1_7}	2.76	2.82	2.87	V	The power supply voltage is rising.
			2.70	2.76	2.81	V	The power supply voltage is falling.
		V_{det1_8}	2.61	2.66	2.71	V	The power supply voltage is rising.
			2.55	2.60	2.65	V	The power supply voltage is falling.
		V_{det1_9}	2.45	2.50	2.55	V	The power supply voltage is rising.
			2.40	2.45	2.50	V	The power supply voltage is falling.
		V_{det1_A}	2.35	2.40	2.45	V	The power supply voltage is rising.
			2.30	2.35	2.40	V	The power supply voltage is falling.

Table 31.47 LVD1 characteristics (2 of 2)

Conditions: VPDR ≤ VCC ≤ 5.5 V, VSS = 0 V, Ta = -40 to +125°C

Parameter		Symbol	Min	Typ	Max	Unit	Test Conditions
Detection voltage	Supply voltage level	V_{det1_B}	2.25	2.30	2.34	V	The power supply voltage is rising.
			2.20	2.25	2.29	V	The power supply voltage is falling.
		V_{det1_C}	2.15	2.20	2.24	V	The power supply voltage is rising.
			2.10	2.15	2.19	V	The power supply voltage is falling.
		V_{det1_D}	2.05	2.09	2.13	V	The power supply voltage is rising.
			2.00	2.04	2.08	V	The power supply voltage is falling.
		V_{det1_E}	1.94	1.98	2.02	V	The power supply voltage is rising.
			1.90	1.94	1.98	V	The power supply voltage is falling.
		V_{det1_F}	1.84	1.88	1.91	V	The power supply voltage is rising.
			1.80	1.84	1.87	V	The power supply voltage is falling.
		V_{det1_10}	1.74	1.78	1.81	V	The power supply voltage is rising.
			1.70	1.74	1.77	V	The power supply voltage is falling.
		V_{det1_11}	1.64	1.67	1.70	V	The power supply voltage is rising.
			1.60	1.63	1.66	V	The power supply voltage is falling.
Minimum pulse width		t_{LW1}	500	—	—	µs	—
Detection delay time		t_{det1}	—	—	500	µs	—
LVD1 detection voltage stabilization time (after changing the LVD1 detection voltage)		$t_{d(E-A)}$	—	—	1500	µs	—



Note: n = 0, 1

Figure 31.35 Voltage detection circuit timing

31.6.5 Power Supply Voltage Rising Slope Characteristics

Table 31.48 Power supply voltage rising slope characteristics

Conditions: VSS = 0 V, Ta = -40 to +125°C

Parameter	Symbol	Min	Typ	Max	Unit	Test Conditions
Power supply voltage rising slope	S_{VCC}	—	—	54	V/ms	—

Note: Make sure to keep the internal reset state by the LVD0 circuit or an external reset until VCC reaches the operating voltage range shown in AC characteristics.

31.7 RAM Data Retention Characteristics

Table 31.49 RAM data retention characteristics

Conditions: VSS = 0 V, Ta = -40 to +125°C

Parameter	Symbol	Min	Typ	Max	Unit	Test Conditions
Data retention supply voltage	V_{CCDR}	1.43*1	—	5.5	V	—

Note 1. This voltage depends on the POR detection voltage. When the voltage drops, the data in RAM are retained until a POR is applied, but are not retained following a POR.

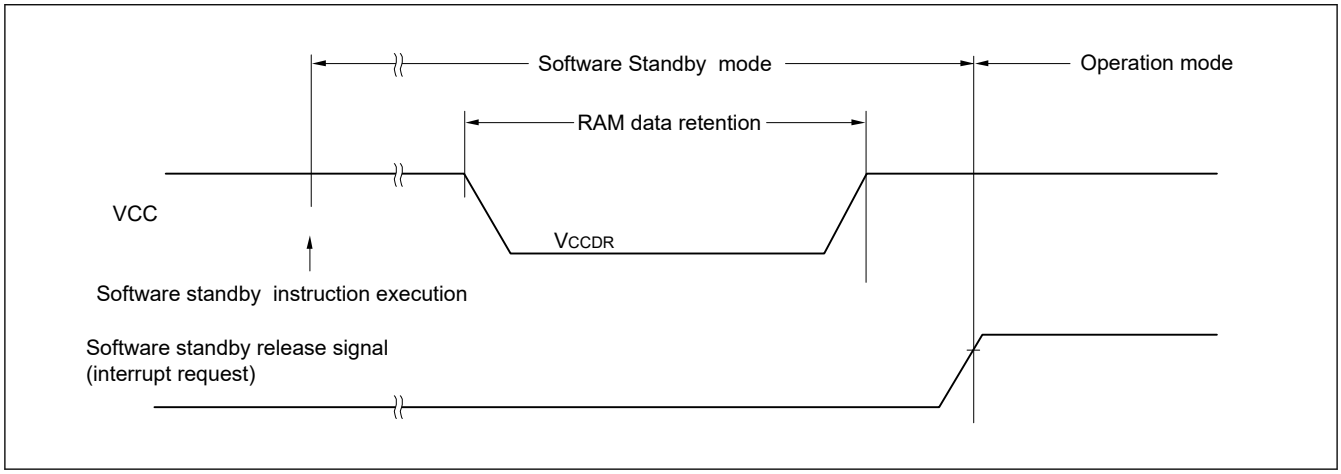


Figure 31.36 RAM data retention

31.8 Flash Memory Programming Characteristics

Table 31.50 Flash memory programming characteristics

Conditions: 1.8 V ≤ VCC ≤ 5.5 V, VSS = 0 V, Ta = -40 to +125°C

Parameter	Symbol	Min	Typ	Max	Unit	Test Conditions
CPU/peripheral hardware clock frequency	I_{CLK}	1	—	32	MHz	—
Number of code flash rewrites*1 *2 *3	Cerwr	10000	—	—	Times	Retained for 10 years Ta = 85°C
		1000	—	—		Retained for 20 years Ta = 85°C
Number of data flash rewrites*1 *2 *3		—	1000000	—		Retained for 1 year Ta = 25°C
		100000	—	—		Retained for 5 years Ta = 85°C
		10000	—	—		Retained for 20 years Ta = 85°C

Note 1. 1 erase + 1 write after the erase is regarded as 1 rewrite. The retaining years are until next rewrite after the rewrite.
Note 2. The listed numbers of times apply when using the flash memory programmer and self-programming.
Note 3. These are the characteristics of the flash memory and the results obtained from reliability testing by Renesas Electronics Corporation.

Table 31.51 Code flash memory characteristics

Conditions: 1.8 V ≤ VCC ≤ 5.5 V, VSS = 0 V, Ta = -40 to +125°C

Parameter		Symbol	ICLK = 1 MHz			ICLK = 2 MHz, 3 MHz			4 MHz ≤ ICLK < 8 MHz			8 MHz ≤ ICLK < 32 MHz			ICLK = 32 MHz			Unit
			Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
Programming time	4 bytes	t_{P4}	—	74.7	656.5	—	51.0	464.6	—	41.7	384.8	—	37.1	346.2	—	34.2	321.9	µs
Erasure time	2 Kbytes	t_{E2K}	—	10.4	312.2	—	7.7	258.5	—	6.4	231.8	—	5.8	218.4	—	5.6	214.4	ms
Blank checking time	4 bytes	t_{BC4}	—	—	38.4	—	—	19.2	—	—	13.1	—	—	10.2	—	—	8.3	µs
	2 Kbytes	t_{BC2K}	—	—	2618.9	—	—	1309.5	—	—	658.3	—	—	332.8	—	—	234.1	µs
Time taken to forcibly stop the erasure		t_{SED}	—	—	18.0	—	—	14.0	—	—	12.0	—	—	11.0	—	—	10.3	µs
Security setting time		t_{AWSSAS}	—	18.0	525.5	—	14.3	468.7	—	12.5	440.7	—	11.6	426.7	—	11.3	422.3	ms
Time until programming starts following cancellation of the Software standby instruction		—	20	—	—	20	—	—	20	—	—	20	—	—	20	—	—	µs
Flash memory mode transition wait time 1		t_{DIS}	2	—	—	2	—	—	2	—	—	2	—	—	2	—	—	µs
Flash memory mode transition wait time 2		t_{MS}	15	—	—	15	—	—	15	—	—	15	—	—	15	—	—	µs

Note: The listed values do not include the time until the operations of the flash memory start following execution of an instruction by software.

Table 31.52 Data flash memory characteristics

Conditions: 1.8 V ≤ VCC ≤ 5.5 V, VSS = 0 V, Ta = -40 to +125°C

Parameter		Symbol	ICLK = 1 MHz			ICLK = 2 MHz, 3 MHz			4 MHz ≤ ICLK < 8 MHz			8 MHz ≤ ICLK < 32 MHz			ICLK = 32 MHz			Unit
			Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
Programming time	1 byte	t_{P4}	—	74.7	656.5	—	51.0	464.6	—	41.7	384.8	—	37.1	346.2	—	34.2	321.9	µs
Erasure time	256 bytes	t_{E2K}	—	7.8	259.2	—	6.4	232.0	—	5.8	218.5	—	5.5	211.8	—	5.4	209.7	ms
Blank checking time	1 byte	t_{BC4}	—	—	38.4	—	—	19.2	—	—	13.1	—	—	10.2	—	—	8.3	µs
	256 bytes	t_{BC2K}	—	—	1326.1	—	—	663.1	—	—	335.1	—	—	171.2	—	—	121.0	µs
Time taken to forcibly stop the erasure		t_{SED}	—	—	18.0	—	—	14.0	—	—	12.0	—	—	11.0	—	—	10.3	µs
Time until programming starts following cancellation of the Software standby instruction		—	20	—	—	20	—	—	20	—	—	20	—	—	20	—	—	µs
Time until reading starts following setting DFLEN to 1		t_{DSTOP}	0.25	—	—	0.25	—	—	0.25	—	—	0.25	—	—	0.25	—	—	µs
Flash memory mode transition wait time 1		t_{DIS}	2	—	—	2	—	—	2	—	—	2	—	—	2	—	—	µs
Flash memory mode transition wait time 2		t_{MS}	15	—	—	15	—	—	15	—	—	15	—	—	15	—	—	µs

Note: The listed values do not include the time until the operations of the flash memory start following execution of an instruction by software.

31.9 Serial Wire Debug (SWD)

Table 31.53 SWD characteristics (1) (1 of 2)

Conditions: VCC = 2.4 to 5.5 V

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
SWCLK clock cycle time	$t_{SWCKcyc}$	80	—	—	ns	Figure 31.37
SWCLK clock high pulse width	t_{SWCKH}	35	—	—	ns	
SWCLK clock low pulse width	t_{SECKL}	35	—	—	ns	
SWCLK clock rise time	t_{SWCKr}	—	—	5	ns	
SWCLK clock fall time	t_{SWCKf}	—	—	5	ns	

Table 31.53 SWD characteristics (1) (2 of 2)

Conditions: VCC = 2.4 to 5.5 V

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
SWDIO setup time	t_{SWDS}	16	—	—	ns	Figure 31.38
SWDIO hold time	t_{SWDH}	16	—	—	ns	
SWDIO data delay time	t_{SWDD}	2	—	70	ns	

Table 31.54 SWD characteristics (2)

Conditions: VCC = 1.6 to 2.4 V

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
SWCLK clock cycle time	$t_{SWCKcyc}$	250	—	—	ns	Figure 31.37
SWCLK clock high pulse width	t_{SWCKH}	120	—	—	ns	
SWCLK clock low pulse width	t_{SECKL}	120	—	—	ns	
SWCLK clock rise time	t_{SWCKr}	—	—	5	ns	
SWCLK clock fall time	t_{SWCKf}	—	—	5	ns	
SWDIO setup time	t_{SWDS}	50	—	—	ns	Figure 31.38
SWDIO hold time	t_{SWDH}	50	—	—	ns	
SWDIO data delay time	t_{SWDD}	2	—	170	ns	

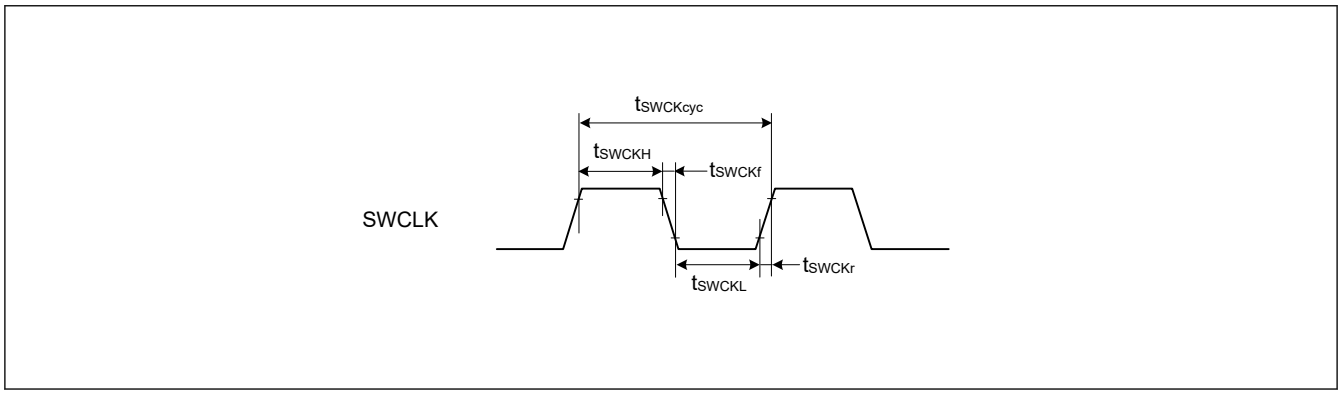


Figure 31.37 SWD SWCLK timing

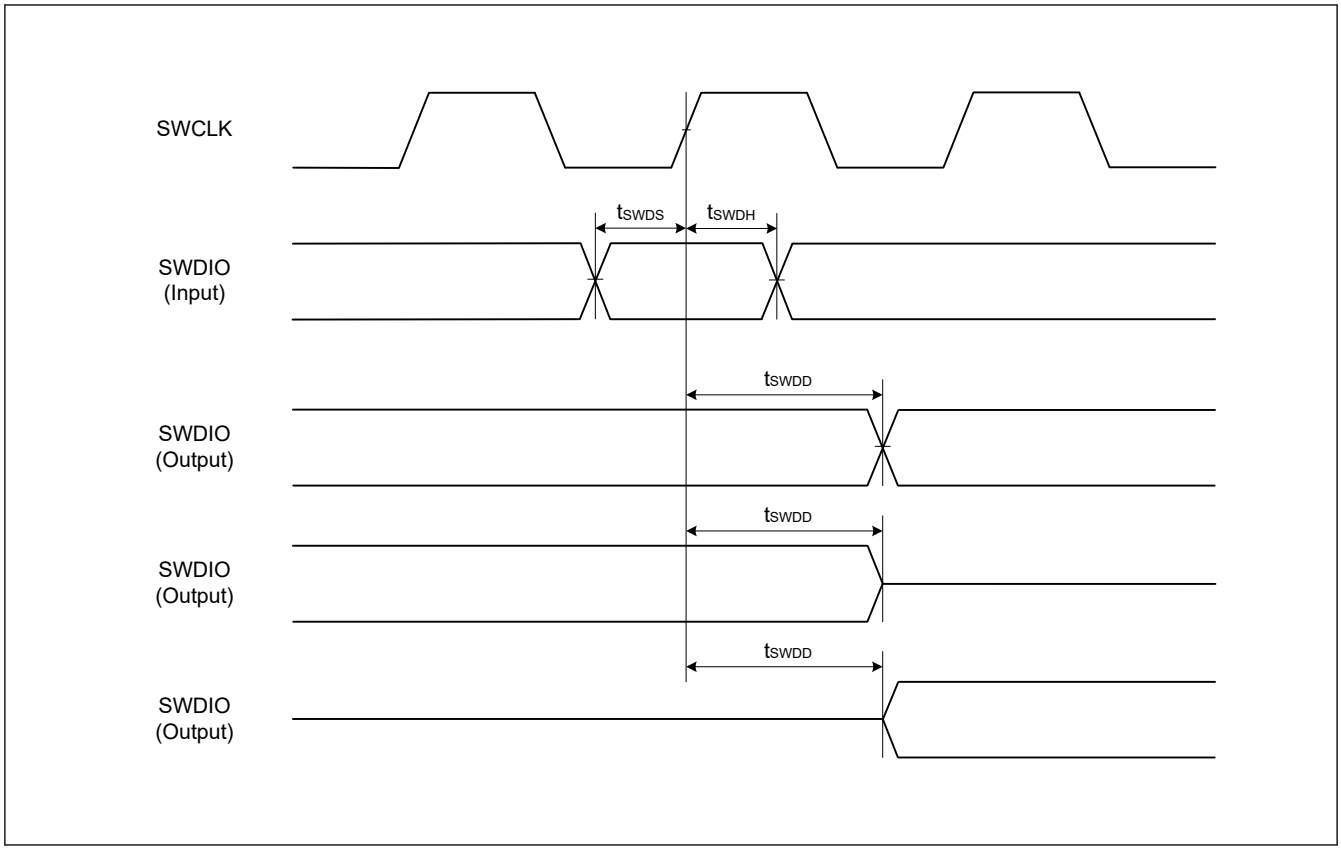


Figure 31.38 SWD input/output timing

Appendix 1. Port States in each Processing Mode

Table A1.1 Port states in each processing mode (1 of 5)

Port name	Reset	Software Standby Mode
P000/AN008/IRQ6_D	Hi-Z	[IRQ6_D selected] IRQ6_D input [Other than the above] Keep-O
P001/AN009/IRQ7_A	Hi-Z	[IRQ7_A selected] IRQ7_A input [Other than the above] Keep-O
P002/AN010/IRQ7_C	Hi-Z	[IRQ7_C selected] IRQ7_C input [Other than the above] Keep-O
P003/AN011	Hi-Z	Keep-O
P004/AN012/IRQ2_E	Hi-Z	[IRQ2_E selected] IRQ2_E input [Other than the above] Keep-O
P008/AN002	Hi-Z	Keep-O
P009/AN003	Hi-Z	Keep-O
P010/VREFH0/AN000	Hi-Z	Keep-O
P011/VREFL0/AN001	Hi-Z	Keep-O
P012/AN004	Hi-Z	Keep-O
P013/AN005	Hi-Z	Keep-O
P014/AN006	Hi-Z	Keep-O
P015/AN007/IRQ1_A	Hi-Z	[IRQ1_A selected] IRQ1_A input [Other than the above] Keep-O
P100/AN022/IRQ2_A/TI04_A/TO04_A/TI01_B/TO01_B/RXD0_A/SI00_A/SDA00_A/RXDA0_D/SCLA0_D	Hi-Z	[IRQ2_A selected] IRQ2_A input [RXDA0_D selected] RXDA0_D input [SCLA0_D selected] SCLA0_D input/output [Other than the above] Keep-O
P101/AN021/IRQ3_A/TI07_A/TO07_A/TI00_C/TXD0_A/SO00_A/TXDA0_D/SDAA0_D	Hi-Z	[IRQ3_A selected] IRQ3_A input [TXDA0_D selected] TXDA0_D output [SDAA0_D selected] SDDA0_D input/output [Other than the above] Keep-O
P102/PCLBUZ0_B/IRQ4_A/TI06_A/TO06_A/TO00_C/RTCOUT_C/SCK00_A/SCL00_A/RXDA1_A/SCLA1_B	Hi-Z	[PCLBUZ0_B selected] PCLBUZ0_B output [IRQ4_A selected] IRQ4_A input [RTCOUT_C selected] RTCOUT_C output [RXDA1_A selected] RXDA1_A input [SCLA1_B selected] SCLA1_B input/output [Other than the above] Keep-O

Table A1.1 Port states in each processing mode (2 of 5)

Port name	Reset	Software Standby Mode
P103/IRQ5_A/TI05_A/TO05_A/SSI00_A/TXDA1_A/SDAA1_B	Hi-Z	[IRQ5_A selected] IRQ5_A input [TXDA1_D selected] TXDA1_D output [SDAA1_B selected] SDAA1_B input/output [Other than the above] Keep-O
P104/IRQ6_C/TI02_D/TO02_D/TI00_D/SCK10_A/SCL10_A	Hi-Z	[IRQ6_C selected] IRQ6_C input [Other than the above] Keep-O
P105/IRQ1_D/TI01_D/TO01_D/TO00_D/SI10_A/SDA10_A/RXDA1_B	Hi-Z	[IRQ1_D selected] IRQ1_D input [RXDA1_B selected] RXDA1_B input [Other than the above] Keep-O
P106/IRQ0_E/SO10_A/TXDA1_B	Hi-Z	[IRQ0_E selected] IRQ0_E input [TXDA1_B selected] TXDA1_B output [Other than the above] Keep-O
P107/IRQ7_D	Hi-Z	[IRQ7_D selected] IRQ7_D input [Other than the above] Keep-O
P108/SWDIO/TI03_B/TO03_B	Pull-up	Keep-O
P109/PCLBUZ1_B/IRQ4_B/TI02_A/TO02_A/TXD2_A/SO20_A/TXDA0_C/SDAA0_C	Hi-Z	[PCLBUZ1_B selected] PCLBUZ1_B output [IRQ4_B selected] IRQ4_B input [TXDA0_C selected] TXDA0_C output [SDAA0_C selected] SDAA0_C input/output [Other than the above] Keep-O
P110/IRQ3_B/TI01_A/TO01_A/RXD2_A/SI20_A/SDA20_A/RXDA0_C/SCLA0_C	Hi-Z	[IRQ3_B selected] IRQ3_B input [RXDA0_C selected] RXDA0_C input [SCLA0_C selected] SCLA0_C input/output [Other than the above] Keep-O
P111/IRQ1_C/TI07_B/TO07_B	Hi-Z	[IRQ1_C selected] IRQ1_C input [Other than the above] Keep-O
P112/IRQ2_B/TI03_A/TO03_A/SCK20_A/SCL20_A/SSI00_C	Hi-Z	[IRQ2_B selected] IRQ2_B input [Other than the above] Keep-O
P113/SO21_B	Hi-Z	Keep-O
P114/SI21_B/SDA21_B	Hi-Z	Keep-O
P115/SCK21_B/SCL21_B	Hi-Z	Keep-O

Table A1.1 Port states in each processing mode (3 of 5)

Port name	Reset	Software Standby Mode
P200/IRQ0_A/NMI	Hi-Z	[NMI/IRQ0_A selected] NMI/IRQ0_A input [Other than the above] Hi-Z
P201/PCLBUZ0_A/IRQ5_B/TI05_B/TO05_B/RTCOUT_B/SCK11_B/SCL11_B/SSI00_B	Hi-Z	[PCLBUZ0_A selected] PCLBUZ0_A output [IRQ5_B selected] IRQ5_B input [RTCOUT_B selected] RTCOUT_B output [Other than the above] Keep-O
P204/SCK01_A/SCL01_A	Hi-Z	Keep-O
P205/PCLBUZ1_A/IRQ5_C/SI01_A/SDA01_A/RXDA1_E/SCLA1_E	Hi-Z	[PCLBUZ1_A selected] PCLBUZ1_A output [IRQ5_C selected] IRQ5_C input [RXDA1_E selected] RXDA1_E input [SCLA1_E selected] SCLA1_E input/output [Other than the above] Keep-O
P206/RES*1	Pull-up	[RES (OFS1.PORTSELB = 1) selected] RES input [P206 (OFS1.PORTSELB = 0) selected] Keep-O
P206/IRQ0_C/SO01_A/TXDA1_E/SDAA1_E*2	Pull-up	[IRQ0_C selected] IRQ0_C input [TXDA1_E selected] TXDA1_E output [SDAA1_E selected] SDAA1_E input/output [Other than the above] Keep-O
P207/IRQ2_C/TO00_B/SI01_B/SDA01_B/RXDA0_A/SCLA1_A	Hi-Z	[IRQ2_C selected] IRQ2_C input [RXDA0_A selected] RXDA0_A input [SCLA1_A selected] SCLA1_A input/output [Other than the above] Keep-O
P208/IRQ3_C/TI00_B/SCK01_B/SCL01_B/TXDA0_A/SDAA1_A	Hi-Z	[IRQ3_C selected] IRQ3_C input [TXDA0_A selected] TXDA0_A output [SDAA1_A selected] SDAA1_A input/output [Other than the above] Keep-O
P212/X1/IRQ1_B/TO00_A/TI03_C/TO03_C/RXD1_A/SI11_A/SDA11_A/RXDA0_B/SCLA0_B	Hi-Z	[IRQ1_B selected] IRQ1_B input [RXDA0_B selected] RXDA0_B input [SCLA0_B selected] SCLA0_B input/output [Other than the above] Keep-O

Table A1.1 Port states in each processing mode (4 of 5)

Port name	Reset	Software Standby Mode
P213/X2/EXCLK/IRQ0_B/TI00_A/TI02_B/TO02_B/TXD1_A/SO11_A/TXDA0_B/ SDAA0_B	Hi-Z	[IRQ0_B selected] IRQ0_B input [TXDA0_B selected] TXDA0_B output [SDAA0_B selected] SDAA0_B input/output [Other than the above] Keep-O
P214/XCOUT	Hi-Z	[Sub-clock Oscillator selected] Sub-clock Oscillator is operating [Other than the above] Hi-Z
P215/XCIN	Hi-Z	[Sub-clock Oscillator selected] Sub-clock Oscillator is operating [Other than the above] Hi-Z
P300/SWCLK/TI04_B/TO04_B	Pull-up	Keep-O
P301/IRQ6_A/TI06_B/TO06_B/SI21_A/SDA21_A/RXDA1_C/SCLA1_C	Hi-Z	[IRQ6_A selected] IRQ6_A input [RXDA1_C selected] RXDA1_C input [SCLA1_C selected] SCLA1_C input/output [Other than the above] Keep-O
P302/IRQ0_D/TI05_C/TO05_C/SCK21_A/SCL21_A/TXDA1_C/SDAA1_C	Hi-Z	[IRQ0_D selected] IRQ0_D input [TXDA1_C selected] TXDA1_C output [SDAA1_C selected] SDAA1_C input/output [Other than the above] Keep-O
P303/SO21_A	Hi-Z	Keep-O
P304	Hi-Z	Keep-O
P400/SCLA1_D	Hi-Z	[SCLA1_D selected] SCLA1_D input/output [Other than the above] Keep-O
P401/SDAA1_D	Hi-Z	[SDAA1_D selected] SDAA1_D input/output [Other than the above] Keep-O
P402/IRQ2_D/TXD2_B/SO20_B/TXDA0_F	Hi-Z	[IRQ2_D selected] IRQ2_D input [TXDA0_F selected] TXDA0_F output [Other than the above] Keep-O
P403/IRQ4_E/RXD2_B/SI20_B/SDA20_B/RXDA0_F	Hi-Z	[IRQ4_E selected] IRQ4_E input [RXDA0_F selected] RXDA0_F input [Other than the above] Keep-O

Table A1.1 Port states in each processing mode (5 of 5)

Port name	Reset	Software Standby Mode
P407/PCLBUZ0_C/IRQ4_C/RTCOUT_A/SCK11_A/SCL11_A/SDAA1_F	Hi-Z	[PCLBUZ0_C selected] PCLBUZ0_C output [IRQ4_C selected] IRQ4_C input [RTCOUT_A selected] RTCOUT_A output [SDAA1_F selected] SDAA1_F input/output [Other than the above] Keep-O
P408/IRQ7_B/TI04_C/TO04_C/SCLA1_F	Hi-Z	[IRQ7_B selected] IRQ7_B input [SCLA1_F selected] SCLA1_F input/output [Other than the above] Keep-O
P409/IRQ6_B/TI03_E/TO03_E/SCK11_C/SCL11_C	Hi-Z	[IRQ6_B selected] IRQ6_B input [Other than the above] Keep-O
P410/IRQ4_D/TI02_C/TO02_C/SCK20_B/SCL20_B/SSI00_D/RXDA1_D/SCLA0_E	Hi-Z	[IRQ4_D selected] IRQ4_D input [RXDA1_D selected] RXDA1_D input [SCLA0_E selected] SCLA0_E input/output [Other than the above] Keep-O
P411/IRQ3_D/TI01_C/TO01_C/SCK11_D/SCL11_D/TXDA1_D/SDAA0_E	Hi-Z	[IRQ3_D selected] IRQ3_D input [TXDA1_D selected] TXDA1_D output [SDAA0_E selected] SDAA0_E input/output [Other than the above] Keep-O
P500/TI03_D/TO03_D/SCK00_B/SCL00_B	Hi-Z	Keep-O
P501/TI04_D/TO04_D/TXD0_B/SO00_B/TXDA0_E/SDAA0_F	Hi-Z	[TXDA0_E selected] TXDA0_E output [SDAA0_F selected] SDAA0_F input/output [Other than the above] Keep-O
P502/IRQ5_D/RXD0_B/SI00_B/SDA00_B/RXDA0_E/SCLA0_F	Hi-Z	[IRQ5_D selected] IRQ5_D input [RXDA0_E selected] RXDA0_E input [SCLA0_F selected] SCLA0_F input/output [Other than the above] Keep-O
P913/SDAA0_A	Hi-Z	[SDAA0_A selected] SDAA0_A input/output [Other than the above] Keep-O
P914/SCLA0_A	Hi-Z	[SCLA0_A selected] SCLA0_A input/output [Other than the above] Keep-O
P915/SO01_B	Hi-Z	Keep-O

Note: Hi-Z: High-impedance
Keep-O: Output pins retain their previous values. Input pins become high-impedance.

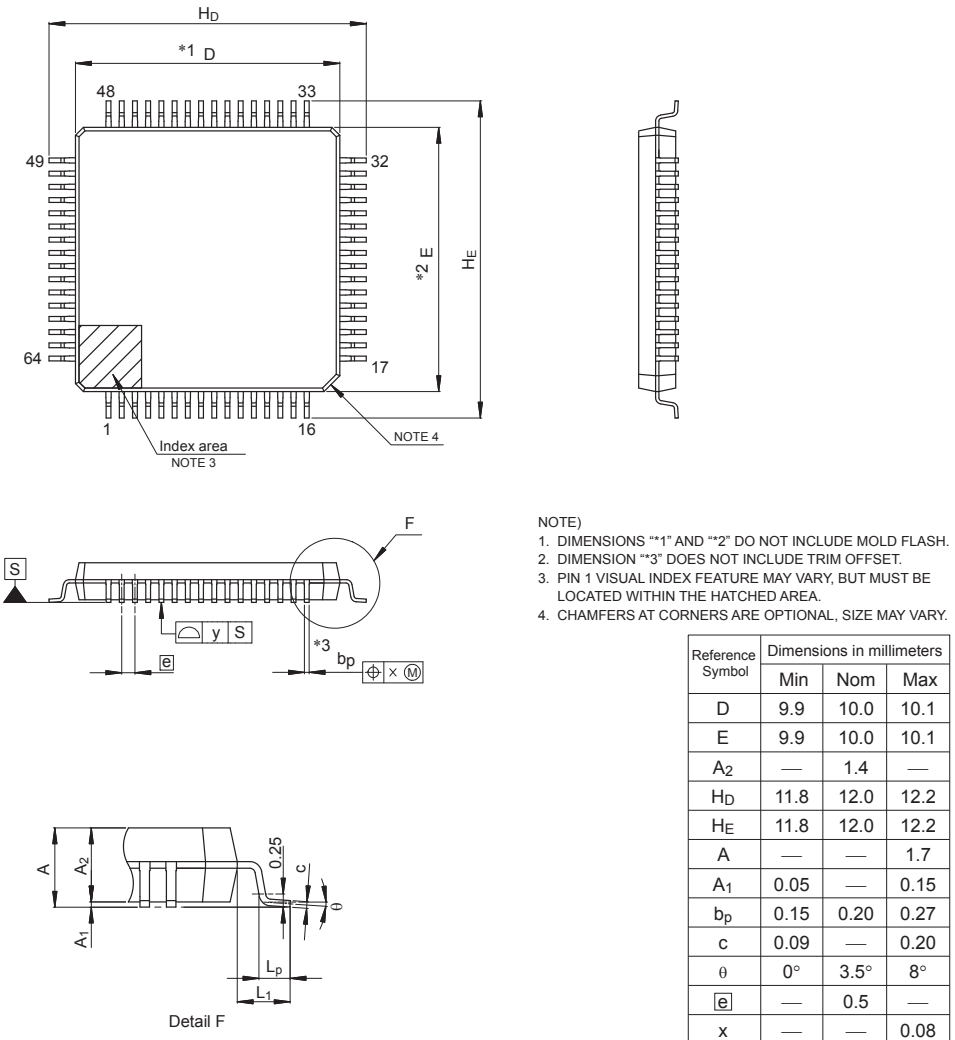
Note 1. "P206" is available only for 32 pin product.
Note 2. "P206/IRQ0_C/SO01_A/TXDA1_E/SDAA1_E" is available for 48 and 64 pins product.

Appendix 2. Package Dimensions

Information on the latest version of the package dimensions or mountings is displayed in “Packages” on the Renesas Electronics Corporation website.

JEITA Package Code	RENESAS Code	Previous Code	MASS (Typ) [g]
P-LFQFP64-10x10-0.50	PLQP0064KB-C	—	0.3

Unit: mm



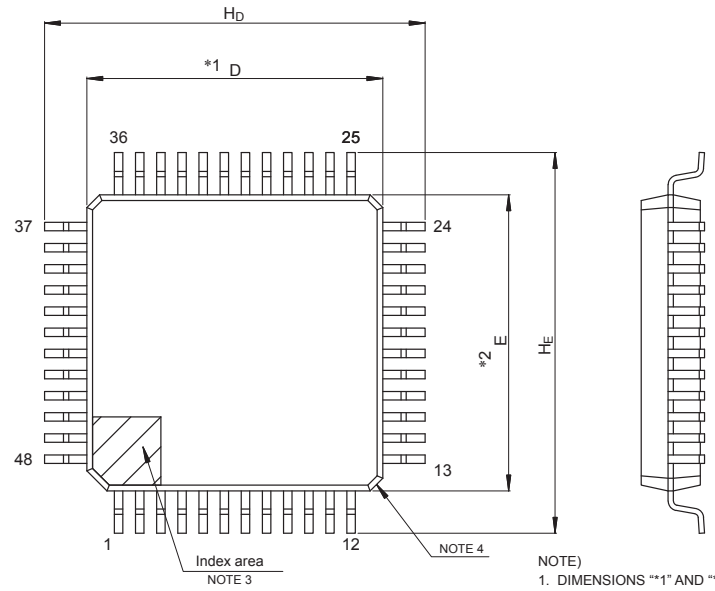
NOTE)
1. DIMENSIONS “*1” AND “*2” DO NOT INCLUDE MOLD FLASH.
2. DIMENSION “*3” DOES NOT INCLUDE TRIM OFFSET.
3. PIN 1 VISUAL INDEX FEATURE MAY VARY, BUT MUST BE LOCATED WITHIN THE HATCHED AREA.
4. CHAMFERS AT CORNERS ARE OPTIONAL, SIZE MAY VARY.

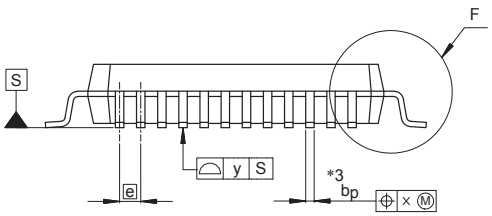
Reference Symbol	Dimensions in millimeters		
	Min	Nom	Max
D	9.9	10.0	10.1
E	9.9	10.0	10.1
A_2	—	1.4	—
H_D	11.8	12.0	12.2
H_E	11.8	12.0	12.2
A	—	—	1.7
A_1	0.05	—	0.15
b_p	0.15	0.20	0.27
c	0.09	—	0.20
θ	0°	3.5°	8°
e	—	0.5	—
x	—	—	0.08
y	—	—	0.08
L_p	0.45	0.6	0.75
L_1	—	1.0	—

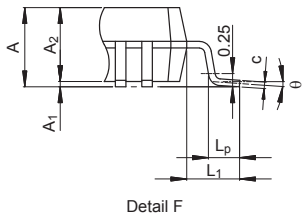
Figure A2.1 LFQFP 64-pin 0.5mm pitch

JEITA Package Code	RENESAS Code	Previous Code	MASS (Typ) [g]
P-LFQFP48-7x7-0.50	PLQP0048KB-B	—	0.2

Unit: mm







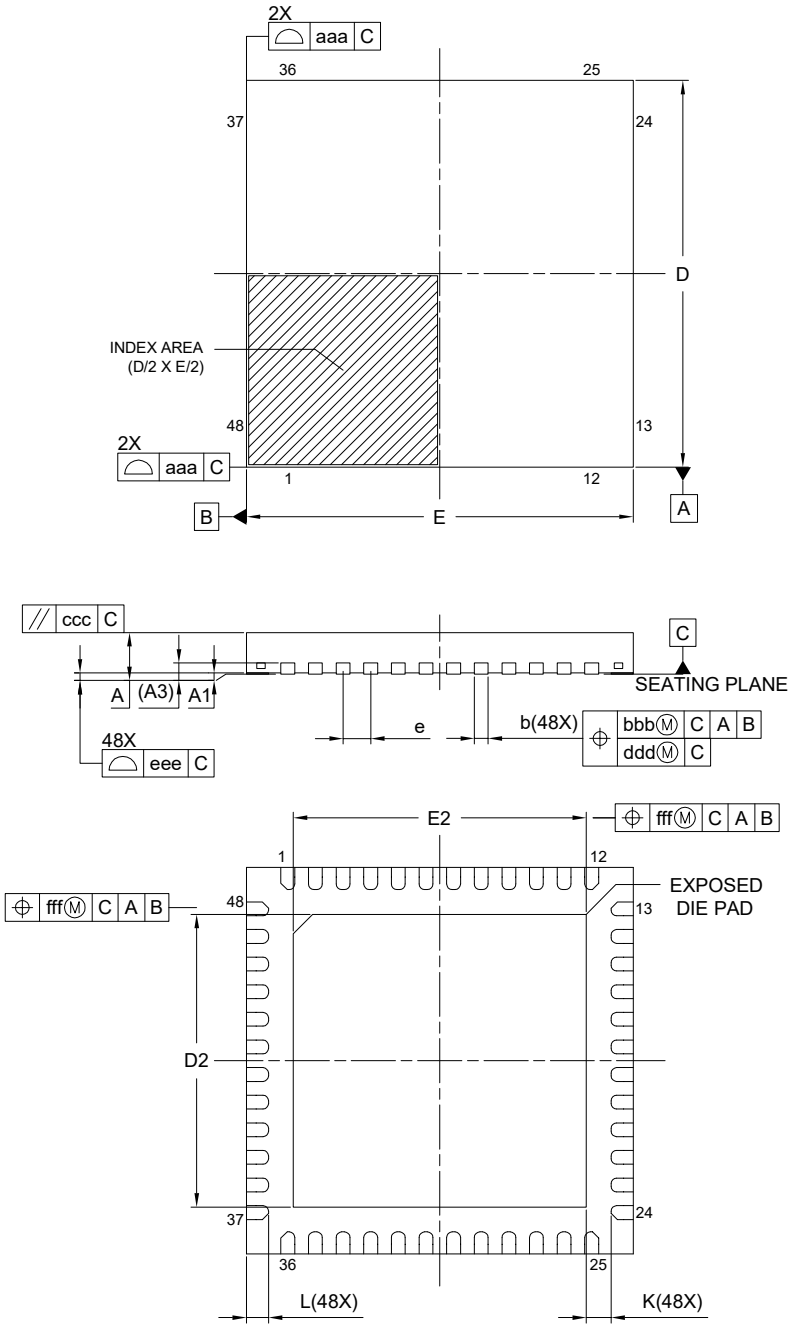
Detail F

NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.
3. PIN 1 VISUAL INDEX FEATURE MAY VARY, BUT MUST BE LOCATED WITHIN THE HATCHED AREA.
4. CHAMFERS AT CORNERS ARE OPTIONAL, SIZE MAY VARY.

Reference Symbol	Dimensions in millimeters		
	Min	Nom	Max
D	6.9	7.0	7.1
E	6.9	7.0	7.1
A_2	—	1.4	—
H_D	8.8	9.0	9.2
H_E	8.8	9.0	9.2
A	—	—	1.7
A_1	0.05	—	0.15
b_p	0.17	0.20	0.27
c	0.09	—	0.20
θ	0°	3.5°	8°
e	—	0.5	—
x	—	—	0.08
y	—	—	0.08
L_p	0.45	0.6	0.75
L_1	—	1.0	—

Figure A2.2 LFQFP 48-pin

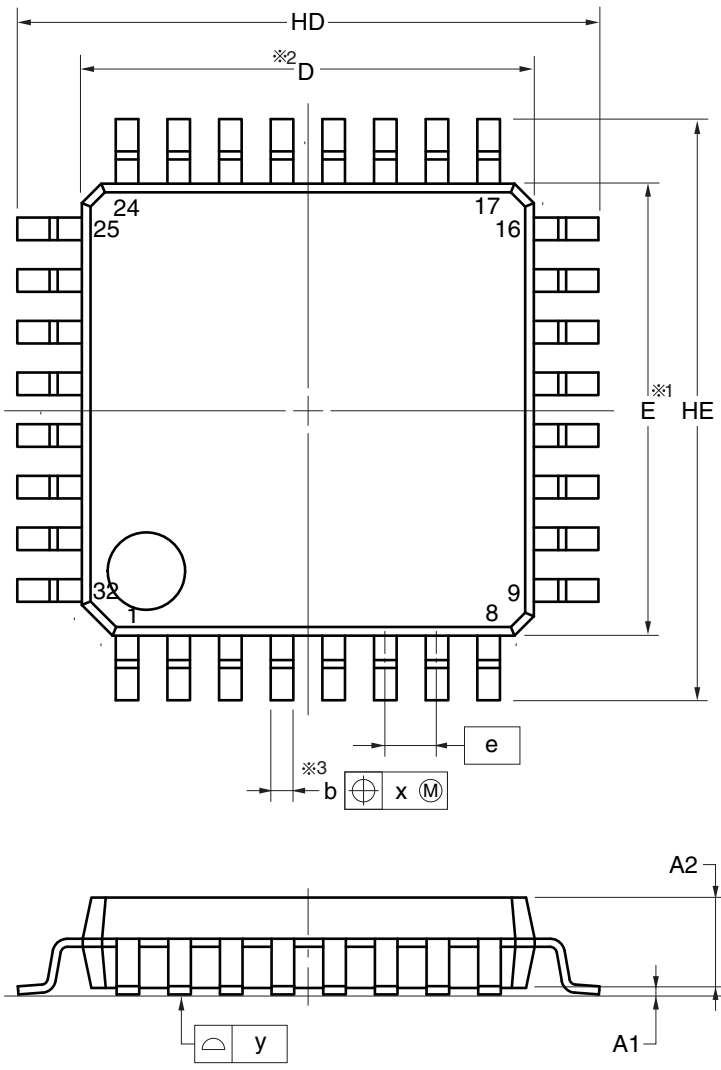
JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-HWQFN048-7x7-0.50	PWQN0048KC-A	0.13 g

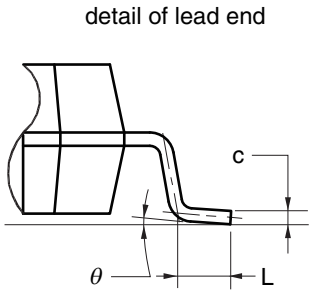


Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
A	—	—	0.80
A_1	0.00	0.02	0.05
A_3		0.203 REF.	
b	0.20	0.25	0.30
D		7.00 BSC	
E		7.00 BSC	
e		0.50 BSC	
L	0.30	0.40	0.50
K	0.20	—	—
D_2	5.25	5.30	5.35
E_2	5.25	5.30	5.35
aaa		0.15	
bbb		0.10	
ccc		0.10	
ddd		0.05	
eee		0.08	
fff		0.10	

Figure A2.3 HWQFN 48-pin

JEITA Package Code	RENESAS Code	Previous Code	MASS (TYP.) [g]
P-LQFP32-7x7-0.80	PLQP0032GB-A	P32GA-80-GBT-1	0.2





(UNIT:mm)

ITEM	DIMENSIONS
D	7.00±0.10
E	7.00±0.10
HD	9.00±0.20
HE	9.00±0.20
A	1.70 MAX.
A1	0.10±0.10
A2	1.40
b	0.37±0.05
c	0.145±0.055
L	0.50±0.20
θ	0° to 8°
e	0.80
x	0.20
y	0.10

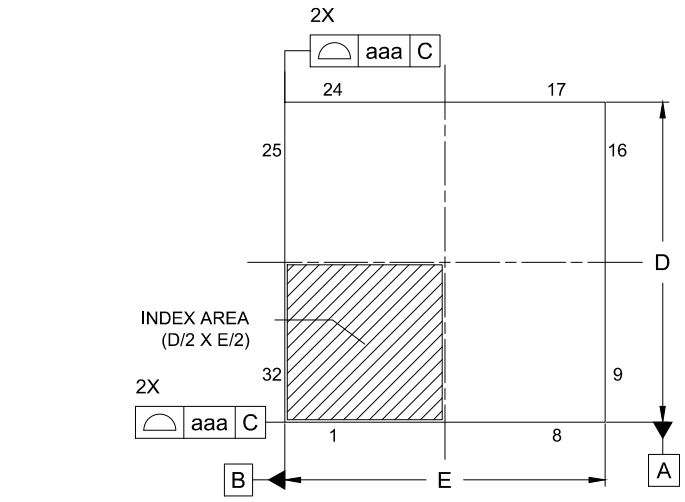
NOTE

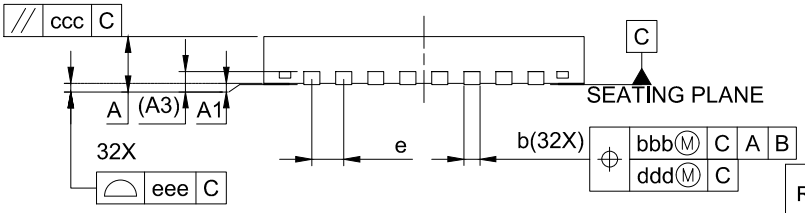
1.Dimensions “※1” and “※2” do not include mold flash.

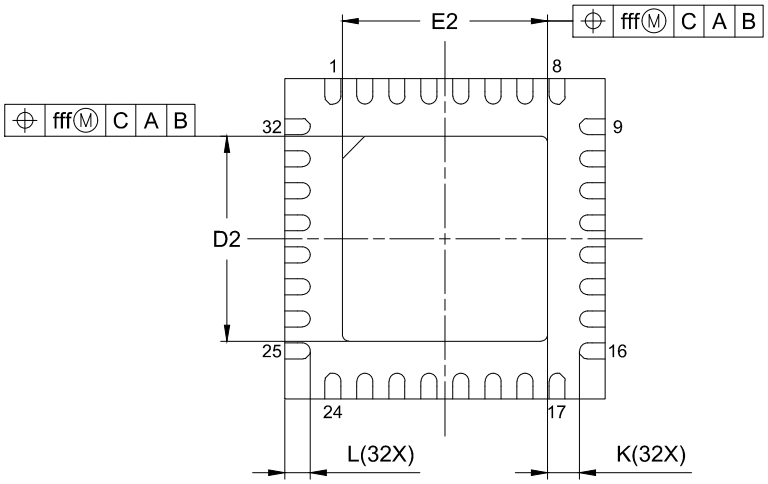
2.Dimension “※3” does not include trim offset.

Figure A2.4 LQFP 32-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 A2.5 HWQFN 32-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 A3.1 shows the name, description, and the base address of each peripheral.

Table A3.1 Peripheral base address

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
DBG	Debug Function	0x4001_B000
SYSC	System Control	0x4001_E000
ELC	Event Link Controller	0x4004_1000
IWDT	Independent Watchdog Timer	0x4004_4400
MSTP	Module Stop Control	0x4004_7000
CRC	CRC Calculator	0x4007_4000
PORT0	Port 0 Control	0x400A_0000
PORT1	Port 1 Control	0x400A_0020
PORT2	Port 2 Control	0x400A_0040
PORT3	Port 3 Control	0x400A_0060
PORT4	Port 4 Control	0x400A_0080
PORT5	Port 5 Control	0x400A_00A0
PORT9	Port 9 Control	0x400A_0120
PFS_A	Pmn Pin Function Select	0x400A_0200
PORGA	Product Organize	0x400A_1000
ADC_D	12-bit A/D Converter	0x400A_1800
SAU0	Serial Array Unit 0	0x400A_2000
SAU1	Serial Array Unit 1	0x400A_2200
TAU	Timer Array Unit	0x400A_2600
RTC_C	Realtime Clock	0x400A_2C00
IICA	I^2C Bus Interface	0x400A_3000
UARTA	Serial Interface UARTA	0x400A_3400
TML32	32-bit Interval Timer	0x400A_3800
PCLBUZ	Clock Output/Buzzer Output Controller	0x400A_3B00
TRNG	True Random Number Generator	0x400D_1000
FLCN	Flash I/O Registers	0x407E_C000

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 A3.2:

- 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.

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 A3.2 shows the register access cycles.

Table A3.2 Access cycles

Peripherals	Address		Number of access cycles			
	From	To	Read	Write	Cycle unit	Related function
SRAM, BUS, DTC, ICU, DBG	0x4000_2000	0x4001_BFFF	3		ICLK	Memory Protection Unit, SRAM, Buses, Data Transfer Controller, Interrupt Controller, CPU, Flash Memory
SYSC	0x4001_E000	0x4001_E6FF	2		ICLK	Low Power Modes, Resets, Low Voltage Detection, Clock Generation Circuit, Register Write Protection
ELC, IWDT, MSTP	0x4004_0000	0x4004_7FFF	3		PCLKB	Event Link Controller, Watchdog Timer, Module Stop Control
CRC	0x4007_4000	0x4007_4FFF	3		PCLKB	CRC Calculator
PORT, PFS_A, PORGA, ADC12, SAU0, SAU1, TAU, RTC, IICA, UARTA, TML32, PCLBUZ	0x400A_0000	0x400A_3FFF	2		PCLKB	I/O Ports, 12-bit A/D Converter, Serial Array Unit 0, Serial Array Unit 1, Timer Array Unit, Real time Clock, I^2C Bus Interface, Serial Interface UARTA, 32-bit Interval Timer, Clock/Buzzer Output Controller
TRNG	0x400D_1000	0x400D_1FFF	3		PCLKB	True Random Number Generator
FLCN	0x407E_C000	0x407E_FFFF	7		ICLK	Data Flash, Flash Control

Appendix 4. Peripheral Variant

Table A4.1 shows the correspondence between the module name used in this manual and the Peripheral Variant.

Table A4.1 Module name vs Peripheral Variant

Module name	Peripheral Variant
ADC12	ADC_D
RTC	RTC_C

Revision History

Revision 1.00 — Dec 27, 2024

Initial release

Revision 1.10 — Nov 28, 2025

Features:
- Updated Memory.

1. Overview:
- Updated Figure 1.1 Block diagram.
- Updated Table 1.13 Function comparison.
- Updated Table 1.14 Pin functions.
- Updated Table 1.15 Pin list.

2. CPU:
- Updated Table 2.2 SWD pins.

5. Resets:
- Updated Table 5.1 Reset names and sources.

8. Clock Generation Circuit:
- Updated Figure 8.1 Clock generation circuit block diagram (64-, 48-, 32-pin).
- Updated 8.2.8 SOSCCR : Sub-clock Oscillator Control Register and 8.2.9 LOCOCR : Low-speed On-chip Oscillator Control Register.
- Updated 8.2.11 MOCOCR : Middle-speed On-chip Oscillator Control Register.
- Updated 8.2.13 OSTS : Oscillation Stabilization Time Select Register.

9. Low Power Modes:
- Updated 9.1 Overview.
- Updated 9.2.1 SBYCR : Standby Control Register.
- Updated 9.5.1 Setting Operating Power Control Mode.
- Added 9.9.11 Sleep-on-exit Function.

14. Data Transfer Controller (DTC):
- Updated 14.9 Low Power Consumption Function.

16. I/O Ports:
- Updated Table 16.3 Handling of unused pins.
- Updated Table 16.6 Examples of register settings for port and alternate functions (2/6).
- Updated 16.6 Peripheral Select Settings for Each Product.

17. Timer Array Unit (TAU):
- Updated 17.1 Overview.
- Updated 17.2.4 TMR0n : Timer Mode Register 0n (n = 0, 2, 4, 5, 6, 7).
- Updated 17.2.5 TMR0n : Timer Mode Register 0n (n = 1, 3).
- Updated 17.2.16 ISC : Input Switch Control Register.

21. Serial Array Unit (SAU):
- Updated 21.1 Overview.
- Updated 21.3.3 SMRmn : Serial Mode Register mn (mn = 01, 03, 11).
- Updated 21.3.25 SSC0 : Serial Standby Control Register 0 and 21.3.26 ISC : Input Switch Control Register.
- Updated Table 21.71 Selection of operation clock for simplified SPI, UART and simplified I2C.

22. I^2C Bus Interface (IICA):
- Updated Table 22.1 I2C specifications.
- Updated 22.2.7 IICWLn : IICA Low-level Width Setting Register n (n = 0, 1) and 22.2.8 IICWHn : IICA High-level Width Setting Register n (n = 0, 1).
- Updated 22.3.2 Setting Transfer Clock Using IICWLn and IICWHn Registers.
- Updated 22.3.10 Timing of Generation of the Interrupt Request Signal (IICAn_TXRXI) and Control of Clock Stretching.
- Updated 22.3.17 Usage Notes.

23. Serial Interface UARTA (UARTA):
- Updated 23.2.4 ASIMAn1 : Operation Mode Setting Register n1 (n = 0, 1).
- Updated Table 23.3 Step of communication procedure.
- Updated 23.3.4 Baud Rate Generator.

25. 12-bit A/D Converter (ADC12):
- Updated 25.7.1 A/D Conversion by Inputting a Hardware Trigger.

31. Electrical Characteristics:
- Updated Table 31.23 In UART communications with devices operating at the same voltage levels.
- Updated 31.6.1 A/D Converter Characteristics.
- Updated Table 31.44 Temperature sensor/internal reference voltage characteristics.

Revision 1.10 — Nov 28, 2025

Appendix 1. Port States in each Processing Mode:
- Updated Table A1.1 Port states in each processing mode.

RA0E2 Group User’s Manual: Hardware

Publication Date: Rev.1.10 Nov 28, 2025
Rev.1.00 Dec 27, 2024

Published by: Renesas Electronics Corporation

32-Bit MCU
RA0E2 Group

Renesas Electronics Corporation

R01UH1090EJ0110